Doctoral theses at NTNU, 2019:247

Sofia Papavlasopoulou

Making-based coding activities for young students: Design meaningful learning experiences

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of Philosophiae Doctor
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Doctoral thesis

Sofia Papavlasopoulou

Doctoral theses at NTNU, 2019:247

NTNU
Norwegian University of
Science and Technology

ISBN 978-82-326-4090-4 (printed ver.)
ISBN 978-82-326-4091-1 (electronic ver.)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

NTNU

Sofia Papavlasopoulou

Making-based coding activities for young students: Design meaningful learning experiences

Thesis for the Degree of Philosophiae Doctor

Trondheim, September 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Computer Science

ISBN 978-82-326-4090-4 (printed ver.)
ISBN 978-82-326-4091-1 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2019:247

Printed by NTNU Grafisk senter

There is a crack in everything, that's how the light gets in….
Leonard Cohen

Abstract

Over the last few years, there is growing evidence supporting the acquisition of 21st-century skills from young students and the introduction of computer science and computational thinking into K-12 education. *Making* offers new opportunities, as it involves constructing activities and related ways using technological tools to create technology-based meaningful artifacts. In an informal setting, different benefits arise when young students are engaged in coding and *making* in the context of a learning experience; they are given the opportunity to enhance their understanding of programming concepts, to collaborate with friends, to use their curiosity, imagination, and creativity, and to change their attitudes toward coding. Additionally, during *making*-based coding activities, when young students focus on the continuous exploratory process of creating something meaningful, pure learning obtains a fertile ground. To overcome various barriers in the learning process (e.g., difficulty, boredom, and confusion), we need appropriately designed and engaging coding activities for young students. Despite the growing research, there is still considerable need to further and systematically investigate those kinds of activities and provide evidence on their appropriate design.

The overall research aims to explore *how making can help us design meaningful coding learning experiences for young students*, considering that learning via technology is a complex process associated with various aspects of interaction, containing variations in the way young students handle coding tasks and how they manage the learning process. The research context of this PhD work is framed between the fields of Child-Computer Interaction, Technology Enhanced Learning, and K-12 Computing Education.

To this end, this PhD research follows a design-based research approach to understand *making*-based coding learning experiences for young students through design, exploration, enactment, evaluation, and redesign. Three iterations (cycles), which represent the field studies of this PhD research, were designed and evaluated in total, with participants aged 8–17 years old, using mixed methods. Young students, as participants in our workshops, used a block-based programming environment (i.e., Scratch) and collaboratively created a socially meaningful artifact (i.e., a game).

The resulting contributions include a systematic literature review, producing substantive findings regarding the *making* approach, and its applications to *making*-based coding activities. Additionally, this PhD research contributes in the investigation of new methods to evaluate coding activities for young students, as the eye-tracking method was used to pervasively track the gaze of children of young ages in a coding activity. Moreover, focusing on enhancing young students' engagement and attitudes, together with examining the potential differences in how young students of different ages and genders handle the learning process during coding activities, the next contribution is to provide

an improved understanding of the learning process to support young students in *making*-based coding activities. Finally, this PhD research's contribution lies in the theoretical grounding of the findings and identification of design principles that can shed light on best practices in the design of *making*-based coding activities for children.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) in partial fulfillment of the requirements for the degree of Philosophiae Doctor.

The PhD work was performed at the Department of Computer Science, NTNU, Trondheim, under the supervision of Associate Professor Michail Giannakos (main supervisor) and Professor Letizia Jaccheri (co-supervisor).

During this PhD project, I have been involved in the Kodeløypa project (which has been my main source of data collection), initiated by Letizia Jaccheri, and in the Learner Computer Interaction Lab, led by Michail Giannakos

Acknowledgements

I feel privileged for the opportunity to carry out a PhD project and I would like to thank everyone who has contributed to this research journey. First, I would like to express my deepest gratitude to my supervisors, Michail Giannakos and Letizia Jaccheri, for believing in me more than I believed in myself and for always being supportive. I have been extremely lucky to have supervisors who cared so much about my work. Many thanks to Michail Giannakos for his patient guidance, encouragement and advice. Special thanks to Letizia Jaccheri for her continuous inspiration, availability and constructive suggestions which were determinant for the accomplishment of my PhD project.

I am very grateful to Monica Divitini. Thank you for your support and all the beneficial advice and the discussions we had. I also want to thank Kshitij Sharma for sharing his knowledge and for collaborating with me in several research papers. You have been a wonderful colleague and above all, a friend.

Very special thanks to Kristin Susanne Karlsen and all the teaching assistants of Kodeløypa: Amanda Jørgine Haug, Lidia Luque Fernandez, An Nguyen, Ton Mangmee, Marjeris Sofia Romero, Eline Stenwig, Kristoffer Venæs Monsen and master student Uyen Dan Nguyen. Working with all of you has been fundamental.

I extend my gratitude to all the colleagues from the Department of Computer Science, ISSE group and the fellow members of the LCI Lab. A special mention to Chara, Ilias, Katerina, Nektaria, Patrick, Serena and Stella for all the engaging discussions and support, also, to Fufen for being a very supportive officemate.

I thank my lovely Italian gang, Elena, Simone and Francesco for being there from day zero in my PhD journey and my life in Norway, helping me with every possible way to survive from all the low moments but also truly share my happiness with them.

A big thank you to all my friends from Corfu, my supporters for so many years now, for making me forget my PhD struggles. They made me see things differently and more optimistically.

Last but not least, I want to express my gratitude to my family. First, to my parents who always supported my decisions including the one to move to Norway. Thanks to my sisters, Maria and Anneta, my nieces, Xenia and Emilia and my nephew, Alexandros who remind me the importance of simple things in life, fulfilling my heart with love and joy every time I see them.

Table of Contents

Part I: Synopsis

1 Introduction

1.1 Introduction and problem statement

Increasing attention has been given to young students' acquisition of 21st-century skills and digital competences. As citizens of a digital world, they should be fluent in technology; i.e., know how the use of technology meets their needs and how it can be changed. In accordance with this need, teaching coding to young students is currently gaining momentum in classrooms and informal learning spaces (e.g., coding clubs, fairs, labs, etc). Several countries have introduced a coding curriculum for all school students; Estonia, Israel, Finland, and the United Kingdom are only a few nations making efforts to integrate coding as a new literacy and support students in creative problem-solving tasks (Hubwieser, Armoni, Giannakos, & Mittermeir, 2014). Similarly, organizations such as "code.org" and "codeacademy.com" offer fruitful coding learning environments. In addition, ACM, the Computer Science Teachers Association, the National Math and Science Initiative, and the K-12 Computer Science Framework provide guidelines for teaching computer science and building learning communities. The simultaneous rise of the maker movement, and increased maker culture-based initiatives (e.g., Makerspaces, Fablabs, Techshops), have evolved in the sense that the maker movement is a technological and creative evolution that has limitless implications for the education world. In different informal learning spaces, like science centers, libraries, and museums, more and more young students develop their own projects and get a different perspective on the learning process since they have the opportunity to control their own learning, instead of being passive knowledge recipients.

Pioneered by Seymour Papert, whose constructionism demonstrated the need to learn through creative making processes (Papert, 1980), and his Logo programming environment in the 1960s, coding in education has received a lot of interest from educators and researchers seeking alternative ways to teach complex problem-solving skills and provide dynamic learning experiences (Kalelioğlu, 2015); (Lye & Koh, 2014). Today, there is a large amount of software available, along with child-friendly programming environments, such as Alice, Scratch, Greenfoot, and Kodu. Therefore, coding and making activities in the context of informal learning experiences have become a more intuitive and engaging experience for young students. Previous research shows that different approaches can combine physical fabrication and coding (Y. B. Kafai & V. Vasudevan, 2015), while others, such as Buechley, Eisenberg, Catchen, and Crockett (2008), used LilyPad Arduino to make coding attractive to girls. By using the Logo-based environment and an interactive white board from kindergarten age, children developed mathematical concepts and social interaction through enjoyable learning activities (Fessakis, Gouli, & Mavroudi, 2013). By participating in these kinds of activities, children are exposed to computational thinking (Wing, 2006), which involves, but is not limited to, critical thinking and problem solving. Apart from that, different benefits arise when young students are engaged in coding and making in the context of a learning experience; they are given the opportunity to enhance their understanding of programming concepts, to collaborate with friends, to use their curiosity, imagination, and creativity, and to change their attitudes toward coding (Fessakis et al., 2013; Maloney, Peppler, Kafai, Resnick, & Rusk, 2008).

Several studies have focused on how to introduce computational literacy to children. Previous research has described practices to motivate and engage children in coding through making (Denner, Bean, & Martinez, 2009). Combining computers with meaningful programmable objects, such as interactive robots, can provide a valuable coding learning experience in a fun and playful manner (Bers, Flannery, Kazakoff, & Sullivan, 2014). Robertson and Howells (2008) argued that making a game was an authentic learning activity, which provided motivation, engagement, and enthusiasm for learning. When making is combined with block-based programming environments, such as Scratch, children's intensive use improved their understanding of concepts, including loops and variables (Maloney et al., 2008). In an informal setting, when making and coding activities involve playful elements, pure learning finds a fertile ground; students find a meaning in their actions, are active, engaged, and use iterative thinking. To overcome the various barriers in the learning process (e.g., difficulty, boredom, confusion, etc.), we need appropriately designed and engaging coding activities for young students. Therefore, there is a need to further and systematically investigate the qualities of *making* in coding activities.

Making involves constructing activities and related ways to fabricate real and/or digital things using technological resources, including fabrication (Katterfeldt, Dittert, & Schelhowe, 2015). The term *making* widely refers to using technological tools, such as 3D printing, coding, microprocessors, electronics, etc. to create technology-based artifacts (Chu, Angello, Saenz, & Quek, 2017). Enriching a making activity with technological resources and focusing on the continuous exploratory process of creating something meaningful, including both the process and the product, is different from the typical hands-on activities and the ones that focus only on task completion (Chu, Quek, Bhangaonkar, Ging, & Sridharamurthy, 2015).

Despite growing research on the design of making-based coding experiences for students, it is still difficult to define the different student populations' needs. There is some variation in the ways students handle coding tasks and how they manage the learning process. There are systems using various affordances to support different age groups (e.g., Scratch and Scratch Junior), but research on how the different age groups use those tools is still in its infancy. In addition, there are relatively few studies focusing on gender issues in making and coding activities for children. Gender discrepancy in science, technology, engineering and mathematics (STEM) exist, and women are underrepresented in the computer science field (Cheryan, Ziegler, Montoya, & Jiang, 2017). In terms of interest, the gender gap starts at elementary school (Ceci & Williams, 2010). Among the different factors that affect women following computer science paths are the lack of positive educational experiences in their childhood (Adya & Kaiser, 2005), their fear of being involved in very technical coding courses, and stereotypes and misconceptions around careers in computing (Teague, 2002). Since more people believe that coding skills are as important as math and writing (Horizon, October 5, 2015), there is an acute need to investigate and get a deep understanding on the benefits and characteristics of making-based coding activities for young students and how to design learning activities.

Interestingly, current research has been focusing on traditional qualitative and/or quantitative measurements, such as observations, interviews, tests, and surveys, to investigate children's engagement, experience, and learning (Papavlasopoulou, Giannakos, & Jaccheri, 2017) in coding and making activities for young students. Learning with technology is a complex process that is associated with many aspects of interaction (e.g., hard mental operations, cognitive friction, etc.). Considering the wide range of multimodal data produced when students interact with a system, there are devices like cameras, wearable sensors, biosensors, infrared imaging, and eye-tracking to enhance the way we collect and analyze data for a deeper understanding of young students' interactions with learning technologies (Noroozi et al., 2018). Despite multimodal data's great potential –and in particular physiological data– for understanding learners' cognition, emotion, attention, and more, research in this direction is limited.

Integrating information from more than a single data stream can give a more complete picture of the learning experience for more holistic understanding. Therefore, research should also focus on using other measures to better understand how young students, of different ages and genders, experience those activities, along with their task-based behavior. With young students, using objective measures such as physiological (e.g., eye-tracking) data, is important because they are generalizable (more than qualitative and subjective measures), real time, and a more reliable way to monitor users' actions. In contrast to other subjective measures, objective measures are independent of perceptual abilities. Using other methods and measures, successfully applied in similar contexts, will better demonstrate how young students learn coding and provide feedback on designing coding experiences to improve learning. To the best of our knowledge, eye-tracking has not yet been used to investigate how students of young ages code.

Considering the above topics, the overall research aim of this PhD thesis is to investigate *how making can help us design meaningful coding learning experiences for young students.*

The research described in this PhD work concerns the study, evaluation, and implementation of making-based coding experiences for young students. Using technological tools to support children's learning has opportunities and challenges dealing with the interaction in real educational settings; in the case of this PhD work, the well-known Scratch programming environment was used. Moreover, enhancing young students' interest in Computing Education and coding in extracurricular activities seems to be very effective in promoting computational concepts and skills, providing creative engagement in various learning activities. Thus, the research context of this PhD work is framed between the fields of Child-Computer Interaction (CCI), Technology Enhanced Learning (TEL), and K-12 Computing Education (Figure 1).

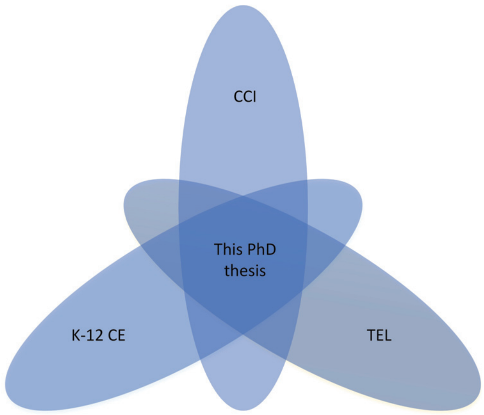

Figure 1:Venn diagram with fields of research

1.2 Goal and research questions

Based on the above, the purpose of this thesis is to support an interactive, engaging approach to informal learning coding activities for young students. Fieldwork was carried out at making-based coding workshops held in conjunction with an initiative organized at the Norwegian University of Science and Technology (NTNU), named Kodeløypa. The focus is on young students' learning experiences while they are introduced to coding to develop a digital artifact. In particular, the motivation is to investigate and flourish the existing practices of informal settings of coding activities, so that become more effective and attractive for young students. An important aspect of this thesis is to investigate how different populations of young students experience the learning process and what are the main characteristics. Therefore, I also focused on possible gender and age differences among the young students. In addition, to have a more holistic view, I collected multimodal data to capture various aspects and phases of the learning experience during these studies. This gives a new perspective to the methods used previously in similar contexts and helped reveal the complex interactions and invisible cognitive learning processes as they occur in challenging learning situations. The research examined theories and concepts that are relevant to Child-Computer Interaction and the learning, motivational, and cognitive processes.

The main aspects of this thesis during the years are: 1) the design of coding workshops to facilitate children's use of the programming tool and introduce them to coding, 2) the use of different methods to evaluate our approach's effectiveness to increase sustainability and scalability, 3) researchers collaborating closely with the participants and assistants who ran the workshops, 4) grounding findings in theory, and 5) identifying general design principles for future similar activities.

The problem foundation of this PhD thesis is expressed by the following main research question and four sub-questions:

> *RQ: How can making help us design meaningful coding learning experiences for young students*?

SQ1. *What is the state-of-the art of making activities related to coding?*

SQ2. *What is the role of young students' attitudes and engagement in their learning experiences during making-based coding activities?*

SQ3. *What is the role of students' ages and genders on how they experience making-based coding activities?*

SQ4. *What principles can enhance making-based coding activities to support students' learning experiences?*

The first question aims to provide a review of how *making* is used in educational settings to support coding activities. The maker movement has gathered much attention recently, due to contemporary technical and infrastructural developments. Scholars and educators have shown a variety of outcomes from *making* as an instructional and learning process; however, there is a need for a clear view of its benefits and challenges. Therefore, we aim to summarize the current findings and identify potential gaps to guide the decisions and future studies in this PhD thesis.

The second question investigates the role of two main aspects that a making-based learning activity for young students involves, which are attitudes and engagement. Motivating, fun, and creative experiences harness young students' sense of excitement, enjoyment, levels of engagement, and personal investment in learning. Exploring those aspects and combining insights from objective measures, we aim to shed light on how young students experience learning during coding.

The third question seeks to explore the potential influences of the young students' ages and genders on their participation in making-based coding activities. Young students are involved in a continuous exploratory procedure that integrates both the process and the product. As they are deeply engaged into creating, experimenting, and collaborating, it is important to better understand how young students of different ages and genders learn to code by focusing on their task-based behaviors and using objective measures. In that way, we can provide feedback on how to design coding experiences to improve the learning process.

The fourth and last question aims to summarize and conceptualize the knowledge gained from the literature and the empirical studies conducted in this PhD work. The answers will help derive design principles that can support making-based coding activities.

1.3 Research approach

The work presented in this dissertation followed the Design-Based Research (DBR) approach. DBR offers a strategy to understand learning processes through design, exploration, enactment, evaluation, and redesign (Anderson, 2005) and is widely used in educational contexts (Anderson & Shattuck, 2012) (Wang & Hannafin, 2005); (Reeves, 2006). DBR is a hybrid method; it does not replace other methodologies but builds on the use of multiple procedures and methods from both design and research methodologies (Wang & Hannafin, 2005). DBR's purpose is to influence real educative interventions and validate theoretical concepts. During the process, researchers are actively involved and maintain constant collaboration with participants, other researchers, and practitioners to manage the research process in real-world settings. The aim is to implement interventions with refined and improved designs that influence practice. In short, there are five basic characteristics of DBR: 1) it refines theory and practice, 2) it happens in real-world settings and is grounded in relevant contexts, 3) it is interactive, iterative, and flexible, 4) it uses mixed methods in accordance with potential new needs and emerging issues, and 5) it is contextual, meaning that the research findings are connected to the design process (Wang & Hannafin, 2005).

More specifically, for my research, I used the DBR approach as it deals with the complexity of real-world educative contexts (in my case making-based coding workshops) and is grounded in theory. In addition, it fits with the long period of research characterized with continuous design, evaluation, and redesign of my interventions. Over two years I conducted three studies, representing the field studies of this thesis, called *cycles* or *iterations*, to evaluate and refine our making-based coding workshops with young students. I applied theoretically and pedagogically aligned tasks to investigate their effectiveness on young students' learning engagement, overall learning experience, and collaboration while developing an artifact. In this way, I also had the opportunity to conduct iterative and flexible revisions of the research design of my studies, applying research methods from both qualitative and quantitative research to get a deeper understanding and more holistic view of the phenomenon. The instruments I used for data collection include attitudinal questionnaires, knowledge acquisition tests, semi-structured interviews, eye-tracking, instructors' reflections, artifact collection, observations, and videos. All data have been respectively analyzed based on their type, founding our results in relevant literature and theory.

Another characteristic of DBR used in this dissertation is the detailed, comprehensive documentation of the whole process. This action helped me analyze data during field studies, especially the retrospective analysis, to contribute to theory and practice. For all four stages of the DBR (Figure 2), constant collaboration with other researchers, experts in the field, and instructors was an essential aspect of my research to improve the interventions' impacts, understand the learning experience processes, advance the initial designs, and provide theoretical and practical impact extracting design principles.

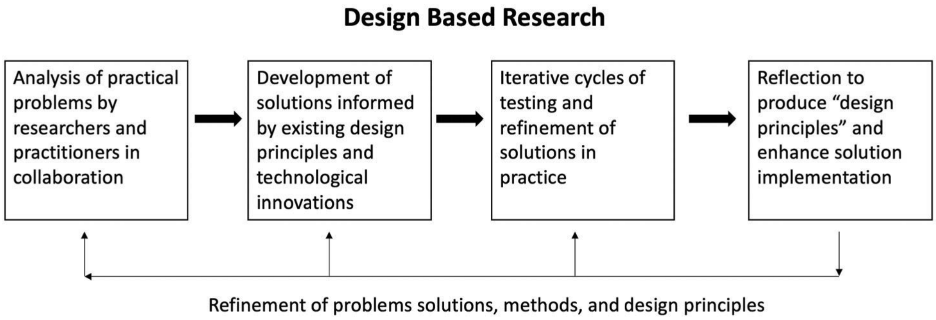

Figure 2: Design-based research description

1.4 Research contributions and papers

This thesis is based on eight papers that are published in international journals and peer-reviewed conference proceedings exploring the research questions. These papers' main contributions are:

C1: *Summarize and conceptualize the state of the art in making practices and their roles in enhancing coding activities for children*. This presents a systematic literature review, producing substantive findings regarding the *making* approach and its applications to coding activities. The review aims to show the state of the art and identify potential research gaps.

C2: *Investigate new methods to evaluate making-based coding activities for young students*. This includes using multimodal data in field studies as well as combining subjective data from various channels (interviews, surveys) with objective physiological data (eye-tracking) to understand the learning process of coding in a deeper way. This contribution refers to using the eye-tracking method to pervasively track the gaze of children of young ages in a coding activity.

C3: *Improved understanding of the learning process to support young students in making-based coding activities*. This includes the design, evaluation, and refinement of coding workshops, in which the field studies of this thesis took place, focusing on enhancing young students' engagement and attitudes during their experience.

C4: *Investigate the needs of different population of young students to support their learning experience*. This investigates the potential differences in how young students of different ages and genders handle the learning process during making-based coding activities.

C5: *Guidelines for designing making-based coding workshop for young students*. This sheds light on best practices in the design of coding activities for children based on *making*. This presents the principles that emerged, representing the knowledge gained from the two years of interventions and the comparative and retrospective analyses of the outcomes, also based on the literature.

C6: *Contribution to theory*. This includes the grounding of the results in the theoretical notions of constructionism with regard to the effects of making-based coding activities on young students' learning experience. Also, combines cognitive load theory and self-determination theory, providing evidence from the use of an objective data-collection method and going deeper into behavior.

The research questions are addressed in the following published research papers. Their connections to the sub-questions are presented in Table 1.

P1: Sofia Papavlasopoulou, Michail N. Giannakos, and Letizia Jaccheri. "*Empirical studies on the Maker Movement, a promising approach to learning: A literature review."* Entertainment Computing 18, (2017): 57-78. https://doi.org/10.1016/j.entcom.2016.09.002

P2: Sofia Papavlasopoulou, Kshitij Sharma, Michail N. Giannakos, and Letizia Jaccheri. "Using Eye-Tracking to Unveil Differences Between Kids and Teens in Coding Activities" In Proc. of Conference on Interaction Design and Children. 171-181. (2017) Stanford, California, US. ACM
http://dx.doi.org/10.1145/3078072.3079740

P3: Sofia Papavlasopoulou, Kshitij Sharma, Michail N. Giannakos. "*How Do You Feel about Learning to Code? Investigating the Effect of Children's Attitudes towards Coding Using Eye-Tracking"* International Journal of Child-Computer Interaction. 17, 50–60 (2018).
https://doi.org/10.1016/j.ijcci.2018.01.004

P4: Sofia Papavlasopoulou, Michail N. Giannakos, and Letizia Jaccheri. *"Discovering children's competences in coding through the analysis of Scratch projects."* In Proc. of IEEE Global Engineering Education Conference (EDUCON) 1127-1133. Tenerife, Spain (2018): IEEE. https://doi.org/10.1109/EDUCON.2018.8363356

P5: Sofia Papavlasopoulou, Michail N. Giannakos, and Letizia Jaccheri. *"Exploring children's learning experience in constructionism-based coding activities through design-based research"* Computers in Human Behavior, 99, 415-427 (2019). https://doi.org/10.1016/j.chb.2019.01.008

P6: Sofia Papavlasopoulou, Kshitij Sharma, and Michail Giannakos. *"Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences."* Computers in Human Behavior (2019). *in press*. https://doi.org/10.1016/j.chb.2019.03.003

P7: Kshitij Sharma, Sofia Papavlasopoulou, and Michail Giannakos. *Coding games and robots to enhance computational thinking: How collaboration and engagement moderate children's attitudes?* International Journal of Child-Computer Interaction, 21, 65-76 (2019). https://doi.org/10.1016/j.ijcci.2019.04.004

P8: Kshitij Sharma, Sofia Papavlasopoulou, and Michail Giannakos. *Joint Emotional State of Children and Perceived Collaborative Experience in Coding Activities"* In Proc. of Conference on Interaction Design and Children. 133-145, Boise, Idaho, US: ACM (2019). https://doi.org/10.1145/3311927.3323145

Table 1: Connections between research papers and sub-questions

	P1	P2	P3	P4	P5	P6	P7	P8
SQ1	•							
SQ2			•		•		•	•
SQ3		•			•	•		
SQ4		•	•	•	•	•	•	•

1.5 Structure of the thesis

This thesis is structured as follows.

Part I: Following the current introduction, Part I includes the background and theories, the methodology used, results, and contributions of the thesis. Those sections are covered in the following chapters:

Chapter 2: Presents the related work and the theoretical grounding of this PhD thesis.

Chapter 3: Shows the research methodology, including the field study settings and the process followed in this PhD work.

Chapter 4: Presents the results by describing the papers attached to this thesis.

Chapter 5: Discusses the results of the PhD thesis in respect to contributions as well as implications, limitations, and evaluation of the research.

Chapter 6: Includes the conclusion remarks and gives suggestions for future work.

Part II: Contains the collection of the eight research papers consisting this thesis in full length

2 Theoretical grounding and related work

This chapter has two goals. First, it presents theories grounding the research in this PhD thesis and second, it provides an overview of related research in the field, in line with the research questions.

2.1 Theoretical Grounding

Stemming from the goal of this PhD thesis presented in Chapter 1, the research conducted is grounded on three theories. First, constructionism theory gave the overall grounding of the studies. Constructionism is a theory of learning, teaching, and design that aligns well with the demands and expectations of computational culture. It emphasizes creating and making shared and meaningful artifacts as a means for gaining knowledge (Papert, 1980) (Papert, 1993). Previous research refers to constructionism as representing both a "theory" (Kafai & Burke, 2015) (Kafai, 2006; Parmaxi, Zaphiris, & Ioannou, 2016) and a "theoretical framework" (Hay & Barab, 2001). In this PhD thesis, constructionism is described as a theory. Our workshops were designed with the goal of creating an artifact, which in our case was a game using the Scratch programming tool. Students worked in teams to develop the artifact. Teaching assistants, as instructors, led the process and assisted the young students. Second, cognitive load theory (CLT) is used as lens through which the young students' cognitive processes were addressed during the coding workshops, using use eye-tracking measures as a proxy for cognition (Eckstein, Guerra-Carrillo, Singley, & Bunge, 2016) to investigate young students' cognitive processes in learning (Mayer, 2010). The coding workshops have an overall cognitive load for the young students, and their working memories can quickly be overloaded by complex tasks. Third, self-determination theory (SDT) allows the elaboration of the concept of attitudes and engagement of the young students during the coding workshops. Self-determination theory (SDT) has been widely used to understand motivation within educational contexts (Guay, Ratelle, & Chanal, 2008). It is centered on the belief that people's needs are the basis of self-motivation; the type of motivation is related to one's goals and attitudes, leading to actions.

2.1.1 Constructionism

Constructionism assumes that knowledge is better gained when young students are deeply and actively involved in building their own meaningful constructions. Based on Piaget (1954) theory, which focuses on how mental constructions are formed in someone's mind (Papert, 1980), constructionism focuses on explaining how construction is a valuable way to create mental constructions. The learners discover their own knowledge, rather than being passive receivers. Papert's constructionism sees the effectiveness of learning as achieved through making, where learners experience the active construction of visible-to-the-world artifacts. Computational culture supports the creation of building those artifacts using digital media and computer-based technologies (Kafai & Resnick, 2012). The vital aspect of constructionism is the requirement of "objects-to-think-with"– *"objects in which there is an intersection of cultural presence, embedded knowledge and the possibility for personal identification"* (Papert, 1980), p. 11). The role of this object in Papert's *Mindstorms* is the "turtle," a digital animal within the Logo programming environment, which can be controlled and moved by giving the appropriate commands. The "turtle" acts as a means to think, supporting and promoting a new way of thinking and learning.

Constructionism is not only valuable for the individual in building knowledge through experience and engagement in creating artifacts, but also for enhancing the social setting (Kafai, 2006). Like in the well-known samba school example, a social setting strengthens the sense of belonging to a group with a common purpose, where learning becomes important for all and connections are made under the learning culture (Papert, 1980). Along the same line, (Kafai & Burke, 2015) mention three dimensions of constructionism involved in making games for learning: personal, social, and cultural. More specifically, "personal" refers to learning and the attitudes related to learning, "social" refers to the collaborative aspects in creating a shared artifact, and "cultural" refers to how gender and race could influence the activity as well as the possible cultural aspects that could influence participation.

In the case of the field studies described in this PhD work; through the process of making computer games, young students plan and manage this complex development, placing themselves in control of their own leaning and thinking (Kafai & Kafai, 1995). Robertson and Howells (Robertson & Howells, 2008) argue that game design is a powerful learning activity that provides motivation, engagement, and enthusiasm. Constructionism's basic idea is that the most effective leaning experiences include active creation, socially meaningful artifacts, interaction with others, and the use of elements that support one's own learning and thinking. Game-making activities not only involve learning how to use technological tools but also using these tools to discover new ways of thinking. In such activities, children are introduced to a culture that permits them to become producers of their own artifacts while building their knowledge in a social context.

2.1.2 Cognitive Load Theory (CLT)

Cognitive load theory (CLT) implies that people have limited working memory; therefore, the amount of information they can process cannot exceed the limit at which they are overwhelmed (Paas, Renkl, & Sweller, 2004). There are three types of cognitive load: intrinsic, extraneous, and germane. Intrinsic load refers to the task and its core characteristics that must be processed. Extraneous load is based on the form of representation and instructional design techniques. Germane load involves information consolidation and refers to schema production for permanent knowledge.

The intrinsic load effort, in the case of the designed coding activities in this thesis, is represented by the performance of the task and its own load due to complexity. The use of the Scratch programming environment to complete the activity and the instructional details relate to the extraneous load. Finally, the germane load consists of the effort and processes from the task which are directed to relevant learning (Sweller, Van Merrienboer, & Paas, 1998).

Cognitive load can influence on visual attention and behavior. Therefore, cognitive load theory was used to support using eye-tracking measures. The eye's different fixations show the distribution of attention (Marcel A Just & Carpenter, 1980), while the cognitive process from graphic and textual visual materials is connected to fixation behavior (locus, duration and sequence) (Marcel Adam Just & Carpenter, 1976). In particular, eye-movement measures, such as number of fixations, fixation duration, duration time, and different scanning paths, can reveal important aspects of the learners' cognitive processes (Rayner, 1998). High fixation duration depicts high cognitive activity (Marcel Adam Just & Carpenter, 1976), and fewer saccades can be related to lower cognitive effort in terms of task performance (Eckstein et al., 2016). In a study about math and physics problems, participants had longer fixations on the more complicated parts of the problem (Hegarty & Just, 1993).

In the case of the field studies conducted for this PhD work, in line with CLT, the designed coding activity has an overall cognitive load that subsequently influences children's cognitive processes and can become overwhelming. We assume that yound students' working memories, especially among novices to coding, can quickly be overloaded by task complexity, and that this will lead to an inefficient learning environment. Thus, we use an eye-tracking technique as a proxy for cognition (Eckstein et al., 2016) to investigate children's cognitive processes in learning (Mayer, 2010) during our creative coding activity.

2.1.3 Self-Determination Theory (SDT)

Motivation is an important aspect of human behavior. Self-determination theory (SDT) has been widely used to understand motivation within educational contexts (Guay et al., 2008) and is centered on the belief that people's needs are the basis of self-motivation. There are three basic psychological needs that SDT supports: competence, autonomy, and relatedness. According to SDT, opportunities to satisfy any of these three needs contribute to people's motivations. The type of motivation is related to one's goals and attitudes, leading to actions. In addition, SDT includes two different types of motivation: intrinsic and extrinsic. When someone is intrinsically motivated, he/she is engaged in an activity, per se, for pleasure and satisfaction from its performance. On the other hand, extrinsic motivation refers to actions from outside sources, leading to separable outcomes (Ryan & Deci, 2000a, 2000b).

In my case, SDT presents a useful theoretical lens through which to view young students' experiences with creative coding activities for learning. In line with the theory, our coding activity is designed to have active participants and satisfy their needs for autonomy (with occasional support from the instructors), competence and relatedness, facilitating higher motivation. We argue that this activity provides intrinsic motivation, a tendency toward learning, and creativity leading to performance, as suggested by Vos, Van Der Meijden, and Denessen (2011). In our study, we provide a creative coding activity that encourages young students to make decisions, act independently, and work collaboratively with their peers. Hence, autonomy and competence are reinforced. Relatedness involves the development of satisfaction in the social context; therefore, we focus on attitudes that students have from and during their participation. In addition, on a given learning activity, motives are important to cognitive learning; the level of motivation influences focus and level of effort. More specifically, it could be argued that, by having the required motivations, young students gain the ability and energy required to sustain positive attitudes toward coding. In turn, positive attitudes facilitate cognitive processing and improve cognitive and affective outcomes. Therefore, I investigated the impact of our coding activity on students' attitudes (e.g., perceived leaning, excitement and intention to participate in a similar activity, team-work, enjoyment, and satisfaction from collaboration) and examined their connection to other measured variables.

2.2 Related work

In this section, I present how previous studies, relevant to the objectives of this PhD thesis, have addressed similar topics. It is essential to provide a brief overview of the-state-of-the-art research to ground the choices made in my studies, identifying challenges that are addressed through the research contributions of this PhD thesis. First, I show the benefits and the importance of making-based coding activities for young students. Those activities provide a fruitful learning environment in which children are stimulated to use a technological tool, affecting their learning experiences in different ways, particularly when the focus is on game-making. Second, I demonstrate how relevant studies have investigated young students' attitudes, motivation, and engagement; all are important issues, central in educational research for several years, related to the success of coding activities and their adoption. Third, I review the use of eye-tracking as a method that provides the researchers with direct access to users' attention patterns. It has been used in multiple educational settings to understand cognitive processes responsible for learning and collaboration among others.

2.2.1 Benefits of making-based coding activities for young students

The various technological tools available nowadays support learning activities based on construction and provide meaningful learning experiences for young students. Motivated by Papert's constructionist approach, today's educational activities are embedding technology tools to provide learning experiences in educational contexts, which occur in environments that are not always learning oriented. In these types of dynamic learning activities, students are at the center, taking control and engaging, at their own will, with a subject. Learning-by-doing, project-based learning, problem-based learning, inquiry-based learning, and challenge-based learning are a few such instructional methods, occurring both inside and outside the classroom (L. Johnson, Adams Becker, Estrada, & Freeman, 2015), and focused on learning tasks that promote computing education, computational thinking, design thinking, collaborative work, and innovation.

Computer game design and development, *making,* and computational textiles/fabrication are among the most successfully applied practices which help students develop coding skills and structure their own learning and thinking by getting involved in the coding process (Papavlasopoulou, Giannakos, et al., 2017) (Buechley et al., 2008). During such learning tasks, successful construction involves a complex process that fosters skills such as problem-solving, confronting "failures," and strategies to explore and decide possible solutions, as well as structure thoughts and actions (Bers et al., 2014). For instance, when children negotiate in the process of making an artifact in a supportive environment, they gain a sense of self-efficacy and belief in their capacities; they learn how to solve a problem, manage difficulties, and communicate with peers (Chu, Schlegel, Quek, Christy, & Chen, 2017); (Çakır, Gass, Foster, & Lee, 2017); (Bers, 2012). Generally, the skills gained in these educational contexts go beyond using a technological tool to make a game and computational thinking. These practices exist in constructionist learning and can be applied in subjects like math, language, arts, and others. The value is in the transferable skills uncovered through the experience of completing a successful project.

Many tools, such as Cricket, Braitenberg Blocks, and Arduino Technologies, can support fruitful learning experiences (Blikstein, 2013), while digital fabrication can provide Bildung (i.e., deep and sustained learning) (Iversen, Smith, Blikstein, Katterfeldt, & Read, 2016). Adams and Webster (2012) reported the results from nine years of coding summer camps for middle and high school students. Block-based visual programming languages (like Scratch) have the advantage of using shapes that fit properly only when they make a logical sequence of orders. This gives relief to users and saves them from much of the heartache traditionally forced on learners by textual languages (Wilson & Moffat, 2010), p. 70). However, even advanced text-based programming languages, like Java, have been used to engage children aged 9–10 in coding (Esper, Foster, Griswold, Herrera, & Snyder, 2014). By analyzing Scratch programs, they investigated the type of blocks students used and how aspects such as project types were related to their choice of blocks. The literature suggests that children can successfully complete and learn by simple, robot-based coding projects (Kazakoff, Sullivan, & Bers, 2013). Robots have the capacity to enhance coding activities and allow children to engage in computational thinking using various programming concepts (Kahn, 2004). In addition, digital game development was found to be beneficial for special education students, increasing their problem-solving skills through a process of representation, planning, execution, and evaluation of an artifact (Ruggiero & Green, 2017).

Regarding the potential effect of gender in coding activities, Bruckman, Jensen, and DeBonte (2002) showed that gender did not affect children's performance level in coding. Similarly, in a study of a game-development task for fourth-grade students, Owston, Wideman, Ronda, and Brown (2009) demonstrated that there were no gender differences in the learning outcomes. No significant gender differences were found in elementary school students' competence, interest at school, and the use of deep learning strategies while constructing a "drag and drop" game (Vos et al., 2011). Another study involving game-making showed that girls focused more on trying to improve their games following their peers' recommendations and that, overall, they achieved higher game quality (Robertson, 2012). In addition, in a study of the use of the code.org website to teach coding to primary school students, it was shown that girls' means of reflective thinking skills toward problems solving were higher than boys, although the results showed no statistically significant difference (Kalelioğlu, 2015).

In a nutshell, making-based coding activities, particularly when the focus is on game-making, provide a fruitful learning environment in which young students are stimulated to use a technological tool, affecting their learning experience. Therefore, there is a need to investigate and get a deep understanding of how we can help learners acquire knowledge, skills, and competences in coding in an engaging and meaningful manner.

2.2.1.1 Computational thinking

Computational thinking can be traced to Papert's strong support of the idea that children who use the Logo programming language develop algorithmic thinking (Papert, 1980). However, the term "computational thinking" was made popular by Wing (2006), who argued that "computational thinking represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" (p. 33). Since then, different efforts to define computational thinking have appeared, with the aim of supporting the importance of research on making computational thinking a 21st-century literacy accessible to all (Guzdial, 2008). Examples include the Computer Science Teachers Association and the International Society for Technology in Education framework (Barr & Stephenson, 2011), and the National Research Council's "Framework for K-12 Science Education" (NRC, 2012).

In our study, we adopted Brennan and Resnick (2012)'s computational thinking framework. With respect to Scratch, their framework suggests three key dimensions to computational thinking: computational concepts (concepts the users engage with when they program, such as parallelism and variables); computational practices (practices that users develop, such as abstraction and debugging); and computational perspectives (perspectives users develop for computation, themselves and the world around them). Brennan and Resnick's computational thinking framework enables the researcher to monitor coding activity and understand how children use the different constructs and deal with the concepts, how they focus on learning and adopt different thinking practices, and, finally, how their perspectives evolve in relation to themselves, others and the technological world. In our making-based coding activity using Scratch, these three dimensions were used to explore and gain insights into children's experience of coding.

2.2.1.2 Supporting and capturing young students' learning in making-based coding activities

Coding is not only a fundamental skill within computer science but is also a demonstration of computational competences (Grover & Pea, 2013) – a way to support computational thinking and develop students' high-order thinking skills. Kids as young as 4–6 can build and code simple robot-based projects and learn ideas from engineering, technology, and coding, thereby enhancing their computational thinking skills [Bers 2008]. Visual programming languages introduce the potential of a broader and younger group of students to learn programming concepts (Sáez-López, Román-González, & Vázquez-Cano, 2016). Various studies have proven the importance of combining coding and physical fabrication to engage students with complex programming concepts (e.g., loops, conditionals, events) and practices (e.g., remixing, testing, debugging) (Denner, Werner, & Ortiz, 2012) (Fields, Vasudevan, & Kafai, 2015) (Kafai & Burke, 2015) found that digital storytelling in a school setting demonstrates competence in several key programming concepts, such as event-driven programming and synchronization. In their project, an early-childhood robotics curriculum, called TangibleK, fostered multiple skills, including problem representation, systematic generation and implementation of solutions, debugging, and strategies to approach difficult problems. Denner et al. (2012) reported results from analyzing 108 games created by middle-school girls to show that it is feasible to learn programming concepts when designing and coding activities are seamlessly combined.

There is some variation in the ways students handle coding tasks and how they manage concepts and practices. For example, novices tend to approach programs in a line-by-line fashion, rather than in blocks (Robins, Rountree, & Rountree, 2003), and are not persistent when debugging their programs (Ericson & McKlin, 2012). In their study of middle-school girls, Denner et al. (2012) reported that students rarely used "variables" to handle coding processes and faced difficulty in joining pieces of code to successfully complete an operation. Students aged 11 to 12 made their own computer games using software called Adventure (Robertson, 2012); they spend the most time adding new content to their code, rather than changing what they had already done, and girls spent more time writing dialog for their games than did boys. The most popular practices used in projects by students of almost the same age–11–14–were reusing and remixing already existing code and addressing problems in an incremental and iterative way (Y. B. Kafai & V. Vasudevan, 2015). Kids aged 5–6 either carefully thought about and tried to predict results before trying the commands or tried different commands to receive immediate feedback (Fessakis et al., 2013).

Many studies have collected the actual code created in children's projects and then analyzed it using Brennan and Resnick's computational thinking framework (Brennan & Resnick, 2012), Bloom's modified taxonomy or solo taxonomy (Biggs & Collis, 2014), or other types of deductive coding schemes to evaluate the projects (Denner et al., 2012) and understand how children learn coding (Laporte & Zaman, 2016). "Fairy assessment," which is based on Alice's programming environment, requires students to modify and add existing code to assess their understanding of algorithm abstraction and code. Other ways of capturing children's progress and understanding include multiple-choice instruments or quizzes that measure their learning of computer science concepts, or even traditional assessments such as tests and grades (Doran, Boyce, Finkelstein, & Barnes, 2012).

Capturing computational thinking skills and the ways in which children learn coding is challenging, and more objective mechanisms are needed to illuminate children's understanding and knowledge gain of computational concepts and other computational thinking skills, such as debugging and problem decomposition (Grover, Cooper, & Pea, 2014). Assessments using coding blocks (akin to Parson's puzzles), where students have to snap them in correct order, are widely used in eBooks (Parsons & Haden, 2006). Assessments in which snippets of basic code are used to test whether children can identify the core constructs are widely used as well (Ericson & McKlin, 2012). Thus, the most common method to capture learning gain in computational thinking and coding is knowledge acquisition tests with combined types of questions (Grover et al., 2014).

2.2.2 Young students' attitudes and engagement toward coding activities

There are many benefits of educational activities in which children use technological tools and digital fabrication to make their own artifacts. They vary from learning programming concepts to behavioral and perceptual changes toward career paths in computing (Sáez-López et al., 2016) (Y. B. Kafai & V. Vasudevan, 2015); (Denner et al., 2012). Visual programming environments provide opportunities for children to be introduced to programming concepts; because of the fun and usefulness of the activity, children are highly motivated and have positive attitudes toward coding (Sáez-López et al., 2016). Motivation appears to be an important key in learning settings, not only for its positive results but also for its aspects of activation, intention (Ryan & Deci, 2000a), and promotion of active learning (Pintrich, 2003). Many studies have shown that students' motivations have an influence on their performance, satisfaction, and well-being (Ryan & Deci, 2000b) (Guay et al., 2008). In general, the aim is to have positive attitudes toward something that is interesting; consequently, the individual will take action according to interest and motivation (Fortus, 2014).

Concerning computing and computer science, students' attitudes and motivation are positive and high when projects and visual programming are involved, highlighting fun, commitment, enthusiasm, and usefulness (Sáez-López et al., 2016). Katterfeldt et al. (2015) conducted a EduWear/TechKreativ workshop, where the students used a smart construction kit that revealed a feeling of empowerment and attitudes that increased students' ability to code. Giannakos and Jaccheri (2018) found that children's positive attitudes regarding an activity's easiness and usefulness significantly affected engagement and their intention to participate. In particular, game-programming activities for children are motivating, support self-esteem, and foster computational thinking (Robertson & Howells, 2008). In collaborative learning activities, the level and quality of collaboration between young students has also been found to have direct influence in the quality of learning processes and persistence (Blunsdon, Reed, McNeil, & McEachern, 2003) as well as in improving students' attitudes (e.g., about mathematics) (Henrie, Halverson, & Graham, 2015). According to Vos et al. (2011), game programming reveals enthusiasm and motivation for learning and determination to accomplish a task.

Gender discrepancy in coding has been related to negative educational experiences in early childhood (Teague, 2002). CS careers still tend to be highly stereotyped, with girls having negative attitudes and being less likely to choose this career path. However, studies have found that both girls and boys who get involved in different kinds of software development practices show a better understanding of, and positive attitudes toward, CS (Bonner & Dorneich, 2016); (Eordanidis, Gee, & Carmichael, 2017); (Robertson, 2013). Scaffolding examples can help girls' engagement and confidence when using a programming environment. Studies specifically focusing on girls have found that game design experiences intended to enhance computational skills affect their perceptions in seeing themselves as able to design computer games and encouraging them to pursue careers in CS-related professions (Stewart-Gardiner, Carmichael, Latham, Lozano, & Greene, 2013). In a study involving middle-school girls creating games, (Denner et al., 2012) found that they were engaged in the process and demonstrated adequate levels of complex programming activity. Thus, designing appropriate activities can be a promising approach to attracting and encouraging girls' interest, ensuring positive attitudes in computing.

To summarize, from the literature it is evident that attitudes are highly associated with the adoption of a learning activity by young students, as well as the learning outcome. Hence, when investigating coding activities for young students, it's important to look into attitudes, motivation, and engagement, which are essential aspects of the design and implementation of successful coding activities.

2.2.3 Eye-tracking in cognitive process of coding

One of the technologies for studying cognitive processes in a deep and subjective way is eye-tracking. Eye movements are strongly related to cognition (Marcel Adam Just & Carpenter, 1984; Rayner, 1998) and have been used to investigate learning (Jarodzka, Scheiter, Gerjets, & Van Gog, 2010), reading [Rayner, 2006] and problem-solving (Tsai, Hou, Lai, Liu, & Yang, 2012). In addition, several studies use eye-tracking data to examine adult programmers' visual attention and explore coding, program comprehension (Aschwanden & Crosby, 2006) (Bednarik & Tukiainen, 2006), and debugging (Bednarik, 2012). The use of different visual attention measures, such as fixations, saccades, or time spent on parts of the screen called Areas of Interest (AOI), can give insights into complex cognition activities.

Romero, Lutz, Cox, and du Boulay (2002) compared the use of different program representation modalities (propositional and diagrammatic) in an expert versus novice debugging study, where experts had a more balanced shift of focus among the different modalities than did the novices. Sharif, Falcone, and Maletic (2012) emphasized the importance of code scan time in a debugging task and concluded that experts perform better and have a shorter code scan time compared to non-experts. Hejmady and Narayanan (2012) compared the gaze shift between different AOIs in a debugging exercise. The authors concluded that good debuggers were switching between code and the expression evaluation and variable window, rather than code and the control structure and data structure window. In another study, Aschwanden and Crosby (2006) defined each line of the code as an AOI and detected how these lines were perceived. Bednarik, Myller, Sutinen, and Tukiainen (2006) related the information types posited by Good and Brna (2004) to the gaze among the four AOIs (Code, Output, Control Panel, and Animation of program). The authors concluded that the presence of information type (e.g., high-level or low-level) in the comprehension summary does not necessarily confirm that that the target program has been comprehended correctly.

Eye-tracking has been widely used to measure collaboration in different dual eye-tracking experiments. In the domain of coding activities, Pietinen, Bednarik, and Tukiainen (2010) provided a new metric to measure joint visual attention in a co-located pair-programming setup, using the number of overlapping fixations, and the fixation duration of overlapping fixation, to assess collaboration quality. In a study by (Pietinen, Bednarik, Glotova, Tenhunen, & Tukiainen, 2008), a possible design for an eye-tracking setup was presented for co-located pair programming, and some of the problems regarding setup, calibration, data collection, validity, and analysis were outlined. In a collaborative task of finding bugs in a program, Stein and Brennan (2004) showed that the pairs who had their gaze displayed to their partners took less time to find the bugs than those pairs who had no information about their partners' gaze.

3 Research Methodology

This chapter aims to present the research methodology adopted during the work described in this thesis. First, I present the DBR approach followed. Then, for each field study, representing the *cycles or iterations* of the DBR, I present the methods used and the design of the coding workshops for young students.

3.1 Design-based research approach

Considering the research context of this thesis, focusing on investigate making-based coding activities for young students in informal learning environments, I chose Design-Based Research (DBR) as the methodological approach of my thesis (Reeves, 2006). The DBR approach advances design, research, and practice concurrently, allowing evidence-based improvements compared to simple observation. Many educational contexts have used DBR as it is a systematic, but agile, methodology that supports understanding the learning processes through design, exploration, enactment, evaluation, and redesign (Anderson, 2005; Anderson & Shattuck, 2012) (Wang & Hannafin, 2005). More precisely, Wang and Hannafin (2005, p. 6) define DBR as: "systematic but flexible methodology aimed to improve educational practices through iterative analysis, design, development, and implementation, based on collaboration among researchers and practitioners in real-world settings, and leading to contextually sensitive design principles and theories." DBR does not replace other methodologies, but it is often defined as a series of methodologies building on the use of multiple procedures and methods from both design and research methodologies (Wang & Hannafin, 2005). The processes that DBR follows are also flexible; researchers, participants, and practitioners collaborate to improve the initial design plan and influence practice through implementation. The knowledge gained throughout the different phases of the study emerges and evolves from the research questions and the factors that influence them during the process. In my case, the DBR methodology was appropriate, as the factors that influence the research questions emerged and evolved, during the process, from the knowledge gained by the researchers throughout the different phases of the study. Another purpose of DBR is to influence real educative interventions and has a theoretical goal, evolving theoretical principles, that support the research work (Barab & Squire, 2004).

The work presented in this thesis complies with the DBR characteristics described below:

- **Iterative, interactive and flexible process:** Following the DBR approach, within the timeframe of the development of the thesis, I conducted three iterations (cycles) (Figure 3) aiming to answer the set of research questions related to the thesis' goals. After each iteration, based on the results obtained from evaluating the learning experiences of the young students in the making-based coding workshops, the next iteration's design was revised together with the goals, research questions, and the next actions of the research.

- **Research in real educational contexts**: Our iterations fit properly with DBR as they happened in the real-world contexts of making-based coding workshops organized for students in the Trondheim region, representing an authentic informal educational setting. The studies, held between Autumn 2016 and Autumn 2017, involved data collected from 157 young students from 3rd to 12th grade (8–17 years old). Situating the research in real educational contexts confirms and ensures that the applied proposals can be used effectively to assess, inform, and improve practice in (at least) those contexts.
- **Collaboration between researchers and practitioners:** In a DBR process, the constant collaboration with other researchers, experts in the field, and instructors is essential. In-depth understanding of these studies requires both the researchers' and the instructors' knowledge. Researchers conduct the rigorous research, and the instructors contribute with their knowledge of the local setting, their pedagogical experience, and interaction with the participants. With the constant collaboration, I managed to monitor all the actions during enactment, provide evidence on the design decisions, and support any emergent proposals.
- **The use of mixed methods:** Mixed research methods are used to maximize the credibility of ongoing research. The methods used vary during different phases as new needs and issues emerge and the research focus evolves. This approach is generally considered an adequate way of exploring the different perspectives and multiple factors that affect learning situations, and it is typically used in DBR (Anderson & Shattuck, 2012). In my case, for my field studies, I used quantitative and qualitative techniques for data collection and analysis; this includes using multimodal data from various channels, which are linked to subjective and objective physiological data. Data were gathered from instructors, young students, and researchers to obtain information about the learning experience provided from our workshops. We used qualitative sources, such as observations, semi-structured interviews and focus groups, students' artifacts, and quantitative sources, such as questionnaires, gaze from eye-tracking, and videos. All data were checked and triangulated.
- **Actual impact on theory and practice:** During the DBR process followed in this thesis, the design, implementation, and evaluation of the interventions are grounded in relevant research, theory, and practice. The aim is to connect the results with the design process and the setting, to refine and improve initial designs, and, ultimately, advance theoretical aims affecting practice. This thesis investigates young students' learning experiences, understanding their needs and practices, and, respectively, design the making-based coding workshops. Based on the lessons learned, the theory's value is appraised by the extent to which principles inform and improve practice.

Starting with DBR, a systematic literature review was conducted and, each time before implementing the field studies, the relevant literature on the respective topics was read to support the research.

For the rest of this chapter, the details of the field studies and their implementation are presented.

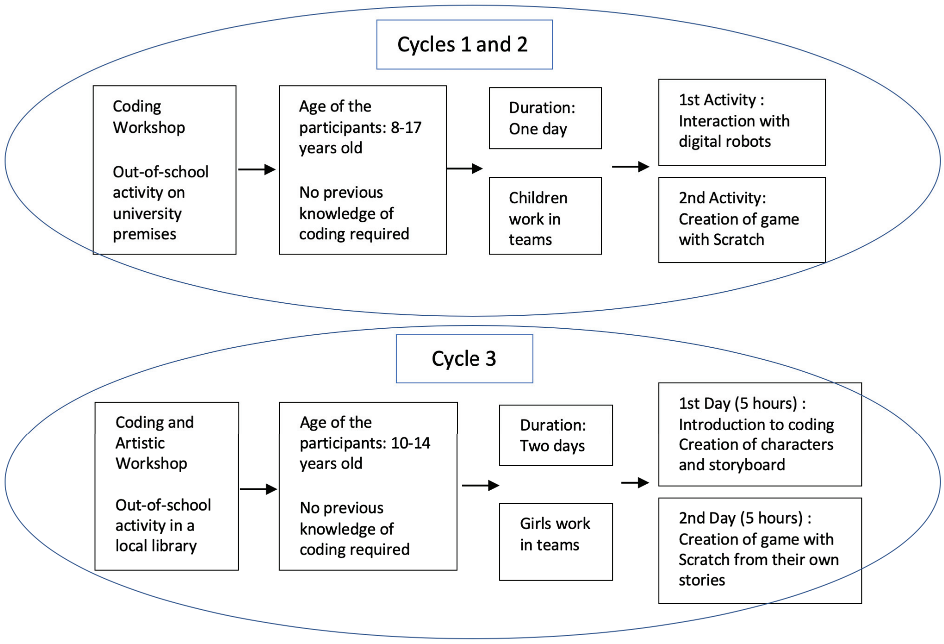

Figure 3: Description of the three DBR Cycles

3.2 Studies design

As field studies of the PhD work presented in this thesis, I selected making-based coding workshops designed and organized in conjunction with "Kodeløypa" (meaning "*the path toward coding*") which is one of the six established science frameworks at NTNU. "Kodeløypa" describes outreach activities offered by the Computer Science Department to all school classes in the Trondheim region (targeting lower secondary schools and mainly 10th grade classes) as out-of-school activities. Those activities include making-based coding workshops using open source software (i.e., Scratch), physical recycled materials, and hardware (i.e., Arduino).

Building on previous efforts and research projects starting from 2012, Kodeløypa has always been influenced by research, having the NTNU's years of conceptual methodological and technical knowledge and experience, conducting field studies following the state-of-the art developments based on societal needs. Kodeløypa workshops are a representative sample of making-based coding activities, following the maker movement approach and rise of coding activities for young students in Europe, in response to international actions to foster computer science education for students from a young age. The workshops' activities focus on the learning processes that occur in an environment that is not merely learning oriented, but promotes design thinking, computational concepts, collaborative work, and innovation, among other things. In particular, it involves making activities and related ways to fabricate real and/or digital products using technological resources, including fabrication, physical computing, and coding.

A characteristic that is also an important aspect of Kodeløypa's workshops is the pleasant and playful environment, supported not only by the activities per se, but also from the instructors' attitude. Targeting young students that have or do not have any previous exposure to coding, offering a making-based-coding workshop for young students invites them to explore and "play" with computers and digital robots to explore and construct their own knowledge. In this way, they gain also a feeling of mastery and pleasure from technology. In addition, introducing concepts and thinking in computer science presents an opportunity to 'plant a seed' of interest that can later develop into increased recruitment to computing.

Although, efforts like coding clubs and different kinds of science learning activities in Norwegian museums started to emerge slowly some years ago, in general, coding for children is still a new phenomenon in formal education. A revision of the national curriculum is currently underway, and the Ministry of Education has published a strategy that began in 2017, offering coding as an elective subject in lower secondary school (Ungdomsskole, Grades 8–10, ages 13–16), which will become a permanent elective subject beginning in 2019.

Among the various informal science learning spaces and practices, much attention has been given to experiences and activities traditionally associated with science museums and centers, zoos, exhibitions, competitions, etc. However, the increasing emergence and proliferation of practices emphasizing making-based-coding activities in a fun and creative way, representing informal learning, have not yet drawn enough attention, while appearing to be one of the new 'big things' in the field. Therefore, based on the goal of this PhD work, to investigate how making can help us design meaningful coding activities for young students, and implementing interventions with refined and improved designs than influence practice, I have selected the aforementioned Kodeløypa coding workshops for the field studies of my research. Below, I describe all details of the design and implementation of the making-based coding workshops.

3.2.1 Description of the coding workshops

During the years of this PhD work, I have conducted three field studies which comprise the three cycles (iterations) as part of the DBR approach followed. For the purpose of this PhD thesis, I refer to all the workshops conducted using the name "Kodeløypa." However, the first two cycles (iterations) of coding workshops took place over two years, with few differences in activity design but with differences in the research design, evidence descriptions, and results of the different data-collection instruments. The third cycle (iteration) was in collaboration with the local library in Trondheim.

The goal of the workshops for young students is to create an artifact, which in our case is a game using the Scratch programming tool. Teaching assistants, specifically trained as activity instructors, led the process and supported the students in achieving their goals. During the workshop, young students were working in teams to develop the artifact. The workshops were designed for children without (or with minimum) previous experience in coding. The design of the activities (interacting with robots and creating games), and the use of Scratch programming language (suitable for all ages) provided flexibility and allowed successful implementation of the workshop. Student participants' ages ranged from 8 to 17 years. Each workshop had a specific age group that were carefully selected within a small age range.

The most influential aspect of our pedagogical approach was what Resnick calls the "kindergarten approach to learning," with a spiral cycle of imagine, create, play, share, and reflect–a process that is repeated over and over (Resnick, 2007). Young students imagine what they want to do and then create a project based on their ideas, play/interact with their own creations, share their creations with others, and reflect on their experiences, leading to new ideas and projects. Adapting Resnick's spiral, our workshops also started with "inspire" to characterize the warming-up and inspiring activities that kicked off the participants' creativity. In addition, to describe the coding process, particularly the use of the Scratch tool, we focused on constant experimentation and iteration: the children developed their artifacts gradually by trying new elements, using different concepts, and revising them.

3.2.2 Cycle 1 and 2

We designed and implemented a coding activity in conjunction with an initiative organized at NTNU in Trondheim. The workshop activities are based on the constructionist approach, following the main principles of *making*. The workshop was conducted in a largely informal setting, as an out-of-school activity, and lasted for four hours. Various student groups, ranging from 8 to 17 years old, were invited to NTNU's specially designed rooms for creative purposes to interact with digital robots and create games using Scratch and the Arduino hardware platform. Specifically, Arduino was attached to the digital robots to connect them with the computer. At that point, an extension of Scratch, called Scratch for Arduino (S4A), provided the extra blocks needed to control the robots. The Scratch programming language uses colorful blocks grouped into categories (motion, looks, sound, pen, control, sensing, operators, and variables), with which children can develop stories, games, and any type of animation. In general, children who attended the workshop worked collaboratively in triads or dyads (depending on the number of children). The workshop was designed for children without (or with minimum) previous experience in coding. During the workshop, student assistants were the responsible supporting each team as needed. Approximately one assistant observed and helped one or two teams. Three researchers were also present throughout the intervention, focusing on observing, writing notes, keeping notes, and taking care of the overall conduct of the workshop. The workshop had two main sections (Figure 4).

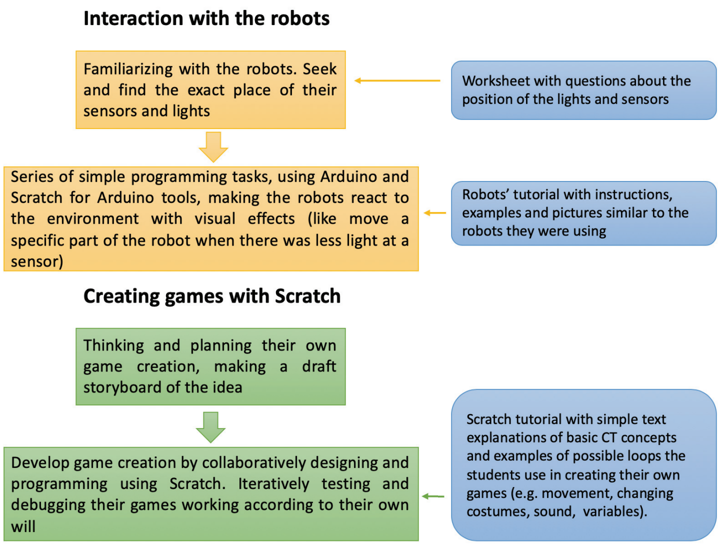

Figure 4: Description of the two activities in the workshop

Interacting with the robots: In the first section, the children interacted with digital robots made from an artist (using recycling materials). Each robot was placed next to a computer (one for each team) (Figure 7 left). When the children entered the room, one assistant welcomed them, told them to be seated, and briefly presented an overview of the workshop. The assistants then advised the children to pay attention to the paper tutorial and the worksheet placed on the desks (one for each student). First, the children filled in the worksheet to answer questions regarding the exact place and the number of sensors and lights on the robots. The tutorial contained instructions with examples and pictures (Figure 5), similar to the robots they were using. The examples had little text and more images, indicating exactly how the children could interact with the robots. The children accomplished a series of simple loops that controlled the robots and made them react to the environment with visual effects (such as turning on a light when sensors detected that the light was below a certain threshold). Children could touch and play with the robots but not change any parts of them. Although the duration of the session was different for each team, it lasted between 45 minutes and 90 minutes and ended with a break before the next session.

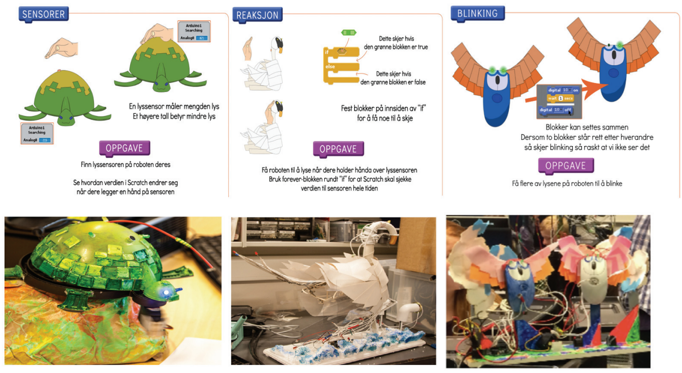

Figure 5: Example of the robots and the instructions

Creating games using Scratch: This session focused on the creative implementation of simple game-development concepts using Scratch (Figure 7 right). All children took another paper-based tutorial containing examples and visualizations to help them ideate their own game (Figure 6). The tutorial comprised simple text explanations, including basic computational thinking concepts and possible loops that the children were supposed to use in their own games. First, the assistants advised the children to concentrate on understanding the game's idea, discuss it with their team members, and then create a draft storyboard. The children then developed their own game by collaboratively designing and coding using Scratch. To accelerate the children's progress, they were given already existing game characters and easy loops. While the children worked on their projects, help was provided whenever they asked for it, and complex programming concepts were introduced on an individual level according to their relevance to the project. Children created their games, step-by-step, by iteratively coding and testing them (Figure 7 middle). After completing the games, all teams reflected and played each other's games. This session lasted approximately three hours.

Figure 6: Example of a game in Scratch and the instructions

Figure 7: The setting with the computer and the robot (left), children collaborating on the creation of their game (middle), example of a created game

3.2.3 Cycle 3

We designed and implemented a two-day workshop in conjunction with the local library in Trondheim, Norway. The workshop activities focused on coding, including artistic elements, and were based on the constructionist approach. The call for participation was made to middle-school girls of the Trondheim region during the autumn 2017 school break. Previous experience was not a prerequisite for participation. Each day's activities were conducted in an informal setting and lasted for approximately five hours, including breaks. Female instructors, with previous experience in similar activities (also involved in Kodeløypa), facilitated the workshop and were responsible for supporting the girls during the process. During the workshop, the girls had to create storyboards based on solving particular environmental problems and then, based on their stories, create games using the Scratch programming language. To develop the storyboards, the girls could use different types of materials, like ribbons, colored cardboard, stickers, drawing pencils, etc., as provided by the library. The girls worked collaboratively in teams of two or three (depending on the number of participants). Two researchers were present for the duration of the workshop, assisting when needed for the smooth execution of the activities, including the data collection. The workshop is described below, based on the two days of activities (Figure 8).

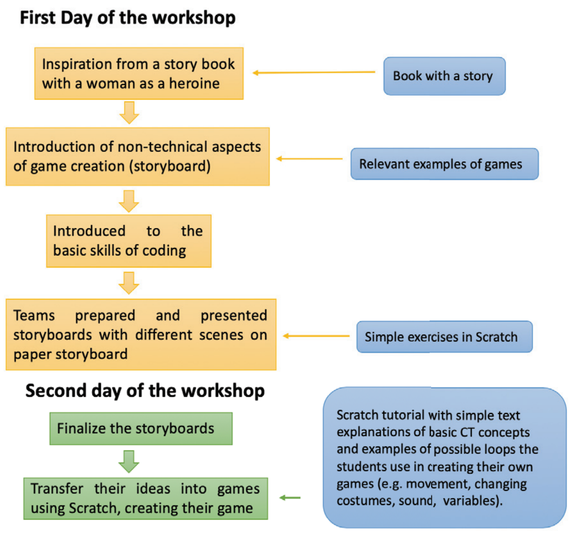

Figure 8: Description of the activities in the two days' workshop

First day of the workshop: On the first day, we introduced basic coding skills and other non-technical aspects of game development, like storyboard creation. The workshop started with a story from a book, based on a woman with children and everyday problems, who was also a mentor and a superhero helping people succeed with their technology projects. The girls were inspired and informed that they had to think of their own characters who needed to save the world from environmental issues of their choosing. As an introduction to coding, the instructors presented an example of a functional game with Scratch on a relevant environmental topic (Figure 9 left). Then, the girls were asked to individually complete basic coding exercises using Scratch. At the end of the first day, the teams prepared and presented their storyboards with three different scenes on paper/cardboard, including the title, theme, character, plot, conflict, and solution.

Second day of the workshop: Starting the second day of the workshop, the girls had to update, if they wanted, their storyboards and finalize them (Figure 9 middle). The rest of the day was dedicated to their game creation using Scratch. The girls completed a paper-based tutorial, created by the instructors, with simple text explanations and examples of basic CT concepts and possible loops that the girls were supposed to use in their own games, all based on Scratch. During the creation of their games, the girls had to use their storyboards exactly and "transfer" their ideas into games using Scratch. At any time, the girls could ask for help from the instructors, who even introduced complicated programming concepts, if necessary, for their games. The girls created their games, step-by-step, continuously testing and coding them. At the end of the day, all teams prepared presentations of their games and everyone played each other's games (Figure 9 right).

Figure 9: Girls participating in the workshop (left), creation of the storyboard (middle), game created using Scratch

3.3 Sampling

All the studies participants for the three iterations (cycles) were students from the Trondheim region whose teachers had applied to participate in our workshops as an out-of-school activity. Two of the field studies took place at the university campus in specially designed rooms and the last one in the local library. All data related to the studies were collected after permission from the National Centre for Research Data (NSD), following all the regulations and recommendations for research with children. Before the execution of the workshops, I contacted the teacher and/or the legal guardian of each child to get written consent for the data collection. The children were informed about the data-collection process and their participation in the study was completely voluntary. They could withdraw their consent for the data collection at any time without affecting their participation in the coding activity.

3.3.1 Participants cycle 1

Children from 3rd to 12th grade (aged 8–17 years old) participated in the coding activity. The activity took place during autumn 2016 with a sample of 12 girls (mean age: 12.64, SD: 2.838) and 32 boys (mean age: 12.35, SD: 2.773). Five workshops took place over two weeks, following the coding activity described in the previous section.

3.3.2 Participants cycle 2

In autumn 2017, children from 8th to 10th grade (aged 13–16 years old) participated in the coding activity. The sample consisted of 105 participants in total, 69 boys and 36 girls (mean age: 14.55, SD: 0.650). Kodeløypa workshops were conducted every Friday for six weeks.

3.3.3 Participants cycle 3

The sample of the third study consisted of eight girls from 6th to 10th grade (aged 10–14 years old) (mean age: 12.135, SD: 1.389). Girls participated in the two-day workshop during autumn 2017, following all the activities of the workshop described in the previous section at the local library.

3.4 Data collection

In this section, I describe the data-collection process for each study in respect of its cycle.

In all stages of the methodology a literature review was performed, and I collected data from different sources following a mixed-method methodology (e.g., questionnaires, interviews, gaze, observations, and field notes) using quantitative and qualitative methods. Mixed-method enables joint analysis and triangulation of quantitative and qualitative data (Creswell & Clark, 2017). In our case, all quantitative and qualitative data were triangulated and cross-referenced to warrant our interpretations during the three DBR cycles. Mixed methods provide benefits, such as a detailed and descriptive view of the situation and data interpretation with divergent views, and supports research conducted in authentic contexts (Creswell & Clark, 2017). *"Combining or integration of qualitative and quantitative research and data in a research study (…) resided in the idea that all methods had bias and weaknesses, and the collection of both qualitative and quantitative data neutralized the weaknesses"* (Creswell, 2014), 563)

Table 2 describes the main instruments used for gathering data during this thesis. In particular, we collected quantitative data in cycle 1. Combining different types of data provided us with a set of indicators for reflecting different mechanisms during the learning process. This allowed us to better understand young students' learning experiences and get another insight into using objective methods, like eye-tracking, to investigate coding activities with young students. In Cycles 2 and 3, we applied a qualitative approach. We collected data from multiple sources, including post-workshop interviews, observational field notes, and participants' Scratch games (artifacts), and performed content analysis.

Apart from data collection from each field study during the two years of the project, the researchers and instructors participated in focus groups after the end of each cycle, discussing and revealing all the growing ideas that emerged from the outcomes of the iteration. This was another data collection and analysis process, which on the one hand helped us to reveal key findings that guided the decision to proceed to the next iteration, and on the other, to synthesize the ideas, conduct an overall evaluation of our coding workshops focusing on young students' engagement from the learning experience with coding, and provide guidelines for designing similar activities.

Table 2 presents an overview of the field studies conducted for this PhD research, the data-collection instrument used, the aim, the cycle, and the relevant paper.

Table 2: Overview of the field studies, data collection, and aim

Field study	**Data collection instrument**	**Aim**	**Cycle**	**Related paper**
Autumn 2016	-Eye tracking -Attitudinal questionnaire -Knowledge acquisition test (pre and post) -Artifact collection -Focus group with researchers and instructors	- Gain understanding on: • How young students code and investigate any potential association between their attitudes and their gaze. • Young student's differences in their coding behavior. • Young students' competences in coding - Identify: • Design elements for the next iteration • Prominent design principles • The most important theoretical aspects - Use the objective method of eye-tracking for new insights on how young students experience coding	1	- P2 - P3 - P4 - P5 - P6 - P7
Autumn 2017	-Semi-structured interviews -Field notes from observations -Artifact collection	- Examine differences between boys and girls coding behavior - Investigate the emotional profile of young students and the quality of collaboration - Identify:	2	- P5 - P6 - P8

	-Focus group with researchers and instructors - Video recordings	• Design elements for the next iteration • Prominent design principles • The most important theoretical aspects		
Autumn 2017	-Semi-structured interviews -Field notes from observations -Artifact collection -Focus group with researchers and instructors	- Investigate girls' strategies and implemented practices during coding - Identify: • Design elements for the next iteration • Prominent design principles	3	-P5

3.4.1 Cycle 1

Attitudinal questionnaire: Young students, as participants of the coding workshop, completed a paper-based attitudinal questionnaire. Participants were asked to rate their experience with the coding activity after the end of the workshop. Attitudes include their perceived learning, excitement, and intent to participate in a similar activity, among others. In all measures, a five-point Likert scale was applied using smiley faces (Figure 10).

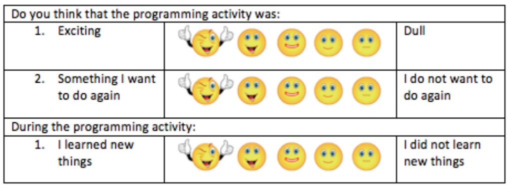

Figure 10: Example of the attitudinal questionnaire

Knowledge acquisition test: The children completed pre- and post-knowledge acquisition tests. These consisted of nine questions of increasing difficulty, based on the workshop's curricula. The questions were adopted from a previous study (Grover et al., 2014) and followed instructors' suggestions for our workshop. The tests took approximately 10 minutes to finish at the start and end of the workshop. The tests were paper-based and manually graded by the researcher. Figure 11 shows two sample questions from the test.

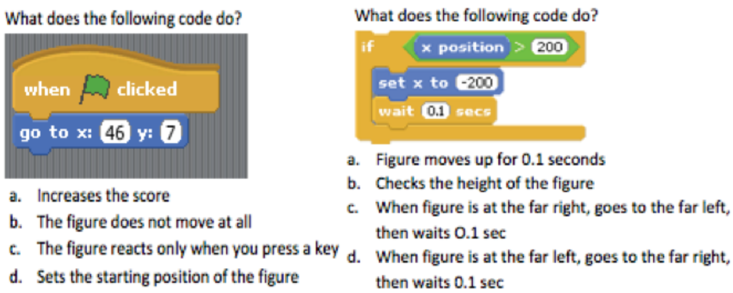

Figure 11: Example of the knowledge acquisition test

Eye-tracking data: During the whole activity, children were wearing eye-tracking glasses. Eye-tracking data were collected using four SMI RED 250 and one TOBII mobile eye-tracker working at 60Hz.

Artifacts and instructors' reflections: Final artifacts from each of the teams were collected after the end of each workshop. In addition, one month after completing the workshops, I conducted a group interview and focus group with the assistants, reflecting on their experience and perspectives for the young students' experience during the workshops.

3.4.2 Cycle 2 and 3

Interviews: Participants completed a demographics questionnaire. After the end of the activity, post-workshop semi-structured interviews were conducted individually with the young students. During the interviews, children were asked about their overall experience, the artifact constriction process, Scratch, collaboration with their team members and the assistants, as well as other subjects that emerged depending on the interviewee. The interviews were conducted in Norwegian, so all the participants could express themselves clearly. Each interview lasted approximately 10 to 15 mins and was audio recorded.

Observations and Artifacts: Artifacts from most teams were collected four times during the workshop session; we collected four versions of the games approximately every hour during the workshop. Some teams were also systematically observed during the process of creating the game. Independent assistants during the workshop kept field notes. Assistants were close to each team of young students and took notes on all tasks to identify what types of help participants were receiving from the instructors. For the systematic observations, we collected field notes, using Franklin et al. (2013) 's coding scheme, of what types of help the participants were receiving from the assistants (see table 3). Also, field notes included other incidents that occurred during the workshop concerning the process the participants followed to successfully complete the tasks, such as whether they had a team leader, if only one was mainly coding, if one member was not participating, and others.

Table 3: The coding scheme for the observations

Number	Explanation
0	**Validation:** Students want confirmation, not information
1	**Where:** Only needed help navigating the Scratch GUI
2	**What:** Only needed a reminder of the name of the concept
3	**How:** Given name of concept, still needed help to complete task
4	**Reteach:** Had to reteach concept and execution

Instructors' reflections: To capture instructors' perspectives and experiences, as described in the previous cycle, we conducted group interviews and focus groups.

3.5 Measurements and data analysis

A thorough description of measurements and data analysis is described in detail in each paper included in this thesis. Below, I present the most important aspects of the data analysis methods used. Table 4 presents a summary of the data analysis method applied, together with the paper, SQ and contribution.

Table 4: Overview of data collection and analysis

Paper and aim	Selected data	Cycle	Data Analysis	Research question	Contribution
Paper 1: - Systematic literature review	- 43 articles	-	- Categorization of their main elements	SQ1	C1
Paper 2:					

- Investigate the role of gaze in the learning gain and collaboration on the different age groups (teens and kids)	- 44 Knowledge acquisition test responses - Gaze from 44 young students captured from mobile eye-trackers	1	-Statistical analysis using ANOVA tests - Statistical analysis using Pearson's correlation	SQ3 SQ4	C2 C4 C5
Paper 3: - Investigate any potential association between children's attitudes and their gaze.	- 44 Attitudinal questionnaire responses - Gaze from 44 young students captured from mobile eye-trackers	1	-Statistical analysis using ANOVA tests	SQ2 SQ4	C2 C3 C5 C6
Paper 4: - This paper examines, young students' competences in coding	- 9 Artifacts (final games)	1	-Statistical analysis using Pearson's correlation	SQ4	C5
Paper 5: - Theoretical grounding of the results in constructionism - Design principles to achieve higher engagement during the coding activity	- Reflections as notes from the studies conducted derived from focus groups with instructors and researchers	1-2-3	- Qualitative content analysis	SQ2 SQ3 SQ4	C2 C3 C5 C6
Paper 6: - Investigate differences between girls' and boys' gaze	- Gaze from 44 young students captured from	1-2		SQ3 SQ4	C2 C3

and learning gain during the coding activity - Investigate differences between girls and boys in the strategies and implemented practices during coding, and in perceptions about those coding activities	mobile eye-trackers - 44 Knowledge acquisition test responses - 44 Interviews audio recorded -Observation field notes from 4 teams -Artifacts (games) from 4 teams		-Statistical analysis using ANOVA tests - Qualitative content analysis		C4 C5
Paper 7: - Investigate if there is a significant moderating effect of collaboration (gaze similarity) and engagement (gaze uniformity) on children's attitude	- Gaze from 44 young students captured from mobile eye-tracker - 44 Attitudinal questionnaire responses	1	-Statistical analysis using Pearson's correlation -Statistical analysis using Moderation	SQ2 SQ4	C2 C3 C5
Paper 8: - Define the emotional profile of a collaborating group of young students - Investigate the relationship between the emotional profile of a group and the quality of collaboration during coding - Identify the components of the emotional profile that affect the quality of	- Videos from 50 young students - 50 Attitudinal questionnaire responses	2	-Statistical analysis using Pearson's correlation -Statistical analysis Linear Regression	SQ2 SQ4	C3 C5

collaboration during learning					

Figure 12 shows the variables used for the quantitative data collected from the different cycles. For the quantitative data analysis, after defining the measurements, we used the statistical analysis of variance (ANOVA) to identify the relationship between the variables, Pearson Correlation to identify any potential correlation between the variables, Moderation, and Regression. These methods are in respect to each paper's research question. During the analysis, various different tests were performed. For example, we checked the assumptions for ANOVA, and if we found variables that did not satisfy the homoscedasticity condition, a version of ANOVA was used for which homoscedasticity is not assumed. This was done using the Welch correction for F-statistic. First, we used Levene's test to examine the homogeneity of variance and the Shapiro–Wilk test to evaluate the normality criterion (Conover & Conover, 1980) (Shapiro & Wilk, 1965) 1965) in order to use ANOVA analysis. Afterwards, one-way independent ANOVA tests were conducted to examine our research question. For all tests, we did not assume equality of variance across groups. The p-values for the main and post-hoc tests are computed in accordance with the Bonferroni correction for repeated tests. Also, since we did not assume equal variance across groups, the F-values are adjusted according to the Welch correction for the partial degrees of freedom. In addition, to analyze the video recordings, we used computer vision for face detection and machine learning for face recognition.

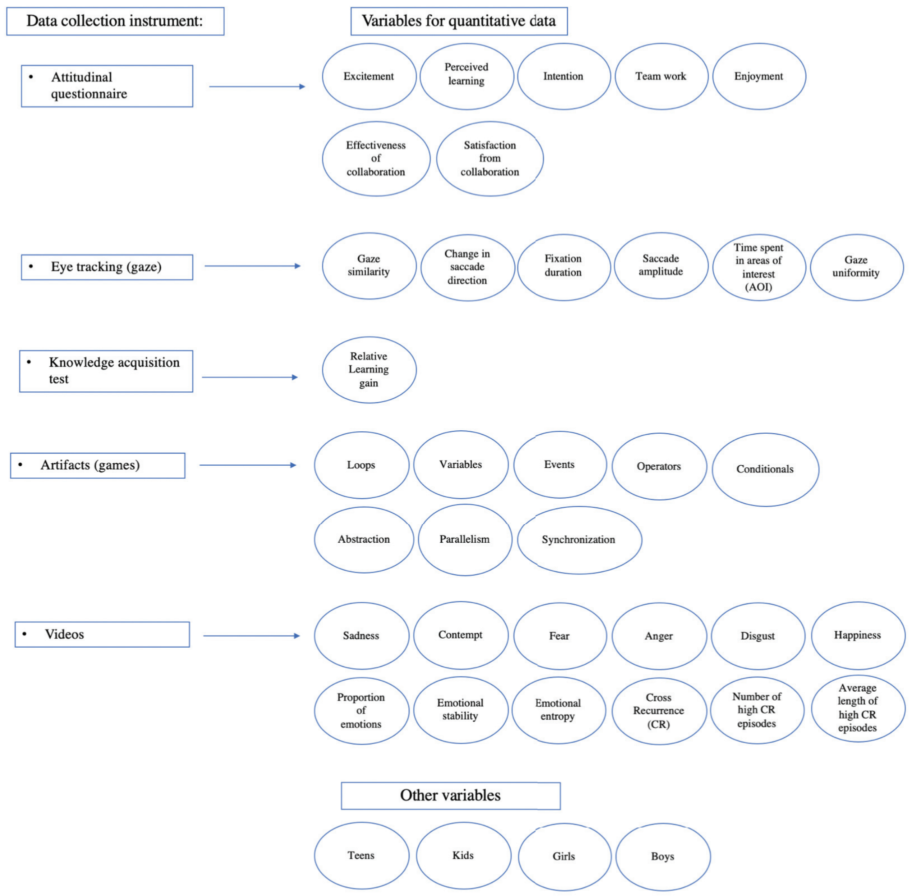

Figure 12: Overview of data-collection instrument and variables

Qualitative data analysis involves Cycles 2 and 3 and the overall analysis of researchers and instructors notes from the focus groups and discussions during the two years of the studies. First, all interviews and focus groups were fully transcribed and translated into English, if needed. Qualitative analysis was manually conducted by the researchers using both inductive and deductive approaches, based on Saldaña (2015) and Mayring, Mayring (2014). In all cases, the inductive coding process was also enriched with theoretically driven deductive elements, based on the respective paper's research question.

To analyze the transcribed interviews, two researchers followed the coding method proposed by Saldaña (2015) for qualitative inquiry. Saldaña's coding method describes a cyclical model that moves from codes to categories and themes. Analysis of the semi-structured interviews focused on identifying categories, and then the overall themes forming the codes emerged from participants' answers. Each transcript was first individually reviewed by two researchers and then, after a focus group and discussion, the two researchers agreed on the major themes that had emerged.

Similar processes were followed for all ideas that were connected to the results of the respective iteration (Cycles 1, 2 and 3), representing the codes for the qualitative analysis. During the two years of the project, after the end of each iteration (cycle), the researchers and instructors participated in focus groups, discussing and revealing all the growing ideas emerged from the iteration outcomes. To synthesize the ideas and formulate themes, we focused mainly on the students' engagement in the coding activities. Consequently, the most prominent themes emerged. It was an iterative process, with constant refinement and reflection on the ideas and themes during the three cycles. This helped us not only to see the connections and make decisions for the design, but also to identify the most important theoretical aspects in our studies. The final step of the analysis, after removing similar themes, involved categorization to identify the most important findings.

Each observation note (one set for each team) was reviewed by two researchers. Using content analysis, the main actions indicating a specific behavior were identified, and the frequencies of help levels were calculated. Finally, artifacts (games) developed by the teams were evaluated in terms of the learning opportunities related to computer science and computational thinking concepts offered by coding a game. The evaluation of the artifacts included loading and playing the game to ensure its functionality and playability. To analyze each version of the games, we analyzed the games based on computational thinking components (i.e., flow control, data representation, abstraction, user interactivity, parallelism, and logic), giving a score for each of them from 0 to 3 (a rubric in which 3 shows proficiency, and 0 means that the skill is not evident). Artifacts were used as an extra source to determine the main characteristics (such as the game's theme, aesthetics, and storytelling) of young students' codes and their use of specific concepts related to the learning objectives of our workshop, as well as to discover any unexpected learning outcomes.

Figure 13 presents a mapping between the various data collections/measurements and the analyses employed.

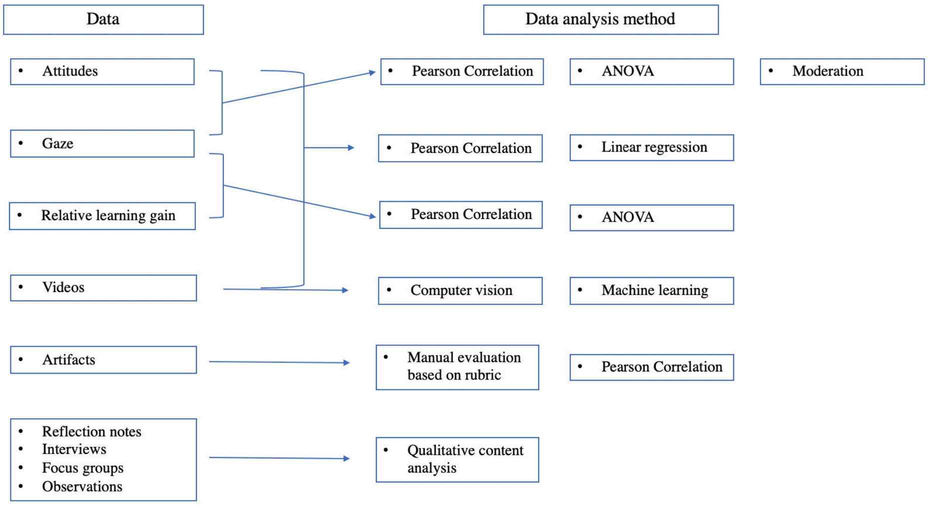

Figure 13: Overview of the data analysis

4 Results

The research work, consisting of this thesis, has been published as five journal papers and three conference papers. Each of the eight papers has been peer-reviewed and, therefore, accepted by other researchers as providing a significant contribution to the body of knowledge. The papers are reprinted in full length in Part II of this thesis following publishing permissions from the editors. The paper summaries are ordered in publishing chronological order. This chapter summarizes the papers that contain the results for the conducted research. The papers presented in the following section include:

- Title
- Authors' names
- Authors contributions in the paper
- Where the paper is published
- Abstract of the paper
- A short description of the main findings of each paper
- The paper's relation to the research questions

4.1 Paper 1

Title: Empirical Studies on the Maker Movement, a Promising Approach to Learning: A Literature Review

Authors: Sofia Papavlasopoulou, Michail N. Giannakos, Letizia Jaccheri

Paper published in: Entertainment Computing, Volume 18, pages 57-78, 2017.

Authors' contributions: Papavlasopoulou led the paper writing and was the main author. Also, Papavlasopoulou collected and analyzed all the articles included in the literature review. Constant consensus meetings of Papavlasopoulou and Giannakos approved each step of the analysis of all studies conducted. Giannakos and Jaccheri provided general supervision of the research and the paper writing.

Abstract: The Maker Movement has gathered much attention recently and has been one of the fastest-growing topics, due to contemporary technical and infrastructural developments. The maker culture can be described as a philosophy in which individuals or groups of individuals create artifacts that are recreated and assembled using software and/or physical objects. Typical topics of interest in maker culture include engineering-oriented pursuits such as electronics, robotics, 3D printing, and computer numerical control tools, as well as more traditional activities such as sewing or arts and crafts. Scholars and educators have reported a variety of outcomes from the Maker Movement as an instructional process;

however, the lack of a summary of these empirical studies prevents stakeholders from having a clear view of the benefits and challenges of this instructional culture. The purpose of this article is to provide a review of the Maker Movement approach in order to summarize the current findings and guide future studies. Forty-three peer-reviewed articles were collected from a systematic literature search and analyzed based on a categorization of their main elements. The results of this survey show the direction of Maker Movement research during recent years and the most common technologies, subjects, evaluation methods, and pedagogical designs. Suggestions for future research include a further investigation into the benefits of using a specific technological tool and analysis of the Maker Movement approach, particularly in classrooms. These future research efforts will allow us to better indicate which aspects and ingredients of "making" work better for which circumstances and student groups. The findings will ultimately allow us to form best practices and a unified framework for guiding/assisting educators who want to adopt this teaching style.

Main findings presented in the paper: This paper is a systematic literature review consisting of 43 studies that implemented *making*. The most common subject areas using a making approach to learning are coding and STEM curricular areas. The results show that studies tend to use a qualitative methodology to assess their work. All the studies used some type of digital material to support making activities, which highlights the need to familiarize users with technology, expanding participants' perspectives and interest in computer science in general. The goal is to achieve better understanding and enhance skills related to the subject areas through digital fabrication devices, producing objects, and modeling tools. Making sessions are promising approaches to engaging students in the design and fabrication process, in thinking and problem solving, as well as in coding. This provides evidence for the success of making in influencing learners' behaviors. When their self-efficacy increases, workshop participants gain confidence, enjoyment, and interest in programming and technology. Furthermore, the studies evaluated reveal a variety of technology tools used in making. Given the large amount of different software available, and the possibility for it to be used for educational reasons, it is difficult to define the best choice for a specific activity.

Relation to the research questions: The paper addresses SQ1 in order to show the state of the art on how making types of instruction assist making-based coding activities.

4.2 Paper 2

Title: Using Eye-Tracking to Unveil Differences Between Kids and Teens in Coding Activities

Authors: Sofia Papavlasopoulou, Kshitij Sharma, Michail N. Giannakos, Letizia Jaccheri

Paper published in: Proceedings of the conference on Interaction Design and Children (IDC), Stanford 2017 (ACM).

Authors' contributions: Papavlasopoulou led the paper writing and was the main author. Papavlasopoulou, Sharma, and Giannakos designed and supervised the study and data collection. Sharma performed the data analysis and contributed to writing the paper. Giannakos and Jaccheri provided general supervision of the research and the paper writing.

Abstract: Computational thinking and coding is gradually becoming an important part of K-12 education. Most parents, policy makers, teachers, and industrial stakeholders want their children to attain computational thinking and coding competences, since learning how to code is emerging as an important skill for the 21st century. Currently, educators are leveraging a variety of technological tools and programming environments, which can provide challenging and dynamic coding experiences. Despite the growing research on the design of coding experiences for children, it is still difficult to say how children of different ages learn to code, and to cite differences in their task-based behavior. This study uses eye-tracking data from 44 children (here divided into "kids" [age 8–12] and "teens" [age 13–17]) to understand the learning process of coding in a deeper way, and the role of gaze in the learning gain and the different age groups. The results show that kids are more interested in the appearance of the characters, while teens exhibit more hypothesis-testing behavior in relation to the code. In terms of collaboration, teens spent more time overall performing the task than did kids (higher similarity gaze). Our results suggest that eye-tracking data can successfully reveal how children of different ages learn to code.

Main findings presented in the paper: The paper investigates the differences between kids' and teens gaze during coding and how their gaze are associated with their learning. First, one interesting feature of the results is that the teens spent more time looking at the scripts, output, and command areas of interest (AOIs), while kids spent more time on the sprites AOIs. The sprites control the aesthetic part of the problem, for example, what the main animated character or the different costumes look like. Spending more time on the appearance of the output proved to be detrimental to the kids' RLG. Our results showed that teens attained higher RLG. Second, the teens had a higher number of transitions among scripts and output/command/robot compared to the kids. The higher number of transitions between scripts and outputs indicates behavior that is either caused by a debugging activity or a desire to verify a hypothesis. Finally, there is a relation between the gaze and age groups, since the teens had higher gaze similarity than the kids did. This study successfully used eye-tracking data as a new means of analyzing children's learning processes in coding and discover any differences in their task-based behavior according to their ages.

Relation to the research questions: The paper investigates SQ3 and contributes to SQ4. It addresses how young students' ages (kids vs. kids) influence their experience during making-based coding activities.

4.3 Paper 3

Title: How do you feel about learning to code? Investigating the effect of children's attitudes towards coding using eye-tracking

Authors: Sofia Papavlasopoulou, Kshitij Sharma**,** Michail N. Giannakos

Paper published in: International Journal of Child-Computer Interaction, Volume 17, Pages 50-60, 2018.

Authors' contributions: Papavlasopoulou led the paper writing and was the main author. Papavlasopoulou, Sharma, and Giannakos designed and supervised the study and data collection. Sharma performed the data analysis and contributed to writing the paper. Giannakos provided general supervision of the research and the paper writing.

Abstract: Computational thinking and coding for children are attracting increasing attention. There are several efforts around the globe to implement coding frameworks for children, and there is a need to develop an empirical knowledge base of methods and tools. One major problem for integrating study results into a common body of knowledge is the relatively limited measurements applied, and the relation of the widely used self-reporting methods with more objective measurements, such as biophysical ones. In this study, eye-tracking activity was used to measure children's learning and activity indicators. The goal of the study is to utilize eye-tracking to understand children's activity while they learn how to code and to investigate any potential association between children's attitudes and their gaze. In this contribution, we designed an experiment with 44 children (between 8 and 17 years old) who participated in a full-day construction-based coding activity. We recorded their gaze while they were working and captured their attitudes in relation to their learning, excitement and intention. The results showed a significant relation between children's attitudes (what they think about coding) and their gaze patterns (how they behaved during coding). Eye-tracking data provide initial insights into the behavior of children, for example if children have difficulty in extracting information or fail to accomplish an expected task. Therefore, further studies need to be conducted to shed additional light on children's experience and learning during coding.

Main findings presented in the paper: This paper investigates potential relations between children's attitudes (excitement, intention to participate in a similar activity, perceived learning) and their gaze during coding activities. Regarding the gaze measures connected to cognition used for this study, it worth noticing that: 1) high fixation duration depicts that the participant is having difficulty extracting information, 2) longer saccades show meaningful transitions in terms of attention, 3) for change in saccade direction: an angle between two lines that is greater than 90◦ reflects a change of plans, revision, or a failed expectation/hypothesis/anticipation. In particular, the children who reported higher excitement had lower fixation duration, lower saccade direction change, and higher saccade amplitude than those who reported lower excitement during the coding task. For gaze and perceived learning, a significant relation was observed between all the gaze variables and learning. The young students, who reported higher perceived learning, had lower fixation duration, lower saccade direction change, and higher saccade amplitude than those who reported lower perceived learning. For gaze and intention, young students who reported higher intention to code had lower fixation duration, lower saccade direction change, and higher saccade amplitude than those who reported lower intention. More specifically, young students, who indicated better management of cognitive load, expressed higher scores in their attitudes. Findings also suggest that young students with higher reported excitement and learning had the same characteristics. This study also demonstrates that eye-tracking provides information about children's approaches to coding tasks.

Relation to the research questions: The paper investigates SQ2 and contributes to SQ4. It addresses the importance of positive attitudes and their relation to cognitive processes during young students' coding learning experience.

4.4 Paper 4

Title: Discovering children's competences in coding through the analysis of Scratch projects

Authors: Sofia Papavlasopoulou, Michail N. Giannakos, Letizia Jaccheri

Paper published in: Proceedings of Global Engineering Education Conference (EDUCON). Tenerife, Spain 2018:

Authors' contributions: Papavlasopoulou led the paper writing, performed the data analysis, and was the main author. Papavlasopoulou and Giannakos designed and supervised the study and data collection. Giannakos and Jaccheri provided general supervision of the research and the paper writing.

Abstract: Computational thinking and coding has received considerable attention over the past several years. Considerable efforts worldwide suggest the need for more empirical studies providing evidence-based practices to introduce and engage children with coding activities. The main goal of this study is to examine which

programming concepts students use when they want to develop a game, and what is the interrelation among these concepts. To achieve our goal, a field study was designed, and data were collected from coding activities. In detail, during a two-week period, one-day workshops were organized almost every day on which 44 children participating in, with ages between 8 to 17. The workshops follow a constructionist approach and comprise of two parts. First the children interact with robots, and then develop a game using Scratch. The findings provide a deeper understanding on how children code by showing the use of specific programming concepts to develop their projects and their correlations. Hence, we improve our knowledge about children's competences in coding.

Main findings presented in the paper: This paper examines young students' competences in coding. Analysis of the young students' artifacts revealed the prominent measures of programming concepts young students used, and which of them were related. The findings show that *event handling* is the basic programming concept and is present in all projects. It can be considered a prerequisite to having operational projects. On the other hand, *variables* are a complicated concept. *Variables* are present in four of the nine games, and the only concept correlated with three out of the six other concepts. *Variables* and *event handling* are strongly correlated. *Threads* and *iteration* are the higher correlated programming concepts. A surprising result is the fact that threads are not correlated with synchronization. In Scratch, synchronization helps to control timing between the sprites which run as parallel threads; therefore, using broadcast/receive blocks (synchronization) is essential for creating communication between them. However, in the analyzed projects, synchronization was very rarely used.

Relation to the research questions: The paper is not particularly related to one of the SQs; however, it contributes to SQ4, provided valuable information on young students' use of specific programming concepts, and guided further research for this PhD research.

4.5 Paper 5

Title: Exploring children's learning experience in constructionism-based coding activities through design-based research

Authors: Sofia Papavlasopoulou, Michail N. Giannakos, Letizia Jaccheri

Paper published in: Computers in Human Behavior, volume 99, pages 415-427, 2019

Authors' contributions: Papavlasopoulou was responsible for the paper writing, data analysis, and documentation. Giannakos and Jaccheri provided general supervision of the research and the paper writing.

Abstract: Over the last few years, the integration of coding activities for children in K-12 education has flourished. In addition, novel technological tools and programming environments have offered new opportunities and increased the need to design effective learning experiences. This paper presents a design-based research (DBR) approach conducted over two years, based on constructionism-based coding experiences for children, following the four stages of DBR. Three iterations (cycles) were designed and examined in total, with participants aged 8–17 years old, using mixed methods. Over the two years, we conducted workshops in which students used a block-based programming environment (i.e., Scratch) and collaboratively created a socially meaningful artifact (i.e., a game). The study identifies nine design principles that can help us to achieve higher engagement during the coding activity. Moreover, positive attitudes and high motivation were found to result in the better management of cognitive load. Our contribution lies in the theoretical grounding of the results in constructionism and the emerging design principles. In this way, we provide both theoretical and practical evidence of the value of constructionism-based coding activities.

Main findings presented in the paper: This paper presents the design-based research (DBR) effort, comprising all the three cycles (iterations) conducted over two years, in my PhD thesis. The prominent results of the studies, with regard to the effects of coding activities on children's learning experience, are grounded on Papert's constructionism. This includes the notion of "bricolage," "learning to think articulately about thinking," "the importance of social norms" "teaching the Turtle to act or to 'think'" and the meaning of "powerful ideas." In general, the young students indicated that they were cognitively engaged during the workshops; they managed to adopt deliberative thinking and to understand and imitate mechanical thinking while coding. Cognitive effort is also linked to young students' behavioral and emotional engagement because positive attitudes have an effect on their load management. Social engagement is important as young students work in front of the computer and reflect on their progress as a team, sharing the same goal to successfully create an artifact. At the end of the workshops, the young students felt competent and proud of their achievements with an increased sense of achievement, self-confidence, and self-efficacy. Lastly, the main principles to facilitate making-based coding activities that support children's learning experience should consider and/or promote the following: 1) Social interaction, 2) Appropriate design according to age, 3) Duration of the activity, 4) Relevance of the activity and meaningful content, 5) Physical and digital artifacts, 6) Young students' attitudes and motivation, 7) Cognitive overload, 8) Appropriate tasks, and 9) Meaningful framework for the involvement of the instructors.

Relation to the research questions: The paper investigates SQ2, SQ3 and addresses SQ4. It shows the role of engagement during young students' coding learning experiences and, since it covers most of the research conducted for this thesis, provides guidelines and lessons learned from our experience from making-based learning activities for young students.

4.6 Paper 6

Title: Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences

Authors: Sofia Papavlasopoulou, Kshitij Sharma, Michail N. Giannakos

Paper published in: Computers in Human Behavior, *in press 2019*

Authors' contributions: Papavlasopoulou led the paper writing and was the main author. Papavlasopoulou, Sharma, and Giannakos designed and supervised the first study included in the paper and the data collection. Sharma performed the data analysis and contributed to writing the paper, referring to the first study. Papavlasopoulou designed and supervised the second study included in the paper and performed the data collection and analysis. Giannakos provided general supervision of the research and the paper writing.

Abstract: Computational thinking and coding are becoming an integral part of K-12 education, with female students being underrepresented in such subjects. The proliferation of technological tools and programming environments offers the opportunity for creative coding activities for children and increases the need for appropriate instructional practices. In this study, we design and evaluate a coding workshop for children. Our goal is to examine differences between boys and girls using eye-tracking as an objective measure and triangulating the findings with qualitative data coming from children's interviews. The results show no statistically significant difference between female and male gaze and learning gain during the coding activity; interestingly, the qualitative data show differences in the strategies and implemented practices during coding, and in perceptions about those coding activities. Our results highlight that further studies need to utilize objective measures and unveil necessary differences in the design and implementation of coding activities. Furthermore, our results provide objective evidence that female students do not lack in competences compared to boys, but simply that they have a different approach during coding activities and different perspectives about coding, an approach that needs to be cultivated and nurtured.

Main findings presented in the paper: This paper investigates gender differences in coding activities for young students using objective measures (gaze) and triangulates them with qualitative data (interviews and observations). First regarding the young students' learning gain, the results showed no significant difference between boys and girls. In addition, there was no significant difference in all the gaze measures used between girls and boys. From the qualitative analysis, it is evident that only girls reported that they did not know what they could do with Scratch or coding. All young students expressed improved confidence and self-efficacy in coding and that they managed to accomplish the tasks required. In mixed teams, when a boy knew about coding, girls stated that a boy had to be the leader, while in girls' teams they appeared to have had equal roles. In the interviews, all boys indicated that they contributed to their teams in terms of coding, whereas girls mentioned that not all of them coded but that they felt a valuable part of collaboration. Both girls and boys had similar difficulties, challenges, and frustrations during the creation of their game. Equally, girls and boys reported that they had fun during the workshop. In terms of help received from the assistants, all teams had approximately the same amount, between five and seven times. However, girls were more persistent than boys in trying on their own before asking for help. Children's games evaluation showed that girls' nature was present in their games, as was clear from their use of female characters and that boys and girls had a similar final performance.

Relation to the research questions: The paper investigates SQ3 and contributes to SQ4. It addresses how young students' gender (girls vs. boys) influences their experience during making-based coding activities and gives a deeper understanding of their perspectives and practices.

4.7 Paper 7

Title: Coding games and robots to enhance computational thinking: How collaboration and engagement moderate children's attitudes?

Authors: Kshitij Sharma**,** Sofia Papavlasopoulou, Michail N. Giannakos

Paper published in: International Journal of Child-Computer Interaction, volume 21, pages 65-76, 2019.

Authors' contributions: Sharma led the paper writing. Papavlasopoulou, Sharma, and Giannakos designed and supervised the study and did the data collection. Sharma performed the data analysis. Papavlasopoulou and Sharma contributed to writing the paper. Giannakos provided general supervision of the research and the paper writing.

Abstract: Collaboration and engagement while coding are vital elements for children, yet very little is known about how children's engagement and collaboration impact their attitudes toward coding activities. The goal of the study is to investigate how collaboration and engagement moderate children's attitudes about coding activities. To do so, we designed a study with 44 children (between 8 and 17 years old) who participated in a full-day coding activity. We measured their engagement and collaboration during the activity by recording their gaze,

and their attitudes in relation to their learning, enjoyment, team-work, and intention by post-activity survey instruments. Our analysis shows that there is a significant moderating effect of collaboration and engagement on children's attitudes. In other words, highly engaging and collaborative coding activities significantly moderate children's attitudes. Our findings highlight the importance of designing highly collaborative and engaging coding activities for children and quantifies how those two elements moderate children's attitudes.

Main findings presented in the paper: This paper investigates how collaboration (gaze similarity) and engagement gaze uniformity) moderate the relationship between children's attitudes when coding (i.e., team-work, intention to participate, perceived learning, and enjoyment). There is a significant moderation effect of engagement and intent to participate on perceived learning. Intention to participate and perceived learning is stronger when the participants experience high engagement than in the case where participants experience low engagement. Thus, data provide strong evidence that children's level of engagement during coding activities moderates the relationships between their intention to participate in the activity and learning. In addition, there is a significant moderating effect of engagement in the relationship between intention to participate and enjoyment. It can be observed that the relation between intention to participate and enjoyment is stronger for the highly engaged participants than for non-engaged ones. Therefore, there is strong evidence that children's level of engagement during coding activities moderates the relationships between their intention to participate in the activity and enjoyment.

Relation to the research questions: The paper investigates SQ2 and contributes to SQ4. It addresses how collaboration (gaze similarity) and engagement (gaze uniformity) moderates the relationship between young students' attitudes.

4.8 Paper 8

Title: Joint Emotional State of Children and Perceived Collaborative Experience in Coding Activities

Authors: Kshitij Sharma, Sofia Papavlasopoulou, Michail N. Giannakos

Paper published in: Proceedings of the conference on Interaction Design and Children (IDC), Boise 2019 (ACM).

Authors' contributions: Sharma led the paper writing. Papavlasopoulou designed and supervised the study included in the paper and the data collection. Sharma performed the data analysis. Papavlasopoulou and Sharma contributed to writing the paper. Giannakos provided general supervision of the research and the paper writing.

Abstract: This paper employs facial features to recognize emotions during a coding activity with 50 children. Extracting group-level emotional states via facial features, allows us to understand how emotions of a group affect collaboration. To do so, we captured joint emotional state using videos and collaborative experience using questionnaires, from collaborative coding sessions. We define groups' emotional state using a method inspired from dynamic systems, utilizing a measure called cross-recurrence. We also define a collaborative emotional profile using the different measurements from facial features of children. The results show that the emotional cross-recurrence (coming from the videos) is positively related with the collaborative experience (coming from the surveys). We also show that the groups with better experience than the others showcase more positive and a consistent set of emotions during the coding activity. The results inform the design of an emotion-aware collaborative support system.

Main findings presented in the paper: This paper shows the relation between groups' joint emotional profile (proportions of emotions, emotional entropy and consistency, and emotional togetherness) and their collaborative experience during the coding activity. Emotions such as happiness and contempt co-occur with high perceived effectiveness and high satisfaction from the collaborative coding experience. Negative emotions co-occur with low perceived effectiveness and low satisfaction from the collaboration. Correlations among collaborative experiences, emotional entropy, and emotional consistency show that the positive experiences are accompanied by a low range and consistent set of emotions. Finally, emotional togetherness is observed to be higher for the groups with high perceived effectiveness and high satisfaction than for the groups with low perceived effectiveness and low satisfaction. In terms of the most important components of the joint emotional profile to explain the collaborative experiences, we observe that the proportion of anger and emotional entropy appear in both the effectiveness and satisfaction models. Moreover, these two measurements have negative coefficients showing that high proportions of anger and having a wide range of emotions in a given time period has a detrimental effect on both the perceived effectiveness and satisfaction.

Relation to the research questions: The paper investigates SQ2 and contributes to SQ4. It addresses how young students' emotional status influences their experience during making-based coding activities and gives a deeper understanding of their collaboration.

5 Discussion

This chapter discusses the research contributions, the implications and limitations of this PhD work.

5.1 Contributions

C1: *Summarizes and conceptualizes the state of the art in making practices and their role in enhancing coding activities for children*. The first contribution of this PhD thesis presents a systematic literature review (*Paper 1*) (Papavlasopoulou, Giannakos, et al., 2017), producing substantive findings regarding the *making* approach and its applications to making-based coding activities. The review aims to show the state of the art and identify potential research gaps.

Nowadays, students need to acquire skills and competences to be prepared for their future work and everyday life. Therefore, learning approaches need to be adjusted to teach 21st century skills successfully. The work is based on the emerging developments in the area of coding and learning technologies, creating momentum for adoption of *making*. We draw on the research literature to consider the trends and possibilities within this movement, investigating different types of making activities that are related to a successful learning experience in terms of learning, interest, and engagement. Furthermore, we consider how these practices could help students improve their performance in coding, computational skills, and problem solving. This has been achieved by evaluating empirical studies from the last five years.

In particular, *making* involves constructing activities and related ways to fabricate real and/or digital things using technological resources, including fabrication (Katterfeldt et al., 2015), physical computing, and coding. *Making* focuses on the process that occurs in an environment that is not always merely learning oriented, but promotes design thinking, computational concepts, collaborative work, and innovation, among other things. Using a systematic methodology (Kitchenham, 2004), 43 papers were reviewed. First, recognizing the most common subject areas for implementing making types of instructions, our literature review confirmed that a making approach to learning is being taken, most notably in coding, and also in STEM curricular areas. Almost all of the studies had as their main subject coding or a combination of coding and math (Garneli, Giannakos, Chorianopoulos, & Jaccheri, 2013).

Some type of digital material is important in *making* activities so the users can familiarize themselves with technology and broaden their perspectives and interest in computer science in general. The goal is to achieve better understanding and enhance skills related to the subject areas through digital fabrication devices, their ability to produce objects, and modeling tools. Students' involvement in several computer science concepts enhances their ability to achieve goals (Franklin et al., 2013). A study by Franklin et al.

(2013) showed that it is possible for students to attain competence in event-driven programming, state initialization, and message passing after just a two-week interdisciplinary camp, which was not entirely focused on computer science. A class that used Lilypad Arduino successfully promoted computing concepts and practices, while perceptions of computing were extended (Kafai et al., 2013). Other studies using the same tool showed how craft materials support a more understandable approach to creating technology, and the results of this process can be more transparent and expressive. Using the same concept, workshop participants managed not only to think about, but also to create, interface designs using conductive fabrics and craft materials (Perner-Wilson, Buechley, & Satomi, 2011). When crafts and technology are tightly connected by conceiving and realizing different artifacts, people become more engaged and develop different skills compared to getting involved with traditional development or electronic toolkits (Mellis, Jacoby, Buechley, Perner-Wilson, & Qi, 2013).

Making activities are expected to be promising approaches to engaging students in coding and design, as well as in the fabrication process, in thinking, and in problem solving. In many studies, the combination of coding and physical fabrication resulted in engagement in complex programming concepts (e.g., loops, conditionals) and practices (e.g., remix, testing, and debugging) (Denner et al., 2012) (Searle, Fields, Lui, & Kafai, 2014). Furthermore, even young students aged 9–10 years have been engaged in Java programming by playing and making games (Esper et al., 2014). Another main core of learning, as reported by Katterfeldt et al. (2015), is self-efficacy. Some of the analyzed studies showed that, after the making activities, the participants' self-efficacy was affected. This provides evidence for the success of *making* in influencing learners' behaviors. When their self-efficacy increases, workshop participants gain confidence, enjoyment, and interest in programming and technology (Qiu, Buechley, Baafi, & Dubow, 2013). Moreover, when actions are motivated with enthusiasm and self-regulatory feedback, self-efficacy ratings are higher (Lane et al., 2013).

All of the studies report the positive effects of making activities on students' perceptions and engagement. Students' experiences with computer game coding could also change their attitudes toward computer science, preparing them for computer science courses and careers (Denner et al., 2012). In general, no matter what the age of the group and which tool was used, making proved to be a successful process in all the different areas of interest. One surprising outcome is the absence of negative results. Almost none of the studies reported negative effects in the research. Even though this is an obvious positive conclusion, it does not provide an in-depth understanding of how to prevent poor practices that hinder students' engagement and performance.

Furthermore, the evaluated studies reveal a variety of technology tools used in making. Given the large amount of different software available, and the possibility for it to be used for educational reasons, it is difficult to define the best choice for a specific activity. It was not surprising that the most-used tool was Scratch, as this is one of the popular visual programming languages, followed by Lilypad Arduino. In the remaining studies, other, less common technologies were used, identifying the need for further investigation in this area. Moreover, combining digital and tangible materials in the creating process has been proven to be valuable (Kafai et al., 2014).

Collaboration among the participants was present in the large majority of studies. We know how important collaboration is to motivating and promoting learning, which is why many efforts concentrated on testing different types of offline or even online collaborative methods (Y. Kafai & V. Vasudevan, 2015). Although collaboration appears to be an important aspect of making activities, we saw very few descriptions of collaborative strategies and how they contribute to individual learning. An interesting approach came from Fields et al. (2015), who proposed a collective design process for coders, deriving from participatory models. They illustrated that collaboration supported learning through the exchange of ideas and mentoring and led to deeper engagement. A surprising result from our review is that few studies focused on gender issues. We expected more studies to provide insights on how making activities benefit females specifically since the main subject areas applied are coding and STEM.

C2: *Investigate new methods to evaluate making-based coding activities for young students*. This includes using multimodal data in the field studies; combining data from various channels which are subjective (interviews, surveys) and objective physiological data (eye-tracking) to understand the learning process of coding in a deeper way. This contribution refers to using the eye-tracking method to pervasively track the gaze of children of young ages in a coding activity.

Collecting data from wearable eye-tracking glasses in a coding activity with young students is a first step toward using gaze to unveil their experience in the coding process. Several studies have successfully shown a clear relation between gaze patterns and performance, learning strategy, and other personality factors (Jermann & Nüssli, 2012). That makes our approach an important contribution in eye-tracking, child-computer interaction community. For instance, gaze provides information about young students' cognitive loads, levels of attention, collaboration, engagement, and the main areas the children looked at, allowing instructional designers and technology developers to design activities and products tailoring to young students' needs. Thus, applying such advanced sensing devices in learning settings may transform how young students learn and how we design activities and technologies to support teaching and learning.

In detail, in contributions C3, C4, and C5, we discuss the results from using young students' gaze as an objective measurement to depict children's processes while coding.

C3: *Improved understanding of the learning process to support young students in making-based coding activities*. This includes the design, evaluation, and refinement of coding workshops, in which the field studies of this thesis took place, focusing on enhancing young students' engagement and attitudes during their experience. This contribution provides insights on the elements of engagement existing in making-based coding activities for young students, as presented in *Paper 5* (Papavlasopoulou, Giannakos, & Jaccheri, 2019) included in this thesis, the importance of positive attitudes investigated in *Paper 3,* (Papavlasopoulou, Sharma, & Giannakos, 2018) *Paper 6* (Papavlasopoulou, Sharma, & Giannakos, 2019)*,* and *Paper 7* (Sharma, Papavlasopoulou, & Giannakos, 2019a)*,* and the importance of emotions in collaboration among the students and their attitudes in *Paper 8* (Sharma, Papavlasopoulou, & Giannakos, 2019b).

The students indicated that they were cognitively engaged during the workshops; they managed to adopt deliberative thinking and to understand and imitate mechanical thinking while coding. In order to achieve this, they had to use an appropriate cognitive strategy (e.g., a "hypothesis-testing" gaze pattern, as shown by the eye-tracking data) to approach the task and achieve some level of self-regulation. There are different ways to approach a problem, and it takes time to learn the necessary skills. In our workshops, we used a visual programming tool (Scratch); one of the strengths of such tools is that computational practices become less cognitively challenging (Kelleher & Pausch, 2005), so students can focus on problem solving and creative thinking (Lin & Liu, 2012). Even with the use of such tools, the cognitive load during the coding process can be critical as students use the "bricolage" style by constantly experimenting and trying different patterns. Instructors can help students manage their learning and thinking to adopt an effective approach to coding. This is not a new practice, as previous studies with Logo have used precise instructions for computational practices, such as testing and debugging (Fay & Mayer, 1994) (Carver & Mayer, 1988).

Cognitive effort, as shown in our studies, is also linked with students' behavioral and emotional engagement because positive attitudes have an effect on their load management. Students should be persistent, make efforts, and deal with difficulties; therefore, having positive attitudes and keeping themselves motivated result in better management of their cognitive load. In particular, our findings suggest that gaze patterns and attitudes can be correlated (Paper 3 and 7) (Papavlasopoulou et al., 2018) (Sharma et al., 2019a). The three different gaze measures used represent children's difficulties in extracting information during a coding activity (fixation duration), the number of trials needed to learn something during coding (saccade direction change), and children's goals and expectations during coding (saccade amplitude). As was expected, children who had fewer difficulties and could handle the cognitive load better had higher scores in their attitudes (i.e., perceived learning, excitement, and intention to participate in a similar activity). In the same vein, Robertson and Howells (2008) argue that the game design experience is a powerful learning environment that supports motivation, engagement, and enthusiasm. Using a visual programming environment, students can be introduced to programming concepts in a fun and useful way through a design activity, making them highly motivated and positive toward coding (Giannakos & Jaccheri, 2018) (Sáez-López et al., 2016).

Social engagement is important as students work in front of the computer and reflect on their progress as a team, sharing the same goal to successfully create an artifact. In our workshops, the students, working as a team, built a group identity and, at the same time, engaged in social comparison with their peers. Students, especially novices to coding, usually have difficulties with simple coding actions, from relating different commands together to completing more advanced actions, like debugging; collaboration helped the students in this study to confront those difficulties. In a similar study with girls creating games, good collaboration in debugging resulted in the girls being more persistent when coding on their own, without help from the instructors (Denner, 2007). Helping each other and sharing their challenges and successes were critical for our students, nurturing social engagement, and avoiding a sense of isolation. Collaboration and reflection lead to better learning and powerful thinking. Reflection relates to their own learning experience or reflecting on their peers' code and actions. Positive experiences in the coding workshops are accompanied by a low range and consistent set of emotions among the young students who collaborated as a team. Previous studies have shown that students performed better when they were working in pair programming (Lye & Koh, 2014) (Werner, Denner, Campe, & Kawamoto, 2012); in a game-making study, when taking into account peers' recommendations and spending time applying these changes, girls produced higher-quality games (Robertson, 2012). Over time, the students in our workshops were able to understand more about coding and became more behaviorally and emotionally engaged. They were able to reflect on the more complex aspects of their own thinking accordingly by making decisions and controlling the outcomes. Students who are actively part of game-making activities strengthen their problem-solving, critical thinking, and CT skills (Grover & Pea, 2013). During construction, students have to investigate different strategies, negotiate, make decisions about possible solutions, confront problems, and organize their thoughts and actions (Bers et al., 2014).

One of the core aspects of a learning activity is the fact that the problem should be meaningful to the learners. In our case, they constructed shared artifacts that mattered to them. Different studies have used problems like designing games (Denner & Werner, 2007) or stories (Burke, 2012). A "powerful idea" must be both personally and epistemologically useful to ensure engagement. The students in our workshops saw themselves gaining a powerful quality by organizing a new way of thinking, building on their previous knowledge and skills. Nowadays, significant value is placed on transferable skills related to digital technology since they are vital for children's roles in the digital world and should be enhanced through activities that are connected to their lives (Iversen, Smith, & Dindler, 2018). In *making*, students deal with difficulties, learn step-by-step to solve problems, develop belief in their skills, and share ideas with peers (Çakır et al., 2017) (Chu, Schlegel, et al., 2017). In our study, this was confirmed: the students increased their sense of achievement, self-confidence, and self-efficacy. At the end of the workshops, the students felt competent and proud of their achievements. After the workshop, compared to the boys, the girls expressed lower self-efficacy (a belief in one's capacity to succeed in tasks), possibly because most of them did not have any previous coding experience. A sense of self-efficacy is important and should be enhanced, as it is related to cognitive strategies, effort, and persistence in learning environments (Bandura, 1997).

C4: *Investigate the needs of different population of young students to support their learning experience*. The fourth contribution of this PhD work investigates the potential differences that exist in how young students of different ages and genders handle the learning process during coding activities. More specifically, in *Paper 2* (Papavlasopoulou, Sharma, Giannakos, & Jaccheri, 2017) we discuss the results from our study in which children (8–17 years old) performed certain coding tasks in groups (dyads and triads) while their gaze was recorded and their RLG measured (using knowledge acquisition tests). We divided the sample into kids (8–12 years) and teens (13–17 years) to analyze the difference in gaze patterns and RLG across age groups. In addition, in *Paper 6,* (Papavlasopoulou, Sharma, et al., 2019) we investigated gender issues in the coding activities, using their gaze as an objective measure and qualitative data (i.e., semi-structured interviews, and observations) to get a deeper understanding of children's experiences during the workshop.

Differences between teens and kids

The key motivation behind our contribution was to establish the relations between gaze, different RLGs, and different age groups. The teens outperformed the kids in terms of RLG. We established certain key differences in the gaze patterns of kids and teens to investigate the reasons why/effects of the fact that teens outperformed kids in the RLG from the coding tasks. First, one interesting feature of the results is that the teens spent more time looking at the scripts, output, and command AOIs, while kids spent more time on the sprites AOIs. The sprites control the aesthetic part of the problem at hand, such as what the main animated character or the different costumes look like. Spending more time on the appearance of the output proved to be detrimental to the kids' RLG. On the other hand, the scripts, output, and commands control the actual functionality of the coding environment and the main areas of attention in the coding process. These are the areas in which the coder must choose the appropriate command, then add it to the scripts area, and see the outcome of the executed code. Our results showed that teens, who were spending more time on these areas, attained higher RLG. In addition, we found positive and significant correlations between the RLG and the proportion of time spent on the scripts, output, and commands AOIs and the negative and significant correlation between the RLG and the proportion of time spent on the sprites. In a study by Lee, Kafai, Vasudevan, and Davis (2014), all participants aged 10–12 spent significant time on aesthetics; however, the authors identified differences in the time spent on aesthetics in terms of gender. Girls spent more time on aesthetics and also tried harder to balance technical functionality.

Second, the teens had a higher number of transitions among scripts and output/command/robot compared to the kids. The higher number of transitions between scripts and outputs indicates behavior that is either caused by a debugging activity or a desire to verify a hypothesis. In addition, a higher number of transitions between script and robot shows similar behavior. For example, moving back and forth between script and output might result from frequent changes in the code and a need to check the output. Thus, if the output matches the student's hypothesis after executing the desired code, the student moves on to the next step to continue with a new task. If the output does not match the hypothesis, he or she refines the code and rechecks the output. This is a typical hypothesis verification cycle, often associated with the novice coding style (Sharma, Caballero, Verma, Jermann, & Dillenbourg, 2015). Novices who tend to use this style often perform better. Since the teens used this style in coding, it might explain why they outperformed the kids. Young students who are novices usually do not try to debug their code, and thus face the difficult task of solving a poorly executed block of code, or successfully joining pieces of code (Denner et al., 2012). The finding of a positive correlation between the RLG and the number of transitions between script and output supports this explanation as well. Moreover, the higher number of transitions between script and command areas shows the process of choosing the appropriate command to follow the current script. The teens spent more time finding the correct command, and trying different ones, than did the kids. Thus, the teens learned more than the kids; again, a significant and positive correlation between the RLG and the number of transitions between the script and command AOIs support our explanation. A study involving kids as young as 5–6 showed they can plan their actions and think two or three commands ahead; those who do so concentrate hard on the screen, do not pay much attention on other's comments, and have more confidence in their actions and knowledge (Fessakis et al., 2013).

Finally, regarding the differences between teens and kids, we found a relation between the gaze and age groups since the teens had higher gaze similarity than the kids did. One plausible explanation for this could be that groups with high gaze similarities were able to reflect together on their progress and deal with the coding tasks by making decisions together. This might have helped them create a shared understanding of the problem at hand. Having a higher level of shared understanding helped them attain a higher average RLG (Sharma et al., 2015). This can also be verified based on observations and assistants' comments during the activity that the teen teams helped each other more, while the children quarreled more about who would take the lead role in coding. On the other hand, the groups with low levels of gaze similarity mostly focused on the different parts of the program within the given time frame, and this might have had a detrimental impact on their level of shared understanding, and in turn, their average RLG. The significantly positive correlation between the gaze similarity and the groups' average RLG further strengthens this explanation. This result is in line with several other studies showing high levels of cross-recurrence or gaze similarity being correlated with task-based performance (Ericson & McKlin, 2012) and/or learning gains.

Differences between boys and girls

Our research findings reveal that gender issues in coding activities for children are a multifaceted phenomenon. According to the quantitative findings, there are no gender differences concerning RLG and gaze behavior in boys and girls. On the other hand, qualitative results from interviews, observations, and the created games showed that some gender differences exist in children's approaches, as revealed by their behavior during the workshop and their perceptions.

There was no difference in the RLG between girls and boys. Therefore, children in our study showed no differences in their performances, which supports previous studies on children using other evaluation methods (Owston et al., 2009) (Vos et al., 2011). Therefore, our findings provide more evidence that girls are not less competent than boys. Although more girls than boys in our interviews said that they had not known about coding before, or that they were afraid of it, they managed to be equally good as the boys. Moreover, the activities offered in our workshop were appropriate independently from the participants' gender and their previous knowledge. Furthermore, in the interviews, young students reported they had fun during the workshop, even though some of them had prior knowledge of coding. This can be attributed to the fact that Scratch is not limited: it provides many possibilities for making more advanced creations, so users can find it interesting and learn more, whatever their existing knowledge level. In addition, the collaborative notion of the workshop enabled students to learn from each other and not to have their own individual performance as their main goal. As shown in other studies, students perform better when working in pair programming than when working alone (Lye & Koh, 2014) (Werner et al., 2012).

A noticeable result is that there is no difference in the gaze behavior of girls and boys. We used the objective measure of eye-tracking data and, by examining different measures, we found no difference in any of them. This indicates that, regarding the actual micro-level experiences of boys and girls during coding with Scratch, there is no difference in their approach based on their gaze, and hence no difference in their cognitive processes (Eckstein et al., 2016). From measures of time spent on different AOI, gaze uniformity, and transitions among the different AOI, results showed that both male and female participants were able to navigate the Scratch interface, had a meaningful thinking process, and were engaged. Similarly, from the other measures used, results show that both genders had equivalent difficulties in extracting information (fixation duration), challenges in learning something (saccade direction change), and goals and expectations in coding (saccade amplitude). Cazzato, Basso, Cutini, and Bisiacchi (2010) found weak gender differences in the gaze behavior of participants when trying to solve visuo-spatial problems, but that women used more cognitive resources. Other studies have found that girls face difficulties in coding when they had a lot of elements (Denner et al., 2012) or when they put more effort into having good functionality (Lee et al., 2014).

Although the results of young students' gaze behavior and performance show that there are no important gender differences, the qualitative results of our study reveal that gender differences exist in the practices used by boys and girls and in their perceptions. In general, girls approached the coding activity in a different way from boys. For example, girls were more organized in terms of collaboration, splitting the responsibilities and focusing on a more systematic approach in the tasks, and they also paid more attention to the tutorials. In addition, girls seemed to like more collaboration with others and to share the social part of the activity. Previous studies have shown that female students have a more trusting and sociable approach compared to male students, who are more independent and focused on themselves (Rosenberg-Kima, Plant, Doerr, & Baylor, 2010). In the computer-supported collaborative learning environment, Bruckman et al. (2002) found that girls spent more time than boys in communicating. Girls' games were richer in aesthetics and graphical representation, and they also had a more “girly” approach. This is similar to other findings that show girls spend more time on dialogs (Robertson, 2012) and aesthetics. Similar to the finding of Denner and Werner (2007), our study shows that girls' teams were more persistent in attempting the tasks on their own before asking for help. Whereas girls' games had simpler tasks (like catching falling objects), boys' games had more competitive characteristics. This observation is similar to the finding of Owston et al. (2009) study, in which teachers reported that boys enjoyed playing games more competitively against others. Our observation notes confirm this finding, as boys were also asking the assistants about how interesting their games were.

One of the goals of our workshops was to build children's belief that coding is something that they can do, and that it is not something that only boys would be interested in. After their participation, boys and girls reported they felt competent to code. Another interesting result from our qualitative study is that, even though both girls and boys reporting improved confidence and self-achievement, we find that girls have less self-efficacy. One example is that when girls were among boys in the teams, they chose a boy to be the leader, indicating less confidence. They also expressed that they did not know what coding was before, that they had not tried it, that they did not know whether they could do it, and that they thought it was only for geeks. The stereotype of boys being better than girls at robotics and coding exists from the young age of six-years-old (Master, Cheryan, Moscatelli, & Meltzoff, 2017). A possible reason why they split the roles during their collaboration is that the girls were less confident; in addition, none of the girls was trying to take control. In solo programming, men have been found to be more confident than women (McDowell, Werner, Bullock, & Fernald, 2006). In the study of Beckwith et al. (2005), females' self-efficacy was lower than men's, and women did not easily accept new debugging features.

5.2 Design guidelines and implications for practice

The research contributions presented have potential implications for scholars, educators, and practitioners in the fields of CCI, TEL, and K-12 Computing Education.

C5: *Guidelines for designing making-based coding workshops for young students*. This presents the principles emerged showing the knowledge gained from the two years of interventions and the comparative and retrospective analysis of the outcomes based also on the literature. All papers included in this thesis contributed.

Below, I present the practical implications for the design and execution of effective coding activities for young students.

From the DBR study conducted for this PhD work, nine guidelines to facilitate making-based learning environments that support young students learning experience emerged. Those shed light on best practices in the design of coding activities for young students. The following guidelines represent the knowledge gained from the interventions and the comparative and retrospective analysis of the outcomes, also based on the literature:

1) *Social interaction:* Collaboration between team members is a vital part of coding activities. It is essential to enhance this and to ensure that there is a sense of equality of effort, involvement, and participation between team members and among teams.

2) *Appropriate design according to age*: Different age groups (teens and kids) need different approaches and designs in order to engage with a coding activity. The instruction should consider the characteristics of each age group. One example is to promote a focus on functionality, rather than graphics, from the beginning of the activity to aid younger participants. Instructors should ensure that children receive guidelines on where to focus their attention when they code (such as commands and output in Scratch).

3) *Duration of the activity:* According to constructionism (Papert, 1980), when having children use technological tools, duration is key for them to become personally, intellectually, and emotionally involved. Workshops with longer hours can enable children to learn strategies, gain technological skills, make connections with their own practices, and engage with coding, helping to increase their knowledge.

4) *Relevance of the activity and meaningful content:* Offering a supportive theme for the artifact creation process, in which participants can meaningfully participate in real-life settings, is a key factor supporting the psychological and sociocultural elements for

effective learning. Children become engaged and actively involved in the artifact creation process when it is meaningful for them and related to a real-life context.

5) *Physical and digital artifacts:* The results of the present study showed that including physical tasks was engaging and enabled the participants to enhance their skills. The initial task of designing and drawing in the traditional way (using pen and paper, as well as other tangible materials) immediately put players into action and created a physical and emotional peak in the process.

6) *Children's attitudes and motivation:* The learning process should be supported by providing tasks that encourage children to reflect, motivate them to collaborate, and give them meaningful reasons to complete their artifacts. In this vein, Papert (1980, p. 42) highlighted a resemblance to juggling: *"in a learning environment with the proper emotional and intellectual support, the 'uncoordinated' can learn circus arts like juggling and those with 'no head for figures' learn not only that they can do mathematics but that they can enjoy it as well."*

7) *Cognitive overload:* Coding activities for children can have a high cognitive load, which affects their performance and overall experience with the tasks. Proper organization and integration of the learning materials, with a coherent representation and instruction of the related digital tools, tasks, and activities, are required to avoid unnecessary streams of information and cognitive overload.

8) *Appropriate tasks:* To effectively implement a coding workshop, the tasks should make the children both interested and able to learn. The process should afford participants the opportunity to apply aspects of problem solving, coding, debugging, collaborating, planning, communicating, and reflecting on their work. The tasks should support children's and instructors' abilities to work through the process of creating an artifact and benefit from an appropriate sequence of tasks that allows the maximum use of their abilities. The proposed tasks are: 1) a warm-up activity and an inspiring introduction, 2) explore/design, 3) construct/create the digital artifact, and 4) evaluate/get feedback from peers, all alongside collaborating with team members and receiving support from assistants/instructors.

9) *Meaningful framework for the involvement of the instructors:* In the construction of an artifact, children are not alone: practitioners (e.g., teachers and assistants) and anyone else who is responsible for the learning task are also involved. Therefore, they should strive to create more articulate and honest teaching relationships. Working with digital tools allows the teacher and the learner to share a common goal by trying to get the computer to do what they want and trying to understand what it does. As they create the artifact and encounter "bugs," children engage in conversations and develop the appropriate language to ask for help when they need it. As each artifact process is unique, new situations might occur that neither the teacher nor the learner has faced before. So, the teacher should be

dynamically involved in the creation and the discussions that occur. In that way, there is an opportunity to find new ways to explain and show in real time the concepts needed to the children.

Other general implications based on our studies' findings demonstrate recommendations on how making-based coding activities for young students can become attractive and effective. Instruction should align with human cognitive architecture (Sweller et al., 1998) as well as enhance the learners' motivation (Paas, Tuovinen, Van Merrienboer, & Darabi, 2005). Motivation and positive consequences are related, and young students with the feeling of excitement when performing a task may be more likely to repeat the task in the future. Supportive instructive methods should provide guidance to help children distinguish the relevant factors to complete the task, preventing them from becoming overwhelmed by irrelevant information and actions. For example, they should help them focus on specific parts of the screen to find the respective code segments, split the code into meaningful chunks, and trace the coding process in an effective way. In parallel, during learning activities, instructors should foster students' self-confidence in their abilities to complete the task successfully and ensure a pleasant and motivated environment. Moreover, there is a need for properly designed tools to help reduce cognitive overload and facilitate a more intuitive experience. The design of the aesthetics of the visual coding tool is important to give a pleasant sense for children's use, but it should also help them indicate, in a clear way, the input and output values while coding. One example could be the clear representation of code segments and less complexity in scripting (e.g., fewer sprites and stacks of code). Another thought might be the design of dynamic coding tools that could be further developed according to children's progress in the coding task, such as starting with fewer code segments and gradually providing more advanced coding possibilities in relation to progress. In short, during coding activities for children, it is important to take the motivational and cognitive effects equally into consideration in order to support effective and efficient learning environments.

One of the aspects that this PhD work has focused on is the gender differences. Findings indicate that there are no gender differences in children's actual performance and gaze behavior while coding, and that the main differences are in their practices. This suggests that practitioners should focus on characteristics that will influence girls and change their limited participation in computer science. Our results show that educators should foster girls' self-efficacy (i.e., the belief in one's capacity to succeed in tasks (Bandura, 1997) and make them believe that they do not lack in competences; therefore, educators should be careful to avoid discriminating behaviors. Qiu et al. (2013), found that participants' confidence, enjoyment, and interest in coding and technology increase when self-efficacy grows. According to Bandura (1997), self-efficacy is important in problem solving since it affects the individual's cognitive strategies, effort, persistence, and, consequently, the learning outcome. Coding activities should take into consideration special gender characteristics and facilitate appropriate workshops. Activities focused on collaboration can also be a method to narrow the gender gap in coding activities and to view partnership as a key factor for fostering both learning and positive attitudes in students.

5.3 Implications for theory

Having discussed the main findings focusing on the contributions of this PhD work, this section follows to discuss the derived theoretical implications. This supports the fifth in line contribution, mainly supported from papers 3 (Papavlasopoulou et al., 2018) and paper 5 (Papavlasopoulou, Giannakos, & Jaccheri, 2019) (**C6:** *includes contributions to theories that the research was based*). First, the contribution on Papert's constructionism is extensively discussed since it represents the main theory in which this PhD work is grounded. To facilitate understanding and repetition, five themes are created according to the most important links to constructionism. Second, implications on CLT and SDT are also presented and discussed.

Themes for constructionism:

Learning to learn (different coding approaches result in different learning gain): According to Papert (1980), in a constructionist learning environment, the child is able to construct their own knowledge and build on what they already know. Findings showed that depending on their ages, the young students used different gaze patterns in the coding process, had different approaches to coding, and had different learning gain from the activity. Eventually, "*by deliberately learning to imitate mechanical thinking, the learner becomes able to articulate what mechanical thinking is and what it is not. The exercise can lead to greater confidence about the ability to choose a cognitive style that suits the problem"* and *"what is most important in this is that through these experiences these children would be serving their apprenticeships as epistemologists, that is to say learning to think articulately about thinking"* (Papert, 1980). Young students' coding processes represent their way of "thinking mechanically" and adopting the educational advantage of this way of deliberately thinking. Using a simple description of the process, trying to make a game is a way to combine appropriate orders and create programs to tell the computer what to do, step-by-step. This process includes logic, math, problem-solving, and critical thinking skills. In order for children to achieve their goals in such environments, they should find the appropriate cognitive style that will support them in the coding process of creating a shared artifact. This shows the importance of having appropriate tools and instructions for each age group. Different age groups differently organize their thinking and consequently their coding, so the way they approach the process of creating an artifact can be instrumental to their learning and successful completion of the artifact. This notion is in accordance with Papert, as he presents a resemblance with juggling: *"It always takes time to learn necessary component skills. What can be eliminated are wasteful and in- efficient methods. Learning enough juggling skill to keep three balls going takes many hours when the learner follows a poor learning* *strategy. When a good one is adopted the time is greatly reduced, often to as little as twenty or thirty minutes"* (Papert, 1980). Finding the appropriate methods to help children of different age groups will result in efficient and effective learning processes.

Cognitive effort and affective engagement: Positive attitudes and motivation are important to cognitive learning. There is a relation between children's attitudes and their cognitive processes while coding. Highly motivated children with positive attitudes can handle cognitive load and better manage construction of their artifacts. This idea appeared in our findings from the measures used to examine cognition through the eye-tracking data and the relation with attitudes of perceived learning (seen as confidence, the degree that children indicate their performance), intention to perform coding again, and excitement. The children who were highly engaged and motivated during the construction of the artifact exhibited gaze behavior that showed lower cognitive overload. Papert (1980) describes the notion of "bricolage," which represents a qualitative way of organizing and planning when problem solving by constantly experimenting until finalizing the artifact. Effort and difficulty are prominent during the whole coding process and require motivational goals and determination from a child to commit themselves to the learning. This is an expected notion, as *"You can't learn bread-and-butter (basic) skills if you come to them with fear and the anticipation of hating them"* (Papert, 1980). The design of the coding activity for our workshop had an overall cognitive load that could become overwhelming for children, especially those who were novices to coding. From the complexity of the task, children might reach a point of feeling overloaded, which can lead to a critical condition where, without the proper pleasant and motivating environment, the learning experience can fail. It is not a surprising result that the children with more difficulties and cognitive load had lower scores in their attitudes.

Social aspect of creating an artifact: The "social" dimension refers to the role of collaboration in the coding activity. Collaboration and social interaction for a common goal have many benefits, including interacting with others, examining different perspectives, expressing understandings, and interpreting things differently. During the coding activity, young students were encouraged to work collaboratively and create a shared artifact that was meaningful for their peers, too. The results showed that, most of the time, collaboration was efficient. An important aspect of the good collaboration was the fact that the team members knew each other from before and/or had done other projects together. Nevertheless, there were some indications of bad collaboration that caused frustration. Papert's (1980) notion of the importance of social norms and interaction in learning is reflected in his research on samba schools: *"These are not schools as we know them; they are social clubs with memberships that may range from a few hundred to many thousands."* The construction of games and other artifacts is not an isolated action but happens in a social context. This resonates with Papert's (Papert, 1980) notion of social interaction: *"Although the work at the computer is usually private it increases the children's desire for interaction. These children want to get together with* *others engaged in similar activities because they have a lot to talk about. And what they have to say to one another is not limited to talking about their products: Logo is designed to make it easy to tell about the process of making them."*

Powerful thinking (or learning about thinking): Papert (1980) argues that children are able to recognize different procedures in code, understand when the code does not run as expected, use debugging strategies, and act intentionally to improve the code. To construct their artifacts during the coding activity, the children worked with programming concepts and practices to successfully complete their task. Making a game requires deep engagement and strategy use to successfully manage the completion of the task. The children iteratively organized and documented their code. As described by Papert (1980, p. 28) regarding the Logo environment: *"teaching the Turtle to act or to 'think' can lead one to reflect on one's own actions and thinking. And as children move on, they program the computer to make more complex decisions and find themselves engaged in reflecting on more complex aspects of their own thinking."* To construct the artifacts, young students had the opportunity to plan, solve problems, code, debug, collaborate, communicate, and reflect on their coding experience using Scratch. The participants realized that this was an iterative process, and for some it appeared to be difficult and challenging. Some found it fun to try out the different blocks, discovering the different functionalities. Whatever they made seemed to be suitable for their code; at the same time, the need to add a new function changed everything and triggered a new thinking and debugging process. Coding in Scratch enables young students to articulate their thoughts and watch the outcomes of their own decisions. After the initial trials with coding, by being more and more engaged in the process, the children had the opportunity to clarify their thinking and interpret the immediate feedback, acting accordingly.

Use of powerful ideas: "Powerful ideas," as described by Papert (1980), are central concepts of learning and should be a necessary part of constructionist activities. A "powerful idea" must be both personally and epistemologically useful, giving the opportunity to organize a way of thinking, appropriate time for each specific task, and build on previously gained skills and knowledge. Learners need to be highly explorative before they gain expertise; therefore, the task they are required to do needs to be engaging enough for them to commit to the learning process. In his book *Mindstorms*, Papert shows the importance of powerfulness and the powerful nature of children's use of computers as tools and the Logo programming language, as well all the powerful ideas that emerge from children's engagement with computer-based activities. It is important to make a powerful idea part of intuitive thinking (Papert, 1980). "Powerful" was a quality gained from the girls, as they were allowed to engage closely with the creation of the artifacts in multiple stages, using Scratch. This process brought the learners in touch with some powerful general ideas, such as planning an exciting project, using programming instructions, debugging, and designing, to mention a few. The participants had an experience outside of the classroom, collaborating with students of similar ages but with varied interests and background knowledge, which was in contrast with a single classroom experience. The workshops' durations were critical, not only for learning purposes but also because they allowed participants to bond and exchange interests, giving them the proper amount of time to interact, negotiate, learn from each other, and finally achieve the goal of creating the artifact. Young students gradually discovered the Scratch tool and how they could use it. Becoming more engaged in the process and seeing their artifact become a reality enhanced their feelings of self-achievement and self-confidence. This highlights the importance of finding themselves confronting difficulties and learning things that they did not know about game design. Using Scratch, created new possibilities and made the users "walk it through" and relate their personal knowledge to thinking, effectively and happily, to achieve the artifact construction.

Implications for SDT and CLT

Based on previous research and the theoretical grounding, as presented in Chapter 2 regarding Cognitive load theory (CLT) and Self-Determination Theory (SDT), we assume that cognitive load is related to young student's attitudes and motivation in creative coding activities. Our findings demonstrate that the way children perceive the cognitive load from the learning process is related to their attitudes. According to CLT's relation to learning, instruction should align with human cognitive architecture (Sweller et al., 1998) as well as enhance the motivation of learners (Paas et al., 2005). Motivation and positive consequences are related (Ryan & Deci, 2000a), so self-determined children who are excited when performing a task may have a higher possibility of repeating the task in the future.

Our studies also verify and extend the work of (Abeysekera & Dawson, 2015), who suggest combining CLT and SDT to create a theoretical model for the flipped classroom, which investigates increasing motivation to better manage cognitive load. We confirm the fact that motivated young students with positive attitudes have better management of cognitive load, as was represented by their eye movements. Indeed, we examine the two theories in the different context of children's coding activity, providing empirical support. Moreover, including eye-tracking data in our study design expands the scope of the theories, providing evidence from the use of an objective data-collection method. In addition, other studies using eye-tracking have mainly focused on multimedia learning theories directly related to vision (Mayer, 2010) but from our perspective, including SDT shows evidence that goes deeper into users' behavior.

Measuring cognitive load is widely used in the intersection of eye-tracking and learning. It is often associated with learning and is an important proxy of learning progress. This is an indication of the importance of relaxation of mental effort during learning and is grounded on the importance of the learning process. For meaningful learning to occur, the cognitive process/load of the activity should be moderate; or, as Csikszentmihalyi (1997) demonstrated, achieving the optimal or flow experience that neither frustrates nor demurs the learner. This can be achieved by selecting relevant information, organizing it into coherent mental representations, integrating it with prior knowledge, and also helping children to relax while designing the activity in a way that mixes mentally heavy with playful and social elements.

5.4 Limitations

This thesis has some limitations. First, our workshops were designed for young students who had no previous experience of coding. The participants were randomly selected; therefore, the sample was not consistent in terms of the students' prior knowledge and interaction with coding. Even though we had an indication in our data collection to measure the young students' previous knowledge, we could have used other methods to be more accurate. Second, the workshops of cycles one and two varied in duration because of time constraints related to whether the participants had been recruited form schools or local coding clubs. In all workshops, the participants completed activities successfully. Third, the factors that might affect children's self-perceptions are much more complex than we might assume. Fourth, although the participants of the third cycle were committed for the two days' workshop and gave us high quality data, the sample is not large; this is due to unexpected matters from the participants' side prior to the scheduled dates of the workshop. In addition, the age range of the students in the studies conducted during this PhD work is wide (8–17); focusing on a smaller range may have given a different perspective. Demographic variables and other characteristics (cognitive and motivational) that distinguish them from the rest of the population could have confounded the findings. More precisely, the participants in our study were randomly selected volunteers from our region in Norway; other sampling methods and demographic variables might have a different effect on students' experience. In addition, one specific aspect of the Norwegian reality is that the Ministry of Education has an ongoing process of integrating coding as an elective subject at schools; this started as a pilot program in 2017. Seventh, artifacts like games might be imperfect examples of what children learn, especially when they receive help during the process. Despite the fact that we observed the teams and made notes on the help they received, we might have underestimated or overestimated their understanding of programming concepts.

In addition, limitations due to the types of data-collection methods and instruments used apply in our case. One limitation is related to the eye-tracking: the young age of the participants, their enthusiasm during the activities, and the fact that eye-trackers are designed for adults made it difficult to gather very good quality data. Moreover, in the coding activities, we used Scratch as a programming environment to develop the artifact: another technological tool might have had a different impact on the children's experience. In general, the specific design and context of the activity (i.e., the use of the Scratch tool, the coding tasks, the duration, and the other characteristics), as well as the sampling method used, limit the generalization of our findings. Although we tried to apply all aspects of the DBR methodology in our study, showing the relationship between theory and practice, there were still some limitations. The data were extensive and comprehensive, requiring extended time for collection and analysis; consequently, because time and resources were limited, some data might have been discarded or received less attention. Also, thinking that all methods have bias and weaknesses, using mixed methods, we tried to neutralize those weaknesses. Lastly, we defined in detail the setting of our study and how theory was linked with the context; by default, this has a bias, as it presents our own understanding of contextualizing the theory.

5.5 Evaluating the research

5.5.1 Internal validity of the experiments

Internal validity is considered the degree to which an experimental design controls extraneous variables (Gall, 1993). An extraneous variable would influence and weaken the internal validity inferring the causal relationships of the variables examined in a study. During experiments, researchers should be aware of aspects that may threaten the validity of their study and control them. Below, I discuss how actions taken in the studies conducted in this PhD research preserved their internal validity depending on the threats relevant in my case.

History threat: This threat is present when events other than the treatments that occur during the experimental period can influence results. All young students, as the participants in our making-based coding workshops - following the basic requirements of the study - received the exact same treatment. Participants were randomly assigned, and the instructors had the exact same attitude, using the same materials. However, in some of the studies, deviation from the usual duration of the coding workshop was necessary due to issues like the time constraints or the young ages of the participants.

Maturation threat: It is more relevant to long-term studies since it refers to physical or psychological changes taking place within the subjects as they age. For our short-term studies, the maturation threat is reflected on whether the young students would have had the exact same outcome even if they had never experienced our coding workshops. For the purpose of the studies, we ensured that the variables used were referring to the experience the young students had and were clearly measuring outcomes deriving from young students' learning experiences during their participation in our coding workshops.

Testing and instrumentation threat: Those threats refer to variables measured from a pre-post assessment, and inconsistent use of testing instruments, i.e., if a pre-test influences post-test performance and if there was inconsistent use of testing instruments or testing conditions during the experiment. In my case, the instruments and testing used were performed in a consistent manner, without any changes to the instruments in all studies conducted. Regarding the pre-post-knowledge acquisition test used to measure young students' relative learning gain from the coding workshops, there might be a chance that the young students performed better in the post test due merely to practice. However, this chance was decreased as the participants didn't know that they will fill in a post test at the end of the activity. In addition, the pre-test was given to them at the very beginning, before they started the activity, and knew that they would be using Scratch, or what the tasks would be about.

Mortality threat: This threat refers to the loss of subjects from one or more treatments during the period of the study, which would bias the results. In our studies, to prevent possible dropouts, we took care informing the subjects of the studies about the data-collection process. This was an action taken also, as we had to collect the consent forms from the children and the guardians. After the data collection, particularly for the pre-post-test measures, we checked for any not-filled in tests. There was no major dropout, only very few cases in which either the pre- or post-test was not completed. Those cases were just removed from the data analysis process and didn't affect the study results.

5.5.2 Construct validity

Construct validity questions whether the methods measure what they are intended to measure (Robson & McCartan, 2016). More particularly, researchers first essentially try to understand, based on theory, how constructs and measures behave and relate to each other. Then, the challenge is to provide evidence, through observation, that the programs or measures actually behave that way in reality.

When designing and conducting the studies in this PhD work, other researchers, HCI, TEL experts and instructors were consulted in the development of the research design and have also evaluated the data collected from the studies, finding them representative of real-life situations. In our case, we collected data using multiple sources (e.g., interviews, questionnaires, observations, artifacts created) including objective measures (e.g., eye-tracking). The data collection increased construct validity and guarded against researcher bias. Statements in the questionnaires and the interviews were inspected and agreed upon by at least two researchers. Peer reviews were conducted to ensure that data about relevant concepts would be collected, and that words were unambiguous, specific, and objective. Both qualitative and quantitative data were analyzed and cross checked for triangulation.

There were more than 250 participants in the coding workshops, and even more during the previous years in which the researchers and instructors have been experimenting with the specific concept. Our aim was to avoid subjective judgments during the periods of research design and data collection. During the design and implementation of the studies, the researchers were in close and personal contact with the instructors and the participants involved. This helped establish construct validity by 1) having a set of constructs with the meaning we had chosen, 2) having control of the operationalization of a construct and that it looks the way it should theoretically look, and 3) supporting our theoretical view of the relations among constructs.

5.5.3 External validity

External validity is related to generalizing. In simple words, external validity is the degree to with the conclusions from a study would hold for other people in other places at other times.

In order to improve external validity in our studies, we followed several actions. First, we used a random selection of our sample. The young students who participated in our workshops were randomly selected and volunteered, coming from Trondheim region in Norway. Once the participants were selected, we tried and managed to have almost zero dropouts. The studies were conducted with different students at different places (occasionally) and times; in that way external validity became stronger the more we replicated our study. However, the fact that the study took place in Norway has some limitations as demographic variables (i.e., educational level, family status) might have an effect on young students' behavior. Also, our coding activity is designed for students who have no previous experience in coding, so everyone can attend. Nevertheless, we cannot know the actual level of children's coding skills and exactly how much they had been exposed to coding before at school and/or in-home activities. We followed a very detailed description of the sample at each study; this allowed us to collect as much data as possible for specific characteristics of the participants (if any) that we took under consideration in the data analysis process and report them in the results.

During the data-collection process, our efforts helped us increase external validity. The participants in our studies were not aware of anticipated outcomes. The young ages of our participants made them react very honestly. Our activity was not a part of school assessment process and that made it easy for the young students to relax and forget about the data-collection process and the presence of the researchers. To assure that they were allowed to speak and answer frankly when needed, the researchers repeatedly reminded them that they could express anything they thought of without fearing any consequences. When conducting the questionnaires and interviews, we collected raw data, but the credibility of said data depended on the participants. However, the age gap between the participants and the researchers and instructors was not that evident, possibly contributing to a strengthened trust between them.

5.5.4 Reliability

Reliability in a research study is about the trustworthiness of the procedures. In simple words, reliability is the "consistency" and "repeatability" of the process. It refers to how consistent the results of a study would be if other researchers reconstructed the research process under similar conditions (B. Johnson & Christensen, 2008). In order to secure reliability, we need to have transparency about the research process.

To overcome researcher bias, two or more researchers were involved in the data collection and analysis process. Multiple researchers were continually communicating about methodological decisions. For example, in the qualitative data, researchers were verifying how much agreement there is about findings and analysis. In addition, we had intensive engagement with the data. This action helped us to link the data with our interpretation and then move backward and forward, if needed, as well as conduct retrospective analysis. Also, we kept detailed notes on the decisions made throughout the process, explaining and justifying our choices. Furthermore, transcriptions, artifacts, field notes, audio, and video have been kept and secured to confirm the findings of this study (in accordance with the permissions of the Norwegian Research Council).

Data in real-life events, like the ones collected in our studies, can raise some problems as subjectivity exists. Even if there is a precise guide for the steps followed, data collected by different researchers may not converge into one consistent picture.

5.5.5 Ecological validity

Ecological validity refers to whether the research is representative of a situation in real-life situations.

Our studies have high ecological validity as all happened in real-life settings. The making-based coding workshops were designed and conducted having as a main goal to offer a fruitful learning experience to participants who were young students from Trondheim region in Norway. The research design in the studies was carefully decided, having, as much as possible, a non-invasive process and collecting as much data as possible without affecting the real experience children had during their participation. It is worth mentioning that regarding the collection of young students' gaze during the coding activity, we used eye-tracking glasses, which are non-invasive and allow us to track children's on-screen and off-screen gaze. To automatically extract the necessary features from our data, we also put QR codes in the main Areas of Interest (AOIs). This data-collection method was the one that possibly affected the realism of a real-life situation.

6 Conclusions and future work

6.1 Concluding remarks

The overall research aim of this PhD thesis was to investigate how making can help us design meaningful coding learning experiences for young students. This was conducted over a four-year research process, using a DBR research approach, with three cycles. Based on literature review and theories, we designed and evaluated coding activities for young students through field studies. Those happened in real-world contexts of coding workshops organized for students in the Trondheim region. The studies, held between Autumn 2016 and Autumn 2017, involved data from 157 young students from 3rd to 12th grade (8–17 years old). Quantitative and qualitative data were gathered from instructors, young students, and researchers to obtain information about the learning experience provided from our workshops. We used qualitative sources such as observations, semi-structured interviews and focus groups, students' artifacts, and quantitative sources such as questionnaires and gaze from eye-tracking, among others. Data were respectively analyzed, and cross checked for triangulation. We focused on how our making-based coding activities enhanced participants' knowledge of basic programming concepts, their coding behavior, their social interaction and collaboration, and how they perceived their coding experience as a whole when introduced to coding.

Throughout the PhD work, the aim was to answer the main research question: *How can making help us design meaningful coding learning experiences for young students?* and the following research sub-questions. In the previous chapter, the contributions of this PhD thesis were discussed while, in this section, the main conclusions related to the research questions are presented.

SQ1. What is the state-of-the art of making activities related to coding?

Through a systematic literature review methodology, we selected and finally analyzed papers from searches over the past five years. An increasing number of papers showed that *making* is gaining momentum in covering a wide range of approaches in the area of learning in both formal and informal education settings. More specifically, we focused on the areas that include, among others, the types of technologies used to support making types of instructions, sample size, ages of the participants, types of methodology, and areas of interest in each study.

A clear finding of the review is that many of the studies that have integrated making sessions are mainly in the area of coding and other STEM subjects. In addition, we regarded making only in a positive way, as almost none of the studies reported negative effects, but also, there is a highlighted need for a more in-depth analysis. In addition, we showed that the effects of making are examined in various aspects as they have been approached in the different studies, like engagement, self-efficacy, performance, and collaboration. The findings showed that there is a direction in which different types of making activities could have an impact on educational approaches and lead to a more effective way of teaching and learning.

SQ2. *What is the role of young students' attitudes and engagement on their learning experience during making-based coding activities?*

The studies conducted helped us to explore the roles of attitudes and engagement in our coding workshops with young students. The results demonstrate a significant relation between attitudes and children's gaze patterns during the coding activity. More specific, children who indicated better management of cognitive load expressed higher scores in their attitudes. Findings also suggest that children with higher reported excitement and perceived learning had the same characteristics. Also, we demonstrated that using eye-tracking provides information about children's approaches to handling coding tasks. Appropriate teaching methods and tools should focus on providing support, avoiding unnecessary disruptions that can become overwhelming.

It is important to have appropriate educational designs aiming to promote active learning with the support of making activities. Including components like a balance of individual and social involvement, and the use of a visual programming language, all employed under the common goal of creating an artifact, fosters young students' deeper transferable CT skills, which are vital for our society's information revolution. Behavioral, emotional, and cognitive engagement are interrelated within the individual and are important elements in the interaction with the instructor, the learning tool and interaction with other students in the creation of an artifact. Engaging young students in a learning environment that embraces creative design, problem solving, collaboration, and communication strengthens their sense of competence and confidence. Together with achieving a significant improvement in students' understanding of computational knowledge, like programming concepts and practices, it is essential to create high levels of motivation, fun, and commitment as part of an efficient pedagogical design, as reflected in our studies.

SQ3. What is the role of student's age and gender on how they experience making-based coding activities?

Recording young students' gaze and measuring their learning gain, we analyzed young students' learning processes in coding and discover any differences in their task-based behavior according to their ages. The teens (13–17 years old) in our sample had higher RLG compared to the kids and tackled coding in a different way. The kids (8–12 years old) focused more on the appearance of the characters, while teens followed more structured behavior, comprised of basic programming practices such as debugging and testing. In addition, the gaze showed that the teen teams collaborated better during the activity, a fact that might have led to their higher learning gain. Moreover, we introduced new means and measures by which to understand how children learn coding.

Regarding gender differences, our findings provide objective evidence that female students do not lack in competences compared to boys, but simply that they have a different approach during coding activities and different perspectives about coding, an approach that needs to be cultivated and nurtured. More precisely, there are no statistically significant differences between female and male gaze and learning gain during the coding activity; interestingly, the qualitative data show differences in the strategies and implemented practices during coding and in perceptions about those coding activities. Our results highlight that further studies need to use objective measures and unveil necessary differences in the design and implementation of coding activities.

SQ4. What principles can enhance making-based coding activities to support students' learning experience?

Based on the knowledge gained from the studies, following the DBR research approach and grounding our results on the literature, we identified design principles to facilitate making-based coding activities that offer a fruitful learning experience to the students.

To conclude, the key design principles are: (1) provide proper assistance: a) help students focus on structured coding behavior, b) avoid cognitive load during the activities that can cause frustration, c) keep them motivated, and d) have instructors that can act as role models, potentially to attract more females; (2) enhance social interaction and collaboration, not only within, but also among the teams and their members; (3) allow an adequate amount of time for engagement during the workshop, focusing on the design part of the game in a structured way; (4) include physical and digital artifacts; and (5) relate the activity to meaningful content and include the appropriate tasks.

6.2 Directions for future work

The work presented in this thesis suggests several streams for future research. Future plans could include conducting our coding workshops in school settings to explore their effects under a traditional teaching approach. Among other aspects, researchers could explore the correlations with students' performance in the form of grades. Another extension of this contribution can be to look into the dynamics of the gaze behavior to observe how engagement and collaboration evolve for different groups of children with different characteristics (e.g., competence in coding, experience, age groups etc.). One possibility would be to investigate, in more detail, specific gaze patterns of boys and girls; another would be to examine collaborative eye-tracking measures and group dynamics in both mixed and non-mixed teams of boys and girls. An interesting approach is to compare the effects of different learning environments on gender. Furthermore, other objective measures could be used to gain a deep understanding of the relationship between coding behavior and gender, ages, or other characteristics.

In terms of learning-specific computer science concepts and how they are related to young students' engagement, attitudes, and different gaze patterns. In addition, the study could be extended to compare the results from children's gaze patterns in other attitudes as well as comparing alternative coding learning environments. In addition, regarding theory, it would be interesting to see more studies in the area that ground their findings in constructionism or/and the theories used in this PhD work. This would bring researchers in the same area together to build a common ground regarding outcomes.

Learning with technology is a complex process that is associated with many aspects of interaction (e.g., cognitive load, attention, etc.). The complexity of this process means that it is likely that no single data modality can paint a complete picture of the learning experience, requiring multiple data streams to complement each other. Therefore, future research should focus more on the great potential of multimodal data–and in particular physiological data–to understand users' cognition, emotion, attention, information acquisition, and more.

7 References

Abeysekera, L., & Dawson, P. (2015). Motivation and cognitive load in the flipped classroom: definition, rationale and a call for research. *Higher Education Research & Development, 34*(1), 1-14.

Adams, J. C., & Webster, A. R. (2012). *What do students learn about programming from game, music video, and storytelling projects?* Paper presented at the Proceedings of the 43rd ACM technical symposium on Computer Science Education.

Adya, M., & Kaiser, K. M. (2005). Early determinants of women in the IT workforce: a model of girls' career choices. *Information Technology & People, 18*(3), 230-259.

Anderson, T. (2005). Design-based research and its application to a call centre innovation in distance education. *Canadian Journal of Learning and Technology/La revue canadienne de l'apprentissage et de la technologie, 31*(2).

Anderson, T., & Shattuck, J. (2012). Design-based research: A decade of progress in education research? *Educational Researcher, 41*(1), 16-25.

Aschwanden, C., & Crosby, M. (2006). *Code scanning patterns in program comprehension.* Paper presented at the Proceedings of the 39th Hawaii International Conference on System Sciences.

Bandura, A. (1997). *Self-efficacy: The exercise of control*: Macmillan.

Barab, S., & Squire, K. (2004). Design-based research: Putting a stake in the ground. *The Journal of the learning sciences, 13*(1), 1-14.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *Acm Inroads, 2*(1), 48-54.

Beckwith, L., Burnett, M., Wiedenbeck, S., Cook, C., Sorte, S., & Hastings, M. (2005). *Effectiveness of end-user debugging software features: Are there gender issues?* Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

Bednarik, R. (2012). Expertise-dependent visual attention strategies develop over time during debugging with multiple code representations. *International Journal of Human-Computer Studies, 70*(2), 143-155.

Bednarik, R., Myller, N., Sutinen, E., & Tukiainen, M. (2006). *Program visualization: Comparing eye-tracking patterns with comprehension summaries and performance.* Paper presented at the Proceedings of the 18th Annual Psychology of Programming Workshop.

Bednarik, R., & Tukiainen, M. (2006). *An eye-tracking methodology for characterizing program comprehension processes.* Paper presented at the Proceedings of the 2006 symposium on Eye tracking research & applications.

Bers, M. U. (2012). *Designing digital experiences for positive youth development: From playpen to playground*: Oxford University Press.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education, 72*, 145-157.

Biggs, J. B., & Collis, K. F. (2014). *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome)*: Academic Press.

Blikstein, P. (2013). *Gears of our childhood: constructionist toolkits, robotics, and physical computing, past and future.* Paper presented at the Proceedings of the 12th international conference on interaction design and children.

Blunsdon, B., Reed, K., McNeil, N., & McEachern, S. (2003). Experiential learning in social science theory: An investigation of the relationship between student enjoyment and learning. *Higher Education Research & Development, 22*(1), 43-56.

Bonner, D., & Dorneich, M. (2016). *Developing game-based learning requirements to increase female middle school students interest in computer science.* Paper presented at the Proceedings of the human factors and ergonomics society annual meeting.

Brennan, K., & Resnick, M. (2012). New frameworks for studying society computer science exemplification project. *Paper presented at AERA*.

Bruckman, A., Jensen, C., & DeBonte, A. (2002). *Gender and programming achievement in a CSCL environment.* Paper presented at the Proceedings of the conference on computer support for collaborative learning: Foundations for a CSCL community.

Buechley, L., Eisenberg, M., Catchen, J., & Crockett, A. (2008). *The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education.* Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.

Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education, 4*(2), 121-135.

Çakır, N. A., Gass, A., Foster, A., & Lee, F. J. (2017). Development of a game-design workshop to promote young girls' interest towards computing through identity exploration. *Computers & Education, 108*, 115-130.

Carver, S., & Mayer, R. (1988). Learning and transfer of debugging skills: Applying task analysis to curriculum design and assessment. *Teaching and Learning Computer Programming: Multiple Research Perspectives. RE Mayer. Hillsdale, NJ, Lawrence Erlbaum Associates*, 259-297.

Cazzato, V., Basso, D., Cutini, S., & Bisiacchi, P. (2010). Gender differences in visuospatial planning: An eye movements study. *Behavioural Brain Research, 206*(2), 177-183.

Ceci, S. J., & Williams, W. M. (2010). Sex differences in math-intensive fields. *Current Directions in Psychological Science, 19*(5), 275-279.

Cheryan, S., Ziegler, S. A., Montoya, A. K., & Jiang, L. (2017). Why are some STEM fields more gender balanced than others? *Psychological bulletin, 143*(1), 1.

Chu, S. L., Angello, G., Saenz, M., & Quek, F. (2017). Fun in Making: Understanding the experience of fun and learning through curriculum-based Making in the elementary school classroom. *Entertainment Computing, 18*, 31-40.

Chu, S. L., Quek, F., Bhangaonkar, S., Ging, A. B., & Sridharamurthy, K. (2015). Making the Maker: A Means-to-an-Ends approach to nurturing the Maker mindset in elementary-aged children. *International Journal of Child-Computer Interaction, 5*, 11-19.

Chu, S. L., Schlegel, R., Quek, F., Christy, A., & Chen, K. (2017). *'I Make, Therefore I Am': The Effects of Curriculum-Aligned Making on Children's Self-Identity.* Paper presented at the Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.

Conover, W. J., & Conover, W. J. (1980). Practical nonparametric statistics.

Creswell, J. W. (2014). *A concise introduction to mixed methods research*: Sage Publications.

Creswell, J. W., & Clark, V. L. P. (2017). *Designing and conducting mixed methods research*: Sage publications.

Csikszentmihalyi, M. (1997). Flow and the psychology of discovery and invention. *HarperPerennial, New York, 39*.

Denner, J. (2007). The Girls Creating Games Program: An innovative approach to integrating technology into middle school. *Meridian: A Middle School Computer Technologies Journal, 1*(10).

Denner, J., Bean, S., & Martinez, J. (2009). The girl game company: Engaging latina girls in information technology. *Afterschool Matters, 8*, 26-35.

Denner, J., & Werner, L. (2007). Computer programming in middle school: How pairs respond to challenges. *Journal of Educational Computing Research, 37*(2), 131-150.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education, 58*(1), 240-249. doi:10.1016/j.compedu.2011.08.006

Doran, K., Boyce, A., Finkelstein, S., & Barnes, T. (2012). *Outreach for improved student performance: a game design and development curriculum.* Paper presented at the Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education.

Eckstein, M. K., Guerra-Carrillo, B., Singley, A. T. M., & Bunge, S. A. (2016). Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development? *Developmental cognitive neuroscience*.

Eordanidis, S., Gee, E., & Carmichael, G. (2017). The effectiveness of pairing analog and digital games to teach computer science principles to female youth. *Journal of Computing Sciences in colleges, 32*(3), 12-19.

Ericson, B., & McKlin, T. (2012). *Effective and sustainable computing summer camps.* Paper presented at the Proceedings of the 43rd ACM technical symposium on Computer Science Education.

Esper, S., Foster, S. R., Griswold, W. G., Herrera, C., & Snyder, W. (2014). *CodeSpells: bridging educational language features with industry-standard languages.* Paper presented at the Proceedings of the 14th Koli Calling International Conference on Computing Education Research.

Fay, A. L., & Mayer, R. E. (1994). Benefits of teaching design skills before teaching logo computer programming: Evidence for syntax-independent learning. *Journal of Educational Computing Research, 11*(3), 187-210.

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education, 63*, 87-97. doi:10.1016/j.compedu.2012.11.016

Fields, D., Vasudevan, V., & Kafai, Y. B. (2015). The programmers' collective: fostering participatory culture by making music videos in a high school Scratch coding workshop. *Interactive Learning Environments, 23*(5), 613-633.

Fortus, D. (2014). Attending to affect. *Journal of research in science teaching, 51*(7), 821-835.

Franklin, D., Conrad, P., Boe, B., Nilsen, K., Hill, C., Len, M., . . . Kiefer, B. (2013). *Assessment of computer science learning in a scratch-based outreach program.* Paper presented at the Proceeding of the 44th ACM technical symposium on Computer science education.

Gall, W. R. B. J. P. G. M. D. (1993). Applying educational research (3rd Ed.). *New York: Longman.*

Garneli, B., Giannakos, M. N., Chorianopoulos, K., & Jaccheri, L. (2013). *Learning by playing and learning by making.* Paper presented at the International Conference on Serious Games Development and Applications.

Giannakos, M. N., & Jaccheri, L. (2018). From players to makers: An empirical examination of factors that affect creative game development. *International Journal of Child-Computer Interaction*.

Grover, S., Cooper, S., & Pea, R. (2014). *Assessing computational learning in K-12.* Paper presented at the Proceedings of the 2014 conference on Innovation & technology in computer science education.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher, 42*(1), 38-43. doi:10.3102/0013189X12463051

Guay, F., Ratelle, C. F., & Chanal, J. (2008). Optimal learning in optimal contexts: The role of self-determination in education. *Canadian Psychology/Psychologie canadienne, 49*(3), 233.

Guzdial, M. (2008). Education Paving the way for computational thinking. *Communications of the ACM, 51*(8), 25-27.

Hay, K. E., & Barab, S. A. (2001). Constructivism in practice: A comparison and contrast of apprenticeship and constructionist learning environments. *The Journal of the learning sciences, 10*(3), 281-322.

Hegarty, M., & Just, M.-A. (1993). Constructing mental models of machines from text and diagrams. *Journal of memory and language, 32*(6), 717-742.

Hejmady, P., & Narayanan, N. H. (2012). *Visual attention patterns during program debugging with an IDE.* Paper presented at the Proceedings of the Symposium on Eye Tracking Research and Applications.

Henrie, C. R., Halverson, L. R., & Graham, C. R. (2015). Measuring student engagement in technology-mediated learning: A review. *Computers & Education, 90*, 36-53.

Horizon, M. (October 5, 2015). Horizon Media study reveals Americans prioritize STEM subjects over the arts; science is "cool," coding is new literacy. PR Newswire. . *Retrieved from* http://www.prnewswire.com/news-releases/horizon-mediastudy-reveals-americans-prioritize-stem-subjects-over-the-arts-science-is-cool-coding-is-new-literacy-300154137.html.

Hubwieser, P., Armoni, M., Giannakos, M. N., & Mittermeir, R. T. (2014). Perspectives and visions of computer science education in primary and secondary (K-12) Schools. *ACM Transactions on Computing Education (TOCE), 14*(2), 7.

Iversen, O. S., Smith, R. C., Blikstein, P., Katterfeldt, E.-S., & Read, J. C. (2016). Digital fabrication in education: Expanding the research towards design and reflective practices. *International Journal of Child-Computer Interaction, 5*, 1-2. doi:10.1016/j.ijcci.2016.01.001

Jarodzka, H., Scheiter, K., Gerjets, P., & Van Gog, T. (2010). In the eyes of the beholder: How experts and novices interpret dynamic stimuli. *Learning and Instruction, 20*(2), 146-154.

Jermann, P., & Nüssli, M.-A. (2012). *Effects of sharing text selections on gaze cross-recurrence and interaction quality in a pair programming task.* Paper presented at the Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work.

Johnson, B., & Christensen, L. (2008). *Educational research: Quantitative, qualitative, and mixed approaches*: Sage.

Johnson, L., Adams Becker, S., Estrada, V., & Freeman, A. (2015). *The NMC Horizon Report: 2015 Museum Edition*: ERIC.

Just, M. A., & Carpenter, P. A. (1976). The role of eye-fixation research in cognitive psychology. *Behavior Research Methods, 8*(2), 139-143.

Just, M. A., & Carpenter, P. A. (1980). A theory of reading: From eye fixations to comprehension. *Psychological review, 87*(4), 329.

Just, M. A., & Carpenter, P. A. (1984). Using eye fixations to study reading comprehension. *New methods in reading comprehension research*, 151-182.

Kafai, Y., & Vasudevan, V. (2015). *Hi-Lo tech games: crafting, coding and collaboration of augmented board games by high school youth.* Paper presented at the Proceedings of the 14th International Conference on Interaction Design and Children.

Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and culture, 1*(1), 36-40.

Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational psychologist, 50*(4), 313-334. doi:10.1080/00461520.2015.1124022

Kafai, Y. B., & Kafai, Y. B. (1995). *Minds in play: Computer game design as a context for children's learning*: Routledge.

Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE), 14*(1), 1.

Kafai, Y. B., & Resnick, M. (2012). Introduction. In *Constructionism in practice* (pp. 13-20): Routledge.

Kafai, Y. B., Searle, K., Kaplan, E., Fields, D., Lee, E., & Lui, D. (2013). *Cupcake cushions, scooby doo shirts, and soft boomboxes: e-textiles in high school to promote computational concepts, practices, and perceptions.* Paper presented at the Proceeding of the 44th ACM technical symposium on Computer science education.

Kafai, Y. B., & Vasudevan, V. (2015). *Constructionist gaming beyond the screen: Middle school students' crafting and computing of touchpads, board games, and controllers.* Paper presented at the Proceedings of the Workshop in Primary and Secondary Computing Education.

Kahn, K. (2004). Toontalk-steps towards ideal computer-based learning environments. *A Learning Zone of One's Own: Sharing Representations and Flow in Collaborative Learning Environments*, 253-270.

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. *Computers in Human Behavior, 52*, 200-210.

Katterfeldt, E.-S., Dittert, N., & Schelhowe, H. (2015). Designing digital fabrication learning environments for Bildung: Implications from ten years of physical computing workshops. *International Journal of Child-Computer Interaction, 5*, 3-10.

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal, 41*(4), 245-255.

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR), 37*(2), 83-137.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University, 33*(2004), 1-26.

Lane, H. C., Cahill, C., Foutz, S., Auerbach, D., Noren, D., Lussenhop, C., & Swartout, W. (2013). *The effects of a pedagogical agent for informal science education on learner behaviors and self-efficacy.* Paper presented at the International Conference on Artificial Intelligence in Education.

Laporte, L., & Zaman, B. (2016). *Informing Content-driven Design of Computer Programming Games: a Problems Analysis and a Game Review.* Paper presented at the Proceedings of the 9th Nordic Conference on Human-Computer Interaction.

Lee, E., Kafai, Y. B., Vasudevan, V., & Davis, R. L. (2014). Playing in the arcade: Designing tangible interfaces with MaKey MaKey for Scratch games. In *Playful user interfaces* (pp. 277-292): Springer.

Lin, J. M.-C., & Liu, S.-F. (2012). An investigation into parent-child collaboration in learning computer programming. *Journal of Educational Technology & Society, 15*(1).

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51-61.

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). *Programming by choice: urban youth learning programming with scratch* (Vol. 40). Proceedings of the 39th SIGCSE technical symposium on Computer science education ACM.

Master, A., Cheryan, S., Moscatelli, A., & Meltzoff, A. N. (2017). Programming experience promotes higher STEM motivation among first-grade girls. *Journal of experimental child psychology, 160*, 92-106.

Mayer, R. E. (2010). Unique contributions of eye-tracking research to the study of learning with graphics. *Learning and Instruction, 20*(2), 167-171.

Mayring, P. (2014). Qualitative content analysis: theoretical foundation, basic procedures and software solution.
McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM, 49*(8), 90-95.
Mellis, D. A., Jacoby, S., Buechley, L., Perner-Wilson, H., & Qi, J. (2013). *Microcontrollers as material: crafting circuits with paper, conductive ink, electronic components, and an untoolkit.* Paper presented at the Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction.
Noroozi, O., Alikhani, I., Järvelä, S., Kirschner, P. A., Juuso, I., & Seppänen, T. (2018). Multimodal data to design visual learning analytics for understanding regulation of learning. *Computers in Human Behavior*.
NRC. (2012). A framework for K-12 science education: Practices, crosscutting concepts, and core ideas. *The National Academies Press*.
Owston, R., Wideman, H., Ronda, N. S., & Brown, C. (2009). Computer game development as a literacy activity. *Computers & Education, 53*(3), 977-989.
Paas, F., Renkl, A., & Sweller, J. (2004). Cognitive load theory: Instructional implications of the interaction between information structures and cognitive architecture. *Instructional science, 32*(1-2), 1-8.
Paas, F., Tuovinen, J. E., Van Merrienboer, J. J., & Darabi, A. A. (2005). A motivational perspective on the relation between mental effort and performance: Optimizing learner involvement in instruction. *Educational Technology Research and Development, 53*(3), 25-34.
Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2017). Empirical studies on the Maker Movement, a promising approach to learning: A literature review. *Entertainment Computing, 18*, 57-78. doi:10.1016/j.entcom.2016.09.002
Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2019). Exploring children's learning experience in constructionism-based coding activities through design-based research. *Computers in Human Behavior*, 99, 415-427.
Papavlasopoulou, S., Sharma, K., & Giannakos, M. (2019). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. *Computers in Human Behavior*. in press.
Papavlasopoulou, S., Sharma, K., Giannakos, M., & Jaccheri, L. (2017). Using Eye-Tracking to Unveil Differences Between Kids and Teens in Coding Activities. *In Proc. of Interaction Design and Children Conference.* 171-181. (2017) Stanford, California, USA. ACM
Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2018). How do you feel about learning to code? Investigating the effect of children's attitudes towards coding using eye-tracking. *International Journal of Child-Computer Interaction*, 17, 50–60.
Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2018). Discovering children's competences in coding through the analysis of Scratch projects. Paper presented at the Proceedings of the IEEE Global Engineering Education Conference (EDUCON) 1127-1133.
Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*: Basic Books, Inc.

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*: ERIC.

Parmaxi, A., Zaphiris, P., & Ioannou, A. (2016). Enacting artifact-based activities for social technologies in language learning using a design-based research approach. *Computers in Human Behavior, 63*, 556-567.

Parsons, D., & Haden, P. (2006). *Parson's programming puzzles: a fun and effective learning tool for first programming courses.* Paper presented at the Proceedings of the 8th Australasian Conference on Computing Education-Volume 52.

Perner-Wilson, H., Buechley, L., & Satomi, M. (2011). *Handcrafting textile interfaces from a kit-of-no-parts.* Paper presented at the Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction.

Piaget, J. (1954). Language and thought from a genetic perspective. *Acta Psychologica, 10*, 51-60.

Pietinen, S., Bednarik, R., Glotova, T., Tenhunen, V., & Tukiainen, M. (2008). *A method to study visual attention aspects of collaboration: eye-tracking pair programmers simultaneously.* Paper presented at the Proceedings of the 2008 symposium on Eye tracking research & applications.

Pietinen, S., Bednarik, R., & Tukiainen, M. (2010). *Shared visual attention in collaborative programming: a descriptive analysis.* Paper presented at the proceedings of the 2010 ICSE workshop on cooperative and human aspects of software engineering.

Pintrich, P. R. (2003). A motivational science perspective on the role of student motivation in learning and teaching contexts. *Journal of educational Psychology, 95*(4), 667.

Qiu, K., Buechley, L., Baafi, E., & Dubow, W. (2013). *A curriculum for teaching computer science through computational textiles.* Paper presented at the Proceedings of the 12th International Conference on Interaction Design and Children.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological bulletin, 124*(3), 372.

Reeves, T. (2006). Design research from a technology perspective. In *Educational design research* (pp. 64-78): Routledge.

Resnick, M. (2007). *All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten.* Paper presented at the Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition.

Robertson, J. (2012). Making games in the classroom: Benefits and gender concerns. *Computers & Education, 59*(2), 385-398. doi:10.1016/j.compedu.2011.12.020

Robertson, J. (2013). The influence of a game-making project on male and female learners' attitudes to computing. *Computer science education, 23*(1), 58-83.

Robertson, J., & Howells, C. (2008). Computer game design: Opportunities for successful learning. *Computers & Education, 50*(2), 559-578.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education, 13*(2), 137-172. doi:10.1076/csed.13.2.137.14200

Robson, C., & McCartan, K. (2016). *Real world research*: John Wiley & Sons.

Romero, P., Lutz, R., Cox, R., & du Boulay, B. (2002). *Co-ordination of multiple external representations during Java program debugging.* Paper presented at the

Human Centric Computing Languages and Environments, 2002. Proceedings. IEEE 2002 Symposia on.
Rosenberg-Kima, R. B., Plant, E. A., Doerr, C. E., & Baylor, A. L. (2010). The influence of computer-based model's race and gender on female students' attitudes and beliefs towards engineering. *Journal of Engineering Education, 99*(1), 35-44.
Ruggiero, D., & Green, L. (2017). Problem solving through digital game design: A quantitative content analysis. *Computers in Human Behavior, 73*, 28-37.
Ryan, R. M., & Deci, E. L. (2000a). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology, 25*(1), 54-67.
Ryan, R. M., & Deci, E. L. (2000b). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist, 55*(1), 68.
Sáez-López, J.-M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education, 97*, 129-141. doi:10.1016/j.compedu.2016.03.003
Saldaña, J. (2015). *The coding manual for qualitative researchers*: Sage.
Searle, K. A., Fields, D. A., Lui, D. A., & Kafai, Y. B. (2014). *Diversifying high school students' views about computing with electronic textiles.* Paper presented at the Proceedings of the tenth annual conference on International computing education research.
Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika, 52*(3/4), 591-611.
Sharif, B., Falcone, M., & Maletic, J. I. (2012). *An eye-tracking study on the role of scan time in finding source code defects.* Paper presented at the Proceedings of the Symposium on Eye Tracking Research and Applications.
Sharma, K., Caballero, D., Verma, H., Jermann, P., & Dillenbourg, P. (2015). *Looking AT versus looking THROUGH: A dual eye-tracking study in MOOC context.* Paper presented at the Exploring the material conditions of learning: opportunities and challenges for CSCL," the Proceedings of the Computer Supported Collaborative Learning (CSCL) Conference.
Sharma, K., Papavlasopoulou, S., & Giannakos, M. (2019). Coding games and robots to enhance computational thinking: How collaboration and engagement moderate children's attitudes? *International Journal of Child-Computer Interaction*, 21, 65-76.
Sharma, K., Papavlasopoulou, S., & Giannakos, M. (2019). Joint Emotional State of Children and Perceived Collaborative Experience in Coding Activities. *In Proc. of Interaction Design and Children Conference*, 133-145. (IDC' 19), Boise, Idaho, USA, ACM.
Stein, R., & Brennan, S. E. (2004). *Another person's eye gaze as a cue in solving programming problems.* Paper presented at the Proceedings of the 6th international conference on Multimodal interfaces.
Stewart-Gardiner, C., Carmichael, G., Latham, J., Lozano, N., & Greene, J. L. (2013). Influencing middle school girls to study computer science through educational computer games. *Journal of Computing Sciences in colleges, 28*(6), 90-97.

Sweller, J., Van Merrienboer, J. J., & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational psychology review, 10*(3), 251-296.
Teague, J. (2002). Women in computing: What brings them to it, what keeps them in it? *ACM SIGCSE Bulletin, 34*(2), 147-158.
Tsai, M.-J., Hou, H.-T., Lai, M.-L., Liu, W.-Y., & Yang, F.-Y. (2012). Visual attention for solving multiple-choice science problem: An eye-tracking analysis. *Computers & Education, 58*(1), 375-385.
Vos, N., Van Der Meijden, H., & Denessen, E. (2011). Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education, 56*(1), 127-137.
Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development, 53*(4), 5-23.
Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). *The fairy performance assessment: measuring computational thinking in middle school.* Paper presented at the Proceedings of the 43rd ACM technical symposium on Computer Science Education.
Wilson, A., & Moffat, D. C. (2010). Evaluating Scratch to introduce younger schoolchildren to programming. *Proceedings of the 22nd Annual Psychology of Programming Interest Group (Universidad Carlos III de Madrid, Leganés, Spain*.
Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Part II: Research papers

Paper 1:

Empirical studies on the Maker Movement, a promising approach to learning: A literature review

Sofia Papavlasopoulou, Michail N. Giannakos, and Letizia Jaccheri

Entertainment Computing

Entertainment Computing 18 (2017) 57–78

Contents lists available at ScienceDirect

Entertainment Computing

journal homepage: ees.elsevier.com/entcom

Empirical studies on the Maker Movement, a promising approach to learning: A literature review

Sofia Papavlasopoulou *, Michail N. Giannakos, Letizia Jaccheri

Department of Computer & Information Science, Norwegian University of Science and Technology (NTNU), Norway

ARTICLE INFO

Article history:
Received 17 January 2016
Revised 14 July 2016
Accepted 4 September 2016
Available online 6 September 2016

Keywords:
Maker Movement
Making, Literature review
Learning by making
Creativity
Constructionist

ABSTRACT

The Maker Movement has gathered much attention recently, and has been one of the fastest-growing topics, due to contemporary technical and infrastructural developments. The maker culture can be described as a philosophy in which individuals or groups of individuals create artifacts that are recreated and assembled using software and/or physical objects. Typical topics of interest in maker culture include engineering-oriented pursuits such as electronics, robotics, 3D printing, and computer numerical control tools, as well as more traditional activities such as sewing or arts and crafts. Scholars and educators have reported a variety of outcomes from the Maker Movement as an instructional process; however, the lack of a summary of these empirical studies prevents stakeholders from having a clear view of the benefits and challenges of this instructional culture. The purpose of this article is to provide a review of the Maker Movement approach in order to summarize the current findings and guide future studies. Forty-three peer-reviewed articles were collected from a systematic literature search and analyzed based on a categorization of their main elements. The results of this survey show the direction of Maker Movement research during recent years and the most common technologies, subjects, evaluation methods, and pedagogical designs. Suggestions for future research include a further investigation into the benefits of using a specific technological tool and analysis of the Maker Movement approach, particularly in classrooms. These future research efforts will allow us to better indicate which aspects and ingredients of "making" work better for which circumstances and student groups. The findings will ultimately allow us to form best practices and a unified framework for guiding/assisting educators who want to adopt this teaching style.

1. Introduction

During the last few years, the Maker Movement has appeared as a new trend that derives from the general maker culture, which is also described as a philosophy or phenomenon. The definition of the Maker Movement is very broad and builds on an individual's ability to be a creator of things, a "maker." There is a growing community of hobbyists and professionals with diverse skills and interests who make their own functional devices, from technological gadgets to home decorating. The rapid growth of this movement derives from advances in technology and new digital fabrication technologies that allowed the appearance of tools such as wearable computing e-textiles, robotics, 3D printing, microprocessors, and programming languages. Moreover, online communities permit the sharing of tools and ideas that enhance collaboration and the globalization of problem-solvers.

The rise of the Maker Movement and the increase of various maker culture-based initiatives (e.g., Makerspaces, Fablabs, Techshops) have evolved in the sense that the Maker Movement is a technological and creative evolution that has limitless implications for the world of education. In addition, prominent funding agencies support research efforts related to the Maker Movement.[1] The role of making in education in particular has been studied for many decades. First, Papert's constructionism demonstrated the need to learn through creative making processes, as well as discovering knowledge rather than receiving it passively [27]. Making provides

* Corresponding author at: Sem Sælands vei 7-9, NO-7491 Trondheim, Norway.
E-mail addresses: spapav@idi.ntnu.no (S. Papavlasopoulou), michail.giannakos@idi.ntnu.no (M.N. Giannakos), letizia@idi.ntnu.no (L. Jaccheri).

[1] For instance funding from NSF: http://www.nsf.gov/awardsearch/showAward?AWD_ID=1216994&HistoricalAwards=falsehttp://grantome.com/grant/NSF/DRL-1433770.

http://dx.doi.org/10.1016/j.entcom.2016.09.002
1875-9521/

a different perspective in the learning process, as it gives learners the opportunity to have control over their own knowledge, instead of being passive recipients. Martinez and Stager [48] presented three ways of learning, based on the constructionist approach: making, tinkering, and engineering. Their approach distinguishes making from tinkering. Making has very wide content, characterized as "a stance" that is more like an attitude toward learning that sets the learner in the center of the educational process and creates opportunities that students may never have encountered themselves. On the other hand, tinkering appears more specific, as it is characterized as a problem-solving process through discovery. Honey and Kanter [30] include making as one of the main components of learning, in accordance with designing and playing. Making is defined as "building or adapting objects by hand for the simple personal pleasure of figuring out how things work."

The interest in making is mainly focused in educational settings centered around science, technology, engineering, and mathematics (STEM) concepts. In addition, many studies, starting from Papert's Logo programming language [52] and Lego Mindstorms [58], showed the connection between making and the learning principles of engineering, design, and computer programming [57,35,32].

In this study, we assess the recent research concerning the Maker Movement and its emerging role in formal and informal education. For the purpose of this review, we define what we consider as making in order to determine our approach to the Maker Movement. In particular, making involves constructing activities and related ways to fabricate real and/or digital things using technological resources, including fabrication [39], physical computing, and programming. Making focuses on the process that occurs in an environment that is not always merely learning oriented, but promotes design thinking, computational concepts, collaborative work, and innovation, among other things.

Our motivation for this work is based on emerging developments in the area of manufacturing, coding, and learning technologies, creating momentum for adoption of the making approach. We draw on the research literature to consider the trends and possibilities within this movement for teaching and learning. The aim of this literature review is to provide a review of research on the Maker Movement approach in order to summarize the findings, and understand how different types of making activities are related to a successful learning experience in terms of learning, interest, and engagement. Furthermore, we consider how these practices could help students to improve their performance in coding, computational skills, and problem solving. This has been achieved by evaluating empirical studies from the last five years.

Key areas covered in the current review include the types of technology used, previous literature reviews, benefits and challenges of the Maker Movement, methodological concerns, and suggestions for future research. In line with the scope of this review and the fact that it is a maturing field, defining Berlin Model [29] areas is important. Despite using making types of instructions in teaching and learning is not novel, there has been a lack of evidence to identify which are the most commonly used subject areas ("contents" in the Berlin Model) that could be applied. Investigating technologies ("media" in the Berlin Model) to support making activities so far, will provide an overview of the already widely used tools, as well as the promising potential existence of new ones. Also, evaluation is an important aspect since evidence of learning laid the foundation of learning science and decision-makers as well as practitioners and researchers rely their future decisions on it. To guide our research, we used two of the main categories of the Berlin Model (i.e. content and media) and also assessment because it is critical for policy makers in order to adopt making. Consequently, we posed the following initial research questions:

1. What are the most common technologies used in making approaches?
2. Which are the most common subject areas in which to implement a making type of instruction?
3. What are the main measures used to evaluate making types of instruction?

In addition to producing substantive findings regarding the making instructional approach, the review aims to identify potential research gaps as well as make suggestions for future research. Future research efforts will allow us to better indicate which aspects and ingredients of making-type instructions achieve better results in learning, and for which circumstances and student groups.

The rest of the paper is organized as follows. In the next section we present the related work; the third section describes the methodology used for our literature review and how we analyzed the studies selected. The fourth section presents the research findings derived from the data analysis based on our specific areas of focus. The last section discusses the results and identifies gaps, while making suggestions for future research.

2. Background: related work

The Maker Movement depicts the appearance of spaces that enable all kind of users, including hobbyists, engineers, hackers, artists, and students, to express themselves creatively by designing and building digital or tangible objects [46]. Furthermore, these spaces offer the possibility of being used to support different classes, away from the traditional classroom environment. The establishment of *Maker* magazine and the first Maker Faire in 2006 led to the name and the idea of the Maker Movement, which can be characterized as an iteration of the do-it-yourself culture. It is clear that making activities and their importance in learning are not novel. Making, playing, building, and interacting with the real world have been argued to be valuable ways of learning [54,67].

The promising use of maker spaces for educational purposes is reflected in their appearance in libraries and universities. Barrett et al. [1] reviewed existing maker spaces in universities in the United States in terms of their use, the equipment they have, their location in the university, who has access, and so on. The most frequently used tools are 3D printers, and the location is usually the library. However, the influence of these maker spaces is an issue that is open for further investigation. Blikstein [3] presented and analyzed old and new physical computing platforms that support educational concepts. Various tools, such as Cricket, Braitenberg Blocks, and Arduino technologies, can be further developed and used in multiple ways to support fruitful learning experiences [3].

Vossoughi and Bevan [66], in their review of making and tinkering, sorted the current published literature into three categories: "making as entrepreneurship and/or community creativity, making as STEM pipeline and workforce development, and making as inquiry-based educative practice" page 5. Most of the studies they examined were linked to the third category. In addition, they mainly focused on investigating: (1) the effect of making and tinkering, particularly in STEM activities for school-aged students; (2) the characteristics of making and tinkering programs according to their organization; (3) the emerging design principles and pedagogies that characterize tinkering and making programs; and (4) how this movement could influence equity-oriented teaching and learning. In addition, a framework for the theoretical roots of the Maker Movement and its relevance to formal or informal educational practices has been provided by Halverson and Sheridan [26]. Their work introduced the need to rethink what could be considered as the learning environment, learning in general, and under which circumstances learning is effective. Furthermore, as they presumed that maker spaces will use twenty-first-century technologies, particular attention should be paid to the product and its development. Martin [46] stressed the need for K-12 engineering instruction to be developed through design activities that provide knowledge and playful experiences. The Maker Movement could play a helpful role in achieving this. In recognition of that fact, Martin [46] described three crucial ways in which the Maker Movement could likely be connected to education: (1) tools that could be involved in making projects; (2) any kind of events, online or not, promoting making activities; and (3) values, beliefs, and the making culture in general, which has to be associated with research recommendations for learning environments.

Despite the interest in the Maker Movement and its connection to formal or informal education, there has been little research concerning the direction it is taking, the opportunities it could present for education, and why. A special issue on digital fabrication in education in the *International Journal of Child–Computer Interaction* [31] identifies the need for new types of curricula, facilitation, tools, and social protocols to empower making in education. In addition, the authors within this issue showed that digital fabrication technologies can be a great opportunity not only for gaining valuable skills, but also for providing Bildung (i.e. deep and sustained learning). Furthermore, they distinguished making activities in public fablabs, hackerspaces, and makerspaces from digital fabrication in formal education in terms of using the latter as a teaching and learning resource. This review builds on the results of a number of studies that have been carried out in the area of learning through making types of instructions, since it gives a deeper understanding of their ingredients. Previous reviews have mainly focused on more theoretical approaches to the Maker Movement [66,26,46], while some have focused on the description and potential use of technological tools for educational purposes [3]. The current review aims to analyze the 43 relevant studies from 2011 to 2015, offering a systematic comprehensive analysis of their main elements concerning the common technologies used for educational purposes, subjects, evaluation methods, and pedagogical designs. A wide-ranging approach is adopted to better explore the opportunities to improve future research by providing knowledge on the learning outcomes from the making approach within or outside of the classroom.

3. Methodology

To carry out our literature review, we followed several steps based on Kitchenham [42]. The entire literature review strategy was accomplished under continual consultation with computer science education specialists. We defined a review protocol that helped us to indicate the research questions, data collection, inclusion, exclusion, and quality criteria, and, finally, data analysis. The procedures in the remainder of this section will describe the method used for each of the actions taken.

3.1. Data collection

With the aim of collecting high-quality data, we conducted a search in the following international online bibliographic databases: Association for Computing Machinery Digital Library (ACM), Science Direct, and EBSCO Education Source including ERIC. Moreover, we independently searched key educational journals and conferences including *Computers and Education* (Elsevier), *International Journal of Child–Computer Interaction* (Elsevier), Conference on Human Factors in Computing Systems (SIGCHI), ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE), ACM Conference on Creativity & Cognition (C&C), ACM technical symposium on Computer Science Education (SIGCSE), and Interaction Design and Children (IDC). In order to find other relevant publications and collect as many studies as possible, we used the reference section of articles as well as Google Scholar. The period examined was from 2011 to 2015 (November). The key search terms were "maker AND movement," "maker AND movement AND education," "school AND makerspaces," "makerspaces AND education," "hackerspaces AND education," "informal AND education AND making," and "informal science education AND making," as well as derivatives of these terms. That was the first stage of the review process, and provided search results of 2930 articles in total.

3.2. Inclusion and exclusion criteria

The next step of the review process was to exclude short papers, posters, workshops, work-in-progress studies, and any non-peer-reviewed papers. This step decreased the number of papers in order to yield a manageable amount for detailed analysis. In this stage, papers with irrelevant topics were excluded, mainly based on their titles. We took into consideration only studies that showed empirical evidence. Both qualitative and quantitative studies were included. Furthermore, we excluded studies that did not have a learning/teaching purpose or did not involve children or students in their sample. In addition, we focused only on studies that described any kind of making experience or testing process from users. It should be noted that we included studies referring to both formal and informal teaching environments, such as schools, museums, and summer camps. In particular, at this stage we focused on titles and abstracts to indicate whether a paper was inside or outside the scope of this review. Finally, we ended up with 223 studies (Fig. 1).

3.3. Quality criteria

To assess each of the remaining 223 studies, we defined criteria proposed for the Critical Appraisal Skills Programme, and more particularly those for assessing qualitative research [25]. The criteria were adjusted to our review context. In general, the studies had to meet the following three criteria; they had to be (1) *rigorous*; (2) *credible*; and (3) *relevant*. "Rigorous" refers to the appropriate

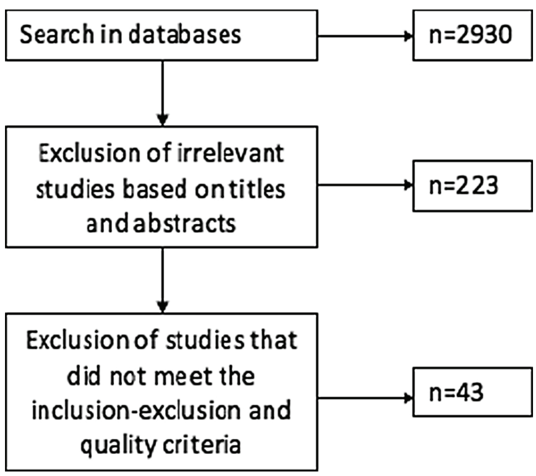

Fig. 1. Study selection process.

research method applied to the study, "credible" points to the presentation and validity of the findings, and "relevant" indicates whether the findings of each study were suitable for education science, as well as computer science education research communities. Specifically, we adopted nine criteria to evaluate the quality of the studies. The scope of this evaluation was to ensure that only high-level studies would contribute to our literature review.

All studies were assessed as to whether they met the following criteria [13]:

1. They comprised empirical research.
2. The aims and objectives were clearly reported.
3. The research design was appropriate to address the aims of the research.
4. There was an adequate description of the context in which the research was carried out.
5. The research design was appropriate to address the aims of the research.
6. Data were collected in a way that addressed the research issue.
7. Data analysis was sufficiently rigorous.
8. There was a clear statement of the findings.
9. They added value for research or practice.

Finally, 43 studies met the inclusion–exclusion, as well as the quality, criteria. Then, we coded these studies according to specific areas of focus. These areas allow the description of the main focus of the studies. With regard to our critical examination of the papers, it was useful to define categories that represent their content. These categories derived from the consideration of different types of organizing making activities and workshops, as well as their evaluation process. This categorization enabled us to record all the details needed from the papers of our literature review and use them to address our research questions.

3.4. Data analysis

An analysis of the studies collected was conducted based on the following areas of focus: location (e.g., school, university lab, museum), materials used (e.g., Arduino, recycled materials), subject/area of study (e.g., programming, mathematics), duration of the workshops or the testing process in each study, age of participants, sample size, type of methodology (qualitative, quantitative), how data analysis was conducted, instruments used (e.g., surveys, interviews, observations), areas of interest (i.e., the main fields that each study wanted to investigate), whether there was collaboration among the participants, the main findings of the studies, research design, and finally experimental design. The authors extracted several attributes of the selected studies and the final areas of focus were resolved by discussion. Constant consensus meetings of all three researchers approved each step of the analysis of all studies conducted mainly from the first author. All 43 studies were analyzed in detail according to the coding scheme and data were extracted to better answer our research questions. Details on the paper coding are shown in Appendices A and B.

4. Research findings

4.1. Type of research

Regarding the "type of methodology," we refer to whether qualitative, quantitative, or mixed methods were used. According to this categorization, the majority of studies were qualitative (22), fewer studies were mixed (11), and even fewer studies were quantitative (6) (Fig. 2, Table 1). It is worth mentioning that four of the studies had no methodology, of which one study was describing 25 different workshops (8 qualitative, 9 quantitative, and 8 mixed), and one study was a pilot test. Moreover, concerning the "experimental design," we refer to the distinction among the following: between groups, within groups [8], and only a post-test measuring the participants' performance after treatment. Concerning the experimental design described above, most of the studies used a within-groups design (10) and very few used a between-groups design (3). None of the studies used a combination of within- and between-groups designs. Only four studies had no design, one had participatory action research design, and one had a pilot usability test. The studies that remained (24) measured only the results of the treatment by conducting a post-test. When it comes to the "research design," all studies used a "true experiment." In a true experiment, participants are randomly assigned to either the treatment or the control group; in a quasi-experiment they are not assigned randomly.

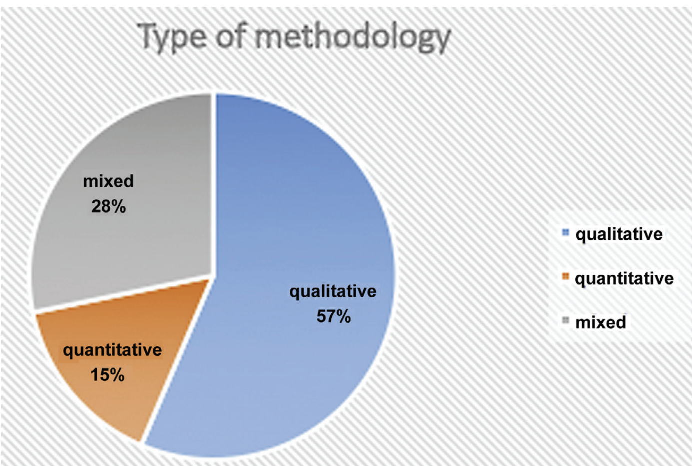

Fig. 2. Breakdown of the types of methodologies present in our review.

Table 1
Type of research.

Type of methodology	Qualitative	Quantitative	Mixed
Number of studies	22	6	11

4.2. Subject area, students, and instruments used

In order to understand the role of making in learning and teaching, it is very important to identify the scope of each study. Thus, we defined the subject on which the studies focused. The largest number of papers (32) aimed to enhance programming skills and computational thinking (Table 2). Specifically, they used programming tools in the sense of making as part of computer science courses at a school, university, or even an independent lab or museum. Other studies suggested that the current trends of learning through making in art, design, and technology practice can provide fertile ground for developing STEM learning (6). Sample size refers to the number of participants in each workshop. Apart from a few cases, the sample size was fewer than 50 participants (34); some workshops (17) had fewer than 20 participants, a small number consisted of more than 50 people (7), and even fewer (four) had more than 100. Looking at the age of the participants, a wide range of ages emerged. The majority of the studies involved ages up to 14 years (22) and a smaller number (12) were conducted with ages over 14 years. Only seven studies included university students at undergraduate, graduate, or master's and PhD level. Five workshops had participants of mixed ages ranging from six to 19 years old, with one study, which took place at a museum, in which participants were in general over six years old. Finally, there were also seven studies on ages over 18 years. In order to capture information to assess the outcome of the workshops, researchers used different instruments, usually in combination. Specifically, 20 studies used interviews, 18 observations, 11 field notes, 13 videos, 6 photographs, 17 code of the games/projects/artifacts, 3 focus groups and 11 surveys (Table 3).

4.3. Materials for making activities

Various types of technologies and objects have been used in the maker philosophy, via which software and/or physical objects are created. In this literature, a total of 18 different digital tools had been used, including diverse software environments. Some of these are quite new and attractive due to their programmable possibilities, and others are more traditional, but can be easily used in a more innovative and creative way. The most frequently used element was Scratch (10), and almost equally popular was Lilypad Arduino (9), followed by Arduino (6) (Table 2). Other tools were employed to almost the same extent, such as 3D printers (2), Raspberry Pi (2), Makey Makey (4), Minecraft (2), Codespells (3), Modkit (1), Circuits (2), and game makers (i.e., yoyogames.com) (1). However, certain technologies appeared only once each; these were Crickets, mobile apps, storytelling kit, software creators, and Tiny programmer. In addition, many tangible objects were offered as supportive material in making activities using technology. Several projects included sewing and conductive materials, Play-Doh, LEDs, batteries, paper, copper tape, recycled materials, and even lemons and potatoes.

Table 2
Areas of study and digital tools.

Subject/area of study		Programming	STEM
Number of studies		32	6
Digital tools	Scratch	Lilypad Arduino	Arduino
Number of studies	10	9	6

5. Discussion and conclusion

After identifying a large number of papers (2930) using our search terms, we can agree on the fact that there has been wide interest in the Maker Movement approach. The current review focused on the outcomes of the Maker Movement for instruction and the benefits of applying the making culture in different environments, including the classroom. Our quality and inclusion and exclusion criteria led us to determine the most relevant and highest-quality papers according to our research questions. Finally, 43 peer-reviewed articles were selected, which were diverse in terms of the making culture for educational or non-educational reasons, including different types of approaches referring to different ages of participants. The number of papers has significantly increased during the last few years, with a peak in 2013 (Fig. 3).

The results show that studies of the Maker Movement applied to instruction tend to use a qualitative methodology to assess their work. Qualitative measures are more suitable for that type of research, as it has special value for investigating complex issues, such as children's attitudes to computer science, mathematics, and engineering [28,63,34], topics such as self-efficacy [41], and general impressions about the process of making activities [55]. In significantly fewer studies a quantitative methodology was mainly used, when an objective view was needed. One example of a quantitative study is Doran et al. [12], who measured students' improvement in school grades. The same method was also used when Esper et al. [16] measured exam performance, skills, and knowledge of computer science and programming intelligence. Studies based on quantitative methodologies were fewer and were mainly adopted for concepts related to school activities. This finding emphasizes that inside the school environment it is easy to collect quantitative data; for example, standardized tests used by default by teachers. On the other hand, one potential difficulty for qualitative studies with young children is the need for special permission to execute long interviews. In addition, in a school environment it is easier to measure students' performance and progress, as the duration of the study lasts for a longer period than only a few days. Qualitative methodology takes precedence over the mixed and quantitative methodologies.

Table 3
Instruments used.

Instrument	INT	OBS	FNO	VID	PHOT	PROJ	FG	SUR
Number of studies	20	18	11	13	6	17	3	11

INT: interviews, OBS: observations, FNO: field notes, VID: videos, PHOT: photos, PROJ: final projects, FG: focus groups, SUR: surveys.

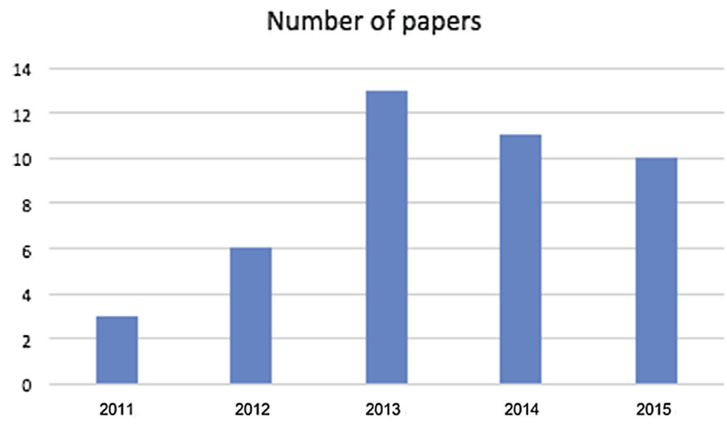

Fig. 3. Number of papers in the literature review, published from 2011 to 2015.

After reviewing 43 studies, we recognized the most common subject areas for implementing making types of instructions. Our literature review confirmed that a making approach to learning is being taken most notably in programming, as well as in STEM curricular areas. This result was expected, because these are the most prominent subjects to which technological resources are applied. It is a challenge to see which other subject areas could benefit from incorporating a making approach. Two potential subject areas could be biology and medicine. One example is the study of Khalili et al. [40], which combined video game development with a focus on neurology, where students could understand molecular processes. Also, nowadays, students need to acquire skills and competences to be prepared for their future work and everyday life. Therefore, learning approaches need to be adjusted to successfully teach the 21st century skills. Making activities could support learning processes that will not only be focused on a specific subject like Maths but also involve the 21st century skills acquisition. Interestingly, an opportunity of incorporating making in other subject areas are design-based activities that teach digital literacy and design thinking as described in the special issue on digital fabrication in education [31]. A surprising result from our review is that few studies focused on gender issues. We expected more studies to provide insights on how making activities benefit females specifically, as the main subject areas applied are STEM and programming.

All the studies used some type of digital material to support making activities, which highlights the need to familiarize users with technology and broaden participants' perspectives and interest in computer science in general. The goal is to achieve better understanding and enhance skills related to the subject areas through digital fabrication devices, ability to produce objects, and modeling tools. Students' involvement in several computer science concepts enhances their ability to achieve goals [20]. In general, making sessions grow participants' competence [9]. A study by Franklin et al. [20] showed that it is possible for students to attain competence in event-driven programming, initialization of state, and message passing after just a two-week interdisciplinary camp, which was not entirely focused on computer science. A class that used Lilypad Arduino successfully promoted computing concepts and practices, while perceptions of computing were extended [36]. Other studies using the same tool showed how craft materials support a more understandable approach to creating technology and that the results of this process can be more transparent and expressive. Using the same concept, participants in workshops managed not only to think about but also to create interface designs using conductive fabrics and craft materials [53]. When crafts and technology are tightly connected by conceiving and realizing different artifacts, people become more engaged and develop different skills compared to getting involved with traditional development or electronic toolkits [50].

We expect making sessions to be promising approaches to engaging students in the design and fabrication process, in thinking and problem solving, as well as in programming. In many studies, the combination of programming and physical fabrication resulted in engagement in complex programming concepts (e.g., loops, conditionals) and practices (e.g., remix, testing, and debugging) [11,63,37]. Furthermore, even young students aged 9–10 years have been engaged in Java programming by playing and making games [16]. Chu et al. [9] examined children's overall general engagement with the experience of using the Maker Theater kit for storytelling, showing that sometimes, apart from fun and excitement, frustration and boredom may arise due to usability problems.

Another main core of learning, as reported by Katterfeldt et al. [39], is self-efficacy. Some of the analyzed studies showed that following the making activities the self-efficacy of the participants was affected. This provides evidence for the success of making in influencing learners' behaviors. When their self-efficacy increases, workshop participants gain confidence, enjoyment, and interest in programming and technology [56]. Moreover, when actions are motivated with enthusiasm and self-regulatory feedback, self-efficacy ratings are higher [43].

All of the above studies report the positive effects of making activities on students' perceptions and engagement. Students' experience with computer game programming could also change their attitude toward computer science and prepare them for computer science courses and careers [11]. In general, no matter what the age of the group and which tool was used, making proved to be a successful process in all the different areas of interest. One surprising outcome is the absence of negative results. Almost none of the studies reported negative effects in the research. Even though this is an obvious positive conclusion, it does not provide an in-depth understanding of how to prevent poor practices that hinder students' engagement and performance.

Furthermore, the studies evaluated reveal a variety of technology tools used in making. Given the large amount of different software available and the possibility for it to be used for educational reasons, it is difficult to define the best choice for a specific activity. It was not surprising that the most-used tool was Scratch, as this is one of the popular visual programming languages, followed by Lilypad Arduino. In the remaining studies, other, less common technologies were used, identifying the need for further investigation in this area. Moreover, combining digital and tangible materials in the creating process has been proven to be valuable [33,34].

The Maker Movement has begun to play a role both inside and outside the classroom, showing that it could be part of the classroom in offering a pattern of simulation [2,64,37,21], and part of outside activities, such as summer camps and libraries [65,68], demonstrating that learning is feasible in any environment as long as it is organized under suitable conditions.

All the studies that took place in schools, except for that of Basawapatna et al. [2], integrated their making sessions into a whole-semester curriculum [63,36] or at least for a few weeks [64]. Longer periods of study offer opportunities for observing the effect over time. Almost all of the studies had as their main subject programming or a combination of programming and math [21]. The scope of the studies was to introduce programming to students and/or examine the learning outcome regarding the enhancement of computational thinking skills [38] or the ability to code [7,59].

Collaboration among the participants was present in the large majority of studies. We know how important collaboration is to motivating and promoting learning, which is why many efforts concentrated on testing different types of offline or even online collaborative methods [18,38]. Although collaboration appears to be an important aspect of making activities, we saw very few descriptions of collaborative strategies and how they contribute to individual learning. An interesting approach came from Fields et al. [19], who proposed a collective design process for programmers, deriving from participatory models. They illustrated that collaboration supported learning through the exchange of ideas and mentoring, and led to deeper engagement. Another aspect of collaboration and its value is between parent and child [51].

5.1. Limitations and further work

The current review has a number of limitations. First, one of the common limitations in every review study is the potential of the limited research terms used, the journals included, and the specific time period covered by the papers published. However, the papers discussed in this literature review provide a snapshot of empirical research on outcomes and impacts of the Maker Movement approach to instruction that is representative of the state of the art at the time. While many aspects of engagement and learning have been discussed, there are other aspects that could be further analyzed. In addition, we analyzed a wide diversity of studies in terms of the age of the participants, duration of different workshops, and scope of the study, and as a consequence face limitations in combining and reflecting more on their results. In the current review, we coded each study with respect to their main outcome or impact, in order to better indicate which aspects and ingredients of making work better and under which circumstances.

If the maker approach contains self-driven discovery, it increases the learning gains [60]. Given the rapid growth of the Maker Movement, there is a need to understand how students can best utilize its strategies to achieve better learning. From the review of prior and ongoing work on the Maker Movement, we can provide recommendations for further research. One recommendation is to focus on technologies and tools that have been used in a limited number of studies but have promising potential; for instance, Raspberry Pi. Another area of investigation could be the analysis of maker instruction as part of the classroom, since most of the studies focused on the extracurricular context. Future work should also focus on collecting relevant studies on participants of a particular age. For example, we did not examine many studies focusing on ages earlier than 11 years. One of the countries that has integrated computer science in its school curriculum is the United Kingdom, in which pupils start from the age of six. We can therefore expect to see more results here. Related to the lack of understanding regarding collaboration and its benefits to participants at a team or individual level, we suggest that future studies investigate how collaborative work supports a successful making learning experience. Lastly, future research would benefit from exploring even more elements of the Maker Movement approach by using another classification.

5.2. Conclusion

In the current review, we analyze 43 peer-reviewed articles, selected from searches of the literature over the past five years, covering a wide range of approaches in the area of learning through making. The aim of this review is to investigate the emerging role of Making in formal and informal education. Therefore, we analyze the studies based on specific areas of focus that could better describe the direction of the Maker Movement research during the recent years. These areas include among other, the types technologies used to support making types of instructions, sample size, age of the participants, type of methodology and areas of interest in each study.

A clear finding of the review is that many of the studies have intergraded making sessions in the classroom, mainly in the area of programming and other STEM areas. We regarded making only in a positive way, as almost none of the studies reported negative effects, but also we highlight the need for a more in-depth analysis. In addition, we define and discuss the effects of making as they have been approached in the different studies, like engagement, self-efficacy, performance, collaboration. Our goal with this article is to summarize the findings and show a direction in which different types of making activities could have an impact on educational approaches and lead to a more effective way of teaching and learning. However, much research on this field is needed for this direction.

Appendix A

No.	Authors, year of publication	Location	Materials: digital	Materials: tangible	Subject/area of study	Duration	Age of the participants	Sample size	Type of methodology	Instruments
1.	Denner et al. [11]	An after-school class	Stagecast Creator software		Programming	4–6 weeks (1–2 h/week)	Middle school	59	Qualitative (case study)	Games made
2.	Harnett et al. [28]	University of Louisville, Kentucky hackerspace	Arduino, Raspberry Pi		Engineering	1 semester	Undergraduate students	9	Ethnographic (qualitative approach)	Interviews and observations
3.	Schwartz et al. [61]	El Pueblo Mágico (EPM), a social design experiment	Minecraft	Sewing materials, Play-Doh, LED 2. Solar cars: e.g., batteries, wheels	STEM	1: 6 days over 2 weeks, 2: 1 semester	Undergraduates, grades K2–5, grades 6–7, researchers	103	Cognitive ethnographies (qualitative approach)	Video data, field notes, student paper, blogs
4.	Moriwaki et al. [51]	Informal education, outside the classroom	Noisemaker circuit: speaker, headphones, digital/analog circuits	Chain, wires, tape, cardboard, bottle of salt water, aluminum foil, wires, adhesive tape, paper tape, stuffed animal, battery and motor, plastic cups, Play-Doh, LEDs	STEAMD		Children 4–11 and their parents (adults)	60	Qualitative	Observations, interviews, informal discussions
5.	Lane et al. [43]	Cahner's Computer Place at the Museum of Science (MoS), Boston, Robot Park	iRobot CreateTM robot, Coach MikE: LEGO Mindstorms		Programming	3–4 min	6+	805 used it but study 1: 269 observations, 223 interviews, 75 questions; study 2: 238 observations)	Mixed	Interview questions, observations, questionnaires
6.	Khalili et al. [40]	Summer program	Game Maker (http://www.yoyogames.com/gamemaker/)		Programming	2–3 h, every day for 4 weeks	High school students, college students	19	Qualitative	Observations, game code, field notes, interviews
7.	Kolko et al. [41]	Semi-formal learning group			Engineering literacy		University students	27	Qualitative	Autoethnographies (self-report), blog posts, and reflection focus group notes
8.	Bruce et al. [5]	University computing curricula in the classroom	Raspberry Pi		Programming (CS courses)	1–2 weeks	University students			

9.	Kafai et al. [34]	E-textile class part of an elective computer science class (high school).	LilyPad Arduino	E.g., LEDs, conductive thread, alligator clips, battery, switch	Programming	10 weeks, 19 meetings, twice a week for 65 min per session	16–18 years	15	Qualitative	Analysis of E-Textile Projects: collected their hand-drawn blueprints of their designs, photographs, Arduino code, interviews
10.	Katterfeldt et al. [39]	Workshops that took place at a university were offered as spare-time activities During holidays or as project work for diverse school classes, under the name of TechKreativ	Lego RCX, Crickets, Arduino, LilyPad Arduino	E.g., fabrics, balloons, paper, cartons	Programming	3–5 days full time	9–15 years	15	8 workshops qualitative, 9 quantitative, 8 mixed	Semi-structured interviews and/or observation or contextual inquiry, questionnaires
11.	Chu et al. [9]	Workshops that took place in a large reconfigurable lab space	Storytelling making kit called the Maker Theater	E.g., paperclips	Storytelling	2 days for half a day (approximately 4 h)	Grades 3–5, 8–11 years	23	Qualitative	Interviews, survey (part of the interview), video/audio recordings, post-reports of the parents
12.	Schwartz et al. [62]	l Pueblo Mágico (EPM) (social design experiment) after-school program	Minecraft, Makey Makey, robot picaxe	Play-Doh, batteries, LEDs, recycled materials, lemons, potatoes, sewing materials, copper tape, paper, conductive materials	STEM	1 day per week for 1 semester, 2 days per week for 3 weeks, 2 days per week for 3 weeks	Undergraduates, grades 2–5, grades 6–8	131 children, 48 students	Qualitative + cognitive ethnographies	Interviews, video, field notes, observations, artifacts
13.	Leduc-Mills and Eisenberg [44]		Ucube (3D printing and fabrication devices)		Spatial cognition		12–14 years	14	"Usability test" initial (and informal) pilot test	

(*continued on next page*)

Appendix A (*continued*)

No.	Authors, year of publication	Location	Materials: digital	Materials: tangible	Subject/area of study	Duration	Age of the participants	Sample size	Type of methodology	Instruments
14.	Fields et al. [19]	High school programming workshop (part of enrichment of curricula in partnership with a local museum)	Scratch		Programming	8 × 2 h weekly sessions in winter	14–15 years and 15–16 years	29	Qualitative	Field notes and video logs, and post hoc interviews
15.	Lee et al. [45]	K-8 neighborhood school (elective courses throughout the year)	Makey-Makey, Scratch	E.g., Play-Doh, pipe cleaners	Programming	9 times, 50 min each	6–8th grades (11–12 years)		Qualitative	Observation notes, photographs and video recordings, code of the games, interviews
16.	Kafai et al. [33]	Over the course of a school year, as part of a partnership between the high school and a local science museum (in the museum space)	LilyPad Arduino	Paper and pencil, textile materials, LED lights	Programming	4–6 weeks/once a week for 2 h sessions, 1 school year	14–15 years	35	Qualitative	Video recordings, photographs, daily field notes, interviews, participant observations
17.	Searle and Kafai [64]	At school, introductory computing activities as the culminating project in an elective class	LilyPad Arduino	Sewable sensors (e.g., temperature sensor, accelerometer) and actuators (e.g., LED lights, sound buzzers	Programming	3 weeks, meeting daily for about 1 h	12–14 years (8th grade)	10	Qualitative	Daily field notes, video, interviews, code screenshots, circuitry blueprints
18.	Kafai and Vasudevan [37]	K-8 public school	Makey-Makey, Scratch	Play-Doh, other textiles (e.g., aluminum foil), fruits	Programming	Twice a week for 50 min each (over 3 years)	11–14 years (middle school students)	28	Qualitative	Field notes, video recordings, final physical artifacts and final code, interviews

19. Searle et al. [63]	Part of school. All participants were enrolled in an elective computer science course	LilyPad Arduino	LEDs, conductive fabric patches	Programming	10 weeks	16–18 years (high school students)	27 (24 interviewed)	Qualitative	Interviews (semi-structured)
20. Kafai et al. [36]	Pre-AP high school class, as part of their elective computer science class	Lilypad Arduino	E.g., LEDs, conductive and non-conductive fabric	Programming	10 weeks (19 meetings) twice a week for 65 min per session	16–18 years	15	Qualitative	Interviews (semi-structured), photographs, code
21. Burke and Kafai [7]	Middle school, an elective course	Scratch		Programming	7 weeks, 11 sessions	12–14 years (middle school students)	10	Mixed	Surveys, field note observations, video, scratch code, interviews
22. Telhan et al. [65]	Free library locations	Website that displayed a virtual representation of the mural and allowed the LEDs to be controlled, Electric Imp hardware module, a 16 × 8 LED matrix controller, and a 4G modem	Physical foam boards, LEDs	Programming		6–19 years	1036	Qualitative	Observations, interviews
23. Schneider et al. [60]	As a requirement of a psychology class	Tabletop interface, camera, speaker	Ear canal, several ear bones, cochlea and 4 neurons	Functioning of the human hearing system	4 consecutive days, ~20 h	College-level students	38	Mixed	Observations, pre-mid-post tests (last question open-ended), log files, informal questions (comments)
24. Qiu et al. [56]	Students invited through a mailing list and through a Learn2Teach, Teach2Learn program	Modkit, LilyPad Arduino	Fabric, switches, conductive thread, LEDs	Programming (computer science and engineering education)	Course of a weekend, lasting for 6 h on both Saturday and Sunday	13–16 years and 12–17 years	37	Quantitative	Survey
25. Delle Monache et al. [10]	SaMPL7 lab in collaboration with the university	Sound Design Toolkit, Arduino	Paper, sensors	Engineering techniques	3 days	MA and PhD	21		

(continued on next page)

Appendix A (*continued*)

No.	Authors, year of publication	Location	Materials: digital	Materials: tangible	Subject/area of study	Duration	Age of the participants	Sample size	Type of methodology	Instruments
26.	Perner-Wilson et al. [53]	Independent workshops	LilyPad Arduino	Cheap toy pianos and T-shirts, sensors, LEDs, fabric, conductive fabric, shawl, beads	Programming, electronics	From a few hours to a few days	20–57 years (those who answered the questionnaire)	Between 5 and 20 participants each (30 answers to the questionnaires)	Quantitative	Questionnaire
27.	Mellis et al. [50]	Independent workshops	TinyProgrammer, TinyUploader software, Arduino, microcontrollers	Conductive ink, paper, LEDs, capacitive electrodes, microphones and light sensors, batteries	Programming		23–60 years	12	Mixed	Surveys, group discussions
28.	Mellis and Buechley [49]	Independent workshops	Speakers, computer mouse, 3D printers, circuit board Laser cutting and 3D printing, circuit board		Fabrication technologies	~6.5 h + 24 h	25–35 years + 21–31 years	7 + 8	Mixed (case studies that included 2 workshops)	Pre-post questionnaires, surveys
29.	Jacobs and Buechley [32]	Independent workshops	Codeable Objects software (a programming environment that was developed as a Java-based library for processing)		Programming, (computational design and digital fabrication)	1: one day 6 h, 2: 10 days	1: Graduate students 2: 11–17 years	9 + 8 (17)	Mixed	Surveys, interviews, and photographs of participants' projects
30.	Qi and Buechley [55]	Independent workshops	ATTINY85 microcontrollers	Paper, circuit components, foil tape and electronic elements, LEDs	Building electronics	Once a week/5 weeks, 2 days	Undergraduates 18–26, 18–37, 19–21, 15–50 years (15–50 years)	11 + 8 + 19 + 16 + 19 (73)	Qualitative	Observations, projects, open-ended questions (written responses)
31.	Wagner et al. [68]	Summer camp for high school students	App Inventor Java, App Inventor block language		Programming	3 weeks, for Java: 35 h over 5 days	Grades 9–12	40	Qualitative	Observations of student work and artifact assessment of student projects
32.	Franklin et al. [20]	Non-academic summer camp	Scratch		Programming	2 weeks	Middle school students	35	Mixed	Field notes, projects
33.	Basawapatna et al. [2]	At a middle school	Simulation Creation Toolkit		Programming (Computational Thinking Patterns [CTPs] to create simulations)	3 days	6th grade	18 + 21 (39) (students from 5th + 6th period)	Quantitative (initial study)	

34. Burge et al. [6]	Summer camp for HS-age girls	iPhone mockup tool, mobile app tool		Programming	1 week	High school students (14–16 years)	25	Quantitative	Survey
35. Kafai and Vasudevan [38]	Partnership between high school and local science center	Makey-Makey, Scratch, digital extensions (e.g., digital dice)		Programming	8 weeks, once a week	High school students, 13–15 years + 14–15 years	17 + 5 (for the case study)	Qualitative + case study with a team of students	Field notes, videos, final projects and code, focus groups observation (from the case study)
36. Esper et al. [16]	At schools	CodeSpells (3D immersive video game using Java)		Programming	8 weeks	4th grade (9–10 years)	50 (for the main study)	Quantitative	Pre-post surveys, final exam and a Growth Mindset about math and programming intelligence
37. Esper et al. [15]		CodeSpells API (Java)		Programming	45-min session		17	Mixed (exploratory study)	Questionnaire, exit interview, video
38. Esper et al. [17]	Summer program	CodeSpells API (Java)		Programming	7-day summer program, 7 sessions, ~10 h in total	8–12 years	16	Qualitative	Audio and video data, logs and video of the games
39. Giannakos and Jaccheri [23]	Workshop that took place in the Norwegian Deaf Museum	Scratch		Programming		12 years	12	Mixed (for quantitative [exploratory evaluation])	Focus group, photographs, videos, observations, surveys
40. Giannakos and Jaccheri [22]	ReMida centers (creative places) and in a university's creative room	Scratch, Arduino	Recycled materials	Programming	2 days, 1 day	12 years	66	Mixed (explorative-qualitative and then quantitative)	Photos, observations, surveys
41. Garneli et al. [21]	At school as part of the normal teaching procedure	Scratch		Programming and math		12–13 years	80	Mixed	Questionnaire, observations, interviews
42. Giannakos et al. [24]	Creative centers of university and ReMida centers	Scratch, Arduino		Programming	2 days, 1 day	12 years, 17 years	37	Quantitative	Survey
43. Rode et al. [59]	In a computer club in a primary school	LilyPad Arduino	LEDs, sensors, speaker, conductive fabric	Programming (computational thinking)		8–10 years	8	Qualitative	Observations, field notes

Appendix B

No. of paper	Areas of interest	Journal/conference	Included team work/collaboration	Data analysis	Main findings	Experimental design
1.	(1) Programming, (2) documenting (3) understanding software, (4) designing for usability	*Computers & Education*, Elsevier	Yes	Coding scheme (based on ISTE National Education Technology Standards for Students and Martin et al. [47]	Computer game programming is a promising approach to engage underrepresented students in the concepts and capabilities that will prepare them for computer science courses and careers	Post test
2.	The Makerspace culture itself, the attitudes and beliefs of the co-op student participants, effect of the experience on their attitudes toward engineering	Frontiers in Education Conference (FIE), 2014 IEEE	Yes	Ethnographic norms	Increased confidence in their problem-solving and project-planning abilities	Within groups
3.	(1) Types of strategic assistance, (2) distributed expertise and roles emerging in activity, and (3) use of key theoretical concepts for mediated praxis	Conference: International Conference of the Learning Sciences (ICLS)	Yes	Overall analysis, coding, ranking scheme and notation (+ descriptive statistics for some)	Positive outcomes regarding the remediation of participation in STEM activities for youth from non-dominant communities when undergraduates allowed children to dive into activity, and when their strategic questioning and assistance distributed thinking to children	Post test
5.	Self-efficacy	Artificial Intelligence in Education, 16th International Conference, AIED 2013	Yes (group visits)	*T*-test, regression analysis (multiple regression model) + descriptive statistics	More time spent programming, increased likelihood of accepting challenges, and less misuse of the exhibit. Findings do not demonstrate the need for an embodied and animated pedagogical agent. No differences in visitor behaviors between conditions. Modest, but significant increase in visitors' self-reported self-efficacy ratings detected when Coach Mike configured to be enthusiastic and to deliver self-regulatory feedback	Between groups (study 1)
4.	Gauging interest level, collaborative practice, completion of workshop activities, and scientific understanding	Integrated STEM Education Conference (ISEC), 2012 IEEE	Yes		STEAMD approach to informal science learning nurtures multigenerational learning by virtue of its scalability, both in concepts learned and personal interest	Post test
6.	Insights on student learning	*Computers in the Schools*, Taylor & Francis Group	Yes	Participatory analysis of the data (review and interpretation of data, patterns discovered, memos written, and then interpretation again)	Despite having no prior knowledge of these neurological concepts, students were able to design reasonably accurate visual representations of the constructs and verbally describe the concepts	Post test

7.		Proceedings of the 12th Participatory Design Conference: Research Papers	Yes		Thematic analysis	Self-efficacy and identity as drivers of technical skill acquisition, connecting outside the academy, experience of blogging cultivates reflection, constraint and scoping, high expectations require significant scaffolding, shared vocabulary	Post test
8.		Proceedings of the IEEE Southeast Conference 2015	Yes				
9.	Computational concepts and practices, computational perceptions, how their project had changed from their early ideas to completion, what they were most proud of and what was most challenging, what they felt they had learned about computer science in the process, whether their ideas about computer science had changed, and whether the project had influenced their future goals focusing on three aspects: personal relevancy of computing, potential study and career path in computing, and expanded understanding of computing at large	ACM Transactions on Computing Education, 2014	Yes		Projects coded based on Brennan and Resnick's [4] 2012 framework for computational thinking, identification, and classification of approaches based on observations, interviews were logged (a very close but not word-for-word transcription) and then analyzed, focusing on three aspects	Students employed computational concepts in writing code for their e-textiles. Students learned about circuitry by remixing the design template to meet the needs of their particular projects. In this way, their aesthetic motivations promoted learning by remixing circuit designs. Reusing and remixing code were key practices. They developed more realistic, personally relevant, and expansive perspectives of computing in the process of making their e-textile artifacts	Post test
10.	Slightly altering perspectives (e.g., gender issues), attitude toward technology, subject formation	*International Journal of Child–Computer Interaction*	Yes				
11.	Self-efficacy, motivation, interest, self-identity/self-concept	*International Journal of Child–Computer Interaction*	Yes		Interview tablet surveys also extracted into a spreadsheet and analyzed, video/audio recording: three-level coding process (also Inscribed software was used for transcription), post reports: open, focused, and axial/thematic analysis	Children wanted to engage in the Making session. Evidence of interest episodes generally occurred. The notion that 'I can Make' was borne out in video coding analysis through a number of recurrent themes	Within groups
12.	(1) Types of strategic assistance, (2) distributed expertise and roles emerging in activity, (3) use of key theoretical concepts for mediated praxis	IJREE–*International Journal for Research on Extended Education*	Yes		Close-up analysis of design discourse. Text analysis, discourse analysis, and analysis of joint problem articulation	Analysis revealed that successful strategies for reorganizing roles and responsibilities, or what is termed distributing expertise in a community of learners and jointly articulating problems, involved the development of design discourse through strategic questioning	Post test

(*continued on next page*)

Appendix B (*continued*)

No. of paper	Areas of interest	Journal/conference	Included team work/collaboration	Data analysis	Main findings	Experimental design
13.		Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction (IDC) ACM, 2012	Yes		Majority of participants were able to take a 2-dimensional representation on the screen and model its 3-dimensional equivalent using the UCube	
14.	Peers' contribution to individual learning, support from leadership (mentors, interns, and teachers), challenges in coordinating with teams, and usefulness of online feedback to projects	Interactive Learning Environments	Yes	Coding scheme (to identify themes)	How the nested nature of the collectives and the structure of the programming task influenced students' designs, gave opportunities for learning and feedback, and resulted in individual students and the collectives overall successfully employing a range of initialization and synchronization strategies to produce cohesive music videos	Post test
15.	To analyze the complexity of computational concepts, design practices	*Playful User Interfaces*, Springer Singapore, 2014	No	Interviews coded using a two-step process to identify themes. Scratch programs Brennan and Resnick's [4]: framework for computational thinking, photos: analysis in respect to functionality and esthetics	Tangible interface designs replicated common controller designs. The functional variations in the controller designs were minor, what did vary was the attention to esthetics, striking differences in how youth mapped out their physical designs as controllers ranged from unformed heaps of Play-Doh to meticulously designed sculptures that mapped tightly to on-screen elements	Post test
16.	Themes of transparency, aesthetics, and gender	*Harvard Educational Review*		Initial, two-step, and analytical coding scheme	It is important to involve students in creating in two distinct modalities of learning: the digital and the material	Post test
17.	What students were learning and what they were struggling with in designing and crafting	International Conference on International Computing Education Research, ACM	No	Field notes and interview transcripts were initially coded using a two-step open coding process, allowing themes to emerge	Highlight the importance of connecting to larger community value systems as a context for doing computing, the importance of allowing space for youth to make decisions within the constraints of the design task, and the value of tangible e-textiles artifacts in providing linkages between home and school spaces	Post test

18.	Approaches to computing and crafting, computational concepts and practices	WiPSCE'15	Yes		Descriptive statistics, Brennan and Resnick's framework, used two-step open coding for interviews	Great deal of personal and creative expression that the novice designers brought to bear not only on the screen but also in their tangible game design	Post test
19.	Engagement with e-textiles on students' attitudes toward and perceptions of computing	ICER'14	Yes		Thematic analysis using a grounded theory approach	Students learning with e-textiles created a link between coding and making that opened up their views of computer science	Within groups
20.	Understand students' learning of concepts, practices, and perceptions of computing	SIGCSE'13	No		Observations: identified different approaches to computing evident, interviews: analyzed focusing on three aspects	The class was successful in promoting a rich array of computing concepts and practices while at the same time broadening perceptions of computing	Within groups
21.	Learn CS concepts and the wider connection between programming and writing as interrelated processes of composition	SIGCSE'12 (ACM technical symposium on Computer Science Education)	No		Interviews – field notes: coded thematically, scratch projects: Scrape technology, video: capturing a moment-by-moment understanding	Students learned the fundamentals of both programming and storytelling, and this is charted in terms of the products (digital stories) they programmed, the processes (debugging and revising) they utilized, and their overall perceptions of the workshop at its close	Within groups
22.		IDC'14					Post test
23.	Isolate the effect of self-driven discovery versus guided exploration on learning, compute learning gains	Proceedings of the Computer Supported Collaborative Learning (CSCL) 2015	Yes		ANOVA + descriptive statistics	Students who built the human hearing system with EarExplorer without guidance improved their learning gain from pre to post test by ~25% compared to students who followed the step-by-step instructions given by a teacher. Additionally, students in this condition were more likely to take advantage of additional resources that provide relevant information to the task. Followers who are less competent than their peers were less likely to involve themselves in the activity, because it might seem that they did not believe that they could make a significant contribution, or because they were put in a clearly passive observer role	Between groups
24.	Students' technological self-efficacy	IDC'13	Yes				Post test
25.		International Conference on Tangible, Embedded and Embodied Interaction. ACM, 2012. TEI 2012	Yes				

(continued on next page)

Appendix B (*continued*)

No. of paper	Areas of interest	Journal/conference	Included team work/collaboration	Data analysis	Main findings	Experimental design
26.	Areas of programming, electronics, sewing, design, and making things	International Conference on Tangible, Embedded, and Embodied Interaction TEI'11	Yes	Descriptive statistics (representation with graphs)	How craft materials support a more understandable approach to creating technology and that the results of this process can be more transparent and expressive. Participants in workshops were able to conceive of and realize original interface designs from the palette of conductive fabrics and craft materials	Post test
27.	Accessibility and appeal of the techniques, affordances of craft processes in making technology, participants' relationship with programming, and relationship between toolkits and off-the-shelf electronic components	International Conference on Tangible, Embedded and Embodied Interaction. TEI'13 ACM, 2013	No		These techniques demonstrate a tight integration of craft and technology, generating different artifacts and engaging different skills and people than traditional embedded development or electronic toolkits	Within groups
28.	Participants' background, experience, and opinions of electronics and laser cutting	Proceedings of the Designing Interactive Systems Conference. ACM, 2012	No		Engaging diverse activities, skills, and people. Combining education and production adds more value to both. The feasibility of collaboration between an engineer and a designer in a lightweight and ad hoc fashion. In collaboration with an engineer, skill in 3D modeling allows for the production of functional electronic devices	Within groups
29.	Feelings, engagement in programming, attitudes toward programming	Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2013	Yes	Analyzed to determine if the essential qualities outlined were achieved (theme analysis)	Sustained engagement in programing, combination of computational design and fabrication can actively support the expression of personal identity in a positive setting and foster feelings of confidence in programming and support aesthetic and technological literacy. Workshops also promoted a deep understanding of computation as evidenced by critiques of the participants as well as demonstrating the importance of physical prototypes in the design process	Within groups

30.	General impression about the process of workshops (e.g., enjoyment, easiness, freedom, and fluency to use the techniques, use of sketchbooks for circuit prototyping)	SIGCHI Conference on Human Factors in Computing Systems. ACM, 2014		Theme analysis	Participants found electronics surprisingly easy to build., wide range of physical projects people could create, Those who were technically advanced were given the freedom to explore their craft from a new perspective. Participants did not use the sketchbooks as a place for prototyping circuits, they did flip through their examples many times in the process of brainstorming ideas and ways to implement their final projects	Post test
31.	1. Writing Android apps; 2. objects; 3. programming environments; 4. events; 5. decision statements; 6. loops; 7. method calls; 8. method creation; 9. being able to interpret documentation; 10. creating a GUI and coding the components of the GUI; 11. using phone sensor components and services	SIGCSE'13	Yes	Project review (whether they met the learning objectives, 23 projects)	Mobile computing provides a powerful new context for motivating computational thinking	Post test
32.	Computer science competence	SIGCSE'13	Yes	Projects: scored manually, then Scratch static analysis tool scored them, field notes: coded according to the level of help of the staff + descriptive statistics	Students attained competence with several computer science concepts	Post test
33.	Program to correctly create the Predator/Prey simulation using the Simulation Creation Toolkit	SIGCSE'13		Descriptive statistics	Programming at the Computational Thinking Pattern level is an important avenue for investigation into simulation creation in the classroom	
34.	Attitudes toward CS, understanding of what computer scientists do, If computer science is relevant to real world issues	SIGCSE'13	Yes	*T*-test	Campers were provided with a positive mini-college experience and showed significant improvement in attitudes about mathematics, computer science, and computing	Within groups
35.	Computational thinking, engagement with computational concepts	IDC'15	Yes	Final projects: Brennan and Resnick's framework (in which they were engaged with computational concepts), thematically coded, descriptive review of all the features on their board games and in their Scratch programs	One cannot design a board game and craft interactive screen elements without thinking what and how to code and vice versa. Game making appeared to be closely intertwined with game playing as the design of the board games progressed. Boundaries between constructionist and obstructionist gaming are more academic in nature	Post test

(*continued on next page*)

Appendix B (*continued*)

No. of paper	Areas of interest	Journal/conference	Included team work/collaboration	Data analysis	Main findings	Experimental design
36.	Skills and knowledge of computer science, fixedness of math and programming intelligence	Koli Calling International Conference on Computing Education Research ACM, 2014	Yes	Descriptive statistics	Students are not only able to engage with Java, they are able to understand and write basic Java code after only 8 h of playing the game and working with the online materials	Within groups
37.	Ability to read, execute, understand, modify, and create spells (programs)	ACM Conference on Innovation and Technology in Computer Science Education ITiCSE'13		Coding the spells made and theme analysis to the interviews	Subjects became immersed in their programming	Post test
38.	Understand and modify basic programming	*Journal of Computing Sciences in Colleges*	Yes	Analysis of game edits, review of video data, theme analysis	Students were fully engaged with their code, they were approaching quests knowing that changing and understanding spells is the expectation	Post test
39.	Enjoyment, control, and easiness with respect to the programming workshop	Entertainment Computing–ICEC 2014	Yes	Content analysis, focus group	Need for improving DHH children's programming experiences	Post test
40.	Exploring children's attitudes to software and hardware-intensive activities	ACM Conference on Creativity & Cognition, C&C'13	Yes	Content analysis, ANOVA	(a) Software and hardware-intensive activities raise awareness of technology, intensify the experience, and invite students to explore boundaries and increase collaboration and the exchange of views and ideas, and (b) the activity's easiness and usefulness significantly affect children's intention to participate	Post test
41.	Students' performance, attitudes	*Serious Games Development and Applications*, Springer, Berlin Heidelberg, 2013	Yes	ANOVA, content analysis	Intention to learn programming in the future was increased. Students who changed the game code did not improve their performance in the math post-test. Use of a serious game seems to be useful for boys who do not really like the usual instruction processes. Low-performance girls improved more by the traditional way than by playing the game in any way. Most important dependent variable in serious game context seems to be attitude of students	Between groups

42.	Effect of enjoyment, happiness, and anxiety on students' intention to participate in similar creative development activities	Learning and Collaboration Technologies. Designing and Developing Novel Learning Experiences	Yes	ANOVA, Pearson's test (also: evaluating structural equation models)	Positive effects of happiness, and the negative effect of anxiety, indicated that students' enjoyment has no relation to students' intention to re-participate in an activity	Post test
43.	Computational thinking skills	International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 2015, UBICOMP'15	Yes	Three rounds of analysis, iterative coding cycles, to produce a grounded model of computational making (Emerson et al. [14], for ethnographic field work; and Wing [69], for computational thinking)	Aesthetics, creativity, constructing, visualizing multiple representations, and understanding materials are five key factors	Participatory action research design

References

[1] T. Barrett, M. Pizzico, B.D. Levy, R.L. Nagel, J.S. Linsey, K.G. Talley, C.R. Forest, W.C. Newstetter, A review of university maker spaces, in: 122nd ASEE Annual Conference & Exposition, June 14–17, Seattle, WA, 2015.

[2] A.R. Basawapatna, A. Repenning, C.H. Lewis, The simulation creation toolkit: an initial exploration into making programming accessible while preserving computational thinking, in: Proceedings of the 44th ACM Technical Symposium on Computer Science Education, ACM, New York, March 2013, pp. 501–506.

[3] P. Blikstein, Gears of our childhood: constructionist toolkits, robotics, and physical computing, past and future, in: Proceedings of the 12th International Conference on Interaction Design and Children, ACM, New York, June 2013, pp. 173–182.

[4] K. Brennan, M. Resnick, New frameworks for studying and assessing the development of computational thinking, in: Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada , April 2012.

[5] R.F. Bruce, J.D. Brock, S.L. Reiser, Make space for the Pi, in: SoutheastCon 2015, IEEE, New York, April 2015, pp. 1–6.

[6] J.E. Burge, G.C. Gannod, M. Doyle, K.C. Davis, Girls on the go: a CS summer camp to attract and inspire female high school students, in: Proceedings of the 44th ACM Technical Symposium on Computer Science Education, ACM, New York, March 2013, pp. 615–620.

[7] Q. Burke, Y.B. Kafai, The writers' workshop for youth programmers: digital storytelling with Scratch in middle school classrooms, in: Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, ACM, New York, February 2012, pp. 433–438.

[8] G. Charness, U. Gneezy, M.A. Kuhn, Experimental methods: between-subject and within-subject design, J. Econ. Behav. Organ. 81 (1) (2012) 1–8.

[9] S.L. Chu, F. Quek, S. Bhangaonkar, A.B. Ging, K. Sridharamurthy, Making the maker: a means-to-an-ends approach to nurturing the maker mindset in elementary-aged children, Int. J. Child-Comput. Interact. 5 (2015) 11–19.

[10] S. Delle Monache, D. Rocchesso, J. Qi, L. Buechley, A. De Götzen, D. Cestaro, Paper mechanisms for sonic interaction, in: Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction, ACM, New York, 2012, pp. 61–68.

[11] J. Denner, L. Werner, E. Ortiz, Computer games created by middle school girls: can they be used to measure understanding of computer science concepts?, Comput. Educ. 58 (1) (2012) 240–249.

[12] K. Doran, A. Boyce, S. Finkelstein, T. Barnes, Outreach for improved student performance: a game design and development curriculum, in: Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education, ACM, New York, 2012, pp. 209–214.

[13] T. Dybå, D. Torgeir, Empirical studies of agile software development: a systematic review, Inf. Softw. Technol. 50 (9) (2008) 833–859.

[14] R. Emerson, R. Fretz, L. Shaw, Writing Ethnographic Fieldwork, University of Chicago Press, Chicago, IL, 1995.

[15] S. Esper, S.R. Foster, W.G. Griswold, CodeSpells: embodying the metaphor of wizardry for programming, in: Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education, ACM, New York, 2013, pp. 249–254.

[16] S. Esper, S.R. Foster, W.G. Griswold, C. Herrera, W. Snyder, CodeSpells: bridging educational language features with industry-standard languages, in: Proceedings of the 14th Koli Calling International Conference on Computing Education Research, ACM, New York, 2014, pp. 5–14.

[17] S. Esper, S.R. Wood, S.R. Foster, S. Lerner, W.G. Griswold, Codespells: how to design quests to teach Java concepts, J. Comput. Sci. Coll. 29 (4) (2014) 114–122.

[18] D.A. Fields, Y.B. Kafai, A. Strommer, E. Wolf, B. Seiner, Interactive storytelling for promoting creative expression in media and coding in youth online collaboratives in Scratch, in: Proceedings of Constructionism, Vienna, Austria, August 19–23, 2014.

[19] D. Fields, V. Vasudevan, Y.B. Kafai, The programmers' collective: fostering participatory culture by making music videos in a high school Scratch coding workshop, Interactive Learn. Environ. 23 (5) (2015) 613–633.

[20] D. Franklin, P. Conrad, B. Boe, K. Nilsen, C. Hill, M. Len, C. Laird, Assessment of computer science learning in a scratch-based outreach program, in: Proceedings of the 44th ACM Technical Symposium on Computer Science Education, ACM, New York, 2013, pp. 371–376.

[21] B. Garneli, M.N. Giannakos, K. Chorianopoulos, L. Jaccheri, Learning by playing and learning by making, in: Serious Games Development and Applications, Springer, Berlin, 2013, pp. 76–85.

[22] M.N. Giannakos, L. Jaccheri, What motivates children to become creators of digital enriched artifacts?, in: Proceedings of the 9th ACM Conference on Creativity & Cognition, ACM, New York, 2013, pp 104–113.

[23] M.N. Giannakos, L. Jaccheri, Code your own game: the case of children with hearing impairments, in: Entertainment Computing–ICEC 2014, Springer, Berlin, 2014, pp. 108–116.

[24] M.N. Giannakos, L. Jaccheri, I. Leftheriotis, Happy girls engaging with technology: assessing emotions and engagement related to programming activities, in: Learning and Collaboration Technologies: Designing and Developing Novel Learning Experiences, Springer, Berlin, 2014, pp. 398–409.

[25] T. Greenhalgh, How to Read a Paper, second ed., BMJ Publishing, London, 2001.

[26] E.R. Halverson, K. Sheridan, The maker movement in education, Harvard Educ. Rev. 84 (4) (2014) 495–504.

[27] I.E. Harel, S.E. Papert, Constructionism, Ablex, Norwood, NJ, 1991.
[28] C.K. Harnett, T.R. Tretter, S.B. Philipp, Hackerspaces and engineering education, in: Frontiers in Education Conference (FIE), 2014 IEEE, IEEE, New York, 2014, pp. 1–8.
[29] P. Heimann, G. Otto, W. Schulz, Unterricht: Analyse und Planung, vol. 1, ICON Group International, 1970.
[30] M. Honey, D. Kanter, Design–Make–Play: Growing the Next Generation of Science Innovators, New York Hall of Science, New York, 2013.
[31] O.S. Iversen, R.C. Smith, P. Blikstein, E.S. Katterfeldt, J.C. Read, Digital fabrication in education: expanding the research towards design and reflective practices, Int. J. Child-Comput. Interact. 5 (2016) 1–2.
[32] J. Jacobs, L. Buechley, Codeable objects: computational design and digital fabrication for novice programmers, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, 2013, pp. 1589–1598.
[33] Y. Kafai, D. Fields, K. Searle, Electronic textiles as disruptive designs: supporting and challenging maker activities in schools, Harvard Educ. Rev. 84 (4) (2014) 532–556.
[34] Y.B. Kafai, E. Lee, K. Searle, D. Fields, E. Kaplan, D. Lui, A crafts-oriented approach to computing in high school: introducing computational concepts, practices, and perspectives with electronic textiles, ACM Trans. Comput. Educ. (TOCE) 14 (1) (2014) 1.
[35] Y. Kafai, K. Peppler, R. Chapman, The Computer Clubhouse: Creativity and Constructism in Youth Communities, Teachers College Press, New York, 2009.
[36] Y.B. Kafai, K. Searle, E. Kaplan, D. Fields, E. Lee, D. Lui, Cupcake cushions, scooby doo shirts, and soft boomboxes: E-textiles in high school to promote computational concepts, practices, and perceptions, in: Proceedings of the 44th ACM Technical Symposium on Computer Science Education, ACM, New York, 2013, pp. 311–316.
[37] Y.B. Kafai, V. Vasudevan, Constructionist gaming beyond the screen: middle school students' crafting and computing of touchpads, board games, and controllers, in: Proceedings of the Workshop in Primary and Secondary Computing Education on ZZZ, ACM, New York, 2015, pp. 49–54.
[38] Y. Kafai, V. Vasudevan, Hi-Lo tech games: crafting, coding and collaboration of augmented board games by high school youth, in: Proceedings of the 14th International Conference on Interaction Design and Children, ACM, New York, 2015, pp. 130–139.
[39] E.S. Katterfeldt, N. Dittert, H. Schelhowe, Designing digital fabrication learning environments for Bildung: implications from ten years of physical computing workshops, Int. J. Child-Comput. Interact. 5 (2015) 3–10.
[40] N. Khalili, K. Sheridan, A. Williams, K. Clark, M. Stegman, Students designing video games about immunology: insights for science learning, Comput. Sch. 28 (3) (2011) 228–240.
[41] B. Kolko, A. Hope, B. Sattler, K. MacCorkle, B. Sirjani, Hackademia: building functional rather than accredited engineers, Proceedings of the 12th Participatory Design Conference: Research Papers, vol. 1, ACM, New York, 2012, pp. 129–138.
[42] B. Kitchenham, Procedures for performing systematic reviews, Keele, UK, Keele Univ. 33 (2004) (2004) 1–26.
[43] H.C. Lane, C. Cahill, S. Foutz, D. Auerbach, D. Noren, C. Lussenhop, W. Swartout, The effects of a pedagogical agent for informal science education on learner behaviors and self-efficacy, in: Artificial Intelligence in Education, Springer, Berlin, 2013, pp. 309–318.
[44] B. Leduc-Mills, M. Eisenberg, The UCube: a child-friendly device for introductory three-dimensional design, in: Proceedings of the 10th International Conference on Interaction Design and Children, ACM, New York, 2011, pp. 72–80.
[45] E. Lee, Y.B. Kafai, V. Vasudevan, R.L. Davis, Playing in the arcade: designing tangible interfaces with MaKey MaKey for Scratch games, in: Playful User Interfaces, Springer, Singapore, 2014, pp. 277–292.
[46] L. Martin, The promise of the maker movement for education, J. Pre-Coll. Eng. Educ. Res. (J-PEER) 5 (1) (2015) 4.
[47] C.K. Martin, S. Walter, B. Barron, Looking at Learning through Student Designed Computer Games: A Rubric Approach with Novice Programming Projects, Stanford University, 2009 (Unpublished Paper).
[48] S.L. Martinez, G. Stager, Invent to Learn: Making, Tinkering, and Engineering in the Classroom, Constructing Modern Knowledge Press, Torrance, CA, 2013.
[49] D.A. Mellis, L. Buechley, Case studies in the personal fabrication of electronic products, in: Proceedings of the Designing Interactive Systems Conference, ACM, New York, 2012, pp. 268–277.
[50] D.A. Mellis, S. Jacoby, L. Buechley, H. Perner-Wilson, J. Qi, Microcontrollers as material: crafting circuits with paper, conductive ink, electronic components, and an untoolkit, in: Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction, ACM, New York, 2013, pp. 83–90.
[51] K. Moriwaki, J. Brucker-Cohen, L. Campbell, J. Saavedra, L. Stark, L. Taylor, Adapting an art and design workshop to support STEM to STEAM learning experiences, in: Integrated STEM Education Conference (ISEC), 2012 IEEE 2nd, IEEE, New York, 2012, pp. 1–6.
[52] S. Papert, Mindstorms, Basic Books, New York, 1980.
[53] H. Perner-Wilson, L. Buechley, M. Satomi, Handcrafting textile interfaces from a kit-of-no-parts, in: Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction, ACM, New York, 2011, pp. 61–68.
[54] J. Piaget, The Psychology of Intelligence, Harcourt Brace, New York, 1950.
[55] J. Qi, L. Buechley, Sketching in circuits: designing and building electronics on paper, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, 2014, pp. 1713–1722.
[56] K. Qiu, L. Buechley, E. Baafi, W. Dubow, A curriculum for teaching computer science through computational textiles, in: Proceedings of the 12th International Conference on Interaction Design and Children, ACM, New York, 2013, pp. 20–27.
[57] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, Y. Kafai, Scratch: programming for all, Commun. ACM 52 (11) (2009) 60–67.
[58] M. Resnick, S. Ocko, S. Papert, LEGO, logo, and design, Child. Environ. Quart. 5 (4) (1988) 14–18.
[59] J.A. Rode, A. Weibert, A. Marshall, K. Aal, T. von Rekowski, H. el Mimoni, J. Booker, From computational thinking to computational making, in: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, New York, 2015, pp. 239–250.
[60] B. Schneider, E. Bumbacher, P. Blikstein, Discovery versus direct instruction: learning outcomes of two pedagogical models using tangible interfaces, Exploring the Material Conditions of Learning: Opportunities and Challenges for CSCL, the Proceedings of the Computer Supported Collaborative Learning (CSCL) Conference, vol. 1, 2015, pp. 364–371.
[61] L. Schwartz, D. DiGiacomo, K. Gutierrez, Diving into practice with children and undergraduates: a cultural historical approach to instantiating making and tinkering activity in a designed learning ecology, in: International Conference of the Learning Sciences (ICLS), Boulder CO, 2013.
[62] L.H. Schwartz, D. DiGiacomo, K.D. Gutiérrez, Designing "contexts for tinkerability" with undergraduates and children within the El Pueblo Mágico Social Design Experiment, Int. J. Res. Extended Educ. 3 (1) (2015) 94–113.
[63] K.A. Searle, D.A. Fields, D.A. Lui, Y.B. Kafai, Diversifying high school students' views about computing with electronic textiles, in: Proceedings of the Tenth Annual Conference on International Computing Education Research, ACM, New York, 2014, pp. 75–82.
[64] K.A. Searle, Y.B. Kafai, Boys' needlework: understanding gendered and indigenous perspectives on computing and crafting with electronic textiles, in: Proceedings of the Eleventh Annual International Conference on International Computing Education Research, ACM, New York, 2015, pp. 31–39.
[65] O. Telhan, Y.B. Kafai, R.L. Davis, K. Steele, B.M. Adleberg, Connected messages: a maker approach to interactive community murals with youth, in: Proceedings of the 2014 Conference on Interaction Design and Children, ACM, New York, 2014, pp. 193–196.
[66] S. Vossoughi, B. Bevan, Making and tinkering: a review of the literature, in: National Research Council Committee on Out of School Time STEM, National Research Council, Washington, DC, 2014, pp. 1–55.
[67] L.S. Vygotsky, Mind in Society, Harvard University Press, Cambridge, MA, 1978.
[68] A. Wagner, J. Gray, J. Corley, D. Wolber, Using app inventor in a K-12 summer camp, in: Proceedings of the 44th ACM Technical Symposium on Computer Science Education, ACM, New York, 2013, pp. 621–626.
[69] J.M. Wing, Computational thinking, Commun. ACM 49 (3) (2006) 33–35.

Paper 2:

Using Eye-Tracking to Unveil Differences Between Kids and Teens in Coding Activities

Sofia Papavlasopoulou, Kshitij Sharma, Michail N. Giannakos, and Letizia Jaccheri

In Proc. of Conference on Interaction Design and Children (IDC)

Paper 3:

How Do You Feel about Learning to Code? Investigating the Effect of Children's Attitudes towards Coding Using Eye-Tracking

Sofia Papavlasopoulou, Kshitij Sharma, and Michail N. Giannakos

International Journal of Child-Computer Interaction

International Journal of Child-Computer Interaction 17 (2018) 50–60

Contents lists available at ScienceDirect

International Journal of Child-Computer Interaction

journal homepage: www.elsevier.com/locate/ijcci

How do you feel about learning to code? Investigating the effect of children's attitudes towards coding using eye-tracking

Sofia Papavlasopoulou, Kshitij Sharma, Michail N. Giannakos *

Department of Computer Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

ARTICLE INFO

Article history:
Received 3 February 2017
Received in revised form 19 December 2017
Accepted 29 January 2018
Available online 3 February 2018

Keywords:
Children's attitudes
Eye-tracking
Coding
Computational thinking
Constructionism

ABSTRACT

Computational thinking and coding for children are attracting increasing attention. There are several efforts around the globe to implement coding frameworks for children, and there is a need to develop an empirical knowledge base of methods and tools. One major problem for integrating study results into a common body of knowledge is the relatively limited measurements applied, and the relation of the widely used self-reporting methods with more objective measurements, such as biophysical ones. In this study, eye-tracking activity was used to measure children's learning and activity indicators. The goal of the study is to utilize eye-tracking to understand children's activity while they learn how to code and to investigate any potential association between children's attitudes and their gaze. In this contribution, we designed an experiment with 44 children (between 8 and 17 years old) who participated in a full-day construction-based coding activity. We recorded their gaze while they were working and captured their attitudes in relation to their learning, excitement and intention. The results showed a significant relation between children's attitudes (what they think about coding) and their gaze patterns (how they behaved during coding). Eye-tracking data provide initial insights into the behaviour of children, for example if children have difficulty in extracting information or fail to accomplish an expected task. Therefore, further studies need to be conducted to shed additional light on children's experience and learning during coding.

1. Introduction

Computational thinking and coding have become an integral part of the K-12 curriculum, as the Common Core Standards, the Computer Science Teachers Association and the International Society for Technology in Education standards have been widely applied. Coding is considered as a new literacy skill, and is integrated into the school curriculum in many countries, such as Estonia, Finland, Israel, Korea and the United Kingdom, to mention a few. Nowadays, governments seek to teach coding to all and to support young students in creative and problem-solving tasks [1]. Although there is a growing body of research in the area, there is still limited evidence on how to design successful coding experiences for children.

Given the large amount of software available and children-friendly programming environments such as Alice, Scratch, Greenfoot and Kodu, teaching coding has become a more intuitive and engaging experience for young students [2]. In addition, organizations such as "codecademy.com" and "code.org" have strengthened their offerings for children's coding experiences. Thus, while new technologies, innovative pedagogies, guidelines and resources in computing education exist, the challenging question arises of how to choose, design and implement the appropriate learning activity for children. Previous studies grounded in constructionist learning [3] have been successfully utilized both inside and outside the classroom. The results have shown increased interest in coding as well as in understanding the fundamental concepts of problem-solving [4,5].

Combining computers with meaningful programmable objects, such as interactive robots, can provide a valuable learning experience of coding in a fun and playful manner [6]. Previous research described practices to motivate and engage children in coding through making and construction [7]. Robertson and Howells [8] argued that making a game was an authentic learning activity; their exploratory research based on qualitative data from sixth-grade students in Scotland showed that this activity provided motivation, engagement and enthusiasm for learning. Especially when making was combined with block-based programming environments such as Scratch, there was intensive use and improved understanding among the children of concepts including loops and variables [9]. Several studies indicated that coding tasks related

* Correspondence to: Sem Sælands vei 7-9, NO-7491 Trondheim, Norway.
E-mail address: michailg@ntnu.no (M.N. Giannakos).

https://doi.org/10.1016/j.ijcci.2018.01.004

to making, construction, game design and development have been found beneficial for children's attitudes towards coding and skills [10]. The current body of knowledge provides several insights into how to design and implement construction-based coding experiences for children; the existing studies have, however, focused on the experience, fun, enjoyment and engagement of the children (e.g. [4,11]) as extracted from qualitative measures such as observations and interviews and/or quantitative measures including surveys (e.g. [12,13]). Focusing on the use of other measures will help to better understand the way children learn how to code and give insights for the design of coding experiences.

Based on recent studies regarding coding and learning [14,15], one important tool that has been used successfully to unveil the cognitive mechanisms underlying coding by adult programmers is eye-tracking. There are studies explaining expertise [16], collaboration quality [15], learning outcome [14] and task-based performance [17] using eye-tracking data. With children, the use of objective measures such as physiological (eye-tracking) data is important because they are generalizable (more than qualitative and subjective measures), real time and provide more reliable monitoring of users' actions. In contrast to other subjective measures, objective measures are independent of perceptual abilities. In addition, during data collection there is no need to interrupt the activity and ask for ratings. To the best of our knowledge, eye-tracking has not yet been used to investigate how children learn to code and any potential relation between children's attitudes and their gaze patterns.

In this contribution, we designed an experiment where children participated in a full-day construction-based coding activity. We recorded their gaze while they were coding and at the end of the day we captured their attitudes in relation to their perceived learning, excitement and intention during the coding activity. Thus, *in this contribution we investigate the relation between children's attitudes and gaze in coding tasks.*

The rest of paper is structured as follows: in the next section, the related work and background theories are outlined; the third section presents the methodology of the study employed in this article; and the fourth section documents the empirical results. The fifth section discusses the results derived, outlining the limitations and recommendations for future research, while the last section concludes the paper.

2. Related work and background theory

2.1. Learning to code through construction

Papert's [3] constructionism states that each child learns more deeply by actively building knowledge through experience. Children should discover knowledge rather than receiving it passively [18]. In the area of computing education, this is also endorsed by the ACM/IEEE Task Force on Computing Curricula [19]. The ACM/IEEE Task Force emphasizes the importance of the development and mastery of problem-solving skills integrated with real-world, group-based construction-learning activities. Motivated by Papert's constructionist approach, today's educational activities are embedding technology tools that provide learning experiences in educational contexts, which occur in environments that are not always learning oriented. In these types of dynamic learning activities, students are at the centre, taking control and engaging at their own will with a subject. Learning-by-doing, project-based learning, problem-based learning, inquiry-based learning and challenge-based learning are a few such instructional methods, occurring both inside and outside the classroom [20] and focused on learning tasks that promote computing education, computational thinking, design thinking, collaborative work and innovation.

Computer game design and development, modding and computational textiles/fabrication are among the most successfully applied practices which help students to develop coding skills and structure their own learning and thinking by getting involved in the process of coding [5,21]. During such learning tasks, successful construction involves a complex process that fosters skills such as problem-solving, confronting "failures", and strategies to explore and decide possible solutions, as well as structure thoughts and actions [6]. Many tools, such as Cricket, Braitenberg Blocks and Arduino technologies, can provide opportunities to support fruitful learning experiences [22], while digital fabrication can provide Bildung (i.e. deep and sustained learning) [23]. Adams and Webster [24] reported the results from nine years of coding summer camps for middle and high school students. By analysing Scratch programs, they investigated the type of blocks students used and how aspects such as project types were related to the choice of these blocks. The literature suggests that children can successfully complete and learn by simple robot-based coding projects [25]. Robots have the capacity to enhance coding activities and allow children to engage in computational thinking using various programming concepts [26].

In a nutshell, construction-based activities create contextual and meaningful learning environments. As such, after designing a creative coding activity for children, we evaluate its effectiveness, with the primary goal being to understand how children learn coding and design those activities accordingly.

2.2. Students' attitudes and motivations towards coding and self-determination theory

Motivation appears as an important key in learning settings, not only for its positive results but also for its aspects of activation, intention [27] and promoting active learning [28]. Many studies throughout the years have shown that students' motivations have an influence on their performance, satisfaction and well-being [27,29]. In general, the aim is to have positive attitudes towards something that is interesting and, consequently, interest and motivation relate to the individual's actions [30]. Concerning computing and computer science, students' attitudes and motivation are positive and high when projects and visual programming are involved, highlighting fun, commitment, enthusiasm and usefulness [4]. Katterfeldt et al. [31] conducted a EduWear/TechKreativ workshop, where the students used a smart construction kit that revealed a feeling of empowerment and attitudes that increased students' ability to code. Giannakos and Jaccheri [32] found that children's positive attitudes regarding an activity's easiness and usefulness significantly affect engagement and their intention to participate. In particular, game-programming activities for children are motivating, support self-esteem and foster computational thinking [8]. According to Vos et al. [33], game programming reveals enthusiasm and motivation for learning and determination to accomplish a task.

Motivation is an important aspect of human behaviour. Self-determination theory (SDT) has been widely used to understand motivation within educational contexts [29] and is centred on the belief that people have needs that are the basis of self-motivation. There are three basic psychological needs that SDT supports: competence, autonomy and relatedness. According to SDT, opportunities to satisfy any of these three needs contribute to people being motivated. The type of motivation is related to one's goals and attitudes, leading to actions. In addition, SDT includes two different types of motivation: intrinsic and extrinsic. When someone is intrinsically motivated, he/she is engaged in an activity per se, for pleasure and satisfaction from its performance. On the other hand, extrinsic motivation refers to actions from outside sources leading to separable outcomes [27,34]

In our approach, SDT presents a useful theoretical lens to represent children's experience with creative coding activities for learning. In line with the theory, our coding activity is designed to have active participants and to satisfy their needs for autonomy (with occasional support from the instructors), competence and relatedness, facilitating higher motivation in the children. We argue that this activity provides intrinsic motivation, a tendency towards learning and creativity leading to performance, as suggested by Vos et al. [33]. In our study, we provide a creative coding activity that encourages children to make decisions, act independently and work collaboratively with their peers. Hence, autonomy and competence are reinforced. Relatedness involves the development of satisfaction in the social context; therefore, we focus on a pleasurable attitude: excitement, in our case.

Based on the theory and the importance of positive attitudes and motivations in coding activities for children, we hypothesize that our coding activity supports the aforementioned three basic psychological needs [29] so children show high intention, performance expectancy and excitement during and after the coding sessions. On a given learning activity, motives are important to cognitive learning; the level of motivation influences focus and level of effort. More specifically, it could be argued that by having the required motivations, children gain the ability and energy required to sustain positive attitudes towards coding. Positive attitudes facilitate cognitive processing and improve cognitive and affective outcomes. Therefore, this study investigates the impact of our coding activity on students' attitudes (i.e. perceived leaning, excitement and intention to participate in a similar activity) and examines the connection with objectively measured variables illustrating cognition (in our study, eye-tracking data).

2.3. Eye movements in cognitive process of coding

One of the objective technologies for studying cognitive processes in a deep and subjective way is eye-tracking. Eye movements are strongly related to cognition [35,36] and have been used to investigate learning [37], reading [38] and problem-solving [39]. In addition, several studies use eye-movement data to examine adult programmers' visual attention and explore coding, program comprehension [40,41] and debugging [42]. The use of different visual attention measures, such as fixations, saccades or time spent on parts of the screen called Areas of Interest (AOI), can give insights to understand complex cognition activities. Romero et al. [43] compared the use of different program representation modalities (propositional and diagrammatic) in an expert versus novice debugging study, where experts had a more balanced shift of focus among the different modalities than did the novices. Sharif et al. [44] emphasized the importance of code scan time in a debugging task and concluded that experts perform better and have a shorter code scan time. Hejmady and Narayanan [45], comparing the gaze shift between different AOIs in a debugging intergraded development environment (IDE), showed that good debuggers were switching between code, expression evaluation and the variable window, rather than code, control structure and the data structure window. In another study, Aschwanden and Crosby [40] defined each line of the code as an AOI and detected how these lines were perceived. Pietinen et al. [46] assessed the quality of collaboration by measuring joint visual attention in a co-located pair programming setup, using the number of overlapping fixations. Bednarik and Tukiainen [41] examined the coordination of different program representations in a program-understanding task. Experts concentrated more on the source code rather than looking at the other representations.

Though many studies have used cognitive neuroscience techniques such as eye tracking [47] to examine the role of eye movements in adults' coding cognition and behaviour, there is a lack of studies using them to assist our understanding of children's cognitive processes in coding activities [48]. Hence, we used eye-tracking to capture children's allocated attention to different sources of information during our creative coding experience.

2.4. Cognitive load theory

Cognitive load theory (CLT) implies that people have a limited working memory and that the amount of information they can process cannot therefore exceed the limit at which they are overwhelmed [49]. There are three types of cognitive load: intrinsic, extraneous and germane. Intrinsic load refers to the task and its core characteristics that must be processed. Extraneous load is based on the form of representation and the techniques used in the instructional design. Germane load involves information consolidation and refers to schema production for permanent knowledge.

The intrinsic load effort in our case of a designed coding activity is represented by the performance of the task and its own load due to complexity. The use of the Scratch programming environment for the completion of the activity and the instructional details relate to the extraneous load. Finally, the germane load consists in the effort and processes from the task which are directed to the relevant learning [50].

Cognitive load can have an influence on visual attention and behaviour. The eye's different fixations show the distribution of attention [51], while the cognitive process from graphic and textual visual materials is connected with fixation behaviour (locus, duration and sequence) [52]. In particular, eye-movement measures such as number of fixations, fixation duration, duration time and different scanning paths can reveal important aspects of the learners' cognitive process [36]. High fixation duration depicts high cognitive activity [53] and fewer saccades can be related to less cognitive effort in terms of task performance [47]. In a study about maths and physics problems, participants had longer fixations in the more complicated parts of the problem [54].

In this study, in line with CLT, the designed coding activity has an overall cognitive load that subsequently influences children's cognitive process and can become overwhelming. We assume that the working memory of children, and especially of novices in coding, can quickly be overloaded by task complexity, and that this will lead to an inefficient learning environment. Thus, we attempt to use an eye-tracking technique as a proxy for cognition [47] to investigate children's cognitive processes in learning [55] during our creative coding activity. The eye measures will show the cognitive overload and we examine their relationship to children's attitudes regarding the activity.

2.5. Goal of the study

Coding activities based on constructionist learning enhance learners' motivations and help them to incorporate knowledge, attitudes and behaviour to achieve effective learning and performance [56]. In addition, there is a need to have the proper instructions and guidance to support self-efficacy for learning [57]. Nevertheless, the cognitive load of these activities can be high and the increased task complexity can become overwhelming. Therefore, to create an effective and efficient learning environment, motivational effects should be considered [57,58].

Based on previous research and the theoretical grounding, we assume that cognitive load is related to children's attitudes and motivation in creative coding activities. In particular, we predict that more highly motivated children with more positive attitudes have better management and a lower cognitive load. The present study fills the gap of using eye movements as an objective measurement to depict children's cognitive processes while coding and examine how they are related to their attitudes.

The aim of this study is summarized by the following research question:

• *What is the relation between children's attitudes and gaze in coding activities?*

3. Methodology

3.1. The coding activity

Based on the constructionist approach and its main principle, learning by doing [59], as well as previous efforts [32], we conducted a coding workshop at the Norwegian University of Science and Technology, in Trondheim, Norway. Our coding workshops are out-of-school activities, in which children from 8 to 17 years old interact with digital robots, using Scratch for Arduino (S4A), and then code their own game using the Scratch programming language. At each workshop the children work in pairs or triads and the activity lasts for approximately four hours. Five assistants with previous experience in similar activities are responsible for instruction and the procedure for the workshops.

The workshop consists of two main parts, interaction with the robots and creating games with Scratch; Fig. 1 depicts the flow of these two parts.

Interaction with the robots: During the first part of the coding activity, the children interact with digital robots built by an artist using recycled materials, mainly from computer parts. First, as the children enter the room and are welcomed by the assistants, they sit in teams next to one robot. The assistants give a brief presentation of the workshop's activities and ask each of the children to pay attention to a worksheet placed on the desk next to them. The goal is to familiarize themselves with the robots by filling in simple questions regarding the exact place and number of the sensors and lights on the robots. Then, the children use a paper tutorial with instructions (Fig. 2) for how to make the robots react to the physical environment with visual effects using simple loops of Scratch for Arduino (e.g. to make the tongue of the snake robot move when there is less light at a sensor). The teams work collaboratively and independently to complete this task (Fig. 3 left). The duration of the first part varies from 45 to 90 min. When all the teams have finished, the children have a break before the next section begins. This part of the workshop offers a smooth start to coding, including tangible objects. The interaction with digital robots provides a better understanding of STEM subjects by showing the connection with the physical world, helping the children to cope with difficult problems [60]. The children are introduced to coding by playfully interacting with the robots while they get motivation and inspiration.

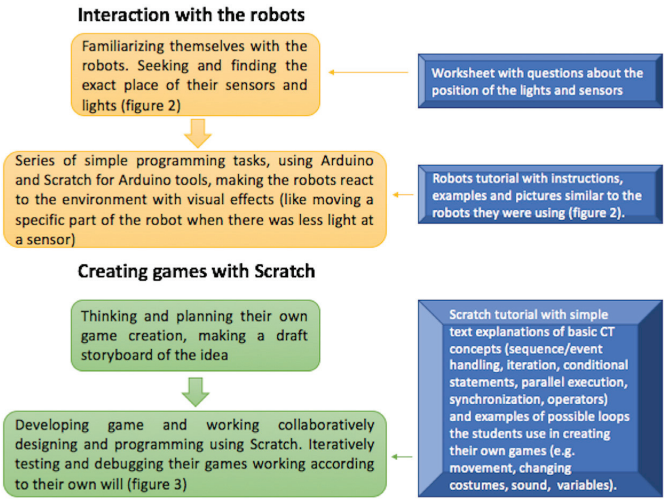

Fig. 1. Description of the two activities in the workshop.

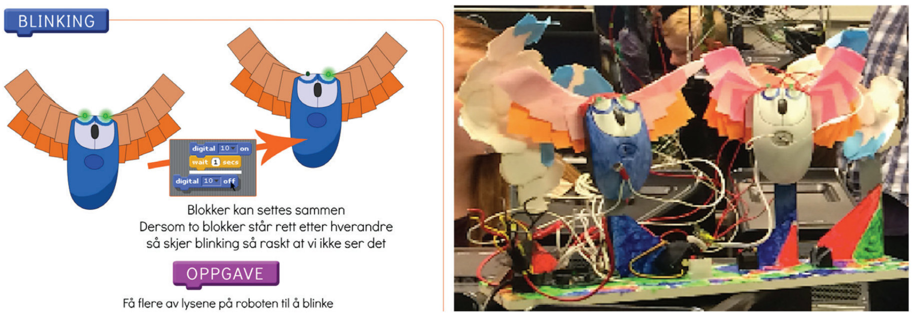

Fig. 2. Example of the robots' tutorial on how children interact with robots.

Fig. 3. Children interacting with the robot (left) and example of developed game (right).

Creating games with scratch: This section is the main activity of the workshop and lasts approximately three hours, without the presence of the robots. The goal is to successfully develop a simple game, coding in Scratch. To achieve this goal, the assistants give another paper tutorial with examples of all the basic Computer Science (CS) concepts and possible loops they should use to complete their own game. The assistants advise the children how to manage the process of game development, working collaboratively. First, they should think about and decide the story for their game and then create a draft storyboard. When they finish that, they start coding using Scratch. The children can ask for support from the assistants whenever they need it throughout the activity. The assistants offer their guidance to the teams, helping them to complete their games and introducing even more complex CS concepts when needed. Finally, after the completion of the games, the children reflect and play each other's games (Fig. 3 right).

3.2. Sampling

We conducted the study at a dedicated lab space at the Norwegian University of Science and Technology, in Trondheim, Norway. Specifically, the study lasted two weeks during Autumn 2016, with 44 children from the eighth to twelfth grades (aged 8–17 years old), 12 girls (mean age: 12.64, standard deviation (SD): 2.838) and 32 boys (mean age: 12.35, SD: 2.773). Five workshops were held in total, all following the same process for the coding activity, addressed to novices in coding. Some of the participants in the sample (13–17 years old) were recruited from the local schools who had applied to take part in our activity (called Kodeløypa in Norwegian, meaning the path to coding). The other set of participants (8–12 years old) were youngsters who attend local coding clubs (Kodeklubben: https://trondheim.kodeklubben.no/) as an after-school activity. The children volunteered their participation in the eye-tracking study and the legal guardians provided a written informed consent form for their child, giving permission for the data collection. In our sample of 44 children in total, 27 children had attended 0–1 workshops about coding before, 15 children 2–5 workshops, and only 2 children more than 5 workshops. In addition, among the children aged 13–17 years, 18 out of 29 participants had chosen less than 3 (mean: 3.06, SD: 1.404) on a seven-point Likert scale measuring their own experience in coding, and only 4 chose more than 5, while none of them chose more than 6.

3.3. Measures

As mentioned before, this study is one of the few so far utilizing children's gaze. We recorded children's gaze while they were coding using the Scratch environment during both parts of the activity. The eye-tracking data was collected using four SMI and one Tobii eye-tracking glasses. The sampling rate for all the eye-tracking glasses was set to be 30 Hz for the binocular eye-tracking. The average accuracy for both SMI and Tobii glasses was 0.5° at a distance of 40 cm.

Many measures have been used to examine cognition. Fixations calculate the time spent on a specific location, reflecting attention and processing time, while saccades represent the shifts between fixations [47].

Based on the literature and prior studies [61], we selected the following gaze measures:

1. Fixation duration: High fixation duration depicts that the participant is having difficulty in extracting information [52]. The authors used a mental rotation task, with 0, 120 and 180°, to study the relation between problem difficulty and gaze patterns. The results showed that with an increase in the rotation angle (increasing difficulty), the fixation duration at the centre of the figure and the arms of the structures increased [52].
2. Saccade amplitude: longer saccades show meaningful transitions in terms of attention [62]. In a web search task, the authors used a set of different tasks on a webpage, so that the participants had to look for particular information to complete the tasks. The results showed that pre-planned eye movements were accompanied by longer saccades [62].
3. Change in saccade direction: the angle between two lines, if more than 90°, reflects a change of plans, revision or a failed expectation/hypothesis/anticipation [63]. In a usability study, the authors found that the change in saccade direction often depicted the behaviour of not finding something which the participants anticipated to find at certain places [63]. This can be translated, in terms of programming behaviour, as having a certain hypothesis and a failed verification.

At the end of the activity, the children completed a paper-based survey. The surveys gathered feedback on the children's attitudes regarding the coding activity. In Table 1, we summarize the operational definitions of these factors, the items and their respective bibliographical sources. The children were asked to rate their experience with the coding activity regarding their learning, excitement and intention. In all measures, a five-point Likert scale was applied using smiley faces [64] (Fig. 4) . Table 1 clearly exhibits the questions put to the children.

3.4. Data analysis

As mentioned above, 44 children were involved in this study. To test our research question the data were separated into three

Table 1
The attitudinal factors and their respective questions, operational definitions and sources.

Factors	Operational Definitions	Item	Source
Learning	Perceived learning (we refer to this as learning in this paper) is the degree to which children indicate their performance.	Please indicate if you learned new things during the coding activity (Not at all — Very much)	[65]
Excitement	Excitement is the degree to which children indicate their excitement for the coding activity	Please indicate how you feel about participating in the coding activity (Dull — Exciting)	[66]
Intention	Intention is the degree of children's willingness to participate in a similar activity.	Please indicate how much you want to attend similar coding activities in the future (Not at all — Very much)	[32]

Table 2
Descriptive statistics of the study.

Variable	Median	Mean	SD	Min	Max
Learning (scale 1–5)	5.0	4.7	0.82	1	5
Intention (scale 1–5)	5.0	4.5	0.76	2	5
Excitement (scale 1–5)	5.0	4.6	0.65	3	5
Fixation duration (milliseconds)	268.46	270.8	90.62	110.0	579.9
Saccade direction change (milliseconds)	36.70	38.76	16.06	12.06	92.47
Saccade amplitude (degrees)	177.24	186.78	61.07	92.81	356.98

Table 3
Testing the effect of children's attitudes in their eye-tracking patterns during coding.

Variables	Learning				Intention				Excitement			
	Mean (SD)			*F*-Value	Mean (SD)			*F*-Value	Mean (SD)			F-Value
	3 or less	4	5		3 or less	4	5		3 or less	4	5	
Fixation duration	488 (79)	328 (43)	244 (65)	16.06**	389 (52)	293 (116)	243 (66)	14.41**	424 (112)	287 (100)	246 (63)	4.83*
Saccade direction change	76.6 (20)	44.6 (6.3)	34.9 (12.1)	4.47*	58.2 (15.8)	45.9 (18.1)	32.9 (11.5)	6.94*	63 (23)	46 (16)	33 (11)	5.13*
Saccade amplitude	115 (14)	143 (38)	198 (59)	19.35***	123 (17)	177 (60)	200 (60)	16.46***	141 (26)	168 (59)	198 (62)	5.32*

*** Significance level $p < .001$.
** Significance level $p < .01$.
* Significance level $p < .05$.

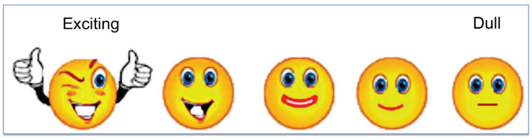

Fig. 4. Example of emoticons used in the survey to measure children's attitudes.
Source: Adopted from Hall et al. [64].

groups, each for one of the three attitudinal factors: learning, intention and excitement. The first group consisted of children who rated the respective attitude 3 or less (relatively low), the second of children who rated it 4 (relatively medium) and the third of children who rated it 5 (relatively high). First, we used Levene's test to examine the homogeneity of variance and the Shapiro–Wilk test to evaluate the normality criterion [67,68] in order to use ANOVA analysis (see the table in the Appendix). Afterwards, 9 separate one-way independent ANOVA tests were conducted to examine our research question. For all tests we did not assume equality of variance across groups. The *p*-values for the main and post-hoc tests are computed in accordance with the Bonferroni correction for the repeated tests. Also, since we are not assuming the variance across groups to be equal, the F-values are adjusted according to the Welch correction for the partial degrees of freedom.

4. Research findings

Children expressed high learning and excitement (4.7/5 and 4.6/5, respectively) for the coding activity. Additionally, they expressed slightly lower intention (4.5/5). High levels of these attitudes indicate positive views concerning their learning performance and beliefs regarding their future engagement with coding activities. The descriptive statistics about children's attitudes and eye-tracking measures are summarized in Table 2.

As mentioned before, to examine our research question one-way ANOVA was used, and the three independent variables (learning, excitement, intention) and the three dependent variables (fixation duration, saccade direction change, saccade amplitude) were included. As can be seen from the outcome data in Table 3, children's learning, excitement and intention exhibited a highly significant relation with their gaze patterns, supporting our research assumption. The results of the 9 separate one-way independent ANOVAs (without assuming equal variances across groups) are summarized in Table 3.

We observe the following relations between the attitudes (learning, intention and excitement) and the gaze variables (fixation duration, saccade amplitude and saccade direction change).

For gaze and perceived learning, we observe a significant relation between all the gaze variables and learning (Fig. 5). In particular, the children who reported higher learning had lower fixation duration ($F[2, 4.37] = 16.06$, $p = .009$), lower saccade direction change ($F[2, 4.47] = 4.47$, $p = .03$) and higher saccade amplitude ($F[2, 8.32] = 19.35$, $p = .0007$) than those who reported lower learning. Furthermore, pairwise comparisons show that the average fixation duration decreases significantly with an increase in perceived learning (3 vs. 4 $F[1, 2.73] = 10.33$, $p = .05$; 3 vs. 5 $F[1, 2.25] = 26.74$, $p = .02$; 4 vs. 5 $F[1, 6.85] = 14.19$, $p = .007$). Considering the average change in the saccade direction, it does not differ significantly from 3 to 5 ($F[1, 2.12] = 12.66$, $p = .09$) or 3 to 4 ($F[1, 2.24] = 7.70$, $p = .06$), but it is significantly higher for the children who reported the perceived learning as 4 than for those who reported the perceived learning as 5 ($F[1, 6.85] = 14.19$, $p = .03$). Finally, concerning the average

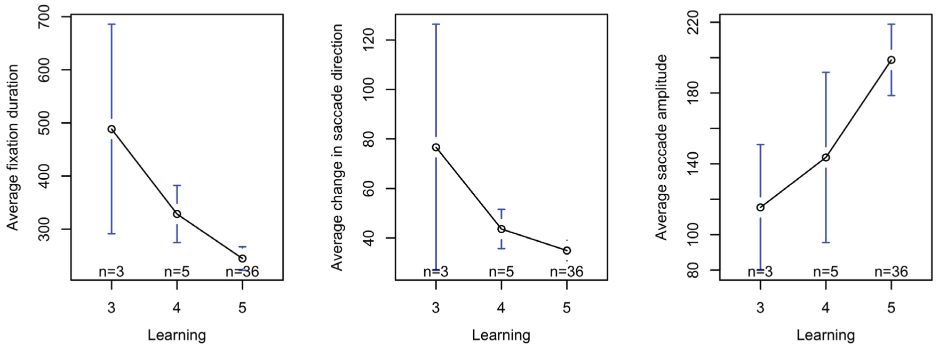

Fig. 5. The influence of learning on children's gaze during the coding activity. The blue bars are the 95% confidence intervals.

saccadic amplitude, it does not differ significantly from the 3 to 4 rating ($F[1, 5, 45] = 2.16$, $p = .19$), but it is significantly higher for 5 than for the other two ratings (3 vs. 5 $F[1, 10.71] = 41.70$, $p = .0005$; 4 vs. 5 $F[1, 7.60] = 6.97$, $p = .02$).

For gaze and intention, we observe a significant relation between all the gaze variables and the intention to program (Fig. 6). In particular, the children who reported higher intention to code had lower fixation duration ($F[2, 10.40] = 14.41$, $p = .001$), lower saccade direction change ($F[2, 8.81] = 6.94$, $p = .01$) and higher saccade amplitude ($F[2, 18.09] = 16.46$, $p = .00008$) than those who reported lower intention. Pairwise comparisons show that the fixation durations do not differ significantly between levels 4 and 5 ($F[1, 11.11] = 1.65$, $p = 0.22$), but do decrease significantly for levels 4 ($F[1, 12.97] = 4.88$, $p = .04$) and 5 ($F[1, 6.47] = 30.46$, $p = .001$) as compared to level 3. When we conducted the pairwise comparisons for the saccade direction change, we did not observe any difference between levels 3 and 4 ($F[1, 9.19] = 1.82$, $p = .20$), but there is a significant decrease in the direction change for level 5 when compared against levels 3 ($F[1, 4.75] = 11.72$, $p = .02$) and 4 ($F[1, 11.61] = 4.54$, $p = .05$). Finally, considering the pairwise comparisons for saccadic amplitude, it increases significantly between levels 3 and 4 ($F[1, 11.60] = 6.94$, $p = .02$) and 3 and 5 ($F[1, 22.85] = 32.89$, $p = .0001$); however, we did not observe any significant differences between levels 4 and 5 ($F[1, 15.68] = 1.12$, $p = .30$).

For gaze and excitement, we observe a significant relation between all the gaze variables and excitement during the coding task (Fig. 7). In particular, the children who reported higher excitement had lower fixation duration ($F[2, 6.48] = 4.83$, $p = .05$), lower saccade direction change ($F[2, 6.50] = 5.13$, $p = .04$) and higher saccade amplitude ($F[2, 11.39] = 5.32$, $p = .02$) than those who reported lower excitement during the coding task. Pairwise comparisons show that the fixation durations are not different for levels 3 and 4 ($F[1, 5.28] = 4.40$, $p = .08$) and 4 and 5 ($F[1, 9.92] = 1.35$, $p = .27$), but fixation durations are significantly lower for level 5 than those for level 3 ($F[1, 3.25] = 9.73$, $p = .04$). The saccadic direction change is significantly lower for level 5 than for level 4 ($F[1, 10.62] = 5.18$, $p = .04$) and we do not observe any other differences between levels 3 and 4 ($F[1, 4.32] = 1, 95$, $p = .22$) or levels 3 and 5 ($F[1, 3.20] = 6.86$, $p = .07$). Finally, the saccade amplitudes are not significantly different between levels 3 and 4 ($F[1, 10.96] = 1.36$, $p = .26$) and levels 4 and 5 ($F[1, 13.52] = 1.62$, $p = .22$); however, saccade amplitudes are significantly higher for level 5 than those for level 3 ($F[1, 8.67] = 11.02$, $p = .009$).

5. Discussion and conclusions

This study is the first attempt to investigate potential relations between children's attitudes and their gaze during coding activities. For that purpose, in addition to the attitudinal survey (learning, excitement and intention), we collected eye-tracking data from children aged 8 to 17 years during our construction-

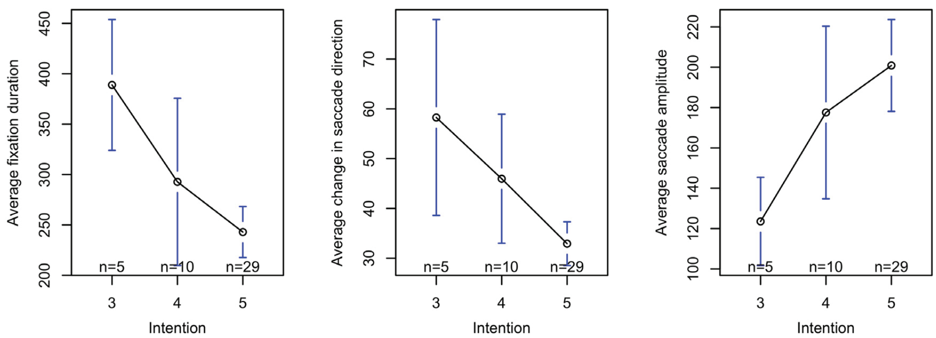

Fig. 6. The influence of intention on children's gaze during the coding activity. The blue bars are the 95% confidence intervals.

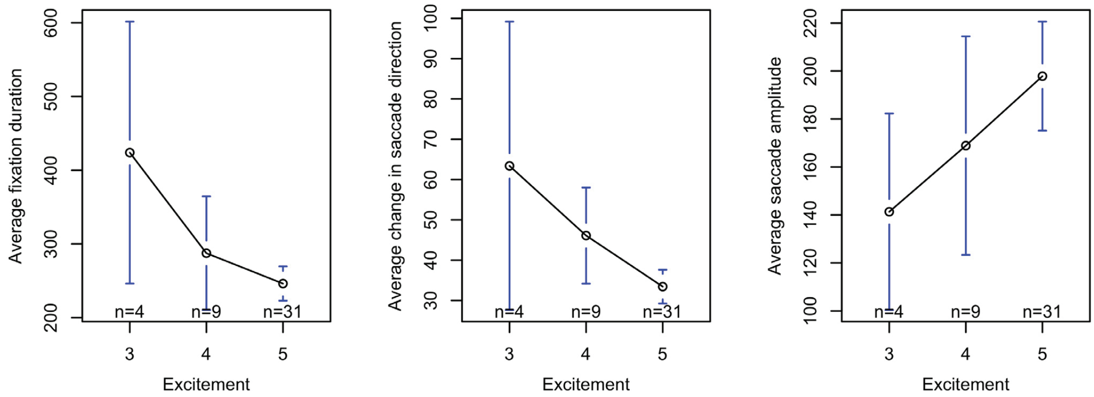

Fig. 7. The influence of excitement on children's gaze during the coding activity. The blue bars are the 95% confidence intervals.

based coding activity. The results showed a significant relation between children's attitudes and their gaze patterns. There are many studies [31,69] focusing on how children interact with digital fabrication and construct games using a programming environment. In our study, we used robots and the Scratch tool.

Change in children's attitudes through game making and engagement with digital media is as important as motivation to learning, since it represents a long-term profit and can be expressed as a later career interest [70]. Our study suggests that gaze patterns and attitudes can be correlated. The three different gaze measures used represent children's difficulties in extracting information during a coding activity (fixation duration), the number of trials needed to learn something during coding (saccade direction change) and children's goals and expectations during coding (saccade amplitude). As was expected, children who had fewer difficulties and could handle the cognitive load better had higher scores in their attitudes. When the instructional conditions enhance their motivations, offer the proper way to manage the tasks' overwhelming conditions and maintain children's focus, there are positive results from their experience. This finding also highlights the importance of proper assistance from the instructor and the materials/tools in coding activities.

In particular, high fixation duration corresponds to children's difficulty in extracting the information needed to accomplish a task. Lower fixation duration depicts the fact that the user (the child in our case) is experiencing less difficulty in extracting information from the stimulus [52]. We found that children who report lower learning have higher fixation duration. That can be attributed to the fact that they possibly put a lot of effort into understanding and choosing the appropriate tools and/or commands in accomplishing the task of creating their game and controlling the robots, resulting in a higher cognitive load. On the contrary, children who believe that they learn more have lower fixation duration, so less of a cognitive load, assuming that they were frequently checking different commands until they found the preferred one and also taking quicker decisions while coding [71]. High saccade amplitude or long saccades show that the transitions in attention are more meaningful than transitions with shorter saccades [63]. In other words, longer saccades depict more of a hypothesis–verification kind of gaze behaviour, and are also indicative of multiple trials to learn a particular topic. This is in accordance with previous studies where young children who are novices in coding rarely try to debug their program and when they do so, find great difficulty in solving issues with a program that is not properly executed [72]. Perkins et al. [73] describe different categories of children while solving a problem: "stoppers", who have no intention of trying different problem-solving methods; "movers", the ones who are willing to try different ways; and "extreme solvers", who try different ways without carefully thinking about them.

One interesting result is that the differences in children's gaze were higher for intention and learning than for excitement. This is possibly related to the fact that excitement derives from intrinsic motivation, driven by interest and enjoyment in the coding activity, and exists within the individual. On the other hand, intention and learning after an educational activity are attitudes closer to the learning tasks than the individual, so are more complicated to effect. Moreover, in terms of the reported excitement, the children with higher levels of excitement had the same characteristics as those who reported high learning. It is not a surprising result that when children experience difficulties in coding they feel less excited, as fun and enjoyment derive from successfully completing functional projects that also give a positive overall experience [12,13].

Expectation confirmation theory [74] asserts that continuance intention is mainly determined by satisfaction with prior experience. To understand this, one has to recall that satisfaction is synonymous with affect (i.e. a positive or negative feeling), and further that affect (as attitude or satisfaction) in prior learning studies is found to be an important predictor of intentions and decisions concerning the use of learning tools and practices (e.g. [75]). Enjoyment and satisfaction affect children's intention to participate in similar activities in the future [76,77]. In our case, children with higher excitement had lower saccade direction change. Likewise, those who reported higher intention had lower saccade direction change. This type of similarity in children's gaze pattern represents that the ones who reported a high level of excitement have also high intention, in accordance with expectation confirmation theory. In the literature, a high amount of saccade direction change depicts sudden changes in short-term goals or expectations [62].

During our study the researchers also collected some notes from observations and assistants' comments, adding some interesting qualitative findings that illustrate children's behaviour during the coding activity. In general, the majority of the children expressed their satisfaction with the activity, and also mentioned a nice atmosphere. Their comments included sentences such as "it is so funny I can make the tail move" or "I like that I am with friends all learning how to code". Enthusiasm was more obvious in younger children's teams, and their willingness to code was expressed even with quarrelling. In conjunction with other studies [11], it was clear among the teams that girls were focusing more on the drawing and the story. In addition, some teams were working more methodically, following the tutorials, while others were working more independently, but asking more frequently for help from the assistants.

5.1. Practical and theoretical implications

Our eye-tracking data analysis in a coding activity with children is a first step towards using eye-tracking to unveil children's experience in the coding process. Several studies have successfully shown a clear relation between gaze patterns and performance, learning strategy and other personality factors [14,17] That makes our approach an important contribution in eye-tracking, child–computer interaction and computer science education communities.

Scholars, educators and practitioners should pay particular attention to children's attitudes, since they heavily influence their experience. A coding activity should not overlook children's excitement, fostering enjoyment and confidence (i.e. high perceived learning). Instructors should focus on presenting support at the appropriate time, to reduce the cognitive overload and help children achieve a fruitful coding experience.

Our study verifies and extends the work of Abeysekera and Dawson [78], who suggest combining CLT and SDT to create a theoretical model for the flipped classroom, which investigates the increase of motivation to better manage cognitive load. This study confirms the fact that motivated children with positive attitudes have better management of cognitive load, as was represented by their eye movements. Indeed, we examine the two theories in the different context of children's coding activity, providing empirical support. Moreover, including eye-tracking data in the design of our study expands the scope of the theories providing evidence from the use of an objective data-collection method. In addition, other studies using eye-tracking have mainly focused on multimedia learning theories directly related to vision [55,79], but from our perspective, including SDT shows evidence that goes deeper into users' behaviour.

Our findings demonstrate that the way children perceive the cognitive load from the learning process is related to their attitudes. According to CLT's relation to learning, instruction should align with human cognitive architecture [50] as well as enhance the motivation of learners [58]. Motivation and positive consequences are related [27], so self-determined children feeling excitement when performing a task may have a higher possibility of repeating the task in the future. Supportive teaching methods should provide guidance to help children distinguish the relevant factors to complete the task, preventing them from becoming overwhelmed by irrelevant information and actions. For example, they should help them focus on specific parts of the screen to find the respective code segments, split the code into meaningful chunks and trace the coding process in an effective way. In parallel, during learning activities instructors should foster students' self-confidence in their ability to complete the task successfully and ensure a pleasant and motivated environment. Moreover, there is a need for properly designed tools to help reduce cognitive overload. The design of the aesthetics of the visual coding tool is important to give a pleasant sense for children's use, but it should also help them indicate in a clear way the input and output values while coding. One example could be the clear representation of code segments and less complexity in scripting (e.g. fewer sprites and stacks of code). Another thought might be the design of dynamic coding tools that could be further developed according to children's progress in the coding task, such as starting with fewer code segments and gradually providing more advanced coding possibilities in relation to progress. In short, during coding activities for children it is important to take the motivational and cognitive effects equally into consideration in order to support effective and efficient learning environments.

5.2. Limitations

The present study is one of the first to offer insights into the relation of gaze patterns and children's attitudes. Nevertheless, some limitations should be mentioned. First, we faced a difficulty in capturing the gaze of 8–12 year-old children, since they were constantly moving their heads during the workshop and the glasses were sometimes irritating, so they had to remove them for some of the time. Their young age and the fact that most of the time they were very excited during the activity and spent a lot of time talking to each other, sharing their experience, made it very difficult to have good-quality data. The data can be corrupted due to many reasons. For instance, some of the participants removed the glasses and wore them again without the experimenters noticing, which resulted in some calibration errors, and thus data from those participants, after we noticed the lack of calibration, were removed from the analysis. Another reason for removing part (or the whole) of the data from a participant is that when they looked directly into a light source, the automatic calibration took a few seconds to recover from the sudden change in luminance. Nevertheless, we could use 75% of the data collected. Lost data was mainly from gaze in places that were not relevant for the experiment; for any other reason data were few and very carefully removed in order not to affect the analysis and provide more valid remained data. In their study, Nevalainen and Sajaniemi [80] reported invalid data of less than 10% of all the collected eye-tracking data from three different tracking devices, while Pernilla and Zhai [81] removed data from three out of fifteen participants in their eye-tracking study. Second, the duration of the activity was not strictly equal every day: children were recruited from the local coding clubs (Kodeklubben: https://trondheim.kodeklubben.no/) and schools, so we had to adjust the activity and sometimes streamline the schedule. However, this adjustment turned out to be constructive, since the children managed to complete sufficient of the workshops' activities and it did not become overwhelming for the majority of them, so that they did not report boredom or decrease their attention. Our coding activity is designed for children who have no previous experience in coding, so everyone can attend. Nevertheless, we cannot know the actual level of children's coding skills and exactly how much they have been exposed to coding before at school and/or in home activities. In addition, at the time of our study the local clubs were just starting their academic year, so the younger participants (8–12 years old) had not had many courses. Another limitation of the study was the lack of structured qualitative data (e.g. observations and interviews). The collection of that type of data could provide valuable insights into our findings and shed some light on children's behaviour during construction-based coding activities. Therefore, qualitative data collection could be taken into account in future studies. Finally, our study took place in Norway and participants voluntarily participated, so other sampling methods and demographic variables (i.e. educational level, family status) might have a contingent effect on children's attitudes.

5.3. Future work

For future work an opportunity will be to collect and analyse eye-tracking data in relation to gender differences. In her study, Robertson [11] identifies differences in game products for boys and girls and in order to investigate these differences she examined the time spent in different types of making process. Eye-tracking measures could be a promising approach to explaining gender differences from another perspective. Furthermore, in future studies attention should be paid to investigating the learning outcomes in terms of learning-specific computer science concepts and how they are related to different gaze patterns. In addition, the study could be extended to compare the results from children's gaze patterns in other attitudes as well as comparing alternative coding learning environments.

5.4. Conclusion

The present study can be regarded as a first step towards the use of eye-tracking method to examine children's learning behaviour in creative coding activities. Based on the constructionist approach we conducted a coding workshop in which children were coding interactive robots and games using the Scratch programming environment. With the goal to examine how children's attitudes and gaze are related, we collected their attitudes via surveys and recorded their gaze via eye-trackers. The examined attitudes include *perceived learning, excitement* and *intention*, all measured in five-point Likert scale using smiley faces. For the gaze we used three different measures connected to cognition, these are: *fixation duration* (showing difficulties in extracting information), *saccade direction change* (efforts needed to learn something) and *saccade direction change* (goals and expectations during the activity).

To support our assumption, that cognitive load relates with children's attitudes and motivation, our approach is grounded on self-determination theory and cognitive load theory. The results demonstrate a significant relation between attitudes and children's gaze patterns during the coding activity. More specific, children who indicated better management of cognitive load, expressed higher scores in their attitudes. Findings also suggest that children with higher reported excitement and learning had the same characteristics. This study demonstrates that the use of eye-tracking provides information about children's approach on handling coding tasks; that can be especially beneficial for the design of successful coding activities for children. Appropriate teaching methods and tools should focus on providing support avoiding unnecessary disruptions that can become overwhelming.

Acknowledgements

The authors would like to express their gratitude to all of the children, teachers and parents for volunteering their time. Our very special thanks go to Letizia Jaccheri, Kristin Susanne Karlsen, Ioannis Leftheriotis, Amanda Jørgine Haug, Lidia Luque, Fernandez, Marjeris Sofia Romero, Eline Stenwig and Kristoffer Venæs Monsen. The project has been recommended by the Data Protection Official for Research, Norwegian Social Science Data Services (NSD), following all the regulations and recommendations for research with children.

This work was funded by the Norwegian Research Council under the projects FUTURE LEARNING (number: 255129/H20) and by NOKUT under the Centre for Excellent IT Education (Excited) (number: 16/02049).

Appendix

See Table A.1.

Table A.1
Normality test (Shapiro–Wilk test, *p*-values, for the three levels of perceived learning, intention and excitement.

Variable	Learn (3, 4, 5)	Intention (3, 4, 5)	Excitement (3,4, 5)
Fixation duration	0.13, 0.74, 0.81	0.45, 0.41, 0.44	0.27, 0.74, 0.08
Saccade amplitude	0.44, 0.42, 0.83	0.56, 0.43, 0.85	0.73, 0.13, 0.58
Saccade direction	0.31, 0.75, 0.54	0.48, 0.31, 0.69	0.45, 0.12, 0.45

References

[1] P. Hubwieser, M. Armoni, M.N. Giannakos, R.T. Mittermeir, Perspectives and visions of computer science education in primary and secondary (K-12) schools, ACM Trans. Comput. Educ. 14 (2014) 7.
[2] M.N. Giannakos, Reviewing the affordances of tangible programming languages: Implications for design and practice, in: IDC Workshop Every Child a Coder? Research Challenges for a 5–18 Programming Curriculum, 2015.
[3] S. Papert, Mindstorms: Children, Computers, and Powerful Ideas, Basic Books, Inc., 1980.
[4] J.-M. Sáez-López, M. Román-González, E. Vázquez-Cano, Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools, Comput. Educ. 97 (2016) 129–141.
[5] S. Papavlasopoulou, M.N. Giannakos, L. Jaccheri, Empirical studies on the Maker Movement, a promising approach to learning: A literature review, Entertain. Comput. 18 (2017) 57–78.
[6] M.U. Bers, L. Flannery, E.R. Kazakoff, A. Sullivan, Computational thinking and tinkering: Exploration of an early childhood robotics curriculum, Comput. Educ. 72 (2014) 145–157.
[7] J. Denner, S. Bean, J. Martinez, The girl game company: Engaging latina girls in information technology, Aftersch. Matters 8 (2009) 26–35.
[8] J. Robertson, C. Howells, Computer game design: Opportunities for successful learning, Comput. Educ. 50 (2008) 559-578.
[9] J.H. Maloney, K. Peppler, Y. Kafai, M. Resnick, N. Rusk, Programming by choice: urban youth learning programming with scratch, in: Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education, ACM, 2008.
[10] J. Denner, What predicts middle school girls' interest in computing? Int. J. Gend. Sci. Technol. 3 (2011).
[11] J. Robertson, Making games in the classroom: Benefits and gender concerns, Comput. Educ. 59 (2012) 385–398.
[12] K. Qiu, L. Buechley, E. Baafi, W. Dubow, A curriculum for teaching computer science through computational textiles, in: Proceedings of the 12th International Conference on Interaction Design and Children, ACM, 2013, pp. 20–27.
[13] K.A. Searle, D.A. Fields, D.A. Lui, Y.B. Kafai, Diversifying high school students' views about computing with electronic textiles, in: Proceedings of the Tenth Annual Conference on International Computing Education Research, ACM, 2014, pp. 75–82.
[14] P. Jermann, M.-A. Nüssli, Effects of sharing text selections on gaze cross-recurrence and interaction quality in a pair programming task, in: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, ACM, 2012, pp. 1125–1134.
[15] K. Sharma, D. Caballero, H. Verma, P. Jermann, P. Dillenbourg, Looking AT versus looking THROUGH: A dual eye-tracking study in MOOC context, in: Exploring the Material Conditions of Learning: Opportunities and Challenges for CSCL, The Proceedings of the Computer Supported Collaborative Learning (CSCL) Conference, 2015, pp. 260–267.
[16] P. Jermann, M.-A. Nüssli, W. Li, Using dual eye-tracking to unveil coordination and expertise in collaborative Tetris, in: Proceedings of the 24th BCS Interaction Specialist Group Conference, British Computer Society, 2010, pp. 36–44.
[17] M.-A. Nüssli, P. Jermann, M. Sangin, P. Dillenbourg, Collaboration and abstract representations: towards predictive models based on raw speech and eye-tracking data, in: Proceedings of the 9th International Conference on Computer Supported Collaborative Learning, Vol. 1, International Society of the Learning Sciences, 2009, pp. 78–82.
[18] S. Papert, I. Harel, Constructionism, Ablex Publishing, Norwood, NJ, 1991.
[19] ACM/IEEE-CS, Joint task force on computing curricula. 2013. ACM/IEEE Computing Curricula 2013 Final Report, 2013. http://www.acm.org/education/CS2013-final-report.pdf.
[20] L. Johnson, S. Adams Becker, V. Estrada, A. Freeman, The NMC Horizon Report: 2015 Museum Edition, ERIC, 2015.
[21] L. Buechley, M. Eisenberg, J. Catchen, A. Crockett, The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2008, pp. 423–432.
[22] P. Blikstein, Gears of our childhood: constructionist toolkits, robotics, and physical computing, past and future, in: Proceedings of the 12th International Conference on Interaction Design and Children, ACM, 2013, pp. 173–182.
[23] O.S. Iversen, R.C. Smith, P. Blikstein, E.-S. Katterfeldt, J.C. Read, Digital fabrication in education: Expanding the research towards design and reflective practices, Int. J. Child-Comput. Interact. 5 (2016) 1–2.
[24] J.C. Adams, A.R. Webster, What do students learn about programming from game, music video, and storytelling projects? in: Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, ACM, 2012, pp. 643–648.
[25] E.R. Kazakoff, A. Sullivan, M.U. Bers, The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood, Early Child. Educ. J. 41 (2013) 245–255.

[26] K. Kahn, Toontalk-steps towards ideal computer-based learning environments, in: A Learning Zone of One's Own: Sharing Representations and Flow in Collaborative Learning Environments, 2004, pp. 253–270.
[27] R.M. Ryan, E.L. Deci, Intrinsic and extrinsic motivations: Classic definitions and new directions, Contemp. Educ. Psychol. 25 (2000) 54–67.
[28] P.R. Pintrich, A motivational science perspective on the role of student motivation in learning and teaching contexts, J. Educ. Psychol. 95 (2003) 667.
[29] F. Guay, C.F. Ratelle, J. Chanal, Optimal learning in optimal contexts: The role of self-determination in education, Can. Psychol. 49 (2008) 233.
[30] D. Fortus, Attending to affect, J. Res. Sci. Teach. 51 (2014) 821–835.
[31] E.-S. Katterfeldt, N. Dittert, H. Schelhowe, Designing digital fabrication learning environments for Bildung: Implications from ten years of physical computing workshops, Int. J. Child-Comput. Interact. 5 (2015) 3–10.
[32] M.N. Giannakos, L. Jaccheri, What motivates children to become creators of digital enriched artifacts? in: Proceedings of the 9th ACM Conference on Creativity & Cognition, ACM, 2013, pp. 104–113.
[33] N. Vos, H. Van Der Meijden, E. Denessen, Effects of constructing versus playing an educational game on student motivation and deep learning strategy use, Comput. Educ. 56 (2011) 127–137.
[34] R.M. Ryan, E.L. Deci, Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being, Am. Psychol. 55 (2000) 68.
[35] M.A. Just, P.A. Carpenter, Using eye fixations to study reading comprehension, in: New Methods in Reading Comprehension Research, 1984, pp. 151–182.
[36] K. Rayner, Eye movements in reading and information processing: 20 years of research, Psychol. Bull. 124 (1998) 372.
[37] H. Jarodzka, K. Scheiter, P. Gerjets, T. Van Gog, In the eyes of the beholder: How experts and novices interpret dynamic stimuli, Learn. Instr. 20 (2010) 146–154.
[38] K. Rayner, K.H. Chace, T.J. Slattery, J. Ashby, Eye movements as reflections of comprehension processes in reading, Sci. Stud. Reading 10 (2006) 241–255.
[39] M.-J. Tsai, H.-T. Hou, M.-L. Lai, W.-Y. Liu, F.-Y. Yang, Visual attention for solving multiple-choice science problem: An eye-tracking analysis, Comput. Educ. 58 (2012) 375–385.
[40] C. Aschwanden, M. Crosby, Code scanning patterns in program comprehension, in: Proceedings of the 39th Hawaii International Conference on System Sciences, 2006.
[41] R. Bednarik, M. Tukiainen, An eye-tracking methodology for characterizing program comprehension processes, in: Proceedings of the 2006 Symposium on Eye Tracking Research & Applications, ACM, 2006, pp. 125–132.
[42] R. Bednarik, Expertise-dependent visual attention strategies develop over time during debugging with multiple code representations, Int. J. Hum.-Comput. Stud. 70 (2012) 143–155.
[43] P. Romero, R. Lutz, R. Cox, B. du Boulay, Co-ordination of multiple external representations during Java program debugging, in: IEEE 2002 Symposia on Human Centric Computing Languages and Environments, 2002 Proceedings, IEEE, 2002, pp. 207–214.
[44] B. Sharif, M. Falcone, J.I. Maletic, An eye-tracking study on the role of scan time in finding source code defects, in: Proceedings of the Symposium on Eye Tracking Research and Applications, ACM, 2012, pp. 381–384.
[45] P. Hejmady, N.H. Narayanan, Visual attention patterns during program debugging with an IDE, in: Proceedings of the Symposium on Eye Tracking Research and Applications, ACM, 2012, pp. 197–200.
[46] S. Pietinen, R. Bednarik, M. Tukiainen, Shared visual attention in collaborative programming: a descriptive analysis, in: Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering, ACM, 2010, pp. 21–24.
[47] M.K. Eckstein, B. Guerra-Carrillo, A.T.M. Singley, S.A. Bunge, Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development? in: Developmental Cognitive Neuroscience, 2016.
[48] S. Papavlasopoulou, K. Sharma, M. Giannakos, L. Jaccheri, Using eye-tracking to unveil differences between kids and teens in coding activities, in: Proceedings of the 2017 Conference on Interaction Design and Children, ACM, 2017, pp. 171–181.
[49] F. Paas, A. Renkl, J. Sweller, Cognitive load theory: Instructional implications of the interaction between information structures and cognitive architecture, Instr. Sci. 32 (2004) 1–8.
[50] J. Sweller, J.J. Van Merrienboer, F.G. Paas, Cognitive architecture and instructional design, Educ. Psychol. Rev. 10 (1998) 251–296.
[51] M.A. Just, P.A. Carpenter, A theory of reading: From eye fixations to comprehension, Psychol. Rev. 87 (1980) 329.
[52] M.A. Just, P.A. Carpenter, Eye fixations and cognitive processes, Cogn. Psychol. 8 (1976) 441–480.
[53] M.A. Just, P.A. Carpenter, The role of eye-fixation research in cognitive psychology, Behav. Res. Methods 8 (1976) 139–143.
[54] M. Hegarty, M.-A. Just, Constructing mental models of machines from text and diagrams, J. Mem. Lang. 32 (1993) 717–742.
[55] R.E. Mayer, Unique contributions of eye-tracking research to the study of learning with graphics, Learn. Instr. 20 (2010) 167–171.
[56] P.A. Kirschner, J. Sweller, R.E. Clark, Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching, Educ. Psychol. 41 (2006) 75–86.
[57] R. Moreno, When worked examples don't work: Is cognitive load theory at an impasse? Learn. Instr. 16 (2006) 170–181.
[58] F. Paas, J.E. Tuovinen, J.J. Van Merrienboer, A.A. Darabi, A motivational perspective on the relation between mental effort and performance: Optimizing learner involvement in instruction, Educ. Technol. Res. Dev. 53 (2005) 25–34.
[59] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, Scratch: programming for all, Commun. ACM 52 (2009) 60–67.
[60] S. Bakker, E. van den Hoven, A.N. Antle, MoSo tangibles: evaluating embodied learning, in: Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction, ACM, 2011, pp. 85–92.
[61] R. Jacob, K.S. Karn, Eye tracking in human–computer interaction and usability research: Ready to deliver the promises, Mind 2 (2003) 4.
[62] J.H. Goldberg, M.J. Stimson, M. Lewenstein, N. Scott, A.M. Wichansky, Eye tracking in web search tasks: design implications, in: Proceedings of the 2002 Symposium on Eye Tracking Research & Applications, ACM, 2002, pp. 51–58.
[63] L. Cowen, L.J. Ball, J. Delin, An eye movement analysis of web page usability, in: People and Computers XVI — Memorable Yet Invisible, Springer, 2002, pp. 317–335.
[64] L. Hall, C. Hume, S. Tazzyman, Five degrees of happiness: Effective smiley face Likert scales for evaluating with children, in: Proceedings of the The 15th International Conference on Interaction Design and Children, ACM, 2016, pp. 311–321.
[65] B. Kuvaas, Work performance, affective commitment, and work motivation: The roles of pay administration and pay level, J. Organ. Behav. 27 (2006) 365–385.
[66] V. Venkatesh, C. Speier, M.G. Morris, User acceptance enablers in individual decision making about technology: Toward an integrated model, Decis. Sci. 33 (2002) 297–316.
[67] W.J. Conover, W.J. Conover, Practical nonparametric statistics, 1980.
[68] S.S. Shapiro, M.B. Wilk, An analysis of variance test for normality (complete samples), Biometrika 52 (1965) 591–611.
[69] J. Denner, L. Werner, S. Bean, S. Campe, The girls creating games program: Strategies for engaging middle-school girls in information technology, Frontiers 26 (2005) 90–98.
[70] Y.B. Kafai, Q. Burke, Constructionist gaming: Understanding the benefits of making games for learning, Educ. Psychol. 50 (2015) 313–334.
[71] G. Fessakis, E. Gouli, E. Mavroudi, Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study, Comput. Educ. 63 (2013) 87–97.
[72] J. Denner, L. Werner, E. Ortiz, Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? Comput. Educ. 58 (2012) 240–249.
[73] D.N. Perkins, C. Hancock, R. Hobbs, F. Martin, R. Simmons, Conditions of learning in novice programmers, J. Educ. Comput. Res. 2 (1986) 37–55.
[74] R.L. Oliver, Effect of expectation and disconfirmation on postexposure product evaluations: An alternative interpretation, J. Appl. Psychol. 62 (1977) 480.
[75] S.-H. Liu, H.-L. Liao, J.A. Pratt, Impact of media richness and flow on e-learning technology acceptance, Comput. Educ. 52 (2009) 599–607.
[76] C.S. Lin, S. Wu, R.J. Tsai, Integrating perceived playfulness into expectation-confirmation model for web portal context, Inf. Manag. 42 (2005) 683–693.
[77] H.-P. Shih, Using a cognition-motivation-control view to assess the adoption intention for web-based learning, Comput. Educ. 50 (2008) 327–337.
[78] L. Abeysekera, P. Dawson, Motivation and cognitive load in the flipped classroom: definition, rationale and a call for research, Higher Educ. Res. Dev. 34 (2015) 1–14.
[79] J. Hyönä, The use of eye movements in the study of multimedia learning, Learn. Instr. 20 (2010) 172–176.
[80] S. Nevalainen, J. Sajaniemi, Comparison of three eye tracking devices in psychology of programming research, in: 6th Annual Psychology of Programming Interest Group, 2004, pp. 170–184.
[81] P. Qvarfordt, S. Zhai, Conversing with the user based on eye-gaze patterns, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2005, pp. 221–230.

Paper 4:

Discovering children's competences in coding through the analysis of Scratch projects

Sofia Papavlasopoulou, Michail N. Giannakos, and Letizia Jaccheri

In Proc. of IEEE Global Engineering Education Conference (EDUCON)

https://doi.org/10.1109/EDUCON.2018.8363356

Paper 5:

Exploring children's learning experience in constructionism-based coding activities through design-based research

Sofia Papavlasopoulou, Michail N. Giannakos, and Letizia Jaccheri

Computers in Human Behavior

Computers in Human Behavior 99 (2019) 415–427

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Full length article

Exploring children's learning experience in constructionism-based coding activities through design-based research

Sofia Papavlasopoulou*, Michail N. Giannakos, Letizia Jaccheri

Department of Computer Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

A R T I C L E I N F O

Keywords:
Constructionism
Coding
Computational thinking
Engagement
Children
Design-based research

A B S T R A C T

Over the last few years, the integration of coding activities for children in K-12 education has flourished. In addition, novel technological tools and programming environments have offered new opportunities and increased the need to design effective learning experiences. This paper presents a design-based research (DBR) approach conducted over two years, based on constructionism-based coding experiences for children, following the four stages of DBR. Three iterations (cycles) were designed and examined in total, with participants aged 8–17 years old, using mixed methods. Over the two years, we conducted workshops in which students used a block-based programming environment (i.e., Scratch) and collaboratively created a socially meaningful artifact (i.e., a game). The study identifies nine design principles that can help us to achieve higher engagement during the coding activity. Moreover, positive attitudes and high motivation were found to result in the better management of cognitive load. Our contribution lies in the theoretical grounding of the results in constructionism and the emerging design principles. In this way, we provide both theoretical and practical evidence of the value of constructionism-based coding activities.

1. Introduction

There is growing evidence supporting the introduction of computer science (CS) and computational thinking (CT) into K-12 education (Hubwieser, Armoni, Giannakos, & Mittermeir, 2014); (Horizon, 2015). According to Wing (2006 p.33) *"CT represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use"*. CT involves problem solving, design of systems and understanding human behavior by employing central concepts of CS (Wing, 2006). Organizations like the Computer Science Teachers Association (CSTA), Informatics Europe, the Cyber Innovation Center, and the National Math and Science Initiative have developed standards applied to coding education (Hubwieser et al., 2015). Increasing interest in learning coding in pedagogical contexts has also been driven and disseminated by organizations like Code.org and Codeacademy, which argue for the need to create skills that support future career opportunities while highlighting the educational advantages that coding offers. CT and coding in education have become integral parts of the school curricula in many countries. For example, the United Kingdom has integrated computer programming as a mandatory course starting from primary school (Jones, 2013), while Denmark promotes digital literacy, focusing on the knowledge gained from building technologies (Tuhkala, Wagner, Nielsen, Iversen, & Kärkkäinen, 2018).

Pioneered by Seymour Papert (Papert, 1980), computer programming in education has received a lot of interest from educators and researchers seeking alternative ways of teaching complex problem-solving skills and providing dynamic learning experiences (Kalelioğlu, 2015; Lye & Koh, 2014). Nowadays, there are a variety of technological tools and child-friendly programming environments (Papavlasopoulou, Giannakos, & Jaccheri, 2017b). Many introductory experiences for K-12 have been designed around the use of block-based programming environments, such as Scratch, Alice, Blocky, and App Inventor (Zhang, Liu, de Pablos, & She, 2014); (Fields, Vasudevan, & Kafai, 2015) (Wagner, Gray, Corley, & Wolber, 2013). These environments do not require any special expertise but do require careful thinking to tell the computer what to do step by step. Papert's (1980) constructionism argues that through coding, children have an "object-to-think-with"; in the process, they learn about their own thinking (Guzdial, 2004). Constructionism-based learning activities have been widely studied in both formal and informal education (Papavlasopoulou, Giannakos, & Jaccheri, 2017a). Integrating coding into pedagogical contexts in an intuitive and engaging experience enhances children's logic, critical thinking, problem-solving, math, and higher-order skills and can change their attitudes towards computing (Sáez-López, Román-González, & Vázquez-Cano, 2016); (Kafai & Burke, 2015). There are strong arguments for children to learn how to code, supported by the

* Corresponding author. Sem Sælands vei 7-9, NO-7491 Trondheim, Norway.
E-mail addresses: spapav@ntnu.no (S. Papavlasopoulou), michailg@ntnu.no (M.N. Giannakos), letizia.jaccheri@ntnu.no (L. Jaccheri).

https://doi.org/10.1016/j.chb.2019.01.008
Received 2 May 2018; Received in revised form 26 December 2018; Accepted 13 January 2019
Available online 16 January 2019
0747-5632/

constructionist approach (Kafai & Burke, 2015) (Gallup, 2015). Children need to acquire 21st-century skills, empowering themselves with the required competences related to the digitalization of our society. Learning how to code has become equally valuable as learning math, reading, and writing (Horizon, 2015).

Several studies have focused on introducing computational literacy to children in different ways (Papavlasopoulou et al., 2017a). Various programmable and interactive objects exist showing the importance of involving children from a young age in learning coding (Fessakis, Gouli, & Mavroudi, 2013). In addition, environments like LiliPad Arduino (Buechley, Eisenberg, Catchen, & Crockett, 2008) have been developed to attract more girls to CS and CT. The combination of physical fabrication and coding has proven valuable for increasing engagement in programming concepts and practices (Kafai & Vasudevan, 2015), especially when it incorporates social and creative dimensions of learning (Giannakos & Jaccheri, 2018). In a study with sixth-grade students in Scotland, Robertson and Howells (Robertson & Howells, 2008) found that making a game is an authentic learning activity offering motivation, enthusiasm, and engagement with learning. Therefore, to overcome the various barriers with learning coding (e.g., difficulty, boredom, confusion, etc.), we need appropriately designed and engaging coding activities for children.

Constructionism theorizes that learner is seen as an active constructor of knowledge rather than being a passive recipient of information (Papert, 1993), with making and coding being the areas that constructionism theory has been widely applied (Kafai & Burke, 2015). Almost three decades after Papert's original ideas on constructionism, the idea remains relevant and has become ubiquitous in how learning theorists and educators aim to empirically ground and revamp constructionism-based teaching (Kao & Harrell, 2017). Such grounding would result in methodological advancements and a comprehensive understanding of children's experience in constructionism-based making activities. In this paper, we present a design-based research (DBR) effort comprising three cycles (iterations) conducted over two years. DBR combines empirical educational research with theory-driven design in learning contexts to understand how, when, and why educational innovations work in real settings (Collins, 1992). The main characteristic of DBR is the systematic and iterative cycle of design, exploration, and redesign (Collective, 2003). Many studies have used DBR in educational contexts (Grover, Pea, & Cooper, 2015; Parmaxi & Zaphiris, 2015) (Sáez-López et al., 2016); (Parmaxi, Zaphiris, & Ioannou, 2016) (Schmitz, Klemke, Walhout, & Specht, 2015), emphasizing the need for well-designed studies characterized by objectivity, reliability, and validity and providing critical evidence to establish outcomes beneficial for others.

This research aims to contribute to the theoretical notions of constructionism with regard to the effects of coding activities on children's learning experience. We designed and evaluated coding workshops for children (aged 8–17 years old). Both qualitative and quantitative methods were employed to evaluate our workshops, including interviews, surveys, observations, and physiological data (eye tracking). The coding activities were designed to impact children's learning outcomes, cognition, and social and emotional development. Thus, the overreaching goal of the study was framed with the following research questions:

- What elements of engagement exist in constructionism-based coding activities?
- What principles can guide us to facilitate constructionism-based learning environments that support children's learning experience?

The rest of the paper is structured as follows: the next section provides an overview of related work on the theoretical framework of constructionism and previous research on similar coding activities. The third section describes the methodology used, the designed coding activities in the three cycles, and the data collection and analysis. The fourth section presents the results based on the theory of constructionism and the main design principles that guided each of the iterations. In the fifth section, we discuss and highlight the design implications, derived from this intervention research. We conclude with the limitations of our study and avenues for future work.

2. Related work

2.1. Theoretical framework: constructionism

Our theoretical grounding is constructionism, which was developed by Papert (Papert, 1997), (Papert, 1980). Constructionism assumes that knowledge is better gained when children are deeply and actively involved in building their own meaningful constructions. Based on Piaget's (1954) theory, which focuses on how mental constructions are formed in someone's mind, Papert (Papert, 1980) focuses on explaining how construction is a valuable way to create mental constructions. The learner discovers their own knowledge, rather than being a passive receiver. Papert's constructionism sees the effectiveness of learning as achieved through making, where learners experience the active construction of visible-to-the-world artifacts. Computational culture supports the creation of building those artifacts by using digital media and computer-based technologies (Kafai & Resnick, 2012). The vital aspect of constructionism is the requirement of "objects-to-think-with" – *"objects in which there is an intersection of cultural presence, embedded knowledge and the possibility for personal identification"* (Papert, 1980), p. 11). The role of this object in Papert's *Mindstorms* is the "turtle", a digital animal within the Logo programming environment that can be controlled and moved by giving the appropriate commands. The "turtle" acts as a means to think, supporting and promoting a new way of thinking and learning. In Papert's (1980, p. 76) words: "*the child's encounter with this theorem is different in several ways from memorizing its Euclidean counterpart 'the sum of the internal angles of a triangle is 180 degrees.' First (at least in the context of Logo computers), the Total Turtle Trip Theorem is more powerful: The child can actually use it. Second, it is more general: It applies to squares and curves as well as to triangles. Third, it is more intelligible: Its proof is easy to grasp. And it is more personal: You can 'walk it through,' and it is a model for the general habit of relating mathematics to personal knowledge.*"

Constructionism is not only valuable for the individual in building knowledge through experience and engagement in creating artifacts but also for enhancing the social setting (Kafai, 2006). Like in the well-known samba school example, a social setting strengthens the sense of belonging to a group with a common purpose, where learning becomes important for all and connections are made under the learning culture (Papert, 1980). In the same line (Kafai & Burke, 2015), mention three dimensions of constructionism involved in the process of making games for learning: personal, social, and cultural. More specifically, "personal" refers to the learning and the attitudes related to learning, "social" refers to the collaborative aspects in creating a shared artifact, and "cultural" relates to how gender and race could influence the activity and the possible cultural aspects that could influence participation.

In the process of making computer games, children plan and manage this complex development, placing themselves in control of their own leaning and thinking (Kafai & Kafai, 1995). Robertson and Howells (Robertson & Howells, 2008) argue that game design is a powerful learning activity that provides motivation, engagement, and enthusiasm. Constructionism's basic idea is that the most effective leaning experiences are those that include active creation, socially meaningful artifacts, interaction with others, and the use of elements that support one's own learning and thinking. Game-making activities not only involve learning how to use technological tools but also using these tools to discover new ways of thinking. In such activities, children are introduced to a culture that permits them to become producers of their own artifacts while building their knowledge in a social context.

2.2. Qualities of constructionism-based coding activities for children

Computer game design and development have been increasingly introduced in both formal and informal educational settings, supporting everything from programming courses and STEM educational topics to broader contexts of problem solving and arts (Papavlasopoulou et al., 2017a). The various technological tools available nowadays allow us to support learning activities based on constructionism and provide meaningful learning experiences for children. In these types of educational activities, children are the protagonists, as they have control of their own products. Coding activities for children not only relate to CS but also allow the development of computational competences and higher-order thinking skills (Grover & Pea, 2013). Children who actively participate in game-making activities enhance, among others, their problem-solving, critical thinking, CT, and collaborative skills (Papavlasopoulou et al., 2017a); (Grover & Pea, 2013).

The benefits of educational activities in which children use technological tools and digital fabrication to construct their own games are many and vary from learning programming concepts to behavioral and perceptual changes towards career paths in computing (Sáez-López et al., 2016) (Kafai & Vasudevan, 2015); (Denner, Werner, & Ortiz, 2012). Making games can be more beneficial than other project-based activities, supporting learning about storytelling, artwork, sound, mechanics, and math (Sung & Hwang, 2013). Moreover, children are familiar with video games from an early age (Granic, Lobel, & Engels, 2014). Visual programming environments provide opportunities for children to be introduced to programming concepts; owing to the fun and usefulness of the activity, children are highly motivated and have positive attitudes towards coding (Sáez-López et al., 2016). Block-based visual programming languages (like Scratch) have the advantage of using shapes that fit properly only when they make a logical sequence of orders. This gives relief to users and saves them from much of the heartache traditionally forced on learners by textual languages (Wilson & Moffat, 2010), p. 70). However, even advanced text-based programming languages like Java have been used to engage children aged 9–10 in coding (Esper, Foster, Griswold, Herrera, & Snyder, 2014). A combination of physical fabrication and coding can engage and enhance children's competences in programming concepts (e.g., loops, conditionals, and events) and practices (e.g., remixing, testing, and debugging) (Kafai & Burke, 2015); (Denner et al., 2012). In addition, digital game development was found to be beneficial for special education students, increasing their problem-solving skills through a process of representation, planning, execution, and evaluation of an artifact (Ruggiero & Green, 2017). Hence, further empirical studies are needed to investigate the different aspects and advantages of constructionism-based activities.

Gender discrepancy in coding has been related to negative educational experiences in early childhood (Teague, 2002). CS careers still tend to be highly stereotyped, with girls being less likely to choose this career path. However, studies have found that both girls and boys who get involved in different kinds of software development practices show a better understanding of and positive attitudes towards CS (Bonner & Dorneich, 2016); (Eordanidis, Gee, & Carmichael, 2017); (Robertson, 2013); (Papavlasopoulou, Sharma, Giannakos, & Jaccheri, 2017). Scaffolding examples can help girls' engagement and confidence when using a programming environment. Studies specifically focusing on girls have found that game design experiences intended to enhance computational skills affect their perceptions in seeing themselves as able to design computer games and encourage them to pursue careers in CS-related professions (Stewart-Gardiner, Carmichael, Latham, Lozano, & Greene, 2013). In a study involving middle-school girls creating games (Denner et al., 2012), found that they were engaged in the process and demonstrated adequate levels of complex programming activity. Thus, designing appropriate activities can be a promising approach to attracting and encouraging girls' interest in computing.

Generally, the skills gained in these educational contexts go beyond the use of a technological tool for making a game and CT. For instance, when children negotiate artifact construction in a supportive environment, they gain a sense of self-efficacy and belief in their capacities; they learn how to solve a problem, manage difficulties, cope with "failure", share resources, and communicate with peers (Chu, Schlegel, Quek, Christy, & Chen, 2017); (Çakır, Gass, Foster, & Lee, 2017); (Bers, 2012). These practices exist in constructionist learning and can be applied in subjects like math, language, arts, and others. The value is in the transferable skills uncovered through the experience of completing a successful project.

In a nutshell, constructionism-based coding activities, particularly when the focus is on game-making, provide a fruitful learning environment in which children are stimulated to use a technological tool, affecting their learning experience. Therefore, there is a need to investigate and get a deep understanding of how we can help learners to acquire knowledge, skills, and competences in coding in an engaging and meaningful manner.

3. Methodology

3.1. Design based research (DBR)

DBR is a systematic but agile methodology widely used in educational contexts (Fig. 1) (Anderson & Shattuck, 2012) (Wang & Hannafin, 2005); (Reeves, 2006). DBR offers a strategy to understand learning processes through design, exploration, enactment, evaluation, and redesign (Anderson, 2005). DBR is a hybrid method, as it is not a replacement of other methodologies but builds on the use of multiple procedures and methods from both design and research methodologies (Wang & Hannafin, 2005). The purpose of DBR is to influence real educative interventions and validate theoretical concepts. The difference between DBR and formative assessment is that it also has a theoretical goal (Barab & Squire, 2004). Researchers are actively involved and maintain constant collaboration with participants, other researchers, and practitioners to manage the research process in real-world settings. Their aim is to implement interventions with refined and improved designs that influence practice. In short, there are five basic characteristics of DBR: 1) it refines theory and practice, 2) it happens in real-world settings and is grounded in relevant contexts, 3) it is interactive, iterative, and flexible, 4) it uses mixed methods in accordance with potential new needs and emerging issues, and 5) it is contextual, meaning that the research findings are connected with the design process (Wang & Hannafin, 2005).

In our approach, based on all the above, we used constructionism theory and applied the DBR methodology to guide our iterations. More specifically, our research process used DBR methodology as it deals with the complexity of real-world educative contexts (in our case coding workshops) and it is grounded in theory (in our case constructionism theory). In addition, DBR approach is in line with the needs of our study, allowing a long period of time with continuous design, evaluation and redesign of our interventions. In this way, we had also the opportunity to conduct iterative and flexible revisions of the research design applying research methods from both qualitative and quantitative research. DBR methodology needs a detailed and comprehensive documentation of the whole process; this action helped

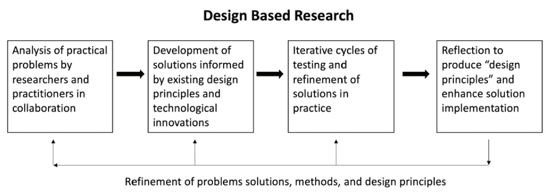

Fig. 1. The research cycle of DBR (Reeves, 2006).

the analysis of our data and especially the retrospective analysis, both to contribute to theory and practice. For all the four stages of the DBR (Fig. 1), constant collaboration with other researchers, experts in the field and instructors is required; this was essential aspect of our study in order to be able to improve the impact of the interventions, understand the learning experience processes, advance the initial designs and provide theoretical and practical impact extracting design principles.

We conducted three cycles (iterations) over two years, evaluating and refining our coding workshops with children. We applied theoretically and pedagogically aligned tasks to investigate their effectiveness on children's learning engagement, overall learning experience, and collaboration while developing an artifact (a game in our approach).

The main aspects of this study were: 1) the design of the coding workshops to facilitate children's use of the programming tool and to introduce them to coding, 2) the researchers working in close collaboration with the participants and assistants who ran the workshops, 3) the use of different methods to evaluate the effectiveness of our approach to increase the sustainability and scalability of this program, 4) grounding our findings in theory, and 5) identifying general design principles for future similar activities.

3.2. Description of the workshops

The participants' goal was to create an artifact, which in our case was a game using the Scratch programming tool. Students worked in teams for the development of the artifact. Teaching assistants, specifically trained, led the process and assisted students in achieving their goals.

Regarding the process of construction in the workshops, the most influential to our pedagogical approach was what Resnick calls the “kindergarten approach to learning”, with a spiral cycle of imagine, create, play, share, and reflect – a process that is repeated over and over (Resnick, 2007). Children imagine what they want to do and then create a project with their ideas, play/interact with their own creations, share their creations with others, and reflect on their experiences, leading to new ideas and projects. Adapting Resnick's spiral, ours also started with “inspire” to characterize the warming-up and inspiring activities that kicked off the children's creativity. In addition, to characterize the coding process and the use of the Scratch tool specifically, we focused on constantly experimenting and iterating: the children developed their artifacts gradually by trying new elements, using different concepts, and revising them (Fig. 2).

3.2.1. Cycles 1 and 2

We designed and implemented a coding activity in conjunction with an initiative organized at the Norwegian University of Science ans Technology (NTNU), in Trondheim, Norway, named Kodeløypa

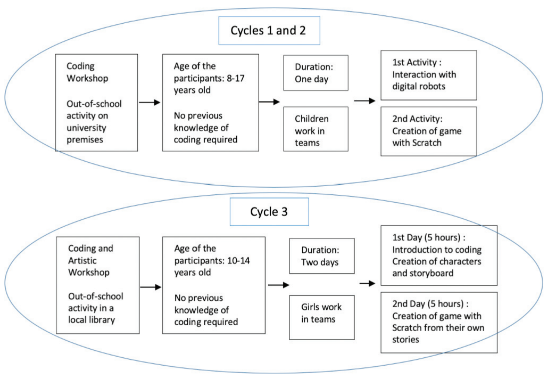

Fig. 2. Description of the three DBR cycles.

(meaning “the path towards coding”). The workshop activities were based on the constructionist approach, as one of the main principles of this is learning by making. The workshop was conducted in a largely informal setting, as an out-of-school activity, and lasted for four hours in total. Various student groups, in the range 8–17 years old, were invited to NTNU's specially designed rooms for creative purposes to interact with digital robots and to create games using Scratch and the Arduino hardware platform. Specifically, Arduino was attached to the digital robots to connect them with the computer. At that point, an extension of Scratch called Scratch for Arduino (S4A) provided the extra blocks needed to control the robots. The Scratch programming language uses colorful blocks grouped into categories (motion, looks, sound, pen, control, sensing, operators, and variables), with which children can develop stories, games, and any type of animation. In general, the children who attended the workshop worked collaboratively in triads or dyads (depending on the number of children). The workshop was designed for children without (or with minimum) previous experience in coding. The design of the activity (interacting with robots and creating games), and the use of Scratch programming language (suitable for all ages) provided flexibility and allowed the successful implementation of the workshop with participants from 8 to 17 years old students. Each of the workshops, had a specific age group of students, carefully selected, being within a small age range. During the workshop, student assistants were responsible for supporting each team as needed. Approximately one assistant observed and helped one or two teams. Three researchers were also present throughout the intervention, focusing on observing, writing notes, and taking care of the overall execution of the workshop. The workshop had two main sections (Fig. 3).

Interacting with the robots: In the first section, the children interacted with digital robots made by an artist (using recycling materials). The different robots were placed next to the computers (one for each team). When the children entered the room, one assistant welcomed them, told them to be seated, and briefly presented an overview of the workshop. The assistants then advised the children to pay attention to the paper tutorial and the worksheets placed on the desks (one for each student). First, the children filled in the worksheet to answer questions regarding the exact places and numbers of sensors and lights on the robots. The tutorial contained instructions with examples and pictures, similar to the robots they were using. The examples had little text and more images and described exactly how the children could interact with the robots. The children accomplished a series of simple loops that controlled the robots and made them react to the environment with visual effects (such as turning on a light when sensors detected that the light was below a certain threshold). Children could touch and play with the robots but not change any parts of them. Although the duration of the session was different for each team, it lasted between 45 min and 1.5 h and ended with a break before the next session.

Creating games using Scratch: This session focused on the creative implementation of simple game development concepts using Scratch. All children took another paper-based tutorial containing examples and visualizations to help them to ideate their own games. The tutorial comprised simple text explanations and included basic CT concepts and possible loops that the children were supposed to use in their own games. First, the assistants advised the children to concentrate on understanding the idea of the game, to discuss it with their team members, and to create a draft storyboard. The children then developed their own games by collaboratively designing and coding using Scratch. To accelerate the children's progress, they were given existing game characters and easy loops. While the children worked on their projects, help was provided whenever they asked for it, and complex programming concepts were introduced on an individual level according to the relevance to their project. Children created their games step by step by iteratively testing and coding them. After completing the games, all teams reflected on and played each other's games. This section lasted approximately three hours.

Fig. 3. Example of an interactive robot (left), children collaborating on game creation (middle), and example of a created game (right).

3.2.2. Cycle 3

We designed and implemented a two-day workshop in conjunction with the local library of Trondheim, Norway. The workshop activities focused on coding including artistic elements and were based on the constructionist approach. The call for participation was made to middle-school girls of the Trondheim region during the autumn 2017 school break. Previous experience was not a prerequisite for the girls' participation. The activities of each day were conducted in an informal setting and lasted for approximately five hours, including breaks. Female instructors, with previous experience in similar activities (also involved in Kodeløypa), facilitated the workshop and were responsible for supporting the girls during the process. During the workshop, the girls had to create storyboards based on solving particular environmental problems and then, based on their stories, create games using the Scratch programming language (Fig. 4). For the development of the storyboards, the girls could use different types of materials, like ribbons, colored cardboard, stickers, drawing pencils, etc., as provided by the library. The girls worked collaboratively in teams of two or three (depending on the number of participants). Two researchers were present for the whole duration of the workshop, assisting when needed for the smooth execution of the activities, including the data collection. The workshop is described below, based on the two days of activities.

First day of the workshop: On the first day, we introduced the basic skills of coding and other non-technical aspects of game development, like storyboard creation. The workshop started with a story from a book, based on a woman with children and everyday problems, who was also a mentor and a superhero helping people to succeed with their technology projects. The girls were inspired and were informed that they had to think of their own characters who needed to save the world from environmental issues of their choosing. Then, in order to give an introduction to coding, the instructors presented an example of a functional game with Scratch on a relevant environmental topic. Then, the girls were asked to individually complete basic coding exercises using Scratch. At the end of the first day, the teams prepared and presented their storyboards with three different scenes on paper/cardboard, including the title, theme, character, plot, conflict, and solution.

Second day of the workshop: Starting the second day of the workshop, the girls had to update, if they wanted, their storyboards and finalize them. Then, the rest of the day was dedicated to their game creation using Scratch. The girls completed a paper-based tutorial, created by the instructors, with simple text explanations and examples of basic CT concepts and possible loops that the girls were supposed to use in their own games, all based on Scratch. During the creation of their games, the girls had to use their storyboards exactly and "transfer" their ideas into games using Scratch. At any time, the girls could ask for help from the instructors, who even introduced complicated programming concepts, if it was necessary for their games. The girls created their games step by step and continuously testing and coding them. At the end of the day, all teams prepared presentations of their games and everyone played each other's games.

Fig. 4. Girls participating in the workshop (left), creation of the storyboard (middle), and game created using Scratch (right).

3.3. Sampling

All the participants of the three cycles were students from Trondheim region. The first two cycles took place at NTNU in specially designed rooms, and the last cycle took place in the local library. The data related to the three cycles were collected after receiving permission from the Norwegian Centre for Research Data (NSD), following all the regulations and recommendations for research with children. When the participants had been selected, a researcher contacted their teachers and parents in order to obtain the necessary consent from both the child and the legal guardian for the data collection. Their participation in the research project was voluntary and they could drop out at any time, with no consequences on their participation in the workshop.

3.3.1. Participants of cycle 1

Children from 3rd to 12th grade (aged 8–17 years old) participated in the coding activity. The activity took place during autumn 2016 with a sample of 12 girls (mean age: 12.64, SD: 2.838) and 32 boys (mean age: 12.35, SD: 2.773). Five workshops took place over two weeks, following the coding activity described in the previous section.

3.3.2. Participants of cycle 2

In autumn 2017, children from 8th to 10th grade (aged 13–16 years old) participated in the coding activity. The sample consisted of 105 participants in total, 69 boys and 36 girls (mean age: 14.55, SD: 0.650). Kodeløypa workshops were conducted every Friday for six weeks.

3.3.3. Participants of cycle 3

The sample of the third study consisted of eight girls from 6th to 10th grade (aged 10–14 years old) (mean age: 12.135, SD: 1.389). Girls participated in the two-day workshop during autumn 2017, following all the activities of the workshop described in the previous section at the local library.

3.4. Data collection

Following the DBR methodology described by (Reeves, 2006), our study involved four stages (Table 1). In stage 1, we conducted a critical literature review to identify theoretical and practical problems in constructionism-based coding activities. Then, in the second stage, after discussions with instructors and with experts in human–computer interaction (HCI) and technology-enhanced learning (TEL), we developed the design of the intervention based on constructionism. Stage 3 involved the testing and refinement of the iterative cycles in practice. Qualitative and quantitative data were collected during the three cycles using various instruments, including pre and post knowledge acquisition tests, attitudinal questionnaires, eye-tracking data, semi-structured interviews, field notes from observations, instructors' reflections, and the artifacts constructed by the children in different phases of the process. All data focused on exploring the children's learning experience

Table 1
Description of the different DBR stages.

Stage	Data collection method	Participants	Purpose
Analysis	Literature review	Researchers HCI experts TEL experts Instructors	Analyze and identify problems and gaps in constructionism-based coding activities
Development	Literature review Discussions Focus groups	Researchers HCI experts TEL experts Instructors	Identify the theoretical framework Design the interventions
Iterative cycles of testing and refinement in practice	**Iteration 1** Eye tracking Attitudinal questionnaire Knowledge acquisition test (pre and post) Artifact collection Instructors' reflections **Iterations 2 and 3** Semi-structured interviews Field notes from observations Artifact collection Instructors' reflections	**Iteration 1** 44 children aged 8–17 years old Instructors **Iteration 2** 105 children aged 13–16 years old Instructors **Iteration 3** 8 girls aged 10–14 years old Instructors	Get a comprehensive view of students' learning experience Design elements for the next iteration
Development of design principles	Focus groups Discussions Reflections and notes from all cycles	Researchers HCI experts TEL experts Instructors	Identify the prominent design principles

in our coding workshops and guided us to the improvement of the design of the next iteration. The fourth stage of DBR is the development of design principles that intend to provide feasible solutions with respect to the theoretical goals. This final stage contains all the reflections from the previous stages, including notes of the design issues that emerged from the analysis of the results at each iteration.

3.5. Data analysis

In the DBR methodology, all stages, from the analysis to the development of design principles, include interactive and iterative formative evaluations. From the beginning of the cycles' implementation, starting with the design, to the execution and evaluation of each workshop, the researchers and instructors were in constant collaboration. Their involvement throughout the project allowed them to gain valuable knowledge and competence in the analysis and interpretation of the data gathered in each cycle. All data collected from the three cycles were respectively analyzed according to their type. For example, quantitative data were analyzed using one-way analysis of variance (ANOVA) and Pearson correlation coefficient among other; while qualitative data were analyzed based on Saldaña (2015). All data were compared and cross-checked for triangulation. For this paper, the qualitative analysis was manually conducted by the researchers using both inductive and deductive approaches, based on (Saldaña, 2015) (Mayring, 2014).

During the two years of the project, after the end of each iteration (cycle), the researchers and instructors participated in focus groups discussing and revealing all the growing ideas emerged from the outcomes of the iteration. All ideas were connected to the results of the respective iteration, representing the codes for our qualitative analysis for this study. In order to synthesize the ideas and formulate themes, we focused mainly on the students' engagement in the coding activities. The students' engagement included interaction with the instructor and the learning tool and interaction with other students in the creation of an artifact. In our approach, we adopt the term "academic engagement" (Turner, Christensen, Kackar-Cam, Trucano, & Fulmer, 2014) to describe how the students were involved in and put effort into learning, understanding, and collaborating with their peers. Engagement during educational activities has many aspects and is connected with other theoretical constructs, like motivation and self-regulation (Henrie, Halverson, & Graham, 2015). According to (Fredricks, Blumenfeld, & Paris, 2004), there are three types of engagement: behavioral, emotional, and cognitive, which are interrelated within the individual. "Behavioral engagement" refers to participation, involvement, and attention, among others. "Emotional engagement" refers to the learner's feelings, like frustration or interest, expressions of positive effects, and social connection. "Cognitive engagement" refers to the learner's investment in understanding what they have been taught, their efforts related to the mind, their strategy use, and their self-regulatory and meta-cognitive behaviors.

Each idea was connected with one of the three types of engagement, depending on its content. For example, ideas representing children's cognitive processes, like the use of different gaze patterns during the coding activity, were placed under cognitive engagement. Respectively, we followed the same procedure to place, if possible, all ideas under the appropriate type of engagement, which also allowed us to see possible interconnections. Consequently, the most prominent themes emerged. It was an iterative process, with constant refinement and reflection on the ideas and themes during the three cycles. This helped us not only to see the connections and make decisions for the design but also to identify the most important theoretical aspects in our studies. The final step of the analysis, after removing similar themes, involved categorization to identify the most important findings. The categories were interpreted according to Papert's (1980) theoretical framework, with the agreement of the instructors and the HCI and TEL experts (Fig. 5).

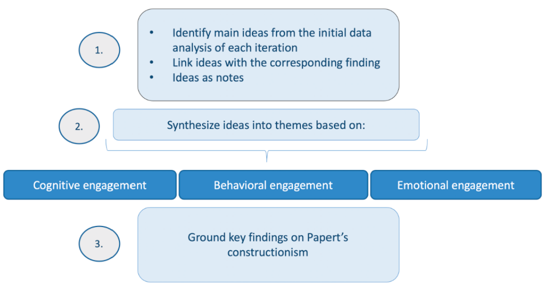

Fig. 5. Data analysis process.

In the next section, we present the findings for the first cycle, showing the important contributions based on the theoretical framework of Papert's constructionism. Then, for cycles 2 and 3, we first present the key findings emerging from the respective previous cycle related to the design of the activities and then the important contributions based on the theoretical framework.

4. Iterative design cycles, theoretical findings, and design elements

For each of the three cycles, we present the most prominent results as linked to Papert's constructionism. Therefore, there is no detailed representation of the results, as they were respectively analyzed according to their type during the process. However, when needed, there is a reference to the findings related to the data collection method in order to help the proper explanation of the specific outcome.

4.1. Cycle 1

Two theoretical ideas emerged from this cycle:

1) Learning to learn (different coding approaches result in different learning gain): According to Papert (Papert, 1980), in a constructionist learning environment, the child is able to construct their own knowledge and build on what they already know. In our workshop, the students produced socially meaningful and engaging artifacts: games. The findings from this study (cycle 1) showed that depending on their age, the students used different gaze patterns in the coding process, had different approaches to coding, and had different learning gain from the activity.

The younger students (kids) focused on the appearance of their games' characters, while the older ones (teens) had more-structured coding behavior. This was evident in the proportion of time that the kids and teens spent in specific areas of interest (based on eye tracking) in the Scratch programming environment and the transitions between them. The teens presented more "hypothesis-testing" behavior during their efforts in making the games and could shift their attention to the more-"meaningful" parts of the Scratch screen. By "the meaningful parts of the screen", we mean specific areas of interest in the Scratch interface that indicate the main areas of attention in coding: scripts, output, and commands. In addition, the teens were able to collaborate better than the kids were (had higher similarity gaze). The teens had a higher level of shared understanding and could communicate better during the coding activity. This confirms the teens' attitude towards helping each other more, contrasting with the kids, who wanted to have greater individual control. Eventually, "*by deliberately learning to imitate mechanical thinking, the learner becomes able to articulate what mechanical thinking is and what it is not. The exercise can lead to greater confidence about the ability to choose a cognitive style that suits the problem"* and *"what is most important in this is that through these experiences these children would be serving their apprenticeships as epistemologists, that is to say learning to think articulately about thinking"* (Papert, 1980). Children's coding processes represent their way of "thinking mechanically" and adopting the educational advantage of this way of deliberately thinking. Using a simple description of the process, trying to create/make a game is a way to combine appropriate orders and create programs to tell the computer what to do, step by step. This process includes logic, math, problem-solving, and critical thinking skills. In order for children to achieve their goals in such environments, they should find the appropriate cognitive style that will support them in the coding process of creating a shared artifact. This shows the importance of having appropriate tools and instructions for each age group. Different age groups differently organize their thinking and consequently their coding, so the way they approach the process of creating an artifact can be instrumental to their learning and the successful completion of the artifact. This notion is in accordance with Papert, as he presents a resemblance with juggling: *"It always takes time to learn necessary component skills. What can be eliminated are wasteful and inefficient methods. Learning enough juggling skill to keep three balls going takes many hours when the learner follows a poor learning strategy. When a good one is adopted the time is greatly reduced, often to as little as twenty or thirty minutes"* (Papert, 1980). Finding the appropriate methods to help children of different age groups will result in efficient and effective learning processes.

2) Cognitive effort and affective engagement: Positive attitudes and motivation are important to cognitive learning. There is a relation between children's attitudes and their cognitive processes while coding. Highly motivated children with positive attitudes have the ability to handle cognitive load and better manage the construction of their artifacts. This idea appeared in our findings from the measures used to examine cognition through the eye-tracking data and the relation with attitudes of perceived learning (seen as confidence, the degree that children indicate their performance), intention to perform coding again, and excitement. The children who were highly engaged and motivated during the construction of the artifact exhibited gaze behavior that showed lower cognitive overload. Papert (1980) describes the notion of "bricolage", which represents a qualitative way of organizing and planning when problem solving by constantly experimenting until finalizing the artifact. Effort and difficulty are prominent during the whole coding process and require motivational goals and determination from a child to commit themselves to the learning. This is an expected notion, as *"You can't learn bread-and-butter (basic) skills if you come to them with fear and the anticipation of hating them"* (Papert, 1980). The design of the coding activity of our workshop had an overall cognitive load that could become overwhelming for children, especially those who are novices to coding. From the complexity of the task, children might reach a point of feeling overloaded, which can lead to a critical condition where, without the proper pleasant and motivating environment, the learning experience can fail. It is not a surprising result that the children with more difficulties and cognitive load had lower scores in their attitudes.

4.2. Cycle 2

The key findings, as design elements, that emerged from cycle 1 and guided the refinement of the design of cycle 2 are described below.

Structured assistance, pleasant environment, and revised learning materials to:

a Guide students to focus on structured coding behavior.

Students should put a lot of effort and thinking into learning the necessary component skills, and they should be cognitively supported during the coding activity. As shown in the results of the eye-tracking data, those who shifted their attention to the meaningful parts of the screen (such as commands and output) had better learning gain, based on their knowledge acquisition tests. Therefore, the design of the activity should support students efficiently to ensure that they can take appropriate actions and know where to pay attention when they code to have an effective approach that is suitable for the task.

b Avoid cognitive overload.

Students can become easily overwhelmed in the process of creating an artifact, especially when they are new to coding. By using the "bricolage" style, in which they are constantly experimenting, students can feel overloaded as they seek to find the appropriate commands in the tool, manage different tasks, and make decisions during the activity. Consequently, supporting students when needed and providing relevant learning materials can reduce their cognitive load and provide a scaffold for managing their learning and thinking.

c Keep the participants motivated.

Students' positive attitudes are related to their cognitive load, as represented by their eye movements, based on the results from cycle 1. Highly motivated students with positive attitudes have better management of cognitive load. Hence, providing a pleasant environment that enhances students' enthusiasm for and engagement with learning will help students to have a fruitful experience.

d Enhance collaboration within the teams.

As students collaborate in teams to create a shared artifact, social interaction in learning during the coding activity is not something we can overlook, as it also unfolds team dynamics. Teams with better collaboration (higher gaze similarity) had higher team average learning gain, as calculated by the knowledge acquisition tests. It is important to encourage collaboration and good communication among team members so that they can benefit from each other's help.

In this cycle, the duration of the workshop for all groups of students was the same, as an out-of-school one-day activity. The results were based on the qualitative analysis of the interviews, observations, and evaluations of the students' artifacts. The children were able to express exactly their struggles and ways of thinking during the artifact creation, allowing us to detect the exact behavior of the children as they expressed it to reflect their cognitive processes (noticing debugging behavior and specific difficulties). Triangulation of the data helped us to validate our findings. The implementation of the Kodeløypa coding workshop took place over two years, with few differences in the design of the activity but with differences in the research design, evidence descriptions, and results of the different instruments used for data collection.

Two theoretical ideas emerged from this cycle:

1) Social aspect of creating an artifact: The "social" dimension refers to the role of collaboration in the coding activity. Children worked in teams of three (or two, depending on the total number of participants) to create a shared artifact. Collaboration and social interaction for a common goal have many benefits, including interacting with others, examining different perspectives, expressing understandings, and interpreting things differently. During the coding activity, the children were encouraged to work collaboratively to create a shared artifact that was meaningful for their peers too. The process also offered the opportunity to all participants to play each other's games and reflect on them. Collaboration was primarily examined between the members of the groups but also among the different teams. From the observations and interviews, the help they got from other team members was important. Half of the children expressed the highest level of satisfaction with the collaborative process in their team, while 72% showed high levels regarding being able to develop skills from the other members of the team. This interaction, which shows collaboration and help among the teams, had various aspects, from practical (what command they should use in Scratch to accomplish a task) to ideas for their games. This finding was confirmed from the artifact analysis: teams who were sitting close to one another had similarities in the programming concepts they used, as well as in their main game ideas (such as a maze or jumping on platforms). In addition, through the different versions of the artifacts, we observed that elements changed based on other teams' suggestions.

"When we didn't find anything, we looked at another group and saw how they did it"

For the team members, the construction of the artifact was not an individual task: it was a social interaction with a shared goal to create a game. The results showed that, most of the time, collaboration was efficient. The children acknowledged and expressed how valuable it was that they were working together to complete their artifacts.

"If I had my own project, I would probably not find anything"

"It is easier to code with someone than to code by yourself; if I had been alone, I wouldn't have managed to do the same"

"We both came up with ideas and equally contributed to the design and coding parts"

"I coded more, while they contributed with ideas on what should be incorporated. We were all important members of the team"

An important aspect of the good collaboration was the fact that the team members knew each other from before and/or had done other projects together.

"We knew each other, and we felt pretty safe around each other. We could discuss and agree easily on what had to be done"

Nevertheless, there were some indications of bad collaboration that caused frustration. This was mainly caused from having a "bad leader" in the group who wanted control. This was expressed from both sides.

"It was maybe that I took too much control. I should have let my partner decide a bit more"

"He didn't let me finish my task; he just wanted to have the control back"

Papert's (Papert, 1980) notion of the importance of social norms and interaction in learning is reflected in his research on samba schools: *"These are not schools as we know them; they are social clubs with memberships that may range from a few hundred to many thousands".* The construction of games and other artifacts is not an isolated action but happens in a social context.

This resonates with Papert's (Papert, 1980) notion of social interaction: *"Although the work at the computer is usually private it increases the children's desire for interaction. These children want to get together with others engaged in similar activities because they have a lot to talk about. And what they have to say to one another is not limited to talking about their products: Logo is designed to make it easy to tell about the process of making them".*

2) Powerful thinking (or learning about thinking): Papert (Papert, 1980) argues that children are able to recognize the different procedures in code, understand when the code does not run as expected, use debugging strategies, and act intentionally to improve the code. For the construction of their artifacts during the coding activity, the children worked with programming concepts and practices to successfully complete their task. Making a game requires deep engagement and strategy use to successfully manage the completion of the task. The children iteratively organized and documented their code. As described by Papert (1980, p. 28) regarding the Logo environment: *"teaching the Turtle to act or to 'think' can lead one to reflect on one's own actions and thinking. And as children move on, they program the computer to make more complex decisions and find themselves engaged in reflecting on more complex aspects of their own thinking."*

For the construction of the artifacts, the children had the opportunity to plan, problem solve, code, debug, collaborate, communicate, and reflect on their coding experience using Scratch. The participants realized that this was an iterative process, and for some it appeared to be difficult and challenging. Some found it fun to try out the different blocks, discovering the different functionalities. Whatever they made seemed to be suitable for their code; at the same time, the need to add a new function changed everything and triggered a new thinking and debugging process.

"The hardest thing was to put the different pieces of code together and make them work as one game"

"It was very challenging when we started to change different things to see what happened with the other code"

The most prominent difficulties related to movement, jumping, the use of loops, and hiding/showing different sprites. These actions were the main problems that the children had to deal with from the beginning of their game creation and defined their thinking processes. This was also indicated by the artifact analysis of the first versions of their games. In order to make a character move and jump in Scratch, you often have to have an event block with a conditional combined with motion blocks for moving the sprite x steps or to place it in a certain y- or x-coordinate in a chosen direction. Observations showed that movement and jumping were the most common reasons the children asked for help, indicating that it was hard for them to articulate their knowledge about conditionals (if _ then; repeat until; and when key is pressed, do this), direction, and the coordinate system to achieve an appropriate order of blocks.

Coding in Scratch enables children to articulate their thoughts and watch the outcomes of their own decisions.

"If you did something, the result wasn't always what you expected"

After the initial trials with coding, by being more and more engaged in the process, the children had the opportunity to clarify their thinking and interpret the immediate feedback, acting accordingly.

"Before, I didn't understand that things wouldn't happen if you didn't explicitly give instructions"

"The ideas and code come really fast when you realize what kind of options you have"

4.3. Cycle 3

The key findings, as design elements, that emerged from cycle 2 and guided the refinement of the design of cycle 3 are described below:

a Allow an adequate amount of time for engagement during the workshop.

The analysis of the interview data revealed that the time the students had to complete the tasks was an important issue for them, as the allocated time was limited. More precisely, at the beginning of the activity, they spent a lot of time trying to familiarize themselves with the tool and the tasks and to bond with team members, especially in teams where they did not know each other from before. Giving additional time for social engagement between the team members will allow students to build common understanding and be more creative.

b Provide a specific theme for the game creation.

As mentioned earlier, the students spent a lot of time at the beginning of the workshop. One of the time-consuming actions was to decide the theme of the game. Time management is very important in such workshops: on the one hand, students need to have the freedom to decide their own themes; on the other hand, it is critical to have an adequate amount of time for the follow-up tasks. Therefore, having a specific theme for the game creation that is sufficiently broad, inspiring, and meaningful will give them the freedom to be creative but at the same time will prevent them from "getting lost." In addition, it will give a meaningful social and personal context to the learning process, foster their interest, and create a common ground for all teams.

c Inspire the participants with an example of a female game hero and a demonstration of a similar game by female assistants (as role models).

From cycle 2, focusing on the analysis of the data collected from the teams consisting only of girls, it is evident that stereotypes exist. Most of them expressed that they had not tried coding before and did not know what to create, as they thought game creation was only for "geeks." In their eyes, only boys like video games. To encourage interest and get the girls inspired and engaged, a storyboard and a game were used as examples, with the main character a heroine who had powers that could "solve problems".

d Focus on the design part of the game in a structured way (i.e. spend sufficient time on creating the storyboard first and having a presentation on it).

The results from the data from cycle 2 (interviews, observations, and game versions) showed that the teams who followed a more-structured approach (creating a draft storyboard with their idea before starting coding) were able to successfully manage and finish on time, as well as being less overwhelmed. Moreover, based on the different versions of the collected games, these students had a greater capacity to develop their initial ideas (designed in the storyboards), and this resulted in higher-quality games (more complete/advanced).

e Introduce coding individually.

The students participating in the workshop did not have the same experience with coding. This approach was geared towards helping the participants individually to familiarize themselves with the tool (in our case, Scratch), gain insights on what they could create, and develop basic skills. Having a common ground of basic knowledge among the team members will make everyone engaged and active. Thus, it is very important to have some individual activities at the beginning that prevent students with experience from dominating their teams, which could disengage novices.

One theoretical idea emerged from this cycle:

1) Use of powerful ideas: "Powerful ideas", as described by Papert (Papert, 1980), are central concepts of learning and should be a necessary part of constructionist activities. A "powerful idea" must be both personally and epistemologically useful, giving the opportunity to organize a way of thinking, appropriate each time for the specific task, building on previously gained skills and knowledge. Learners need to be highly explorative before they gain expertise; therefore, the task they are required to do needs to be engaging enough in order to commit them to the learning process. In his book *Mindstorms*, Papert shows the importance of powerfulness and the powerful nature of children's use of computers as tools and the Logo programming language, as well all the powerful ideas that emerge from children's engagement with computer-based activities.

What is important is to make a powerful idea part of intuitive thinking (Papert, 1980). In the design of the activity in the third cycle, "powerful" was a quality gained from the girls, as they were allowed to closely engage with the creation of the artifacts in multiple stages, using Scratch. This process brought the learners in touch with some powerful general ideas, for example planning an exciting project, using programming instructions, debugging, and designing, to mention a few.

The girls had an experience outside of the classroom in a local library, collaborating with girls of a similar age but with varied interests and background knowledge, which was in contrast with a single classroom experience. The duration of the workshop was critical not only for learning purposes but also because it allowed the participants to bond and exchange interests and gave them the proper amount of time to interact, negotiate, learn from each other, and finally achieve the goal of the creation of the artifact. In addition, by having a concrete context for the game (create a game that reflects an environmental issue) and a tool (Scratch) embedded in a meaningful environment, they could see the project's relevance to their lives.

"It was so fun and exciting to make a game for saving the world with Scratch and with new friends, who taught me so much about computers"

The girls gradually discovered the Scratch tool and how they could use it. As they became more engaged in the process and saw their artifact become a reality, they enhanced their feelings of self-achievement and self-confidence. They found themselves confronting difficulties and learning things that they did not know about game design. The use of Scratch gave them new possibilities and made them "walk it through" and relate their personal knowledge to thinking effectively and happily to achieve the artifact construction.

"I thought it was much harder to make a game, but I could understand how to use it and at the end we managed to do everything we wanted"

"… some things were difficult, but we tried and made things happen"

"… we find out how things worked, and many times we had to go back and change stuff"

"I am so proud of what I did today … When you design a game in a storyboard, you don't think about using a timer, but with Scratch you can … you can do everything you can think of"

5. Discussion

The intended outcomes of this DBR were twofold: 1) to ground the main findings of interventions conducted over two years in constructionism, and 2) to identify reusable design principles that can inform coding activities for children and pedagogical tasks. In this study, we aimed to investigate children's learning experience as they constructed their own knowledge by using a digital programming tool (Scratch) and collaboratively creating socially meaningful artifacts: games.

Analysis of the different data collected from the various instruments over the two-year intervention helped us to explore the effectiveness of our coding workshops on children's engagement. We focused on how they enhanced participants' knowledge of basic programming concepts, their coding behavior, their social interaction and collaboration, and how they perceived their coding experience as a whole when introduced to coding.

It is important to have appropriate educational designs aiming to promote active learning with the support of constructionism. Including components like a balance of individual and social involvement and the use of a visual programming language, all employed under the common goal of creating an artifact, fosters children's deeper transferable CT skills, which are vital for our society's information revolution. Engaging children in a learning environment that embraces creative design, problem solving, collaboration, and communication strengthens their sense of competence and confidence, their compassion for others, and their moral character (Bers, 2010). Together with achieving a significant improvement in students' understanding of computational knowledge, like programming concepts and practices, it is essential to create high levels of motivation, fun, and commitment as part of an efficient pedagogical design, as reflected in our study.

5.1. Engagement in constructionism-based coding activities

Below we summarize the main characteristics of student engagement, as shown in our DBR approach and according to constructionism.

The students indicated that they were cognitively engaged during the workshops; they managed to adopt deliberative thinking and to understand and imitate mechanical thinking while coding. In order to achieve this, they had to use an appropriate cognitive strategy (e.g., a "hypothesis-testing" gaze pattern, as shown by the eye-tracking data) to approach the task and achieve some level of self-regulation (Papavlasopoulou, Sharma et al., 2017). There are different ways to approach a problem, and it takes time to learn the necessary skills. In our workshops, we used a visual programming tool (Scratch); one of the strengths of such tools is that computational practices become less cognitively challenging (Kelleher & Pausch, 2005), so students can focus on problem solving and creative thinking (Lin & Liu, 2012). Even with the use of such tools, during the coding process, cognitive load can be critical, as students use the "bricolage" style by constantly experimenting and trying different patterns. Instructors can help students to manage their learning and thinking and to adopt an effective approach to coding. This is not a new practice, as previous studies with Logo have used precise instructions for computational practices such as testing and debugging (Fay & Mayer, 1994) (Carver & Mayer, 1988).

Cognitive effort, as shown in our study, is also linked with students' behavioral and emotional engagement because positive attitudes have an effect on their load management. Students should be persistent, put effort in, and deal with difficulties; therefore, having positive attitudes and keeping themselves motivated result in better management of their cognitive load (Papavlasopoulou, Sharma, & Giannakos, 2018). In the same vein, Robertson and Howells (2008) argue that the game design experience is a powerful learning environment that supports motivation, engagement, and enthusiasm. Using a visual programming environment, students can be introduced to programming concepts in a fun and useful way through a design activity, making them highly motivated and positive towards coding (Giannakos & Jaccheri, 2018) (Sáez-López et al., 2016).

Social engagement is important as students work in front of the computer and reflect on their progress as a team, sharing the same goal to successfully create an artifact. Working as a team, in our workshops, the students built a group identity and at the same time engaged in social comparison with their peers. Students, especially novices to coding, usually have difficulties with simple coding actions, from relating different commands together to completing more-advanced actions, like debugging; collaboration helped the students in this study to confront those difficulties. In a similar study with girls creating games, good collaboration in debugging resulted in the girls being more persistent when coding on their own, without help from the instructors (Denner, 2007). In the present study, helping each other and sharing their challenges and successes were critical for our students, nurturing social engagement and avoiding a sense of isolation. Collaboration and reflection lead to better learning and powerful thinking. Reflection relates to their own learning experience or reflecting on their peer's code and actions. Previous studies have shown that students performed better when they were working in pair programming (Lye & Koh, 2014) (Werner, Denner, Campe, & Kawamoto, 2012); in a game-making study, when taking into account peers' recommendations and spending time applying these changes, girls produced higher-quality games (Robertson, 2012). Over time, the students in our workshops were able to understand more about coding and became more behaviorally and emotionally engaged. They were able to reflect on the more-complex aspects of their own thinking accordingly by making decisions and controlling the outcomes. Students who are actively part of game-making activities strengthen their problem-solving, critical thinking, and CT skills (Grover & Pea, 2013). During construction, students have to investigate different strategies, negotiate and make decisions about possible solutions, confront problems, and organize their thoughts and actions (Bers, Flannery, Kazakoff, & Sullivan, 2014).

One of the core aspects of a learning activity is the fact that the problem should be meaningful to the learners. In our case, they constructed shared artifacts that mattered to them. Different studies have used problems like designing games (Denner & Werner, 2007) or stories (Burke, 2012). A "powerful idea" must be both personally and epistemologically useful to ensure engagement. The students in our workshops saw themselves gaining a powerful quality by organizing a new way of thinking, building on their previous knowledge and skills. Nowadays, significant value is placed on transferable skills related to digital technology, as they are vital for children's role in the digital world and should be enhanced through activities that are connected to their lives (Iversen, Smith, & Dindler, 2018). In constructionist learning, students deal with difficulties, learn step by step to solve problems, develop belief in their skills, and share ideas with peers (Çakır et al., 2017) (Chu et al., 2017). In our study, this was confirmed: the students increased their sense of achievement, self-confidence, and self-efficacy. At the end of the workshops, the students felt competent and proud of their achievements. After the workshop, compared to the boys, the girls expressed lower self-efficacy (a belief in one's capacity to succeed in tasks), possibly due to the fact that most of them did not have any previous experience with coding. A sense of self-efficacy is important and should be enhanced, as it is related to cognitive strategies, effort, and persistence in learning environments (Bandura, 1997).

5.2. Principles to facilitate constructionism-based learning environments that support children's learning experience

In summary, we identified the following nine principles emerging from our DBR study, which shed light on best practices in the design of coding activities for children based on constructionism. The principles emerged represent the knowledge gained from the two years of interventions and the comparative and retrospective analysis of the outcomes based also on the literature:

1) *Social interaction:* Collaboration between team members is a vital part of coding activities. It is essential to enhance this and to ensure that there is a sense of equality of effort, involvement, and participation between team members and among teams.
2) *Appropriate design according to age*: Different age groups (teens and kids) need different approaches and designs in order to engage with a coding activity. The instruction should consider the characteristics of each age group. One example is to promote a focus on functionality rather than graphics from the beginning of the activity to aid younger participants. Instructors should ensure that children receive guidelines on where to focus their attention when they code (such as commands and output in Scratch).
3) *Duration of the activity:* According to constructionism (Papert, 1980), when having children use technological tools, duration is key for them to become personally, intellectually, and emotionally involved. Workshops with longer hours can enable children to learn strategies, gain technological skills, make connections with their own practices, and engage with coding, helping to increase their knowledge.
4) *Relevance of the activity and meaningful content:* Offering a supportive theme for the artifact creation process, in which participants can meaningfully participate in real-life settings, is a key factor supporting the psychological and sociocultural elements for effective learning. Children become engaged and actively involved in the process of artifact creation when it is meaningful for them and related to a real-life context.
5) *Physical and digital artifacts:* The results of the present study showed that the inclusion of physical tasks was engaging and enabled the participants to enhance their skills. The initial task of designing and drawing in the traditional way (using pen and paper, as well as other tangible materials) immediately put players into action and created a physical and emotional peak in the process.
6) *Children's attitudes and motivation:* The learning process should be supported by providing tasks that encourage children to reflect, motivate them to collaborate, and give them meaningful reasons to complete their artifacts. In this vein, Papert (1980, p. 42) highlighted a resemblance with juggling: *"in a learning environment with the proper emotional and intellectual support, the 'uncoordinated' can learn circus arts like juggling and those with 'no head for figures' learn not only that they can do mathematics but that they can enjoy it as well."*
7) *Cognitive overload:* Coding activities for children can have a high cognitive load, which affects their performance and overall experience with the tasks. Proper organization and integration of the learning materials, with a coherent representation and instruction of the related digital tools, tasks, and activities, are required to avoid unnecessary streams of information and cognitive overload.
8) *Appropriate tasks:* To effectively implement a coding workshop, the tasks should make the children both interested and able to learn. The process should afford participants the opportunity to apply aspects of problem solving, coding, debugging, collaborating, planning, communicating, and reflecting on their work. The tasks should support children's and instructors' ability to work through the process of creating an artifact and benefit from an appropriate sequence of tasks that allows the maximum use of their abilities. The proposed tasks are: 1) a warm-up activity and an inspiring introduction, 2) explore/design, 3) construct/create the digital artifact, and 4) evaluate/get feedback from peers, all alongside collaborating with team members and receiving support from assistants/instructors.
9) *Meaningful framework for the involvement of the instructors:* In the construction of an artifact, children are not alone: practitioners (e.g., teachers and assistants) and anyone else who is responsible for the learning task are also involved. Therefore, they should strive to create more-articulate and -honest teaching relationships. Working with digital tools allows the teacher and the learner to share a common goal by trying to get the computer to do what they want and trying to understand what it does. As they create the artifact and encounter "bugs", children engage in conversations and develop the appropriate language to ask for help when they need it. As each artifact process is unique, new situations might occur that neither the teacher nor the learner has faced before. So, the teacher should be dynamically involved in the creation and the discussions that occur. In that way, there is an opportunity to find new ways to explain and show in real time the concepts needed to the children. As noted by Papert (Papert, 1980): *"sharing the problem and the experience of solving it allows a child to learn from an adult not 'by doing what teacher says' but 'by doing what teacher does.'"*

6. Limitations

This study had some limitations. First, our workshops were designed for children who had no previous experience of coding. The participants were randomly selected; therefore, the sample was not consistent in terms of the children's prior knowledge and interaction with coding. Even though we had an indication in our data collection to measure the children's previous knowledge, we could have used other methods to be more accurate. Second, the factors that might affect children's self-perceptions are much more complex than we might assume. Third, although the participants of the third cycle were committed for the two days' workshop and gave us high quality data, the sample is not large; this is due to unexpected matters from the participants' side prior to the scheduled dates of the workshop. In addition, the age range of the students in the study is big (8–17), maybe, focusing on a smaller range would have given a different perspective. Demographic variables and other characteristics (cognitive and motivational) that distinguish them from the rest of the population could have confounded the findings. Artifacts like games might be imperfect examples of what children learn, especially when they receive help during the process. Despite the fact that we observed the teams and made notes on the help they received, we might have underestimated or overestimated their understanding of programming concepts.

In addition, limitations due to the types of data collection methods and instruments used apply in our case. One limitation related to the eye tracking: the young age of the participants, their enthusiasm during the activities, and the fact that eye trackers are designed for adults made it difficult to gather good-quality data. Moreover, this project used Scratch as a programming environment for the development of the artifact: another technological tool might have had a different impact on the children's experience. Our choice was based on our literature review and the acknowledged benefits of this programming environment, which has been widely used over the last few years. Although we tried to apply all aspects of the DBR methodology in our study, showing the relationship between theory and practice (of the artifact construction activity), there were still some limitations. The data were extensive and comprehensive, requiring extended time for collection and analysis; consequently, because time and resources were limited, some data might have been discarded or received less attention. Lastly, we defined in detail the setting of our study and how theory was linked with the context; by default, this has a bias, as it presents our own understanding of contextualizing the theory.

7. Future work

Future research should further explore gender differences. Although the main focus of our study was not to investigate gender differences in the process of creating an artifact, we found that girls like to make different type games from boys, in terms of both content (story/purpose of the game) and elements (colors and main character), and tend to handle the process slightly differently. In addition, future plans should include conducting our coding workshops in school settings to explore their effects under a traditional teaching approach. Among other aspects, researchers could explore the correlations with students' performance in the form of grades. Finally, in terms of theory, it would be interesting to see more studies in the area that ground their findings in constructionism. This would bring together researchers in the same area to build a common ground regarding outcomes.

Acknowledgements

The authors would like to express their gratitude to all of the children, teachers and parents for volunteering their time. Our very special thanks go to Kshitij Sharma, Uyen Dan Nguyen, Kristin Susanne Karlsen, Ioannis Leftheriotis, Amanda Jørgine Haug, Lidia Luque Fernandez, An Nguyen, Ton Mangmee, Marjeris Sofia Romero, Eline Stenwig and Kristoffer Venæs Monsen. The project has been recommended by the Data Protection Official for Research, Norwegian Centre for Research Data (NSD), following all the regulations and recommendations for research with children.

This work supported from the "Learning science the fun and creative way: coding, making and play as vehicles for informal science learning, in the 21st century" Project, under the European Commission's Horizon 2020 SwafS-11-2017 Program (Project Number: 787476). This article reflects the views only of the authors and it does not represent the opinion of neither the European Commission nor NTNU, and the European Commission and NTNU can not be held responsible for any use that might be made of its content. This work is also supported from the Norwegian Research Council under the project FUTURE LEARNING (number: 255129/H20), and by NOKUT under the Centre for Excellent IT Education (Excited) (number: 16/02049).

References

Anderson, T. (2005). Design-based research and its application to a call centre innovation in distance education. *Canadian Journal of Learning and Technology/La revue canadienne de l'apprentissage et de la technologie, 31*(2).

Anderson, T., & Shattuck, J. (2012). Design-based research: A decade of progress in education research? *Educational Researcher, 41*(1), 16–25.

Bandura, A. (1997). *Self-efficacy: The exercise of control.* Macmillan.

Barab, S., & Squire, K. (2004). Design-based research: Putting a stake in the ground. *The Journal of the Learning Sciences, 13*(1), 1–14.

Bers, M. U. (2010). Beyond computer literacy: Supporting youth's positive development through technology. *New Directions for Student Leadership, 2010*(128), 13–23.

Bers, M. U. (2012). *Designing digital experiences for positive youth development: From playpen to playground.* Oxford University Press.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education, 72*, 145–157.

Bonner, D., & Dorneich, M. (2016). Developing game-based learning requirements to increase female middle school students interest in computer science. *Paper presented at the Proceedings of the human factors and ergonomics society annual meeting*.

Buechley, L., Eisenberg, M., Catchen, J., & Crockett, A. (2008). The LilyPad Arduino: Using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. *Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems*.

Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education, 4*(2), 121–135.

Çakır, N. A., Gass, A., Foster, A., & Lee, F. J. (2017). Development of a game-design workshop to promote young girls' interest towards computing through identity exploration. *Computers & Education, 108*, 115–130.

Carver, S., & Mayer, R. (1988). Learning and transfer of debugging skills: Applying task analysis to curriculum design and assessment. In R. E. Mayer (Ed.). *Teaching and learning computer programming: Multiple research perspectives* (pp. 259–297). Hillsdale, NJ: Lawrence Erlbaum Associates.

Chu, S. L., Schlegel, R., Quek, F., Christy, A., & Chen, K. (2017). 'I make, therefore I Am': The effects of curriculum-aligned making on children's self-identity. *Paper presented at the proceedings of the 2017 CHI conference on human factors in computing systems*.

Collective, D.-B. R. (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher, 32*(1), 5–8.

Collins, A. (1992). Toward a design science of education. *New directions in educational technology* (pp. 15–22). Springer.

Denner, J. (2007). The Girls Creating Games Program: An innovative approach to integrating technology into middle school. *Meridian: A Middle School Computer Technologies Journal, 1*(10).

Denner, J., & Werner, L. (2007). Computer programming in middle school: How pairs respond to challenges. *Journal of Educational Computing Research, 37*(2), 131–150.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education, 58*(1), 240–249. https://doi.org/10.1016/j.compedu.2011.08.006.

Eordanidis, S., Gee, E., & Carmichael, G. (2017). The effectiveness of pairing analog and digital games to teach computer science principles to female youth. *Journal of Computing Sciences in Colleges, 32*(3), 12–19.

Esper, S., Foster, S. R., Griswold, W. G., Herrera, C., & Snyder, W. (2014). CodeSpells: Bridging educational language features with industry-standard languages. *Paper presented at the proceedings of the 14th Koli calling international conference on computing education research*.

Fay, A. L., & Mayer, R. E. (1994). Benefits of teaching design skills before teaching logo computer programming: Evidence for syntax-independent learning. *Journal of Educational Computing Research, 11*(3), 187–210.

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education, 63*, 87–97. https://doi.org/10.1016/j.compedu.2012.11.016.

Fields, D., Vasudevan, V., & Kafai, Y. B. (2015). The programmers' collective: Fostering participatory culture by making music videos in a high school Scratch coding workshop. *Interactive Learning Environments, 23*(5), 613–633.

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School engagement: Potential of the concept, state of the evidence. *Review of Educational Research, 74*(1), 59–109.

Gallup, G. a. (2015). Searching for computer science: Access and barriers. *U.S. K–12 education*.

Giannakos, M. N., & Jaccheri, L. (2018). From players to makers: An empirical examination of factors that affect creative game development. *International Journal of Child-Computer Interaction, 18*, 27–36.

Granic, I., Lobel, A., & Engels, R. C. (2014). The benefits of playing video games. *American Psychologist, 69*(1), 66.

Grover, S., & Pea, R. (2013). Computational thinking in K–12 a review of the state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189X12463051.

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education, 25*(2), 199–237.

Guzdial, M. (2004). Programming environments for novices. *Computer Science Education Research, 2004*, 127–154.

Henrie, C. R., Halverson, L. R., & Graham, C. R. (2015). Measuring student engagement in technology-mediated learning: A review. *Computers & Education, 90*, 36–53.

Horizon, M. (October 5, 2015). *Horizon Media study reveals Americans prioritize STEM subjects over the arts; science is "cool," coding is new literacy.* PR Newswire. Retrieved from http://www.prnewswire.com/news-releases/horizon-mediastudy-reveals-americans-prioritize-stem-subjects-over-the-arts-science-is-cool-coding-is-new-literacy-300154137.html.

Hubwieser, P., Armoni, M., Giannakos, M. N., & Mittermeir, R. T. (2014). Perspectives and visions of computer science education in primary and secondary (K-12) Schools. *ACM Transactions on Computing Education (TOCE), 14*(2), 7.

Hubwieser, P., Giannakos, M. N., Berges, M., Brinda, T., Diethelm, I., Magenheim, J., & Jasute, E. (2015). A global snapshot of computer science education in K-12 schools. *Paper presented at the proceedings of the 2015 ITiCSE on working group reports*.

Iversen, O. S., Smith, R. C., & Dindler, C. (2018). From computational thinking to computational empowerment: A 21st century PD Agenda. *Paper presented at the participatory design conference*.

Jones, S. P. (2013). *Computing at school in the UK.* http://research.microsoft.com/en- us/um/people/simonpj/papers/cas/computingatschoolcacm.pdf.

Kafai, Y. B. (2006). Playing and making games for learning: instructionist and constructionist perspectives for game studies. *Games and Culture, 1*(1), 36–40.

Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational Psychologist, 50*(4), 313–334. https://doi.org/10.1080/00461520.2015.1124022.

Kafai, Y. B., & Kafai, Y. B. (1995). *Minds in play: Computer game design as a context for children's learning.* Routledge.

Kafai, Y. B., & Resnick, M. (2012). Introduction. *Constructionism in practice* (pp. 13–20). Routledge.

Kafai, Y. B., & Vasudevan, V. (2015). Constructionist gaming beyond the screen: Middle school students' crafting and computing of touch pads, board games, and controllers. *Paper presented at the proceedings of the workshop in primary and secondary computing*

education.
Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. *Computers in Human Behavior, 52*, 200–210.
Kao, D., & Harrell, D. F. (2017). MazeStar: A platform for studying virtual identity and computer science education. *Paper presented at the proceedings of the 12th international conference on the foundations of digital games*.
Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR), 37*(2), 83–137.
Lin, J. M.-C., & Liu, S.-F. (2012). An investigation into parent-child collaboration in learning computer programming. *Journal of Educational Technology & Society, 15*(1).
Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51–61.
Mayring, P. (2014). *Qualitative content analysis: Theoretical foundation, basic procedures and software solution*.
Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2017a). Empirical studies on the maker movement, a promising approach to learning: A literature review. *Entertainment Computing, 18*, 57–78. https://doi.org/10.1016/j.entcom.2016.09.002.
Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2017b). Reviewing the affordances of tangible programming languages: Implications for design and practice. *Paper presented at the global engineering education conference (EDUCON), 2017*. IEEE.
Papavlasopoulou, S., Sharma, K., Giannakos, M., & Jaccheri, L. (2017). Using eye-tracking to unveil differences between kids and teens in coding activities. *Paper presented at the proceedings of the 2017 conference on interaction design and children*.
Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2018). How do you feel about learning to code? investigating the effect of children's attitudes towards coding using eye-tracking. *International Journal of Child-Computer Interaction, 17*, 50–60.
Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* Basic Books, Inc.
Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer.* ERIC.
Papert, S. (1997). *Why school reform is impossible (with commentary on O'Shea's and Koschmann's reviews of" the children's machine*. JSTOR.
Parmaxi, A., & Zaphiris, P. (2015). Developing a framework for social technologies in learning via design-based research. *Educational Media International, 52*(1), 33–46.
Parmaxi, A., Zaphiris, P., & Ioannou, A. (2016). Enacting artifact-based activities for social technologies in language learning using a design-based research approach. *Computers in Human Behavior, 63*, 556–567.
Reeves, T. (2006). Design research from a technology perspective. *Educational design research* (pp. 64–78). Routledge.
Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. *Paper presented at the Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*.
Robertson, J. (2012). Making games in the classroom: Benefits and gender concerns. *Computers & Education, 59*(2), 385–398. https://doi.org/10.1016/j.compedu.2011.12.020.
Robertson, J. (2013). The influence of a game-making project on male and female learners' attitudes to computing. *Computer Science Education, 23*(1), 58–83.
Robertson, J., & Howells, C. (2008). Computer game design: Opportunities for successful learning. *Computers & Education, 50*(2), 559–578.
Ruggiero, D., & Green, L. (2017). Problem solving through digital game design: A quantitative content analysis. *Computers in Human Behavior, 73*, 28–37.
Sáez-López, J.-M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education, 97*, 129–141. https://doi.org/10.1016/j.compedu.2016.03.003.
Saldaña, J. (2015). *The coding manual for qualitative researchers.* Sage.
Schmitz, B., Klemke, R., Walhout, J., & Specht, M. (2015). Attuning a mobile simulation game for school children using a design-based research approach. *Computers & Education, 81*, 35–48.
Stewart-Gardiner, C., Carmichael, G., Latham, J., Lozano, N., & Greene, J. L. (2013). Influencing middle school girls to study computer science through educational computer games. *Journal of Computing Sciences in Colleges, 28*(6), 90–97.
Sung, H.-Y., & Hwang, G.-J. (2013). A collaborative game-based learning approach to improving students' learning performance in science courses. *Computers & Education, 63*, 43–51.
Teague, J. (2002). Women in computing: What brings them to it, what keeps them in it? *ACM SIGCSE Bulletin, 34*(2), 147–158.
Tuhkala, A., Wagner, M.-L., Nielsen, N., Iversen, O. S., & Kärkkäinen, T. (2018). Technology comprehension: Scaling making into a national discipline. *Paper presented at the proceedings of the conference on creativity and making in education*.
Turner, J. C., Christensen, A., Kackar-Cam, H. Z., Trucano, M., & Fulmer, S. M. (2014). Enhancing students' engagement: Report of a 3-year intervention with middle school teachers. *American Educational Research Journal, 51*(6), 1195–1226.
Wagner, A., Gray, J., Corley, J., & Wolber, D. (2013). Using app inventor in a K-12 summer camp. *Paper presented at the Proceeding of the 44th ACM technical symposium on Computer science education*.
Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research & Development, 53*(4), 5–23.
Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: Measuring computational thinking in middle school. *Paper presented at the proceedings of the 43rd ACM technical symposium on computer science education*.
Wilson, A., & Moffat, D. C. (2010). Evaluating Scratch to introduce younger schoolchildren to programming. *Proceedings of the 22nd annual psychology of programming interest group*. Leganés, Spain: Universidad Carlos III de Madrid.
Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.
Zhang, J. X., Liu, L., de Pablos, P. O., & She, J. (2014). The auxiliary role of information technology in teaching: Enhancing programming course using Alice. *International Journal of Engineering Education, 30*(3), 560–565.

Paper 6:

Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences

Sofia Papavlasopoulou, Kshitij Sharma, and Michail N. Giannakos

Computers in Human Behavior

ARTICLE IN PRESS

Computers in Human Behavior xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Full length article

Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences

Sofia Papavlasopoulou, Kshitij Sharma, Michail N. Giannakos*

Department of Computer Science (IDI), Norwegian University of Science and Technology (NTNU), Trondheim, Norway

A R T I C L E I N F O

Keywords:
Coding
Computational thinking
Eye-tracking
Gender differences
Learning strategies

A B S T R A C T

Computational thinking and coding are becoming an integral part of K-12 education, with female students being underrepresented in such subjects. The proliferation of technological tools and programming environments offers the opportunity for creative coding activities for children and increases the need for appropriate instructional practices. In this study, we design and evaluate a coding workshop for children. Our goal is to examine differences between boys and girls using eye-tracking as an objective measure and triangulating the findings with qualitative data coming from children's interviews. The results show no statistically significant difference between female and male gaze and learning gain during the coding activity; interestingly, the qualitative data show differences in the strategies and implemented practices during coding, and in perceptions about those coding activities. Our results highlight that further studies need to utilize objective measures and unveil necessary differences in the design and implementation of coding activities. Furthermore, our results provide objective evidence that female students do not lack in competences compared to boys, but simply that they have a different approach during coding activities and different perspectives about coding, an approach that needs to be cultivated and nurtured.

1. Introduction

Increasing attention has been given to children's acquisition of 21st-century skills and digital competences. In accordance with this need, computational thinking and coding have, in recent years, become an integral part of school curricula in many countries. Estonia, Israel, Finland and the United Kingdom are only a few examples of the growing efforts of governments to integrate coding as a new literacy and to support students in creative problem-solving tasks (Hubwieser, Armoni, Giannakos, & Mittermeir, 2014). Similarly, organizations such as "code.org", "codeacademy.com" offer fruitful learning environments to promote coding activities. In addition, ACM, the Computer Science Teachers Association, National Math and Science Initiative and K-12 Computer Science Framework provide guidelines for informing and building communities for the teaching of computer science. While ever more people believe that coding skills are as important as math and writing (Horizon, 2015), there is an acute need for evidence about the design of effective and engaging learning activities for children (Guzdial, 2017; Lye & Koh, 2014).

Since the first appearance of Papert's Logo programming environment introduced in the 1960s, many other programming tools have emerged. Today, there is a lot of child-friendly software that offers an intuitive and pleasant experience while coding. Examples of such software are Scratch, Alice and Kodu. By participating in coding activities, children are exposed to computational thinking (Wing, 2006), which involves, but is not limited to, critical thinking, problem solving and creativity. These activities are apparent in both formal and informal settings and are characterized by different designs, technologies and approaches (Papavlasopoulou, Giannakos, & Jaccheri, 2017). Previous research shows that different approaches can combine physical fabrication and coding (Kafai & Vasudevan, 2015), while others, such as Buechley, Eisenberg, Catchen, and Crockett (2008), used LilyPad Arduino to make coding attractive to girls. By using Logo-based environment and an interactive white board from kindergarten age, children developed mathematical concepts and social interaction, at the same time as enjoying the learning activity (Fessakis, Gouli, & Mavroudi, 2013). Different benefits arise from grounding coding activities on constructionism (Papert, 1980); children are given the opportunity to enhance their understanding of programming concepts, to promote collaboration with friends, and to change their attitudes towards coding (Fessakis et al., 2013; Maloney, Peppler, Kafai, Resnick, & Rusk, 2008; Papavlasopoulou, Giannakos, & Jaccheri, 2019).

Despite the growing research and the many possibilities offered by learning environments to design constructionism-based coding

* Corresponding author. Sem Sælands vei 9, NO-7491 Trondheim, Norway.
E-mail address: michailg@ntnu.no (M.N. Giannakos).

https://doi.org/10.1016/j.chb.2019.03.003
Received 16 July 2018; Received in revised form 15 January 2019; Accepted 3 March 2019

activities for children (Kafai & Burke, 2015), there are relatively few studies focusing on gender issues in making and coding activities for children (Papavlasopoulou, Giannakos, et al., 2017). Gender discrepancy in science, technology, engineering and mathematics (STEM) exist, with women more underrepresented in the field of computer science (Cheryan, Ziegler, Montoya, & Jiang, 2017). In terms of interest, the gender gap starts at elementary school (Ceci & Williams, 2010). Among the different factors that impact on women following computer science paths are the lack of positive educational experiences in their childhood (Adya & Kaiser, 2005), their fear of being involved in very technical coding courses, and stereotypes and misconceptions around careers in computing (Teague, 2002). Girls' interest in computer science from a young age possibly fades because of a gendered or non-appropriate pedagogical approach (Schulte & Knobelsdorf, 2007). With respect to decreasing the gender gap in participation and to attracting more girls to computing, several studies on coding workshops have focused on differences in girls' competences compared to boys' (Kalelioğlu, 2015), while others have explored increasing girls' self-efficacy, interest, attitudes and confidence (Cheryan et al., 2017; Çakır, Gass, Foster, & Lee, 2017). However, the methods used in these studies were traditional qualitative and/or quantitative instruments like surveys, tests and interviews (Papavlasopoulou, Giannakos, et al., 2017). In order to gain knowledge on how to design the coding activities, it is necessary to use new objective methods to investigate the existence of gender differences in aspects like learning performance in coding activities and to discover the main differences between boys' and girls' practices (Papavlasopoulou, Sharma, & Giannakos, 2018; Papavlasopoulou, Sharma, Giannakos, & Jaccheri, 2017).

In this study we designed and evaluated a coding workshop for children aged 8–17 years old. The aim was to investigate if gender differences exist in children's coding behavior. We used objective measures (children's gaze) and triangulated them with qualitative data (interviews with the children) in order to acquire a deeper understanding of children's perspectives and practices. Eye-tracking is a method widely used in computer programming (Obaidellah, Al Haek, & Cheng, 2018), but studies with children are very limited and, to the best of our knowledge, eye-tracking has not been used to discover gender differences in children. Our study addressed the following research questions:

(1) Is there a difference in girls' and boys' gaze patterns in coding activities?
(2) Is there a difference in learning gain among boys and girls in coding activities?
(3) What are the differences in boys' and girls' strategies and implemented practices during the coding activity?

The rest of paper is structured as follows: in the following section, we present the related work and background theories; the third section describes our constructionism-based coding activity and the methodology used in our study; the fourth section presents the research findings; and the fifth and final section discusses the results in relation to the relevant literature, presents the research limitations, and suggests directions for future research.

2. Related work and background theory

2.1. Gender differences and characteristics in children's coding

The gender gap in STEM-related contexts has been examined in recent years (Cheryan et al., 2017). The profile of a computer scientist seems still to be stereotyped, and women show less interest in computer science and less likelihood to consider it properly as a possible future career (Wang & Degol, 2017). As a way to attract more females to computer science and to increase gender equality, educators have focused on offering diverse coding experiences specially for girls (Kelleher, Pausch, & Kiesler, 2007). The aim is to increase their interest in coding, enhance their attitudes, and examine the causes of gender differences (Denner, Werner, & Ortiz, 2012; Robertson, 2012). Çakır et al. (2017) conducted a game-design workshop for girls, showing that, at the end of the workshop, the girls had better attitudes towards computer science, higher confidence and self-reported competence with computers. However, a study by Robertson (2013) that investigated the influence of a game-development project on students' attitudes revealed that the level of enjoyment of the project was higher in boys than in girls, and that the project did not increase the possibility of them studying computer science in the future.

Bruckman, Jensen, and DeBonte (2002) showed that gender did not affect children's performance level in coding. Similarly, in a study of a game-development task for fourth-grade students, Owston, Wideman, Ronda, and Brown (2009) demonstrated that there were no gender differences in the learning outcomes. No significant gender differences were found in elementary school students' competence, interest at school and the use of deep learning strategies while constructing a “drag and drop” game (Vos, Van Der Meijden, & Denessen, 2011). Another study involving game-making showed that girls focused more on trying to improve their games following their peers' recommendations, and that overall they achieved a higher game quality (Robertson, 2012). In addition, in a study of the use of the code.org website to teach coding to primary school students, it was shown that girls' means of reflective thinking skills towards problems solving were higher than boys, although the results showed no statistically significant difference (Kalelioğlu, 2015).

Concerning children's approaches to and practices of coding activities, studies have reported differences depending on the gender of the participants. Robertson (2012) found that girls approached the game-making process differently when using a software called Adventure Author. For example, girls were spending more time than boys in writing dialogues for their games. In addition, girls' greater interest in narration was reflected in the use of Alice software, which is specifically designed for storytelling; nevertheless, this resulted in equal gains to those achieved by the girls who used the generic version (Kelleher et al., 2007). Denner et al. (2012), in an analysis of 108 games created by 59 middle-school girls, found that they were facing difficulties in organizing the design of their game and in handling their code when many pieces were involved. In another study of girls creating games, the results showed that they were very focused and collaborated well in the debugging process; as a result, they were trying to work more on their own before asking for help from the instructors (Denner, 2007). On the other hand, when boys were dealing with needlework, they were nervous when engaging with craft practices and they considered them to be “women's work”. However, they were committed and realized how challenging and demanding it was to complete their task. Further, in the same study, they were able to see their actions in a tangible way and were testing different codes until they managed to accomplish the desired outcome (Searle & Kafai, 2015). A study with young participants aged 10–12 showed that girls spent more time on aesthetics and put more effort into having a good technical functionality (Lee, Kafai, Vasudevan, & Davis, 2014).

2.2. Gaze behavior and gender differences

Various eye-tracking studies in the past have shown results based on gender differences in a variety of contexts such as usability (Pan et al., 2004), Google searches (Lorigo et al., 2008), web design (Djamasbi et al., 2007), advertisement (Hewig, Trippe, Hecht, Straube, & Miltner, 2008), visuo-spatial planning (Cazzato, Basso, Cutini, & Bisiacchi, 2010), visual toy preferences (Alexander, 2006), facial emotion recognition (Schmid, Mast, Bombari, Mast, & Lobmaier, 2011) and color preferences (Moss & Colman, 2001).

In a study in which the participants were asked to observe a set of preselected gaze behavior, Pan et al. (2004) found that the average fixation duration for men was significantly higher than that for the women in the experiment. This depicted a deeper observational gaze behavior by men than by women in relation to webpages. In a comparison of search tasks (informational vs. navigational), Lorigo et al. (2008) did not find any gender differences based on engagement (pupil dilation). In a web-usability study, Djamasbi et al. (2007) found that the color of a specific part of the webpage influences the gaze behavior of women more than men; similar color preferences were found by Moss and Colman (2001). In an eye-tracking study to examine the role of models' gender in an advertisement on the ratings given by men and women, Hewig et al. (2008) found that the gender of the model has more impact on the ratings of men than of women; however, the authors did not find any other gender differences in gaze behavior. Cazzato et al. (2010) conducted an eye-tracking study in which the participants were required to solve visuo-spatial problems by finding the shortest path between a source and a destination. The authors found that females used more cognitive resources than males. However, in terms of gaze behavior, their study found differences between males and females. When Alexander (2006) presented "masculine" and "feminine" toys to participants, the visual behavior of men and women did not reveal any differences, even though there was clear gender bias in the preferences for toys. In another study in which the participants were asked to identify the facial emotions depicted in pictures, Schmid et al. (2011) found that women performed better than men. However, the authors did not find any gender-based gaze differences in the interfeatural saccades.[1] The only difference was that women processed information less locally than men did (Schmid et al., 2011).

Most of these studies show that gender differences are at the preference and/or performance levels. There are only a few gender differences as far as the gaze behavior is concerned. This leads to a working hypothesis that *there are not many gaze behavioral differences between men (boys) and women (girls)*; the differences occur at the attitudinal/strategic level.

2.3. Constructionism and computational thinking framework

Papert's theory of constructionism argues that learning experiences are more powerful when learners are actively involved in the learning process by creating their own projects (Papert, 1980). With the experience of developing a project, children build on their previous knowledge and discover new knowledge without receiving it passively. The core element in constructionism is an "object-to-think-with"; this is what will provide the opportunity for learners to interact and support their own thinking. However, constructionism is more like a synthesis of characteristics that will result in effective learning. Thus, together with the core element to stimulate an individual's thinking, it is necessary for there to be active involvement of the learners, socially meaningful elements and social interaction (Kafai, 2006).

Computer game programming represents an educational strategy of constructionist learning. During the process of making a game, children try to achieve a goal and to master their own ways of learning and thinking. The use of a programming environment offers the possibility of creating socially meaningful artifacts, of communicating with others and of having a pleasant and engaging experience (Robertson & Howells, 2008). Many studies have used constructionism as a support for coding activities in both formal and informal settings in order to promote coding, problem solving, critical thinking and collaborative skills (Papavlasopoulou, Giannakos, et al., 2017). In this study, which is based on constructionism, *we design a coding activity for children of all ages that does not demand any previous experience in coding.*

Computational thinking can be traced to Papert's strong support of the idea that children who use the Logo programming language develop algorithmic thinking (Papert, 1980). However, the term "computational thinking" was made popular by Wing (2006), who argued that "computational thinking represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" (p. 33). Since then, different efforts to define computational thinking have appeared, with the aim of supporting the importance of research on making computational thinking a 21st-century literacy accessible to all (Guzdial, 2008). Examples include the Computer Science Teachers Association and the International Society for Technology in Education framework (Barr & Stephenson, 2011), and the National Research Council's "Framework for K-12 Science Education" (NRC, 2012). In our study, we adopted Brennan and Resnick's (2012) computational thinking framework. With respect to Scratch, their framework suggests three key dimensions to portray computational thinking: computational concepts (concepts the users engage with when they program, such as parallelism and variables); computational practices (practices that users develop, such as abstraction and debugging); and computational perspectives (perspectives users develop for computation, themselves and the world around them). Brennan and Resnick's computational thinking framework enables the researcher to monitor the coding activity and to understand how children use the different constructs and deal with the concepts, how they focus on learning and adopt different thinking practices, and, finally, how their perspectives evolve in relation to themselves, others and the technological world. In our constructionism-based coding activity using Scratch, these three dimensions were utilized to explore and gain insights into children's experience of coding.

2.4. Selectivity theory and gender schema theory

In terms of information processing in task-related circumstances, two theories have been used to shed light on gender differences, selectivity theory and gender schema theory. First, selectivity theory aims to explain gender perceptual differences (Meyers-Levy, 1986; Meyers-Levy & Maheswaran, 1991; Meyers-Levy & Sternthal, 1991). This theory implies that gender perceptual differences rely on how males' and females' brains function. According to this theory, males rely on their right hemisphere, which indicates a "selective" way of processing information. More specifically, selective processing shows a more heuristic approach, focusing on the most prominent signs and visual representations instead of the details, which requires less cognitive effort (Meyers-Levy, 1989). In contrast, females are more likely to employ their left hemisphere, which results in what is named "comprehensive processing" (Goodrich, 2014). Females' approach shows their tension in a detailed analysis of all available information related to the specific task. Females assess and examine all the factors in a given task, involving themselves in more extensive elaboration compared to men.

Second, gender schema theory argues that there are gender differences in the way males and females use schema for cognitive processing (Martin, Ruble, & Szkrybalo, 2002). According to this theory, a schema relates to cognitive structures that we apply to form our perceptions, and this differs depending on gender. Males' schema associates with success, having as a result an attitude more strongly related to risk-taking, ambition and competition than that of females (Noble, Griffith, & Adjei, 2006). On the other hand, females focus more than males on collective actions and tend to care about relationships, sharing information frequently (Putrevu, 2001).

Previous empirical studies have used the theories mentioned above to examine gender differences in contexts of information processing. For example, in a study on the use of websites in e-commerce, Simon (2001) found that males' and females' preferences differ. Regarding the use of programming environments in the industry, Burnett et al. (2010) revealed significant gender differences in using and exploring software features and in users' confidence. In addition, other studies have used

[1] While recognizing the emotions from the facial images, it is necessary to look at the specific points on the face, such as eyebrows, shape of lips and the extension of the eye-opening. The gaze shifts from these features are called interfeatural saccades.

eye-tracking data to examine gender differences. Hwang and Lee (2018) found that gender differences exist in terms of visual attention and attitudes towards the presented products in online shopping environments. Exploring females' and males' characteristics of identifier style in source code reading, Sharafi, Soh, Guéhéneuc, and Antoniol (2012) presented mixed results, with no significant differences in accuracy, time and effort, but gender differences in strategies used in males' and females' approaches.

In our study, we aimed to investigate gender differences in coding activities for children. Given our coding task, children's activity required cognitive processes to successfully complete their goal, using Scratch and social interaction to collaboratively create a game. Therefore, we assumed that there are gender differences in how girls and boys behave during coding activities. In order to investigate the impact of gender, we used eye-tracking measures to generate objective quantitative data and qualitative data to examine different aspects of gender characteristics.

3. Methodology

In this section, we present the details of the two experiments, data collection, variables and analysis. We ran two studies, one in autumn 2016 and one in autumn 2017. For both experiments, the duration of the workshop for all groups of students was the same, as an out-of-school one-day activity. The implementation of the coding workshop was an intervention over two years, with few differences in design of the activity; the main differences were in the research design, the description of evidence and the results from different instruments used for data collection.

3.1. Description of coding activity

Based on the constructionist approach and its main principle, learning by doing (Resnick et al., 2009), we designed and implemented a coding workshop at the Norwegian University of Science and Technology (NTNU), in Trondheim, Norway. Our coding workshop was an out-of-school activity, in which children, novices to coding, from 8 to 17 years old interacted with digital robots, using Scratch for Arduino (S4A), and then coded their own game using the Scratch programming language. At each workshop, the children worked for approximately 4 h. Five assistants with previous experience in similar activities were responsible for instruction and the workshop procedure. The workshop consisted of two main parts: interaction with the robots, and creating games with Scratch.

Interaction with the robots: During the first part of the coding activity, the children interacted with digital robots built by an artist using materials recycled mainly from computer parts. First, once the children had entered the room and been welcomed by the assistants, they sat in teams next to one robot per team. The assistants gave a brief presentation of the workshop's activities and asked each of the children to pay attention to a worksheet placed on the desks next to them. The goal was to familiarize themselves with the robots by filling in simple questions regarding the exact place and number of the sensors and lights on the robots. Next, the children used a paper tutorial with instructions (Fig. 1a) on how to make the robots (Fig. 1b) react to the physical environment with visual effects using simple loops of Scratch for Arduino (e.g., to make the tongue of the snake robot move when there is less light at a sensor). The teams worked collaboratively and independently to complete this task (Fig. 1c). The duration of the first part varied between 45 min and one and a half hours. When all teams had finished, the children had a break before the next section began.

Creating games with Scratch: This section was the main activity of the workshop and lasted approximately 3 h, without the presence of the robots. The goal was to successfully develop a simple game coded in Scratch. To achieve this goal, the assistants gave another paper tutorial with examples of all the basic computer science concepts and possible loops the children should use to complete their own game. Assistants advised the children how to manage the process of game development by working collaboratively. They were advised that, first, they should think and decide the story for their game and then create a draft storyboard. When they had finished, they started coding using Scratch. Throughout the activity, children could ask for support from the assistants whenever they needed it. The assistants offered their guidance to the teams to help them complete their games by introducing even more complex computer science concepts when needed. Finally, after completion of the games, the children reflected on and played each other's games (Fig. 1d).

At each of the workshops, the same parts were conducted, with children participating once after being carefully selected according to their age; participants were of the same grade or within a small age range. The design of the coding activity provided flexibility, and the workshop instructors had the appropriate experience to be able to assist the children properly, taking into account the children's age. In the first part of the coding activity (interaction with robots), the children needed to perform simple tasks, but these were still things they had not done before since the robots and their functionalities were specially designed for our workshop. The design of the activity took into consideration the amount of time and support needed to complete the tasks, as these were likely to differ depending on the group of children. Creating games using Scratch was the second and main section of the coding activity and allowed the possibility for each of the teams to create a functional game using the basics or, depending on the team's capacities, to create a game with more advanced features. Teams of children worked independently with help from the instructors, who had the knowledge and the experience to help with advanced concepts according to the teams' needs. Scratch was used as a programming environment for the coding activity since it does not require any special expertise and children of all ages can quickly learn the basics to start creating in an efficient way. Moreover, Scratch is simple enough for novices and young users, while at the same time having enough power and functional variety to keep users engaged.

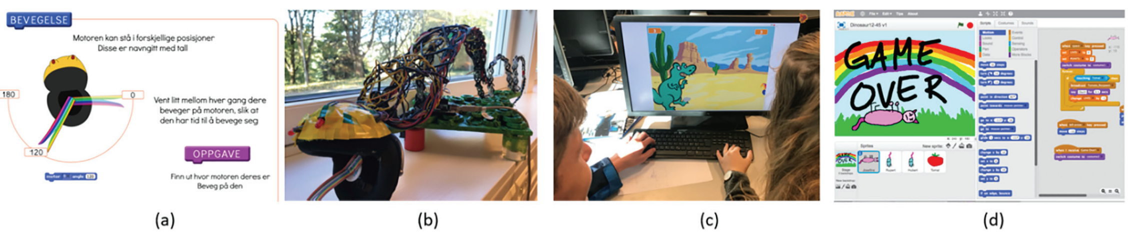

Fig. 1. (a) Example of the paper tutorial; (b) a snake-shaped interactive robot; (c) children collaborating on the creation of their game; (d) example of a created game.

3.2. Sampling

All the participants of the two studies were K-12 students from the mid-Norway region. Both studies took place at the university campus in specially designed rooms. Data related to both studies were collected after permission from the Norwegian Centre for Research Data (NSD), following all the regulations and recommendations for research with children. The children volunteered their participation in the eye-tracking study and the interviews. A researcher contacted the teacher and the legal guardian of each child to get a written consent form that gave permission for the data collection.

3.2.1. Participants: study one

The study lasted two weeks during autumn 2016, with 44 children from the third to 12th grades (aged 8–17 years old), 12 girls (mean age: 12.64; standard deviation [*SD*]: 2.838) and 32 boys (mean age: 12.35; *SD*: 2.773). Five workshops were held in total, all following the same process for the coding activity and designed for novices in coding. Some of the participants in the sample (13–17 years old) were recruited from the local schools that had applied to take part in our activity. The other set of participants (8–12 years old) were youngsters who attended local coding clubs as an after-school activity.

3.2.2. Participants: study two

In autumn 2017, children from eighth to tenth grade (age 13–16 years old) participated in the coding activity. The sample consisted of 105 participants in total, 69 boys and 36 girls (mean age: 14.55; *SD*: 0.650). At the end of each workshop, some of the participants were interviewed. In total, 44 children were interviewed, 23 boys and 21 girls.

3.3. Data collection and analysis

3.3.1. Data collection and analysis for the first study

Eye-tracking measures: As mentioned above, this study is one of the few so far to utilize children's gaze. We recorded children's gaze while they were coding using the Scratch environment during both parts of the activity. The eye-tracking data were collected using four SMI and one Tobii eye-tracking glasses. The sampling rate for all the eye-tracking glasses was set to be 30 Hz for the binocular eye-tracking. The average accuracy for both SMI and Tobii glasses was 0.5° at a distance of 40 cm. The visual field was divided into six areas of interest (AOI), five of which were in the Scratch interface, with the sixth in the physical robot area, as shown in Fig. 2.

From the eye-tracking data, we calculated the following measurements:

Average fixation duration: High fixation duration indicates that the participant is having difficulty in extracting information (Just & Carpenter, 1976). We used a mental rotation task, with angles of 0°, 120° and 180°, to study the relation between problem difficulty and gaze patterns. The results showed that with an increase in the rotation angle (increasing difficulty), the fixation duration at the center of the figure and the arms of the structures increased.

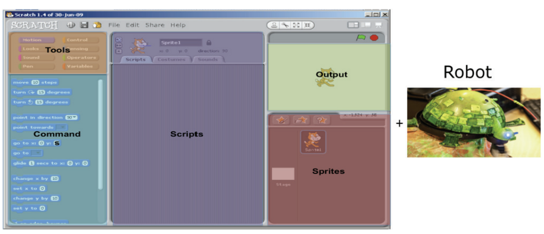

Fig. 2. The five areas of interests (AOI) for the screen; the sixth AOI was the robot.

Average change in saccade direction: Longer saccades show meaningful transitions in attention (Goldberg, Stimson, Lewenstein, Scott, & Wichansky, 2002). In a web search task, the authors used a set of different tasks on a webpage, so that the participants had to look for particular information to complete the tasks. The results showed that pre-planned eye movements were accompanied by longer saccades (Goldberg et al., 2002).

Saccade amplitude: The angle between two lines, if more than 90°, reflects a change of plans, revision or a failed expectation/hypothesis/anticipation (Cowen, Ball, & Delin, 2002). In a usability study, the authors found that the change in saccade direction often depicted the gaze behavior of not finding something which the participants anticipated to find at certain places (Cowen et al., 2002). This can be translated, in terms of programming behavior, as having a certain hypothesis and a failed verification.

Gaze uniformity: This is an individual measurement of engagement. This measure captured the uniformity of the time spent on all AOIs. The distribution was computed as a vector of length six (there are six AOIs) comprising the proportion of time spent on each AOI. The uniformity was computed as the inverse of a Kullback-Leibler divergence between the original proportionality vector and a uniform distribution with the same minimum and maximum limits as the original vector.

Time spent on each AOI: We divided the whole visual field into six AOIs – five on the screen and the sixth as the robot. We used specially made QR codes to scan the robots and the area around them (Fig. 2). The five AOIs on the screen were as follows:

- **Tools:** This area of the screen contained a general categorization of the commands available; for example, commands to control the motion, looks, sound, and other variables.
- **Command:** This area contained all available commands within the currently selected tools.
- **Scripts:** This was the area of the screen in which the coding task was performed.
- **Output:** This area showed participants the output of their scripts.
- **Sprites:** This area controlled the aesthetics of the program. The participants could change the appearance of the animated character using the characters available in this part of the interface.

Transitions among AOIs: We also computed the transitions to and from one AOI to another. This helped us to understand the temporal relationship between the children's gaze patterns and to depict the coding process used by the participants. For example, frequent transitions between script and output, or script and robot, depict the typical behavior of hypothesis verification. The participants made a small change in the program based on a certain hypothesis about the output or the robot's movement; once they had observed the output/robot's behavior, either their hypothesis was confirmed and they moved onto the next step in coding, or they modified the program to reverify their hypothesis. This behavior would result in a high number of transitions between the script and output/robot. We considered only three types of transitions for this analysis based on the literature which says that experts shift their attention between the code and the output more than novices do (Hejmady & Narayanan, 2012). This is why we chose to compare the gaze transitions between the script and robot/output. The third type of transition we included in our analysis was that between the commands and the script areas. These transitions imply a behavior that shows a thinking process of “what comes next in the code?” by the children.

Relative learning gain (RLG): The children completed pre- and post-knowledge acquisition tests. These consisted of nine coding questions of increasing difficulty. The questions were adapted from a previous study (Grover, Cooper, & Pea, 2014) and followed instructors' suggestions. The children took approximately 10 min to finish the tests.

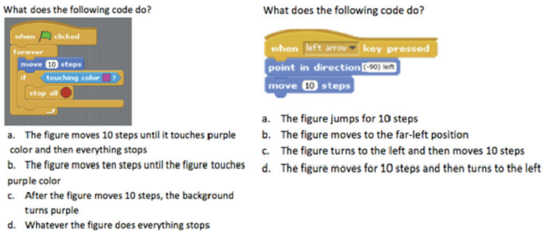

Fig. 3. Example of questions on the knowledge acquisition test.

The tests were paper-based and were manually graded by the researcher. Fig. 3 shows two sample questions from the test.

In our study, we calculated the RLG as defined by Sangin, Molinari, Nüssli, and Dillenbourg (2008). RLG is more accurate compared to learning gain since it takes into consideration the difficulty in gaining more knowledge if the learner is already very knowledgeable in a subject.

$$RLG = \begin{cases} \frac{Posttest - Pretest}{Max \cdot in\ Pretest - Pretest}, & if\ Posttest \geq Pretest \\ \frac{Posttest - Pretest}{Pretest}, & if\ Posttest < Pretest \end{cases}$$

Data analysis: To identify the relationship between gender, gaze, and RLG, we used analysis of variance (ANOVA) to compare the variables across different categories, since all the variables were normally distributed. In addition, we checked the assumptions for ANOVA, and if we found variables that did not satisfy the homoscedasticity condition, a version of ANOVA was used in which homoscedasticity is not assumed. This was done using the Welch correction for *F*-statistic.

3.3.2. Data collection and analysis for the second study

In the second study we utilized a qualitative approach. We collected data from multiple sources, including post-workshop interviews, observational field notes, and participants' Scratch games. All data were compared and cross-checked for triangulation.

Interviews: Participants were interviewed individually in Norwegian after the end of the workshop. The interviews were audio recorded, lasted approximately 10 min, and used a semi-structured protocol. During the interviews, students were asked to discuss their workshop experience, such as what they found to be the easiest, the most difficult and the most frustrating parts of creating the artifact, how they found their team collaboration, what they liked, and how they found their interaction with Scratch.

Interviews were transcribed and translated after the end of the workshop. To analyze the transcribed interviews, two researchers followed the coding method proposed by Saldaña (2015) for qualitative inquiry. Saldaña's coding method describes a cyclical model that moves from codes to categories and themes. Analysis of the semi-structured interviews focused on identifying categories and then the overall themes forming the codes emerged from participants' answers. Each transcript was first individually reviewed by two researchers and then, after a focus group and discussion, the two researchers agreed on the major themes that had emerged. In all codes under each category and then theme, it was indicated whether it derived from a boy's or a girl's interview. This helped us to detect gender differences in the already created themes. Analyzing the interviews allowed us to provide a holistic understanding of girls' and boys' perspectives on coding activities and to identify any potential masculine characteristics as well as girls' hesitation or stereotypes related to their participation in coding activities.

Observations and artifacts: Independent assistants during the workshop kept field notes. Assistants were close to each of the teams and took notes on all tasks. In order to identify what type of help participants were receiving from the assistants (see Table 1), we used Franklin, Conard, Boe, Nilsen, Hill, Len, & Kiefer (2013) coding scheme. In addition, observation notes included other incidents involving the participants that occurred during the workshop and which concerned the process they followed to successfully complete the coding tasks. Examples of such incidents were: how the participants distributed the roles among the team; which aspects they spent most of their time on while coding; and what their reactions were.

Table 1
Observations' coding scheme.

Number	Explanation
0	**Validation:** Students want confirmation, not information
1	**Where:** Only needed help navigating the Scratch GUI
2	**What:** Only needed a reminder of the name of the concept
3	**How:** Given name of concept, still needed help to complete task
4	**Reteach:** Had to reteach concept and execution

Observation field notes helped us monitor girls' and boys' practices during the process, capturing their behavior in all aspects, as well as the type and frequency of help received from the teaching assistants running the workshop.

For the purpose of this study, and to be able to investigate any potential differences between girls' and boys' approaches, we randomly analyzed observations of two teams of girls and two teams of boys (whose members were also interviewed), together with the final games created by these teams. Each of the observation notes (one set for each team) was reviewed by two researchers. Using content analysis, the main actions indicating a specific behavior were identified, and the frequencies of help level were calculated.

Finally, artifacts (games) developed by the teams were evaluated in terms of the learning opportunities related to computer science and computational thinking concepts offered by coding a game. We collected four versions of the games approximately every hour during the workshop. The evaluation of the artifacts included loading and playing the game to ensure its functionality and playability. For the analysis of each version of the games, we analyzed the games based on computational thinking components (i.e., flow control, data representation, abstraction, user interactivity, parallelism, and logic), giving a score for each of them from 0 to 3 (a rubric in which 3 shows proficiency, and 0 means that the skill is not evident). Artifacts were used as an extra source to determine the main characteristics (such as the game's theme, aesthetics, and storytelling) of boys' and girls' codes and their use of specific concepts related to the learning objectives of our workshop, as well as to discover any unexpected learning outcomes.

The analysis of observations and artifacts created by the teams focused on exploring any potential specifications that underline gender differences.

4. Research findings

4.1. Results from the first study

4.1.1. Gender and RLG

To investigate any potential gender difference in the RLG, an ANOVA (without assuming equal variances) with the RLG as the dependent variable and gender as the independent variable was used. The results showed no significant difference between boys and girls, $F(1,18.05) = 0.18$, $p = 0.65$ (see Fig. 4).

4.1.2. Gaze measures and gender

Next, in order to examine any potentially significant differences between the gaze measures of girls and boys, we utilized a one-way ANOVA (without assuming equal variance across gender). Table 2 shows the results, which indicate that there is no significant difference in the gaze behavior between girls and boys.

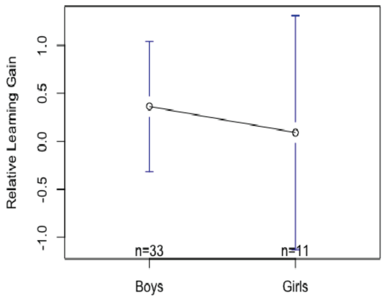

Fig. 4. RLG for boys and girls; the blue bars represent the 95% confidence interval.

Table 2
Testing the effect of gender on gaze behavior using analysis of variance (ANOVA).

	Mean (*SD*) girls	Mean (*SD*) boys	DoF 2	*F*-value	*P*-value
Uniformity	0.56 (0.25)	0.53 (0.25)	17.26	0.13	.72
Fixation duration	279.80 (55.94)	267.85 (100.10)	31.52	0.24	.62
Saccade direction	39.94 (12.33)	39.36 (17.25)	24.09	0.25	.61
Saccade amplitude	170.43 (49.27)	192.23 (64.26)	22.31	1.37	.25
Tools	0.13 (0.09)	0.16 (0.08)	15.10	0.61	.44
Script	0.14 (0.08)	0.16 (0.08)	17.69	0.17	.68
Commands	0.15 (0.06)	0.15 (0.06)	16.51	0.16	.69
Sprites	0.15 (0.11)	0.19 (0.07)	12.65	0.87	.36
Output	0.15 (0.08)	0.15 (0.07)	14.83	0.01	.90
Robots	0.16 (0.06)	0.16 (0.06)	14.83	0.01	.99
Scripts.command	0.23 (0.24)	0.23 (0.26)	18.32	0.32	.57
Scripts.output	0.24 (0.27)	0.21 (0.27)	16.81	1.39	.25
Scripts.robot	0.26 (0.22)	0.20 (0.22)	17.01	0.23	.63

4.2. Results from the second study

4.2.1. Interview results

This section describes the identified themes relating to children's perceptions that are relevant to our research question.

Improved confidence and self-efficacy in coding: In all interviews, children expressed that they managed to accomplish the tasks required. This was also evident from the fact that all teams managed to have a complete and functional game. Some of the comments, belonging in that category, are a clear indication of achievement expressed both by boys and girls. For example:

"*At the end we managed to do everything we wanted" – Thomas*

"We tried and made all the things we wanted to happen, we found how things worked" – Maria

Other comments indicate confidence in the game design and coding:

"Now, I know it is not so difficult to make a game, I can do it again" – Anna

"I knew something about coding before, but I didn't know how easy it was, I thought it was much harder to make a game, I can definitely do it" – Arne

In addition, it is evident that only girls reported that they did not know what they could do with Scratch or what coding is. Comments were similar to the one below:

"I was not at all looking forward to coming to the workshop, I thought it would be some geek stuff, I have never tried something similar" – Ingrid

Perceptions about leadership and collaboration: This theme reveals how boys and girls faced the collaboration process and how much they contributed to their team. In mixed teams, when a boy knew about coding, girls stated that a boy had to be the leader, while in girls' teams they appear to have had equal roles:

"We were lucky to have a boy in our team who has coded before, so he was leading in Scratch and the game creation" – Sonia

"We let Marius lead the team as he was more capable than us in coding" – Olga

"We distributed the roles equally and changed the rotation of control in different tasks" – Cecile

In the interviews, all boys indicated that they contributed to their teams in terms of coding, whereas girls mentioned that not all of them coded but that they felt a valuable part of the team because of their ideas:

"I didn't do much in the coding, but if it wasn't for me, they would have done a very boring game" – Jane

"I was not the one responsible for coding, but I decided how things will look or behave" – Katia

Interaction with Scratch: There were no differences among boys and girls in relation to their experience with Scratch and coding per se. Comments below show that both girls and boys had similar difficulties, challenges and frustrations during the creation of their game.

- General impression of Scratch programming environment:

"You can put together blocks and make a big script" – Daria

"If you put weird things together properly you can actually do whatever you want" – Marius

During the workshop the participants faced difficulties completing their projects. Many aspects of game design and coding appeared to be challenging and sometimes frustrated the children in their attempts to finish their projects.

- Difficulties of coding with Scratch:

"Sometimes the code can become messy" – Bjorn

"We couldn't make something stop when it was touching something else" – Annete and Peter

"It was so difficult to make our timer and score counting" – Sofia and Kevin "Making the character move was quite difficult" -Ines

"When you wanted to add something new in the function, then you had to go back and check everything again" – Arne

"We had to test and fix our game again and again until the end" – Martha

This category indicates what was easy in the coding part with Scratch. Participants liked many parts of Scratch and their interaction with it.

- Easiness of coding with Scratch:

"It was easy to put the blocks together" – Sofia

"I could find easily what I was looking for" – Lukas

"I found it easy that I could make the character as I wanted and then make the platform he was standing on" – Stefano

Some of the children were also able to refer to specific commands in Scratch. For example:

"I could easily use the sensing in a loop, to change the color of the

platform" – Ines

Affective engagement state: Participants reported that they liked coding the game. Equally, girls and boys reported that they had fun during the workshop. One participant commented:

"It was fun to code with colorful blocks that look childish" – Daria

Also, many participants described the workshop activity simply as:

"It was so fun" – Anne, Marius, Cloe

It was interesting to discover that participants with previous experience with coding found the whole workshop experience pleasant, showing that having experience is not a limitation to attending that type of workshop. This was highlighted in the following comment:

"The workshop was more fun because I knew about coding" – Alex

Even though some will not try it again, their experience was quite fun and interesting:

"I will not code again at home, but I had a lot of fun and I liked making the game and playing all the different games at the end" – Singrid

Moreover, they expressed their enjoyment at having an experience outside school:

"We don't do these things at school, I am happy I tried something else" – Daria

4.2.2. Observations and artifacts-analysis results

Our analysis of the observations from girls-only and boys-only teams revealed a few differences between their practices and behavior during the workshop. The results showed that girls had a different approach than boys on how they were organizing their tasks. From the beginning of the task, girls assigned roles and split the responsibilities (i.e., co-operation/dividing labor). For example, girls started thinking and designing the game, and in one of the teams a girl who was very good at drawing started creating a storyboard for their game on paper. On the other hand, boys started immediately navigating in the Scratch interface, trying different commands for a while without having a concrete plan for their actions. The teaching assistants needed to ask the boys' teams to concentrate, think of an idea for their game and make a quick storyboard. Girls looked more at the paper tutorial, trying to find examples of code, whereas boys had the tutorial on the side and only after the assistant's prompting did they start to look at it. Both teams had the same reaction when a team member was not interested: they tried at least to give him/her a task. It was apparent that girls' teams discussed more the decisions that they should take, and all were involved at every stage of the game creation. In addition, they paid attention to all aspects of the game with equal consideration. Boys cared more about the "how to code" part and using the Scratch interface, and they were less interested in the ideas and the aesthetics (e.g., color, what the character would look like, background) of their games. In terms of help received from the assistants, all teams had approximately the same amount, between five and seven times. However, girls were more persistent than boys in trying on their own more before asking for help. Taking into consideration the type of help received each time, the only prominent difference was that boys wanted more confirmation of their actions in Scratch together with the approval of the assistants that they had created an interesting or funny game. Difficult parts that all teams needed a lot of help with were "how to make characters jump", and the insertion of variables, high-score or time.

Children's games evaluation showed that girls' nature was present in their games, as was clear from their use of female characters inspired by the famous "Barbie" or similar figures. Furthermore, girls' games had a simple "goal", like catching objects falling, whereas boys made games involving throwing a ball to a goalkeeper or shooting activities. In the case of the programming concepts and computational thinking components (i.e., flow control, data representation, abstraction, user interactivity, parallelism, and logic), the rubric scores were almost the same in all final versions (see Appendix). This indicates that boys and girls had a similar final performance. However, both teams of boys had better scores in the first and second versions of the games.

5. Discussion and conclusions

In this study our aim was to investigate gender differences in coding activities for children. To this end, we designed and evaluated a one-day coding workshop with participants aged from 8 to 17 years old. During the workshop, children were introduced to coding by interacting with digital robots, specially designed for the activity, and creating a game using Scratch. In all the activities, children worked collaboratively in teams to successfully complete their goal. For the evaluation, we used eye-tracking data as an objective measure, which to the best of our knowledge has not previously been used to capture children's coding gaze behavior. In addition, we collected and analyzed qualitative data (i.e., semi-structured interviews, and observations) to get a deeper understanding of children's experiences during the workshop. A qualitative approach is especially valuable for examining gender issues, since expressing opinions about gender can be vulnerable process (Popper, 1971). Our research findings reveal that gender issues in coding activities for children are a multifaceted phenomenon. According to the quantitative findings, there are no gender differences concerning RLG and gaze behavior in boys and girls. On the other hand, qualitative results from interviews, observations and the created games showed that some gender differences exist in children's approaches, as revealed by their behavior during the workshop and their perceptions.

There was no difference in the RLG between girls and boys. Therefore, children in our study showed no differences in their performances, which supports previous studies on children using other evaluation methods (Owston et al., 2009; Vos et al., 2011). Therefore, our findings provide more evidence that girls do not lack in competence compared to boys. Fisher, Cox, and Zhao (2006), in a study on adults' performance on a program comprehension task, found no gender differences and reported that men and women were equally capable of developing the skills required to be professional developers. Although more girls than boys in our interviews said that they had not known about coding before, or that they were afraid of it, they managed to be equally good as the boys. Moreover, the activities offered in our workshop were appropriate independently from the participants' gender and their previous knowledge. Furthermore, in the interviews, children reported that they had fun during the workshop, even though some of them had prior knowledge of coding. This can be attributed to the fact that Scratch is not limited: it provides many possibilities for making more advanced creations, so users can find it interesting and learn more, no matter their level of existing knowledge. In addition, the collaborative notion of the workshop enabled the students to learn from each other and not to have their own individual performance as their main goal. As shown in other studies, students perform better when working in pair programming than when working alone (Lye & Koh, 2014; Werner, Denner, Campe, & Kawamoto, 2012).

A noticeable result is that there is no difference in the gaze behavior of girls and boys. We used the objective measure of eye-tracking data, and by examining different measures we found no difference in any of them. This indicates that, regarding the actual micro-level experience of boys and girls during coding with Scratch, there is no difference in their approach based on their gaze, and hence also no difference in their cognitive processes (Eckstein, Guerra-Carrillo, Singley, & Bunge, 2016). From measures of time spent on different AOI, gaze uniformity, and transitions among the different AOI, results showed that both male and female participants were able to navigate in the Scratch interface, had a meaningful thinking process, and were engaged. Similarly, from the other measures used, results show that both genders had equivalent difficulties in extracting information (fixation duration), challenges in learning something (saccade direction change), and goals and expectations in coding (saccade amplitude). Cazzato et al. (2010) found weak gender differences in the gaze behavior of participants when trying to solve visuo-spatial problems, but that women used more cognitive resources. Other studies have found that girls face difficulties in coding when they had a lot of elements (Denner et al., 2012) or when they put more effort into having good functionality (Lee et al., 2014).

Although the results of children's gaze behavior and performance show that there are no important gender differences, the qualitative results of our study reveal that gender differences exist in the practices used by boys and girls and in their perceptions. In general, girls approached the coding activity in a different way to boys, verifying the theories of gender differences in information processing (Martin et al., 2002; Meyers-Levy, 1986). For example, girls were more organized in terms of collaboration, splitting the responsibilities and focusing on a more systematic approach in the tasks, and they also paid more attention to the tutorials. In addition, girls seemed to like more collaboration with others and to share the social part of the activity. Previous studies have shown that female students have a more trusting and sociable approach compared to male students, who are more independent and focused on caring about themselves (Rosenberg-Kima, Plant, Doerr, & Baylor, 2010). In the computer-supported collaborative learning environment, Bruckman et al. (2002) found that girls spent more time than boys in communicating. Girls' games were richer in aesthetics and graphical representation, and they also had a more "girly" approach. This is similar to other findings that show girls spend more time on dialogues (Robertson, 2012) and aesthetics. Similar to the finding of Denner and Werner (2007), our study shows that girls' teams were more persistent in attempting the tasks on their own before asking for help. Whereas girls' games had simpler tasks (like catching falling objects), boys' games had more competitive characteristics. This observation is similar to the finding of Owston et al.'s (2009) study, in which teachers reported that boys enjoyed playing games more competitively against others. Our observation notes confirm this finding, as boys were also asking the assistants about how interesting their games were.

One of the goals of our workshop was to build children's belief that coding is something that they can do, and that it is not something that only boys would be interested in. After their participation, boys and girls reported that they felt competent to code. Another interesting result from our qualitative study is that regardless of both girls and boys reporting improved confidence and self-achievement, we find that girls have less self-efficacy. One example is that when girls were among boys in the teams they chose a boy to be the leader, indicating less confidence. They also expressed that they did not know what coding was before, that they had not tried it, that they did not know whether they could do it, and that they thought it was only for geeks. The stereotype of boys being better than girls at robotics and coding exists from the young age of 6-years-old (Master, Cheryan, Moscatelli, & Meltzoff, 2017). A possible reason why they split the roles during their collaboration is that the girls were less confident; in addition, none of the girls was trying to take control. In solo programming, men have been found to be more confident than women (McDowell, Werner, Bullock, & Fernald, 2006). In the study of Beckwith et al. (2005), females' self-efficacy was lower than men's, and women did not easily accept new debugging features.

That there are no gender differences in children's actual performance and gaze behavior while coding, and that the main differences are in their practices, indicates that practitioners should focus on characteristics that will influence girls and change their limited participation in computer science. Our results show that educators should foster girls' self-efficacy (i.e., the belief in one's capacity to succeed in tasks [Bandura, 1997]) and make them believe that they do not lack in competences; therefore, educators should be careful to avoid discriminating behaviors. Qiu, Buechley, Baafi, and Dubow (2013) found that participants' confidence, enjoyment and interest in coding and technology increase when self-efficacy grows. According to Bandura (1997), self-efficacy is important in problem solving, since it affects the individual's cognitive strategies, effort, persistence and, consequently, the learning outcome. Coding activities should take into consideration special gender characteristics and facilitate appropriate workshops. Activities focused on collaboration can also be a method to narrow the gender gap in coding activities and to view partnership as a key factor for fostering both learning and positive attitudes in students.

5.1. Limitations

This study has some limitations. First, it was very challenging to collect eye-tracking data from children. Eye-tracking measurements with children is a very difficult task since it involves small eye pupils, difficulty with the calibration, and the need for equipment tailored to children, so this caused some problems. The large size of the glasses annoyed the participants, especially the younger ones, who, as a result, had to take them off. Consequently, we had to stop the activity to calibrate them again. The young age of the participants combined with the playful environment of the workshop, in which children were very enthusiastic and also wanted to experiment with the glasses, led to us having data that we could not use because they were not from the areas of our interest.

Second, participants in the two studies conducted for this paper had slightly different ages, and we lacked participants aged 8–12 and 17. Third, we analyzed only a small number of observations from the teams and games. Including a larger amount of data could have added more value to our results. In addition, the specific design and context of the activity (i.e., the use of the Scratch tool, the coding tasks, the duration, and the other characteristics), as well as the sampling method used, limits the generalization of our findings. More precisely, the participants in our study were randomly selected volunteers from our region in Norway; other sampling methods and demographic variables might have a different effect on children's experience.

5.2. Future work

Our study suggests new aspects as the subject for follow-on research. One possibility would be to investigate in more detail specific gaze patterns of boys and girls; another would be to examine collaborative eye-tracking measures and group dynamics in both mixed and non-mixed teams of boys and girls. Another interesting approach would be to compare the effect of different learning environments on gender. Furthermore, other objective measures could be used to gain a deep understanding of the relationship between coding behavior and gender. In addition, other quantitative methods, such as surveys, could be used to supplement the collection of data relating to children's perceptions of coding.

Acknowledgments

The authors would like to express their gratitude to all of the children, teachers and parents for volunteering their time. Our very special thanks go to Letizia Jaccheri, Kristin Susanne Karlsen, Ioannis Leftheriotis, Uyen Dan Nguyen, Amanda Jørgine Haug, Lidia Luque Fernandez, Marjeris Sofia Romero, An Nguyen, Ton Mangmee, Eline Stenwig and Kristoffer Venæs Monsen.

The project has been recommended by the Data Protection Official for Research, Norwegian Social Science Data Services (NSD), following all the regulations and recommendations for research with children.

This work supported from the "Learning science the fun and creative way: coding, making, and play as vehicles for informal science learning in the 21st century" Project, under the European Commission's Horizon 2020 SwafS-11-2017 Program (Project Number: 787476). This article reflects the views only of the authors and it does not represent the opinion of neither the European Commission nor NTNU, and the

European Commission and NTNU can not be held responsible for any use that might be made of its content. This work is also supported from the Norwegian Research Council under the projects FUTURE LEARNING (number: 255129/H20) and Xdesign (290994/F20), and by NOKUT under the Centre for Excellent IT Education (Excited) (number: 16/02049).

Appendix

The rubric used for the evaluation of the artifacts:

Computational thinking components:	3 (excellent)	2 (Good)	1(Satisfactory)	0 (Not evident)
Flow control (loops and sequence)				
Data representation (variables)				
Abstraction				
User interactivity (events)				
Parallelism				
Logic (conditionals and operators)				

Are there any specific characteristics?

- Theme of the game
•Aesthetics
•Storytelling
•Other comments (if any)

Artifact's rubrics scores for girls' and boys' teams.

	Computational thinking components						
Team and game version:	Flow control	Data representation	Abstraction	User interactivity	Parallelism	Logic	Total score
GT1V1	1	1	0	1	0	0	3
GT1V2	1	2	0	1	1	1	6
GT1V3	2	2	1	1	1	2	9
GT1V4	2	2	1	1	3	2	11
GT2V1	1	0	0	1	0	1	3
GT2V2	1	1	0	1	1	1	5
GT2V3	2	2	1	2	1	1	9
GT2V4	3	2	1	2	2	2	12
BT1V1	1	0	0	2	0	2	5
BT1V2	2	1	1	2	1	2	9
BT1V3	2	2	1	2	1	2	10
BT1V4	2	2	1	2	2	2	11
BT2V1	1	1	1	1	0	1	5
BT2V2	2	1	1	2	1	1	8
BT2V3	2	1	1	2	1	2	9
BT2V4	2	2	1	2	2	3	12

*GT1: girls' team number 1; *GT2: girls' team number 2; *V1: game version 1; *V2: game version 2; *V3: game version 3; *V4: game version 4; *BT1: boys' team number 1; * BT2: boys' team number 2.

References

Adya, M., & Kaiser, K. M. (2005). Early determinants of women in the IT workforce: A model of girls' career choices. *Information Technology and People, 18*(3), 230–259.
Alexander, G. M. (2006). Associations among gender-linked toy preferences, spatial ability, and digit ratio: Evidence from eye-tracking analysis. *Archives of Sexual Behavior, 35*(6), 699–709.
Bandura, A. (1997). *Self-efficacy: The exercise of control.* Macmillan.
Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48–54.
Beckwith, L., Burnett, M., Wiedenbeck, S., Cook, C., Sorte, S., & Hastings, M. (2005). Effectiveness of end-user debugging software features: Are there gender issues? *Paper presented at the proceedings of the SIGCHI conference on human factors in computing systems*.
Brennan, K., & Resnick, M. (2012). New frameworks for studying society computer science exemplification project. *Paper presented at AERA*.
Bruckman, A., Jensen, C., & DeBonte, A. (2002). Gender and programming achievement in a CSCL environment. *Paper presented at the Proceedings of the conference on computer support for collaborative learning: Foundations for a CSCL community*.
Buechley, L., Eisenberg, M., Catchen, J., & Crockett, A. (2008). The LilyPad Arduino: Using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. *Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems*.
Burnett, M., Fleming, S. D., Iqbal, S., Venolia, G., Rajaram, V., Farooq, U., & Czerwinski, M. (2010). Gender differences and programming environments: Across programming populations. *Paper presented at the Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement*.
Çakır, N. A., Gass, A., Foster, A., & Lee, F. J. (2017). Development of a game-design workshop to promote young girls' interest towards computing through identity exploration. *Computers & Education, 108*, 115–130.
Cazzato, V., Basso, D., Cutini, S., & Bisiacchi, P. (2010). Gender differences in visuospatial planning: An eye movements study. *Behavioural Brain Research, 206*(2), 177–183.
Ceci, S. J., & Williams, W. M. (2010). Sex differences in math-intensive fields. *Current Directions in Psychological Science, 19*(5), 275–279.
Cheryan, S., Ziegler, S. A., Montoya, A. K., & Jiang, L. (2017). Why are some STEM fields more gender balanced than others? *Psychological Bulletin, 143*(1), 1.
Cowen, L., Ball, L. J., & Delin, J. (2002). An eye movement analysis of web page usability. *People and computers XVI-memorable yet invisible* (pp. 317–335). Springer.
Denner, J. (2007). The Girls Creating Games Program: An innovative approach to integrating technology into middle school. *Meridian: A Middle School Computer Technologies Journal, 1*(10).
Denner, J., & Werner, L. (2007). Computer programming in middle school: How pairs respond to challenges. *Journal of Educational Computing Research, 37*(2), 131–150.
Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education, 58*(1), 240–249. https://doi.org/10.1016/j.compedu.2011.08.006.

Djamasbi, S., Tullis, T., Hsu, J., Mazuera, E., Osberg, K., & Bosch, J. (2007). Gender preferences in web design: Usability testing through eye tracking. *AMCIS 2007 Proceedings, 133*.
Eckstein, M. K., Guerra-Carrillo, B., Singley, A. T. M., & Bunge, S. A. (2016). Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development? *Developmental Cognitive Neuroscience, 25*, 69–91.
Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education, 63*, 87–97. https://doi.org/10.1016/j.compedu.2012.11.016.
Fisher, M., Cox, A., & Zhao, L. (2006). Using sex differences to link spatial cognition and program comprehension. *Paper presented at the software maintenance, 2006. ICSM'06. 22nd IEEE international conference*.
Franklin, D., Conrad, P., Boe, B., Nilsen, K., Hill, C., Len, M., ... Kiefer, B. (2013). Assessment of computer science learning in a scratch-based outreach program. *Paper presented at the Proceeding of the 44th ACM technical symposium on Computer science education*.
Goldberg, J. H., Stimson, M. J., Lewenstein, M., Scott, N., & Wichansky, A. M. (2002). Eye tracking in web search tasks: Design implications. *Paper presented at the Proceedings of the 2002 symposium on Eye tracking research & applications*.
Goodrich, K. (2014). The gender gap: Brain-processing differences between the sexes shape attitudes about online advertising. *Journal of Advertising Research, 54*(1), 32–43.
Grover, S., Cooper, S., & Pea, R. (2014). Assessing computational learning in K-12. *Paper presented at the Proceedings of the 2014 conference on Innovation & technology in computer science education*.
Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM, 51*(8), 25–27.
Guzdial, M. (2017). Balancing teaching CS efficiently with motivating students. *Communications of the ACM, 60*(6), 10–11.
Hejmady, P., & Narayanan, N. H. (2012). Visual attention patterns during program debugging with an IDE. *Paper presented at the proceedings of the symposium on eye tracking research and applications*.
Hewig, J., Trippe, R. H., Hecht, H., Straube, T., & Miltner, W. H. (2008). Gender differences for specific body regions when looking at men and women. *Journal of Nonverbal Behavior, 32*(2), 67–78.
Horizon, M. (October 5, 2015). *Horizon Media study reveals Americans prioritize STEM subjects over the arts; science is "cool," coding is new literacy.* PR Newswire. Retrieved from, http://www.prnewswire.com/news-releases/horizon-mediastudy-reveals-americans-prioritize-stem-subjects-over-the-arts-science-is-cool-coding-is-new-literacy-300154137.html.
Hubwieser, P., Armoni, M., Giannakos, M. N., & Mittermeir, R. T. (2014). Perspectives and visions of computer science education in primary and secondary (K-12) Schools. *ACM Transactions on Computing Education (TOCE), 14*(2), 7.
Hwang, Y. M., & Lee, K. C. (2018). Using an eye-tracking approach to explore gender differences in visual attention and shopping attitudes in an online shopping environment. *International Journal of Human-Computer Interaction, 34*(1), 15–24.
Just, M. A., & Carpenter, P. A. (1976). Eye fixations and cognitive processes. *Cognitive Psychology, 8*(4), 441–480.
Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and Culture, 1*(1), 36–40.
Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational Psychologist, 50*(4), 313–334. https://doi.org/10.1080/00461520.2015.1124022.
Kafai, Y. B., & Vasudevan, V. (2015). Constructionist gaming beyond the screen: Middle school students' crafting and computing of touchpads, board games, and controllers. *Paper presented at the proceedings of the workshop in primary and secondary computing education*.
Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior, 52*, 200–210.
Kelleher, C., Pausch, R., & Kiesler, S. (2007). Storytelling alice motivates middle school girls to learn computer programming. *Proceedings of the SIGCHI conference on Human factors in computing system* (pp. 1455–1464). .
Lee, E., Kafai, Y. B., Vasudevan, V., & Davis, R. L. (2014). Playing in the arcade: Designing tangible interfaces with MaKey MaKey for scratch games. *Playful user interfaces* (pp. 277–292). Springer.
Lorigo, L., Haridasan, M., Brynjarsdóttir, H., Xia, L., Joachims, T., Gay, G., & Pan, B. (2008). Eye tracking and online search: Lessons learned and challenges ahead. *Journal of the Association for Information Science and Technology, 59*(7), 1041–1052.
Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51–61.
Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with Scratch. *Proceedings of the 39th SIGCSE technical symposium on Computer science education ACM: Vol. 40*.
Martin, C. L., Ruble, D. N., & Szkrybalo, J. (2002). Cognitive theories of early gender development. *Psychological Bulletin, 128*(6), 903.
Master, A., Cheryan, S., Moscatelli, A., & Meltzoff, A. N. (2017). Programming experience promotes higher STEM motivation among first-grade girls. *Journal of Experimental Child Psychology, 160*, 92–106.
McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM, 49*(8), 90–95.
Meyers-Levy, J. (1986). *Gender differences in information processing: A selectivity interpretation.* Northwestern University.
Meyers-Levy, J. (1989). Priming effects on product judgments: A hemispheric interpretation. *Journal of Consumer Research, 16*(1), 76–86.
Meyers-Levy, J., & Maheswaran, D. (1991). Exploring differences in males' and females' processing strategies. *Journal of Consumer Research, 18*(1), 63–70.
Meyers-Levy, J., & Sternthal, B. (1991). Gender differences in the use of message cues and judgments. *Journal of Marketing Research,* 84–96.
Moss, G., & Colman, A. M. (2001). Choices and preferences: Experiments on gender differences. *Journal of Brand Management, 9*(2), 89–98.
Noble, S. M., Griffith, D. A., & Adjei, M. T. (2006). Drivers of local merchant loyalty: Understanding the influence of gender and shopping motives. *Journal of Retailing, 82*(3), 177–188.
NRC. (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas.* The National Academies Press.
Obaidellah, U., Al Haek, M., & Cheng, P. C.-H. (2018). A survey on the usage of eye-tracking in computer programming. *ACM Computing Surveys (CSUR), 51*(1), 5.
Owston, R., Wideman, H., Ronda, N. S., & Brown, C. (2009). Computer game development as a literacy activity. *Computers & Education, 53*(3), 977–989.
Pan, B., Hembrooke, H. A., Gay, G. K., Granka, L. A., Feusner, M. K., & Newman, J. K. (2004). The determinants of web page viewing behavior: An eye-tracking study. *Paper presented at the Proceedings of the 2004 symposium on Eye tracking research & applications*.
Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2017). Empirical studies on the maker movement, a promising approach to learning: A literature review. *Entertainment Computing, 18*, 57–78. https://doi.org/10.1016/j.entcom.2016.09.002.
Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2019). Exploring children's learning experience in constructionism-based coding activities through design-based research. *Computers in Human Behavior* (in press).
Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2018). How do you feel about learning to code? Investigating the effect of children's attitudes towards coding using eye-tracking. *International Journal of Child-Computer Interaction, 17*, 50–60.
Papavlasopoulou, S., Sharma, K., Giannakos, M., & Jaccheri, L. (2017). Using eye-tracking to unveil differences between kids and teens in coding activities. *Paper presented at the proceedings of the 2017 conference on interaction design and children*.
Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* Basic Books, Inc.
Popper, K. (1971). *The open society and its enemies* (5th ed.). Princeton: Princeton University Press.
Putrevu, S. (2001). Exploring the origins and information processing differences between men and women: Implications for advertisers. *Academy of Marketing Science Review, 2001*, 1.
Qiu, K., Buechley, L., Baafi, E., & Dubow, W. (2013). A curriculum for teaching computer science through computational textiles. *Paper presented at the proceedings of the 12th international conference on interaction design and children*.
Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., & Silverman, B. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60–67.
Robertson, J. (2012). Making games in the classroom: Benefits and gender concerns. *Computers & Education, 59*(2), 385–398. https://doi.org/10.1016/j.compedu.2011.12.020.
Robertson, J. (2013). The influence of a game-making project on male and female learners' attitudes to computing. *Computer Science Education, 23*(1), 58–83.
Robertson, J., & Howells, C. (2008). Computer game design: Opportunities for successful learning. *Computers & Education, 50*(2), 559–578.
Rosenberg-Kima, R. B., Plant, E. A., Doerr, C. E., & Baylor, A. L. (2010). The influence of computer-based model's race and gender on female students' attitudes and beliefs towards engineering. *Journal of Engineering Education, 99*(1), 35–44.
Saldaña, J. (2015). *The coding manual for qualitative researchers.* Sage.
Sangin, M., Molinari, G., Nüssli, M.-A., & Dillenbourg, P. (2008). How learners use awareness cues about their peer's knowledge?: Insights from synchronized eye-tracking data. *Paper presented at the Proceedings of the 8th international conference on International conference for the learning sciences: Vol. 2*.
Schmid, P. C., Mast, M. S., Bombari, D., Mast, F. W., & Lobmaier, J. S. (2011). How mood states affect information processing during facial emotion recognition: An eye tracking study. *Swiss Journal of Psychology, 70*(4), 223–231.
Schulte, C., & Knobelsdorf, M. (2007). Attitudes towards computer science-computing experiences as a starting point and barrier to computer science. *Paper presented at the Proceedings of the third international workshop on Computing education research*.
Searle, K. A., & Kafai, Y. B. (2015). Boys' needlework: Understanding gendered and indigenous perspectives on computing and crafting with electronic textiles. *Paper presented at the ICER*.
Sharafi, Z., Soh, Z., Guéhéneuc, Y.-G., & Antoniol, G. (2012). Women and men—different but equal: On the impact of identifier style on source code reading. *Paper presented at the IEEE 20th international conference on program comprehension (ICPC), 2012*.
Simon, S. J. (2001). The impact of culture and gender on web sites: An empirical study. *ACM SIGMIS - Data Base: The Database for Advances in Information Systems, 32*(1), 18–37.
Teague, J. (2002). Women in computing: What brings them to it, what keeps them in it? *ACM SIGCSE Bulletin, 34*(2), 147–158.
Vos, N., Van Der Meijden, H., & Denessen, E. (2011). Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education, 56*(1), 127–137.
Wang, M.-T., & Degol, J. L. (2017). Gender gap in science, technology, engineering, and mathematics (STEM): Current knowledge, implications for practice, policy, and future directions. *Educational Psychology Review, 29*(1), 119–140.
Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: Measuring computational thinking in middle school. *Paper presented at the proceedings of the 43rd ACM technical symposium on computer science education*.
Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Paper 7:

Coding games and robots to enhance computational thinking: How collaboration and engagement moderate children's attitudes?

Kshitij Sharma, Sofia Papavlasopoulou, and Michail N. Giannakos

International Journal of Child-Computer Interaction

International Journal of Child-Computer Interaction 21 (2019) 65–76

Contents lists available at ScienceDirect

International Journal of Child-Computer Interaction

journal homepage: www.elsevier.com/locate/ijcci

Coding games and robots to enhance computational thinking: How collaboration and engagement moderate children's attitudes?

Kshitij Sharma *, Sofia Papavlasopoulou, Michail Giannakos

Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway

ARTICLE INFO

Article history:
Received 5 November 2018
Received in revised form 8 March 2019
Accepted 16 April 2019
Available online 7 May 2019

Keywords:
Eye-tracking
Computational thinking
Coding activity
Programming
Informal learning
Attitudes
Children programming
Interaction design and children
Collaborative eye-tracking

ABSTRACT

Collaboration and engagement while coding are vital elements for children, yet very little is known about how children's engagement and collaboration impact their attitudes toward coding activities. The goal of the study is to investigate how collaboration and engagement moderate children's attitudes about coding activities. To do so, we designed an study with 44 children (between 8 and 17 years old) who participated in a full-day coding activity. We measured their engagement and collaboration during the activity by recording their gaze, and their attitudes in relation to their learning, enjoyment, team-work and intention by post-activity survey instruments. Our analysis shows that there is a significant moderating effect of collaboration and engagement on children's attitudes. In other words, highly engaging and collaborative coding activities significantly moderate children's attitudes. Our findings highlight the importance of designing highly collaborative and engaging coding activities for children and quantifies how those two elements moderate children's attitudes.

1. Introduction

Children's engagement during a learning activity, is considered the "holy grail of learning" [1]. It is associated with several important aspects of the design and implementation of contemporary learning activities and is clearly associated with students attitudes [2]. Engagement changes over time and is dependent on interventions, social interactions and changing contexts [3]. In collaborative learning activities, the level and quality of collaboration between young students has also been found to have direct influence in the quality of learning processes and persistence [4] as well as in improving students' attitudes (e.g., about mathematics) [5]. Thus, when investigating learning activities for children, it is important to look closely how collaboration and engagement might moderate the interplay of other important attitudes.

Computational thinking and coding activities for young students are becoming an integral part of contemporary informal learning in different contexts (e.g., in makerspaces, after school activities, museums, libraries etc.). It is evident that young students should begin developing computational thinking skills early [6], and thus, more and more organizations design and deliver coding activities, as part of their curriculum or their outreach program. Properly designed coding activities for children have shown to be beneficial, since they enhance problem solving skills, critical thinking and creativity among other [7]. The design of these activities is important to enhance collaboration and engagement in a meaningful way [6], yet very little is known about the role of collaboration and engagement and their connections with other attitudes that empower children's participation (e.g., positive attitudes like enjoyment, intention etc.).

Therefore, in this contribution we seek to investigate how collaboration and engagement moderate the relationship between central attitudes of children when coding (i.e., team-work, intention to participate, perceived learning and enjoyment).

To tackle the aforementioned proposition, we conducted a study with 44 children participating to a full day coding activity. We used eye-tracking techniques to measure their engagement and collaboration during the activity and post-activity surveys to measure their attitudes in relation to learning obtained, sense of enjoyment, team-work and intention to participate in a similar activity in the future. By investigating the role of collaboration and engagement we provide a quantified evidence of how those two important elements moderate other attitudes and enable various insights for the design of future coding activities. In particular, our paper makes the following contributions:

- We present insights from a study that collects data related to children's behaviour (eye-tracking) and attitudes (surveys) during a coding activity.
- We show that collaboration and engagement moderates the relationship between children's attitudes

* Corresponding author.
E-mail address: kshitij.sharma@ntnu.no (K. Sharma).

https://doi.org/10.1016/j.ijcci.2019.04.004

The remainder of the paper is organized as following. The second section presents the related work on investigating the relationship between attitudes and behaviour in primarily educational/organizational settings. Third section highlights the conceptual model and research hypotheses of our paper. The fourth section provides the methodology of the study, the coding activity, participants, variables used for the analysis and the analysis itself. The fifth section shows results from data analysis, and the last section discusses the implications of the results and concludes the paper with future work and limitations.

2. Related work

2.1. The importance of attitudes in learning activities

An important issue related to the success of coding activities is their adoption by children. A number of models and theories have been developed and utilized to understanding the relationships between the attitudes towards a new technologies and the experiences and outcomes of using the technology (e.g., UTAUT or its initial form Technology Acceptance Model-TAM, [8]). TAM is a model connecting the ease of use, intention to use, user behaviour and the usage outcomes (enjoyment, engagement, learning to name a few). Various studies have used this model as a basis for their analyses or extending the basic model given by Davis (1989) [8].

Attitudes have been central in educational research for several years. For instance, in an organizational learning context, humans' intention to use new technology was found to be positively correlated with their motivation to learning and transfer learning [9]. In another study, perceived enjoyment is another element that has been reported to be closely associated with intention. This association has been reported in studies concerning both the teachers [10] and young students [11]. In another study, pre-service teachers showed that the perceived enjoyment was positively associated with their intention to use new technology [10]. Finally, in the context of gaming it is found that the intention to play games had a positive significant correlation with the enjoyment in the games [12].

Enjoyment and learning are also associated, this has been proven through different studies in educational settings [13–15]. For instance, in a face-to-face class about data analysis where the teacher focused on the dialectic relation between theory and data, the students who enjoyed this method, believed that it helped them with their learning [13]. Similarly, based on the surveys in another face to face classroom setting, the results stipulated a positive correlation between enjoyment and learning performance [16]. The results from a survey about a web-based class management system, showed a positive correlation between enjoyment and the learning goal orientation [14]. In a reading study with eighth graders, the authors found the correlation between the enjoyment in reading text and the perception about learning to be significantly positive [15]. In a study based on PISA tests, the perceived enjoyment was positively correlated with the science knowledge, for students across different countries (USA, Columbia, Estonia, Sweden) [17].

One of the most intuitive relations, among the various constructs included in TAM, is between enjoyment and engagement when it comes to technology usage. These studies (mostly using survey data) were conducted at different educational levels, such as pre-university level [18], high school [19], primary and secondary levels [20,21]. Therefore, if an experience provides enjoyment to the participant, it is likely that it would also be engaging in long-term. For example, a study using PISA tests showed ($N > 400{,}000$, 57 countries) a positive correlation between activation enjoyment and engagement with learning science [17]. Considering high school students in different years (10–13 grade) the students showed a positively significant correlation between their enjoyment at and engagement with the school [19]. This correlation was also consistent across the different years. In a study with children aged between 7 and 8 years using educational games, the children who enjoyed the games also showed higher levels of engagement than the children with lower levels of enjoyment [21]. Among pre-university students, the results showed a negative correlation between disengagement and various constructs such as enjoyment at school and class participation [18]. In a study with tangible user interfaces involving children, the results showed a positive correlation between children's engagement with the tangible game and their perceived enjoyment [20]. Further, within a teacher–student laboratory paradigm [22] the students who reported high levels of enjoyment also reported high levels of engagement. To summarize, from the literature it is evident that attitudes are highly associated with the adoption of a learning activity by young students, as well as, the learning obtained from the activity.

2.2. Engagement and collaboration in learning

Many studies have reported a positive relation between collaborative learning and engagement [23–26]. In a collaborative learning scenario with clickers in the classroom, there was a significant positive association between engagement and active learning [23]. The proponents of Computer Supported Collaborative Learning (CSCL) argue that introducing technology to facilitate the collaboration might increase engagement with the learning activities and hence learning outcomes [24,26]. Jarvela and Jarvenoja [25] identified engagement as one of the key factors for the success of self-regulated learning. Kreijns and colleagues [27] argue that there might be two different ways in which mutual engagement and learning are related. First, because of mutual engagement individuals can gain knowledge that could not be done prior to the collaboration. Second, mutual engagement facilitates the co-creation of knowledge and hence leads to better individual learning outcomes.

Furthermore, Lipponen and colleagues [28] highlight the need of engagement in collaborative learning by stating that "just by putting two or more individuals together" one cannot foster collaborative learning, one should make the collaborative task active enough to engage the collaborators. Engagement has also been shown to be related with team work [29–32]. In a group writing study, there were negative effects of restricted communication over engagement of students within different groups [30]. A study with hockey players, showed that with positive attitude towards the team work, novice players showed more willingness to come to practice [29]. Similar results were reported in the context of basketball players [31]. In a study with educational robotics, the authors found a positive correlation between group work and engagement levels with robots [32].

2.3. Eye-tracking as a means to understand engagement and collaboration

Eye-tracking provides a direct access of users' attentional patterns to the researchers. Eye-tracking has been used in multiple educational settings to provide an understanding of cognitive processes responsible for learning and collaboration [33]. Eye-tracking has been historically known as a data source to measure engagement in various research contexts. Shagass et al. [34], Navab et al. [35] and Sanchez et al. [36], used eye-tracking to detect attentional disengagement in psychotic, autistic, and depression-effected patients, respectively. Eye-tracking has also been used to capture the engagement in marketing studies (for a comprehensive review see [37]). Dalzel and colleagues [38] used eye-tracking to compare the engagement patterns in a intelligent agent based learning scenario. Moreover, it has also been used to measure engagement in learning scenarios (for a comprehensive review see [39,40]).

Eye-tracking has been widely used to measure collaboration in different dual eye-tracking experiments. In the past studies, collaboration and engagement measures have been used to correlate the collaboration levels to various constructs like expertise [41], collaboration quality [42], task based performance [43] and learning outcomes [44]. Two synchronous eye-trackers can be used for studying the gaze of two persons interacting to solve a problem. It gives a chance to understand the underlying cognition and social dynamics when people collaborate to solve problems at hand [45]. In a collaborative task of finding bugs in a program, Stein et al. [46] showed that the pairs who had their gaze displayed to their partners took less time in finding the bugs than those pairs who had no information about their partners' gaze. From a collaborative concept map experiment , Liu et al. [47] found that the gaze data of the pair is predictive of the expertise in the collaboration. The authors framed the whole interaction as a sequence of concepts looked at. The authors then use Hidden Markov Models to predict the outcome of post-test and achieved an accuracy of 96.3%.

Eye-tracking has been used to capture communication and referencing in collaborative scenarios, which are essential for creating and maintaining mutual ground among collaborators. Grounding is an essential part of the communication [48]. Clark and Brennan define grounding as the "coordination of process" – which entails sharing information (or common ground) – which includes mutual knowledge, beliefs, assumptions [49,50]. In a dual eye-tracking experiment, the authors measured the time lag between the speakers looking and referring at a specific actor and the listeners looking at the same actor. This time lag was termed as the cross-recurrence between the participants. The average cross-recurrence was found to be between 1200 and 1400 ms. This time was consistent with the additions of eye-voice span [51] and voice eye-span [52]. The cross-recurrence [53] (the amount of time spent by the collaborators while looking at the same object) is one of the most common measurements to assess the collaboration quality. Recently, Sharma and colleagues [43] proposed a temporal and more distributed and robust version of the cross-recurrence known as gaze-similarity (the amount of time spent by the collaborators while looking at the same set of objects in a given time window). Thus, eye-tracking is an established approach to quantify both collaboration and engagement during an activity.

2.4. Eye-tracking as a means to understand cognitive processes during collaborative learning

Collaborative eye-tracking has been used in previous research in collaborative learning scenarios to shed light on the socio-cognitive mechanisms responsible for learning gains such as, joint-attention [42,54] , mutual understanding [55,56], misunderstandings [57], memorization [58]. In a pair programming study with collaborative eye-tracking data, the results depicted that the students which were able to provide correct answers to the comprehension questions had more joint-attention (measured by cross-recurrence or gaze similarity) than the students who could not give correct answers [42,54]. Furthermore, in a collaborative concept map study, the joint-attention was found to be correlated with the learning gains of the pair [59]. In a similar study with collaborative concept maps, participants' gaze on a Knowledge Awareness Tool (KAT) to assess the peer's domain expertise was reported to be correlated with high levels of mutual understanding between the pair [55]. Mutual understanding had been shown to be one of the main socio-cognitive construct responsible for high level collaborative learning outcomes [56,60–62]. Sangin and colleagues [63] used a knowledge awareness tool (KAT) to inform the pair about their partners' knowledge about a certain topic in a collaborative concept map task. From the gaze data analysis, the authors found that there was a positive correlation between the gaze on the KAT and participants' relative learning gain.

In terms of collaborative eye-tracking and dialogues during the collaborative learning situations, Cherubini and colleagues showed that the distance between the places looked at by peers is predictive of their level of misunderstanding [57]. The misunderstanding was measured by the mistakes (made by the listener) in disambiguation the (speaker's) verbal references in a shared learning system, which was a detrimental factor for the learning outcome [57].

In a collaborative learning task, the gaze of the peers was indicative of the processed responsible for memorization and analysis of new concepts [44]. In a similar study the unbalanced participation (division of labour, as measured by eye-tracking) was found to be negatively correlated with learning gains of the collaborating pairs [58]. Moreover, sharing gaze among collaborating peers, resulted in a better division of labour [64], better understanding of the content [65], and better attention spans from the students [66]. In this contribution, we attempt to use the gaze as a measurement of the behaviour of the peers and examine the effect of a certain behaviour on the relationship between the different attitudes of children towards coding.

3. Conceptual model and research hypotheses

As presented in the previous section, relevant literature has shown positive correlations among the different constructs related to attitudes (intention to participate, attitude towards team work, enjoyment and perceived learning) and indicators of behaviour (collaboration and engagement). Behaviour is seldom considered as a factor which can affect these relationships; rather it is considered as a factor in the correlational analyses while most of these studies use subjective questionnaires. In addition, eye-tracking has been widely used to provide a direct access of users' attentional patterns and provide an understanding of cognitive processes responsible for learning and collaboration [33]. In this study, we propose objective measures of behaviour as a pivoting factor, and have a hypothesis that behaviour can affect the strength and/or the polarity of the relationship between attitudes of children in coding activities. Therefore, we measured behaviour using eye-tracking data. Specifically, we used gaze uniformity to measure the level of children's engagement and gaze similarity to measure Children's level of collaboration during the coding activity. Furthermore, our study is guided by the following research question:

How does the gaze behaviour moderates the relationship between different attitudes when it comes to coding activity with children?

In order to investigate the effect of children's behaviour (capture via gaze) in their attitudes (captured via survey responses) during coding activities (see conceptual model in Fig. 1), we divide the overall challenge into smaller hypotheses, as described below. Responding to the following three hypotheses offers important insights into the general feasibility of the problem. Specifically, our study attempts to verify the following research hypothesis:

- H1a: Children's engagement (gaze uniformity) has a significant moderating effect on the relationship between children's Intention and Enjoyment during a coding activity.
- H1b: Children's engagement (gaze uniformity) has a significant moderating effect on the relationship between children's Intention and Learning during a coding activity.

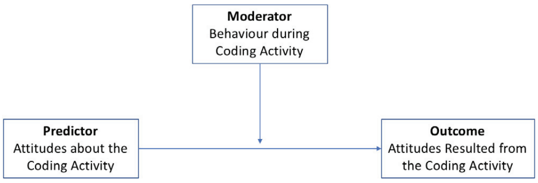

Fig. 1. The Conceptual Model of our Study.

- H2a: Children's level of collaboration (gaze similarity) has a significant moderating effect on the relationship between children's Team Work and Enjoyment during a coding activity.
- H2b: Children's level of collaboration (gaze similarity) has a significant moderating effect on the relationship between children's Team Work and Learning during a coding activity.

In the following diagram presented in Fig. 2, the research hypotheses of our study are summarized.

4. Methodology

In this section, we present the methodological details of our study, like, the measurements used and the data analysis implemented.

4.1. The coding activity

Based on the constructionist approach and its main principle, learning by doing [67], we conducted a coding workshop at the Norwegian University of Science and Technology Trondheim, Norway. Our coding workshops are out-of-school activities, in which children from 8 to 17 years old are invited in a specially designed room in the university's premises to interact with digital robots, using Scratch for Arduino (S4A), and then code their own game using the Scratch programming language. At each workshop the children work for approximately four hours. Five assistants with previous experience in similar activities are responsible for instruction and the procedure for the workshops. The workshop consists of two main parts, interaction with the robots and creating games with Scratch, described below.

Interaction with robots: During the first part of the coding activity, the children interact with digital robots. The assistants give a brief presentation of the workshop's activities. Then, the children use a paper tutorial with instructions (Fig. 3) for how to make the robots react to the physical environment with visual effects using simple loops of Scratch for Arduino (e.g. to make the tongue of the snake robot move when there is less light at a sensor). The first part of the workshop provides a smooth start for the participants as they playfully interact with tangible objects. Showing the connection with the physical world through digital robots, gives an opportunity to the children to understand STEM subjects better and handle difficult problems [68]. For this activity children by using Scratch for Arduino (S4A) are also introduced to Scratch ''logic'' while they get motivation and inspiration. The duration of the first part varies from 45 to 90 min. When all the children have finished, they have a break before the next section begins.

Creating games with Scratch: This section is the main activity of the workshop and lasts approximately three hours, without the presence of the robots. The goal is to successfully develop a simple game, coding in Scratch. To achieve this goal, the assistants give another paper tutorial with examples of all the basic Computer Science concepts, possible loops they should use to complete their own game, and how to manage the process of game development (Fig. 4). They were advised that, first, they should think and decide the story for their game and then create a draft storyboard. When they had finished, they started coding using Scratch. The children can ask for support from the assistants whenever they need it throughout the activity in order to successfully complete their games. Finally, the children reflected on and played each other's games.

4.2. Participants

We conducted the study at a dedicated lab space at the Norwegian University of Science and Technology Trondheim, Norway. Specifically, the study lasted two weeks during Autumn 2016, with 44 children from the eighth to twelfth grades (aged 8–17 years old), 12 girls (mean age: 12.64, standard deviation (SD): 2.838) and 32 boys (mean age: 12.35, SD: 2.773). Five workshops were held in total, all following the same process for the coding activity, addressed to novices in coding. Some of the participants in the sample (13–17 years old) were recruited from the local schools who had applied to take part in our activity. The other set of participants (8–12 years old) were youngsters who attend local coding clubs as an after-school activity. The children were carefully selected regarding their age so at each of the workshops, the participants were at the same grade or within a small age range. All 44 children comprising the sample of this study were eye-tracked volunteering their participation; the legal guardians provided a written informed consent form for their child, giving permission for the data collection.

4.3. Measures

As mentioned before, this study is one of the few so far utilizing children's gaze. We recorded children's gaze while they were coding using the Scratch environment during both parts of the activity. The eye-tracking data was collected using four SMI and one Tobii eye-tracking glasses. The sampling rate for all the eye-tracking glasses was set to be 30 Hz for the binocular eye-tracking. The average accuracy for both SMI and Tobii glasses was 0.5 degrees at a distance of 40 Centimetres. The visual field was divided into six areas of interests (AOIs). Five of them are shown in Fig. 5. Once we have the gaze data on these six AOIs, we extracted the following variables to include in our analysis for this contribution:

Level of Collaboration: To measure the level of collaboration of children during the coding activity, we calculate the gaze similarity. Gaze similarity captures the proportion of the time spent by the participants looking at the similar set of AOIs in a given time window. This is computed as the cosine similarity of the vectors comprising of the proportion of time spent in each AOI within a given time window (see Fig. 6).

$$Similarity(X,Y) = \frac{\sum_{i=1}^{N} X_i Y_i}{\sqrt{\sum_{i=1}^{N} X_i^2}\sqrt{\sum_{i=1}^{N} Y_i^2}} \quad (1)$$

Engagement: To measure engagement of children during the coding activity, we calculate gaze uniformity (see Fig. 7). Gaze uniformity captures the uniformity of the time spent on all AOIs. The distribution is computed as a vector of length six (there are six AOIs) comprising of the proportion of time spent in each AOI. The uniformity is computed as the inverse of a Kullback–Leibler divergence between the original proportionality vector and a uniform distribution with the same minimum and maximum limits as the original vector.

$$Uniformity(X,Y) = \sum_{i=1}^{N} X_i log \frac{X_i}{Y_i} \quad (2)$$

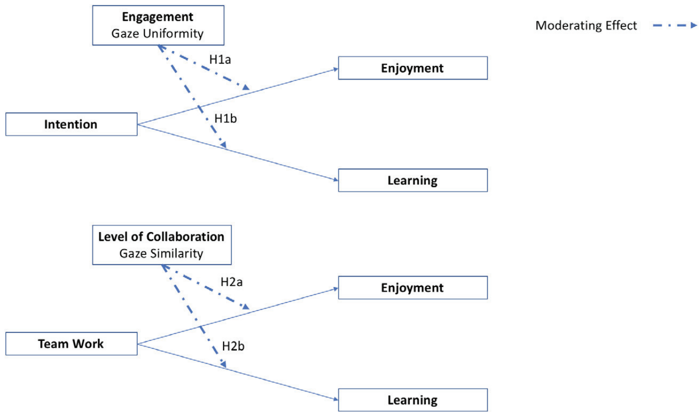

Fig. 2. The research hypotheses of our study.

Fig. 3. Left: Example of the robots' tutorial. Right: example robot.

Fig. 4. Left: example of the tutorial; Middle: children interacting with Scratch; Right: example of developed game.

At the end of the activity, the children completed a paper-based survey. The survey gathered feedback on the children's attitudes regarding the coding activity. The children were asked to rate their experience with the coding activity regarding their four different variables: team work, their intention to participate in future similar activities, their enjoyment during the activity and how much they thing they learned (i.e., perceived learning). For all the measures, we used a five-point Likert-scale questionnaire. Table 1 shows the operational definitions of the four factors.

4.4. Data analysis

In this contribution, we address the following analysis question: *"how does the behaviour moderates the relationship between different attitudes when it comes to coding?"*. Fig. 1 shows the relation between the constructs, measurements and variables used in this study. To find how the behaviour affects the relation between the different attitudes towards coding, we chose to conduct moderator analysis [72]. Moderator is a variable that affects the strength and/or direction of the relationship between

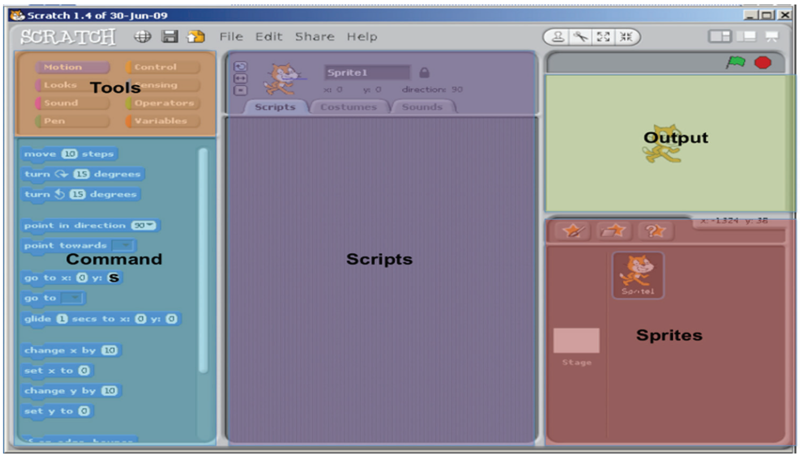

Fig. 5. The five areas of interests (AOI) for the screen, the sixth AOI was the robot.

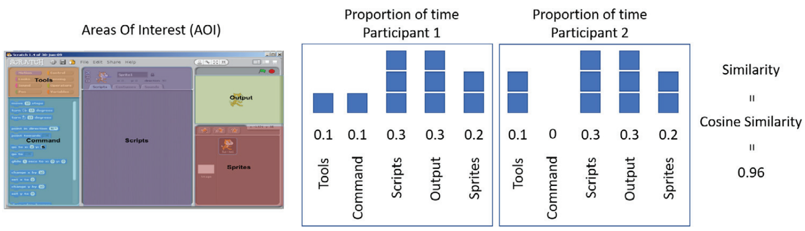

Fig. 6. A typical example of computing gaze similarity from the time spent on the different AOIs.

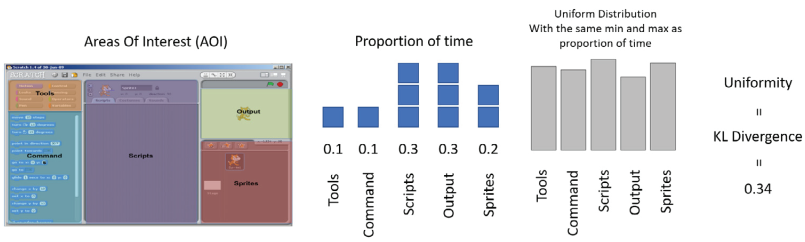

Fig. 7. A typical example of computing gaze uniformity from the time spent on the different AOIs.

Table 1
The attitudinal factors and their respective questions, operational definitions and sources.

Factor	Operational definition	Item/Question	Source
Perceived learning	The degree to which children indicate their performance.	Please indicate if you learned new things during the coding activity (Not at all – Very much)	[69]
Intention	The degree of children 's willingness to participate in a similar activity.	Please indicate how much you want to attend similar coding activities in the future (Not at all – Very much)	[70]
Enjoyment	The degree to which children indicate their enjoyment during the activity	Please indicate how much you enjoyed your participation in the coding activity (Not at all – Very much)	[70]
Team Work	The degree to which children indicate their enjoyment of working in a team during the activity	Please indicate how much you enjoyed working in a team (Not at all – Very much)	[71]

two variables. In terms of ANOVA or correlational analyses, this variable is added as an independent variable that does not have a direct effect on the dependent variable, but when combined with the main independent variable, shows a significant interaction effect. In the present analyses, we use intention and team work as the independent variables; enjoyment and learning as the dependent variables; and gaze behaviour (similarity and uniformity) as the potential moderator variables.

Table 2
Descriptive statistics for the variables used in this contribution.

	Mean	Std. Dev.	Minimum	Maximum
Uniformity (0–1)	0.48	0.35	0.004	0.97
Similarity (0–1)	0.34	0.26	.002	0.96
Intention (scale 1–5)	4.45	0.73	3	5
Team work (scale 1–5)	4.24	0.87	2	5
Enjoyment (scale 1–5)	4.55	0.51	4	5
Perceived Learning (scale 1–5)	4.65	0.98	1	5

Table 3
Pearson correlation matrix for the attitude variables used in the analysis.

		1	2	3	4
Team work	1	–	0.37**	0.32*	0.30*
Intention to participate	2	–	–	0.45**	0.56***
Learning	3	–	–	–	0.65***
Enjoyment	4	–	–	–	–

***p-value $< .001$.
**p-value $< .01$.
*p-value $< .05$.

Table 4
Testing the moderating effect of the level of collaboration, on the team work to enjoyment and team work to learning relationships.

Model for perceived learning					
	Estimate	Error	t-value	p-value	Hypothesis
Intercept	4.62	0.10	38.71	.00001	H2b Accepted
Team work	0.44	0.18	2.42	.01	
Similarity	0.33	0.43	0.77	.44	
interaction	1.20	0.58	2.04	.04	
Model for enjoyment					
Intercept	4.62	0.10	44.34	.00001	H2a Accepted
Team work	0.36	0.15	5.62	.02	
Similarity	−0.45	0.33	−1.38	.17	
interaction	1.43	0.67	2.11	.04	

Table 5
Moderator effect model for perceived learning and enjoyment, using attitude towards intention and gaze uniformity as the independent and moderating variables, respectively.

Model for perceived learning					
	Estimate	Error	t-value	p-value	Hypothesis
Intercept	4.70	0.10	45.34	.00001	H1b Accepted
Intention	0.63	0.14	4.33	.0001	
Uniformity	−0.25	0.36	−0.69	.049	
interaction	1.44	0.48	2.98	.004	
Model for enjoyment					
Intercept	4.06	0.08	53.85	.00001	H1a Accepted
Intention	0.66	0.11	5.62	.0001	
Uniformity	−0.61	0.34	−1.75	.09	
interaction	1.09	0.39	2.77	.008	

5. Results

5.1. Descriptive results

Children expressed high learning and enjoyment (4.7/5 and 4.6/5, respectively) for the coding activity. Additionally, they expressed slightly lower intention and attitudes towards team work (4.4/5 and 4.2/5, respectively). High levels of these attitudes indicate positive views concerning their learning performance and beliefs regarding their engagement with coding activities. The descriptive statistics about children's attitudes and eye-tracking measures are summarized in Table 2.

To assess the correlation between individual items on the questionnaire, Pearson's correlation coefficient between the factors was used. Pearson quantifies the strength of the relationship between the variables. Table 3 shows the pairwise correlations among attitude towards team work, intention to participate, learning, and enjoyment. We observe that all the correlations are significant and positive. This allows us to proceed with the investigation for the moderation effects. In the following subsections we present four different moderation analysis for the different variables measuring attitudes and behaviour.

5.2. Level of collaboration as moderator

First, we focus on the relation between attitude towards team work and learning; we test if this relation is significantly moderated by the level of collaboration of the participants. Table 4 shows the model fitting details and Fig. 8 (left) shows the trends for the main effect (dashed line) the high collaboration (blue line) and the low collaboration categories (red line). We observe a significant interaction effect of collaboration and attitude towards team work on the learning. From Fig. 8 (left) it can be observed that the relation between the attitude towards team work and learning is stronger for participants who experienced high level of collaboration. Thus our data provide strong evidence that children's level of collaboration during coding activities moderates the relationships between their attitude about team-work and learning (accepting H2b).

Second, concerning the moderating effect of the level of collaboration for the relation between attitude towards team work and enjoyment, Table 4 shows the details for the model and Fig. 8 (right) shows the trends similar to that of Section 5.2. We observe a significant moderating effect of collaboration and attitude towards team work on the enjoyment. From Fig. 8 (right) it can be observed that the relation between the attitude towards team work and enjoyment is stronger when the participants experience high levels of collaboration than in the case participants experience low levels of collaboration. Thus our data provide strong evidence that children's level of collaboration during coding activities moderates the relationships between their attitude about team-work and enjoyment (accepting H2a).

5.3. Engagement as a moderator

Investigating how engagement moderates the relation between intention to participate and perceived learning, Table 5 shows the details for the model and Fig. 9 (left) shows the trends similar to that of Section 5.2. We observe a significant moderation effect of engagement and intention to participate on perceived learning. From Fig. 9 (left) it can be observed that the relation between intention to participate and perceived learning is stronger when the participants experience high engagement than in the case the participants experience low engagement. Thus our data provide strong evidence that children's level of engagement during coding activities moderates the relationships between their intention to participate in the activity and learning (accepting H1b).

Finally, we consider how engagement moderates the relation between intention to participate and enjoyment, Table 5 shows the details for the model and Fig. 9 shows the trends similar to that of Section 5.2. We observe a significant moderating effect of engagement in the relationship between intention to participate and enjoyment. From Fig. 9 (right) it can be observed that the relation between intention to participate and enjoyment is stronger for the highly engaged participants than the case of non-engaged ones. Thus our data provide strong evidence that children's level of engagement during coding activities moderates the relationships between their intention to participate in the activity and enjoyment (accepting H1a).

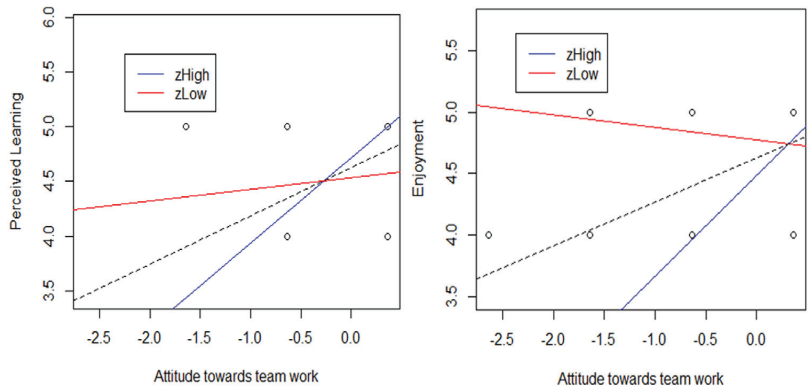

Fig. 8. Trends from the models shown in Table 4. The red and blue lines show fitted model with low and high collaboration (gaze similarity) values, respectively. The dashed line shows the main effect. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

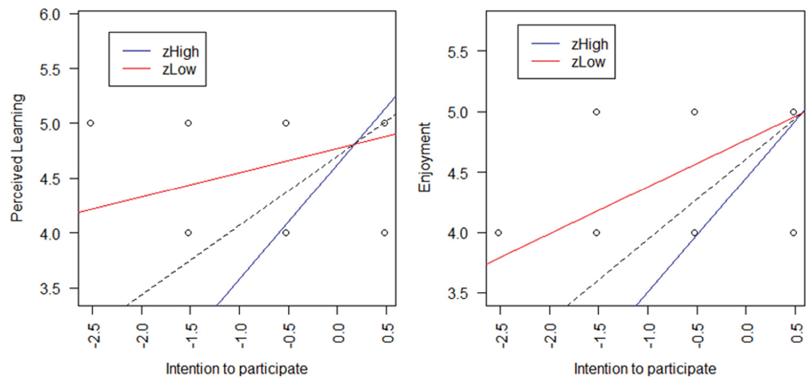

Fig. 9. Trends from the models shown in Table 5. The red and blue lines show fitted model with low and high gaze similarity values, respectively. The dashed line shows the main effect. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

6. Discussion and conclusions

We presented analysis of data from a study with children coding games and interactive robots. We captured children's behaviour while coding using eye-trackers. Moreover, we also captured their attitude towards coding using questionnaires. In this contribution, we investigated the role of gaze-behaviour as a moderator for the relationship between the different attitudes. The results show that gaze-behaviour does moderate the relationship between the attitudes about a coding activity with the ones resulted from the coding activity.

The first behavioural measure is the level of collaboration (measure via gaze similarity). The results show that the level of collaboration affects the relationship between children's attitudes. High level of collaboration shows children's ability to share the learning experience, this fosters their enjoyment from the process. Moreover, high level of collaboration also indicates high level of mutual understanding (common ground) [42,53] and better division of labour [64], that is critical for group learning activities. In addition, through the collaborative process of coding, children share their learning by interacting and making decisions together [73]. This could reinforce learning (as also indicated from the perceived learning measure). A few studies have also reported similar results where the lack of shared gaze among the participants turns out to be detrimental for children's learning (e.g., [74]).

The differences in children's coding level of competence, even if they had positive attitude towards team work, made them feeling that they did not learn enough. Differences in children's coding competence could have also made it difficult to communicate and coordinate with the partner. This can be the reason that they also enjoyed the activity less than those who were in more homogeneous groups, and were able to communicate and coordinate well with peers. These results are inline with the previous work related to learn new concepts and the gaze-togetherness [58,75]; and the lack of gaze-togetherness and the high levels of misunderstandings [57].

The second behavioural measure we used was children's engagement (measured via gaze uniformity). Engagement moderated the relationship between children's attitudes (influence of intention to participate to enjoyment and learning). Higher engagement, shows confidence and children explore different parts of the interface and navigate in all parts of the screen [76]. Also, the activities were designed in a manner that all parts of the interface were equally important for success (task based). Children who pay equal attention to all the feel more successful in learning the concepts than those who did not. Their familiarization with the learning environment (Scratch) and being able to understand all its different parts and their functionalities influenced their enjoyment and learning. The ability of accomplishing a task provides an overall positive experience and offers positive results like fun and enjoyment [77,78].

Collaboration promotes better perspective taking and reflection among students [79,80], which in turn enables higher learning gains and better collaborative learning experiences [81–84]. Moreover, the engagement with collaborative tasks can offer opportunities for the children to learn the domain related [85–87] as well as the collaborative skills [88]. These relations have also been highlighted in the case of pair-programming at a classroom level [89]. Collaboration among the students has also been found to be fruitful in acquiring other computer literacy skills [90] beyond programming skills.

It is shown that co-located collaboration has certain educational benefits [91,92] such as, externalization of thought processes [93] and reduced cognitive load [94]. This supports our results where we found that relation between the attitude towards team work and learning is stronger for participants who experienced high level of collaboration. The groups with high levels of collaborative work and a more positive attitude towards collaboration were able to talk about the programming processes and concepts more than the groups with lower collaborative work and hence they were able to achieve higher learning gains. By designing for these mechanisms one can achieve higher collaborative outcomes [95]. For example, while working together and sharing insights and problems with each other, the peers might benefit from a reflection tool [96].

Other studies with collaborative learning with children have argued about the benefit of collaboration [97–99] specifically, in learning computational thinking skills [97,100,101]. Our results consists of two benefits over the previous studies. First, most of the studies reported in Section 2 addressed the pairwise relations among behaviour and attitudes, while this contribution focuses on more intricate nature of the triumvirate relationship. Second, the behaviour was used in the reported eye-tracking studies [34,35,53,102] more as a process variable for the plausible explanation of the relation between success/expertise/collaboration/perception, while our results show that it could be used as a moderator. This fact will allow us to provide feedback in real time to affect both attitudes and experiences in positive manner.

6.1. Theoretical and practical implications

Our results show that the behaviour is key to understand the relation between attitudes towards learning, specifically when it comes to learning to code. Both gaze similarity and gaze uniformity influenced children's relationship among attitudes. This highlights the importance of both individual and collaborative measures to understand learner's behaviour during coding activities and act accordingly to enhance their learning experiences.

Considering the relationships between the intention, learning and enjoyment; and how they are moderated by gaze uniformity, our results seem to extend the Technology Acceptance Model (TAM) [8]. According to TAM the perceived ease of use, intention to use and the actual usage are correlated [8]. Our results suggest that the children with high gaze uniformity on the interface have a higher correlation between the intention and enjoyment; and between the intention and perceived learning than those children who have low gaze uniformity. This shows that the gaze behaviour moderates the relationship between intention to use technology and the other attitudes (enjoyment and learning). This is inline with TAM, which also shows significant correlations between the intention to use, the behavioural use, and the attitudes towards technology. In this contribution we propose to use the behaviour as a moderator of the relationship between different attitudes. The results enhances our understanding of how children's behaviour can impact their attitudes towards a new technology, since most of the children participating in the workshop were novices.

In practical terms, the gaze uniformity translates to exploring the interface in a uniform manner to learn most of the functions provided by the environment. Gaze uniformity can be calculated in real-time, which could allow us to provide feedback while the children are coding. This might enhance the learning experiences and outcomes for them. Gaze uniformity can also be used to develop post-coding reflection tools as well. One can use the gaze data to show how the children explored the interface and help them understand what they missed. This might help them to have better exploration and understanding in the future coding activities.

In any collaborative scenario, where the coordination of the collaborators is essential for the successful completion of the task such as collaborative programming, collaborative problem solving, collaborative learning, it is essential to have a common ground between the team members [48]. According to the grounding theory in communication [48] – grounding is basic to all the communications – and hence, it is important to have a measurement for the process of grounding the conversations. Mutual gaze is the process by which two or more collaborators initiate and maintain the common ground [103]. Mutual gaze can be initiated by a diactic gesture (verbal or physical) by one of the team members [104]. When John refers (talks about or points) at a certain part of the Scratch interface to initiate a conversation he has to look at that particular part of the screen. At the same time, if following what John's discourse, Susan looks at the same part of screen to make sense of what John is saying. This results in gaze similarity. Our results show that the teams with high gaze similarity had a higher correlation between the attitude towards team work and both learning and enjoyment (Table 4 and Fig. 8). This is inline with results reported in previous research with collaborative processes and conversations [42,43,53,105]. The results in the present contribution highlight the importance of having a common ground among collaborators at young ages as well.

In practical terms, the moderator effect shown by the gaze similarity could be exploited to provide gaze-aware feedback to the collaborating partners. In video based learning scenarios gaze-awareness has been shown to improve learning experiences [106] and outcomes [66]. In collaborative problem solving sharing the gaze of partners leads to better collaborative outcomes [64, 105,107]. Children might benefit from having a additional support for sharing a common ground with their team-mates, since their verbal referencing capabilities might not be as good as adults due to lack of experience.

6.2. Limitations

This study is the one of the few ones (to the best of our knowledge) to explore the relationship between the objective behaviour and the attitudes of children towards coding activities. The eye-tracking data provided us a proxy for the behaviour. However, there were many difficulties faced while collecting the data, which affected the quality of data in certain ways. For example the eye-tracking glasses are made to fit adult sized heads and the participants were eight to seventeen years old. A few of the children obviously had small head sizes. This created a few problems while calibrating and post-processing the data. Another limitation of the current contribution comes from the fact that this was an experiment conducted with a visual programming tool (Scratch) and following specific instructions and learning goals. Although, we would expect the findings to generalize across other visual programming tools and coding activities, it is difficult to generalize for text-based programming environments and formal coding activities.

6.3. Future work

This contribution opens up varied directions for further extension of research. First, this paper focuses on eye-tracking as an objective proxy of behaviour, which is not ideal for ecological validity and hence one should explore other options for behavioural proxy. Some examples are, facial features, interaction patterns with the programming interface, arousal data collected through devices such as wristbands. Second, our results show that there is potential to use eye-tracking data to provide feedback to children while they are learning how to code. Our results can provide a first step towards designing a gaze-aware feedback system to enhance the learning experiences and the learning outcomes. Third, a logical extension of the present contribution can be to look into the temporal dynamics of the gaze behaviour to observe how engagement and collaboration evolve for different groups of children with different characteristics (e.g., competence in coding, experience, age groups etc.).

6.4. Conclusions

In this paper we present analytics to understand the relationships between attitudes and behaviour of children while solving coding problems. We proposed to use the behaviour, as measured by gaze, as a moderator of the relationship between the different attitudes. The results show that the behaviour is an important factor while examining such relations. We found that behaviour does moderates the relation between the intention to learn, attitude towards team work, enjoyment and perceived learning. We also demonstrate that the results are inline with existing theories and contemporary research. This encourages us to work in this direction for future towards enhancing our understanding about kids coding patterns.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.ijcci.2019.04.004.

Acknowledgements

This work supported from the “Learning science the fun and creative way: coding, making, and play as vehicles for informal science learning in the 21st century” Project, under the European Commission’s Horizon 2020 SwafS-11-2017 Program (Project Number: 787476). This article reflects the views only of the authors and it does not represent the opinion of neither the European Commission nor NTNU, and the European Commission and NTNU cannot be held responsible for any use that might be made of its content. This work is also supported from the Norwegian Research Council under the projects FUTURE LEARNING (number: 255129/H20) and Xdesign (number: 290994/F20), and by NOKUT under the Centre for Excellent IT Education (Excited) (number: 16/02049).

References

[1] Gale M. Sinatra, Benjamin C. Heddy, Doug Lombardi, The Challenges of Defining and Measuring Student Engagement in Science, Taylor & Francis, 2015.
[2] Curtis R. Henrie, Lisa R. Halverson, Charles R. Graham, Measuring student engagement in technology-mediated learning: a review, Comput. Educ. 90 (2015) 36–53.
[3] Jennifer A. Fredricks, Michael Filsecker, Michael A. Lawson, Student Engagement, Context, and Adjustment: Addressing Definitional, Measurement, and Methodological Issues, Elsevier, 2016.
[4] Juanjuan Chen, Minhong Wang, Paul A Kirschner, Chin-Chung Tsai, The role of collaboration, computer use, learning environments, and supporting strategies in cscl: a meta-analysis, Rev. Educ. Res. (2018) 0034654318791584.
[5] Fengfeng Ke, Alternative goal structures for computer game-based learning, Int. J. Comput. Support. Collaborat. Learn. 3 (4) (2008) 429.
[6] Jennifer Tsan, Collin F. Lynch, Kristy Elizabeth Boyer, Alright, what do we need?: a study of young coders collaborative dialogue, Int. J. Child Comput. Interact. (2018).
[7] Marina Umaschi Bers, Louise Flannery, Elizabeth R Kazakoff, Amanda Sullivan, Computational thinking and tinkering: exploration of an early childhood robotics curriculum, Comput. Educ. 72 (2014) 145–157.
[8] Fred D. Davis, Perceived usefulness, perceived ease of use, and user acceptance of information technology, Manag. Inf. Syst. Q. (1989) 319–340.
[9] Toby Marshall Egan, Baiyin Yang, Kenneth R. Bartlett, The effects of organizational learning culture and job satisfaction on motivation to transfer learning and turnover intention, Hum. Resour. Dev. Q. 15 (3) (2004) 279–301.
[10] Timothy Teo, Jan Noyes, An assessment of the influence of perceived enjoyment and attitude on the intention to use technology among pre-service teachers: a structural equation modeling approach, Comput. Educ. 57 (2) (2011) 1645–1653.
[11] Abdulhameed Rakan Alenezi, Abdul Karim, et al., An empirical investigation into the role of enjoyment, computer anxiety, computer self-efficacy and internet experience in influencing the students’ intention to use e-learning: a case study from saudi arabian governmental universities, Turk. Online J. Educ. Technol. 9 (4) (2010) 22–34.
[12] Jiming Wu, De Liu, The effects of trust and enjoyment on intention to play online games., J. Electron. Commer. Res. 8 (2) (2007).
[13] Betsy Blunsdon, Ken Reed, Nicola McNeil, Steven McEachern, Experiential learning in social science theory: an investigation of the relationship between student enjoyment and learning, High. Educ. Res. Dev. 22 (1) (2003) 43–56.
[14] Y.Y.i Mun, Yujong Hwang, Predicting the use of web-based information systems: self-efficacy, enjoyment, learning goal orientation, and the technology acceptance model, Int. J. Hum-Comput. Stud. 59 (4) (2003) 431–449.
[15] Mark A. Faust, Nancy Glenzer, “I could read those parts over and over”: eighth graders rereading to enhance enjoyment and learning with literature, J. Adolesc. Adult Lit. 44 (3) (2000) 234–239.
[16] Reinhard Pekrun, Thomas Goetz, Anne C Frenzel, Petra Barchfeld, Raymond P Perry, Measuring emotions in students’ learning and performance: the achievement emotions questionnaire (aeq), Contemp. Educ. Psychol. 36 (1) (2011) 36–48.
[17] Mary Ainley, John Ainley, Student engagement with science in early adolescence: the contribution of enjoyment to students’ continuing interest in learning about science, Contemp. Educ. Psychol. 36 (1) (2011) 4–12.
[18] Andrew J. Martin, The relationship between teachers’ perceptions of student motivation and engagement and teachers’ enjoyment of and confidence in teaching, Asia-Pac. J. Teach. Edu. 34 (1) (2006) 73–93.
[19] Leslie Morrison Gutman, John Vorhaus, The impact of pupil behaviour and wellbeing on educational outcomes, Department for Education, 2012.
[20] Lesley Xie, Alissa N. Antle, Nima Motamedi, Are tangibles more fun?: comparing children’s enjoyment and engagement using physical, graphical and tangible user interfaces, in: Proceedings of the 2nd International Conference on Tangible and Embedded Interaction, ACM, 2008, pp. 191–198.
[21] Stuart MacFarlane, Gavin Sim, Matthew Horton, Assessing usability and fun in educational software, in: Proceedings of the 2005 Conference on Interaction Design and Children, ACM, 2005, pp. 103–109.
[22] Johnmarshall Reeve, Hyungshim Jang, What teachers say and do to support students’ autonomy during a learning activity., J. Educ. Psychol. 98 (1) (2006) 209.
[23] Lorena Blasco-Arcas, Isabel Buil, Blanca Hernández-Ortega, F Javier Sese, Using clickers in class. the role of interactivity, active collaborative learning and engagement in learning performance, Comput. Educ. 62 (2013) 102–110.
[24] Paul Resta, Thérèse Laferrière, Technology in support of collaborative learning, Educ. Psychol. Rev. 19 (1) (2007) 65–83.
[25] Sanna Järvelä, Hanna Järvenoja, Socially constructed self-regulated learning and motivation regulation in collaborative learning groups, Teach. Coll. Rec. 113 (2) (2011) 350–374.
[26] Cindy E. Hmelo, Mark Guzdial, Jennifer Turns, Computer-support for collaborative learning: learning to support student engagement, J. Interact. Learn. Res. 9 (2) (1998) 107.
[27] Karel Kreijns, Paul A. Kirschner, Wim Jochems, Identifying the pitfalls for social interaction in computer-supported collaborative learning environments: a review of the research, Comput. Human Behav. 19 (3) (2003) 335–353.

[28] Lasse Lipponen, Exploring foundations for computer-supported collaborative learning, in: Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community, International Society of the Learning Sciences, 2002, pp. 72–81.
[29] Jean M. Williams, Colleen M. Hacker, Causal relationships among cohesion, satisfaction, and performance in women's intercollegiate field hockey teams, J. Sport Psychol. 4 (4) (1982) 324–337.
[30] Jolene Galegher, Robert E. Kraut, Computer-mediated communication for intellectual teamwork: an experiment in group writing, Inf. Syst. Res. 5 (2) (1994) 110–138.
[31] Daniel M Landers, Michael O Wilkinson, Brad D Hatfield, Heather Barber, Causality and the cohesion-performance relationship, J. Sport Psychol. 4 (2) (1982) 170–183.
[32] Martin Kandlhofer, Gerald Steinbauer, Evaluating the impact of educational robotics on pupils technical-and social-skills and science related attitudes, Robot. Auton. Syst. 75 (2016) 679–685.
[33] Kshitij Sharma, Patrick Jermann, Pierre Dillenbourg, Luis P Prieto, Sarah DAngelo, Darren Gergle, Bertrand Schneider, Martina Rau, Zach Pardos, Nikol Rummel, CSCL and Eye-Tracking: Experiences, Opportunities and Challenges, International Society of the Learning Sciences, Philadelphia, PA, 2017.
[34] Charles Shagass, Richard A. Roemer, Marco Amadeo, Eye-tracking performance and engagement of attention, Arch. Gen. Psychiatry 33 (1) (1976) 121–125.
[35] Anahita Navab, Kristen Gillespie-Lynch, Scott P Johnson, Marian Sigman, Ted Hutman, Eye-tracking as a measure of responsiveness to joint attention in infants at risk for autism, Infancy 17 (4) (2012) 416–431.
[36] Alvaro Sanchez, Carmelo Vazquez, Craig Marker, Joelle LeMoult, Jutta Joormann, Attentional disengagement predicts stress recovery in depression: an eye-tracking study, J. Abnorm. Psychol. 122 (2) (2013) 303.
[37] Michel Wedel, Rik Pieters, A review of eye-tracking research in marketing, in: Review of Marketing Research, Emerald Group Publishing Limited, 2008, pp. 123–147.
[38] Sara Dalzel-Job, Craig Nicol, Jon Oberlander, Comparing behavioural and self-report measures of engagement with an embodied conversational agent: a first report on eye tracking in second life, in: Proceedings of the 2008 Symposium on Eye Tracking Research & Applications, ACM, 2008, pp. 83–85.
[39] Heather L. O'Brien, Elaine G. Toms, The development and evaluation of a survey to measure user engagement, J. Assoc. Inf. Sci. Technol. 61 (1) (2010) 50–69.
[40] Meng-Lung Lai, Meng-Jung Tsai, Fang-Ying Yang, Chung-Yuan Hsu, Tzu-Chien Liu, Silvia Wen-Yu Lee, Min-Hsien Lee, Guo-Li Chiou, Jyh-Chong Liang, Chin-Chung Tsai, A review of using eye-tracking technology in exploring learning from 2000 to 2012, Educ. Res. Rev. 10 (2013) 90–115.
[41] Patrick Jermann, Marc-Antoine Nüssli, Weifeng Li, Using dual eye-tracking to unveil coordination and expertise in collaborative tetris, in: Proceedings of the 24th BCS Interaction Specialist Group Conference, British Computer Society, 2010, pp. 36–44.
[42] Patrick Jermann, Marc-Antoine Nüssli, Effects of sharing text selections on gaze cross-recurrence and interaction quality in a pair programming task, in: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, ACM, 2012, pp. 1125–1134.
[43] Kshitij Sharma, Daniela Caballero, Himanshu Verma, Patrick Jermann, Pierre Dillenbourg, Looking AT Versus Looking THROUGH: A Dual Eye-Tracking Study in MOOC Context, International Society of the Learning Sciences, Inc.[ISLS]., 2015.
[44] Bertrand Schneider, Kshitij Sharma, Sébastien Cuendet, Guillaume Zufferey, Pierre Dillenbourg, Roy Pea, Using mobile eye-trackers to unpack the perceptual benefits of a tangible user interface for collaborative learning, ACM Trans. Comput.-Hum. Interact. 23 (6) (2016) 39.
[45] Marc-Antoine Nüssli, Dual eye-tracking methods for the study of remote collaborative problem solving, 2011.
[46] Randy Stein, Susan E. Brennan, Another person's eye gaze as a cue in solving programming problems, in: Proceedings of the 6th international conference on Multimodal interfaces, ACM, 2004, pp. 9–15.
[47] Yan Liu, Pei-Yun Hsueh, Jennifer Lai, Mirweis Sangin, M-A Nussli, Pierre Dillenbourg, Who is the expert? analyzing gaze data to predict expertise level in collaborative applications, in: Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on, IEEE, 2009, pp. 898–901.
[48] Herbert H. Clark, Susan E. Brennan, et al., Grounding in communication, Perspect. Soc. Shar. Cogn. 13 (1991) (1991) 127–149.
[49] Herbert H. Clark, Thomas B. Carlson, Hearers and speech acts, Language (1982) 332–373.
[50] Herbert H. Clark, Catherine R. Marshall, Definite reference and mutual knowledge, in: Psycholinguistics: Critical Concepts in Psychology, vol. 414, Taylor & Francis, 2002.
[51] Zenzi M. Griffin, Kathryn Bock, What the eyes say about speaking, Psychol. Sci. 11 (4) (2000) 274–279.
[52] Paul D. Allopenna, James S. Magnuson, Michael K. Tanenhaus, Tracking the time course of spoken word recognition using eye movements: evidence for continuous mapping models, J. Mem. Lang. 38 (4) (1998) 419–439.
[53] Rick Dale, Natasha Z. Kirkham, Daniel C. Richardson, The dynamics of reference and shared visual attention, Front. Psychol. 2 (2011).
[54] Kshitij Sharma, Patrick Jermann, Marc-Antoine Nüssli, Pierre Dillenbourg, Understanding collaborative program comprehension: interlacing gaze and dialogues, in: Computer Supported Collaborative Learning (CSCL 2013), 2013.
[55] Mirweis Sangin, Gaëlle Molinari, Marc-Antoine Nüssli, Pierre Dillenbourg, How learners use awareness cues about their peer's knowledge?: insights from synchronized eye-tracking data, in: Proceedings of the 8th International Conference on International Conference for the Learning Sciences-Volume 2, International Society of the Learning Sciences, 2008, pp. 287–294.
[56] Pierre Dillenbourg, Séverin Lemaignan, Mirweis Sangin, Nicolas Nova, Gaëlle Molinari, The symmetry of partner modelling, Int. J. Comput. Support. Collaborat. Learn. 11 (2) (2016) 227–253.
[57] Mauro Cherubini, Marc-Antoine Nüssli, Pierre Dillenbourg, Deixis and gaze in collaborative work at a distance (over a shared map): a computational model to detect misunderstandings, in: Proceedings of the 2008 symposium on Eye tracking research &; applications, in: ETRA '08, ACM, New York, NY, USA, ISBN: 978-1-59593-982-1, 2008, http://dx.doi.org/10.1145/1344471.1344515, http://doi.acm.org/10.1145/1344471.1344515.
[58] Bertrand Schneider, Kshitij Sharma, Sebastien Cuendet, Guillaume Zufferey, Pierre Dillenbourg, Roy Pea, Leveraging mobile eye-trackers to capture joint visual attention in co-located collaborative learning groups, Int. J. Comput. Support. Collaborat. Learn. 13 (3) (2018) 241–261.
[59] Kshitij Sharma, Daniela Caballero, Himanshu Verma, Patrick Jermann, Pierre Dillenbourg, Shaping learners' attention in massive open online courses, Revue internationale des technologies en pédagogie universitaire/Int. J. Technol. Higher Educ. 12 (1–2) (2015) 52–61.
[60] Anne Meier, Hans Spada, Nikol Rummel, A rating scheme for assessing the quality of computer-supported collaboration processes, Int. J. Comput. Support. Collaborat. Learn. 2 (1) (2007) 63–86.
[61] Nicolas Nova, Pierre Dillenbourg, Thomas Wehrle, Jeremy Goslin, Yvan Bourquin, The impacts of awareness tools on mutual modelling in a collaborative video-game, in: International Conference on Collaboration and Technology, Springer, 2003, pp. 99–108.
[62] Séverin Lemaignan, Pierre Dillenbourg, Mutual modelling in robotics: inspirations for the next steps, in: 2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE, 2015, pp. 303–310.
[63] Mirweis Sangin, Gaëlle Molinari, Marc-Antoine Nüssli, Pierre Dillenbourg, Facilitating peer knowledge modeling: effects of a knowledge awareness tool on collaborative learning outcomes and processes, Comput. Hum. Behav. 27 (3) (2011) 1059–1067.
[64] Susan E Brennan, Xin Chen, Christopher A Dickinson, Mark B Neider, Gregory J Zelinsky, Coordinating cognition: the costs and benefits of shared gaze during collaborative search, Cognition 106 (3) (2008) 1465–1477.
[65] Kshitij Sharma, Sarah D'Angelo, Darren Gergle, Pierre Dillenbourg, Visual Augmentation of Deictic Gestures in Mooc Videos, International Society of the Learning Sciences, Singapore, 2016.
[66] Kshitij Sharma, Hamed S Alavi, Patrick Jermann, Pierre Dillenbourg, A gaze-based learning analytics model: in-video visual feedback to improve learner's attention in moocs, in: Proceedings of the Sixth International Conference on Learning Analytics & Knowledge, ACM, 2016, pp. 417–421.
[67] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al., Scratch: programming for all, Commun. ACM 52 (11) (2009) 60–67.
[68] Saskia Bakker, Elise van den Hoven, Alissa N. Antle, Moso tangibles: evaluating embodied learning, in: Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction, ACM, 2011, pp. 85–92.
[69] Bård Kuvaas, Work performance, affective commitment, and work motivation: the roles of pay administration and pay level, J. Organ. Behav. 27 (3) (2006) 365–385.
[70] Michail N. Giannakos, Letizia Jaccheri, From players to makers: an empirical examination of factors that affect creative game development, Int. J. Child Comput. Interact. (2018).
[71] Hyo-Jeong So, Thomas A. Brush, Student perceptions of collaborative learning, social presence and satisfaction in a blended learning environment: relationships and critical factors, Comput. Educ. 51 (1) (2008) 318–336.
[72] Charles E. Lance, Residual centering, exploratory and confirmatory moderator analysis, and decomposition of effects in path models containing interactions, Appl. Psychol. Meas. 12 (2) (1988) 163–175.
[73] Pierre Dillenbourg, What Do You Mean by Collaborative Learning?, Elsevier, Oxford, 1999.

[74] Suleman Shahid, Computer mediated playful interactions: investigating how variations in the level of gaze affect children's gameplay, in: Proceedings of the 17th ACM Conference on Interaction Design and Children, ACM, 2018, pp. 427–433.

[75] Kshitij Sharma, Patrick Jermann, Marc-Antoine Nüssli, Pierre Dillenbourg, Gaze Evidence for different activities in program understanding, in: 24th Annual Conference of Psychology of Programming Interest Group, 2012.

[76] Sofia Papavlasopoulou, Kshitij Sharma, Michail Giannakos, Letizia Jaccheri, Using eye-tracking to unveil differences between kids and teens in coding activities, in: Proceedings of the 2017 Conference on Interaction Design and Children, ACM, 2017, pp. 171–181.

[77] Kanjun Qiu, Leah Buechley, Edward Baafi, Wendy Dubow, A curriculum for teaching computer science through computational textiles, in: Proceedings of the 12th International Conference on Interaction Design and Children, ACM, 2013, pp. 20–27.

[78] Kristin A Searle, Deborah A Fields, Debora A Lui, Yasmin B Kafai, Diversifying high school students' views about computing with electronic textiles, in: Proceedings of the Tenth Annual Conference on International Computing Education Research, ACM, 2014, pp. 75–82.

[79] Toshio Mochizuki, Takehiro Wakimoto, Hiroshi Sasaki, Ryoya Hirayama, Hideo Funaoi, Yoshihiko Kubota, Hideyuki Suzuki, Hiroshi Kato, Effects of Perspective-Taking Through Tangible Puppetry in Microteaching Role-Play, International Society of the Learning Sciences, Philadelphia, PA, 2017.

[80] Joshua A Danish, Noel Enyedy, Asmalina Saleh, Christine Lee, Alejandro Andrade, Science Through Technology Enhanced Play: Designing to Support Reflection Through Play and Embodiment, International Society of the Learning Sciences, Inc.[ISLS]., 2015.

[81] Gerry Stahl, Guiding group cognition in cscl, Int. J. Comput. Support. Collaborat. Learn. 5 (3) (2010) 255–258.

[82] Peter Reimann, Time is precious: variable-and event-centred approaches to process analysis in cscl research, Int. J. Comput. Support. Collaborat. Learn. 4 (3) (2009) 239–257.

[83] Chris Phielix, Frans J Prins, Paul A Kirschner, Gijsbert Erkens, Jos Jaspers, Group awareness of social and cognitive performance in a cscl environment: effects of a peer feedback and reflection tool, Comput. Hum. Behav. 27 (3) (2011) 1087–1102.

[84] Chris Phielix, Frans J. Prins, Paul A. Kirschner, Awareness of group performance in a cscl-environment: effects of peer feedback and reflection, Comput. Hum. Behav. 26 (2) (2010) 151–161.

[85] Ruth Kershner, Neil Mercer, Paul Warwick, Judith Kleine Staarman, Can the interactive whiteboard support young children's collaborative communication and thinking in classroom science activities?, Int. J. Comput. Support. Collaborat. Learn. 5 (4) (2010) 359–383.

[86] Michael A Evans, Eliot Feenstra, Emily Ryon, David McNeill, A multimodal approach to coding discourse: collaboration, distributed cognition, and geometric reasoning, Int. J. Comput. Support. Collaborat. Learn. 6 (2) (2011) 253.

[87] Joan Moss, Ruth Beatty, Knowledge building in mathematics: supporting collaborative learning in pattern problems, Int. J. Comput. Support. Collaborat. Learn. 1 (4) (2006) 441–465.

[88] Elena Durán-López, Leah F. Rosenbaum, Ganesh V. Iyer, Geometris: designing collaborative mathematical interactions for children, in: Proceedings of the 2017 Conference on Interaction Design and Children, ACM, 2017, pp. 697–700.

[89] Matthew Berland, Don Davis, Carmen Petrick Smith, Amoeba: designing for collaboration in computer science classrooms through live learning analytics, Int. J. Comput. Support. Collaborat. Learn. 10 (4) (2015) 425–447.

[90] Eun Mee Lim, Patterns of kindergarten children's social interaction with peers in the computer area, Int. J. Comput. Support. Collaborat. Learn. 7 (3) (2012) 399–421.

[91] Jean Piaget, The language and thought of the child, vol. 10, Psychology Press, 2002.

[92] Lev Semenovich Vygotsky, Mind in Society: The Development of Higher Psychological Processes, Harvard university press, 1980.

[93] Don Norman, Things That Make us Smart: Defending Human Attributes in the Age of the Machine, Diversion Books, 2014.

[94] David Kirsh, The intelligent use of space, Artif. Intell. 73 (1–2) (1995) 31–68.

[95] Philip Heslop, Ahmed Kharrufa, Madeline Balaam, David Leat, Paul Dolan, Patrick Olivier, Learning extended writing: designing for children's collaboration, in: Proceedings of the 12th International Conference on Interaction Design and Children, ACM, 2013, pp. 36–45.

[96] Jochen Rick, Quadratic: manipulating algebraic expressions on an interactive tabletop, in: Proceedings of the 9th International Conference on Interaction Design and Children, ACM, 2010, pp. 304–307.

[97] Jochen Rick, Paul Marshall, Nicola Yuill, Beyond one-size-fits-all: how interactive tabletops support collaborative learning, in: Proceedings of the 10th International Conference on Interaction Design and Children, ACM, 2011, pp. 109–117.

[98] Todd Shimoda, Barbara White, Marcela Borge, John Frederiksen, Designing for science learning and collaborative discourse, in: Proceedings of the 12th International Conference on Interaction Design and Children, ACM, 2013, pp. 247–256.

[99] Panagiotis Apostolellis, Thanasis Daradoumis, Exploring the value of audience collaboration and game design in immersive virtual learning environments, in: Proceedings of the 9th International Conference on Interaction Design and Children, ACM, 2010, pp. 326–330.

[100] Anna Jordan-Douglass, Vishesh Kumar, Peter J. Woods, Exploring computational thinking through collaborative problem solving and audio puzzles, in: Proceedings of the 17th ACM Conference on Interaction Design and Children, ACM, 2018, pp. 513–516.

[101] Tiffany Tseng, Coram Bryant, Paulo Blikstein, Collaboration through documentation: automated capturing of tangible constructions to support engineering design, in: Proceedings of the 10th International Conference on Interaction Design and Children, ACM, 2011, pp. 118–126.

[102] Elliot Soloway, Jack Lochhead, John Clement, Does computer programming enhance problem solving ability? some positive evidence on algebra word problems, in: Computer Literacy, Elsevier, 1982, pp. 171–201.

[103] Daniel C. Richardson, Rick Dale, Looking to understand: the coupling between speakers' and listeners' eye movements and its relationship to discourse comprehension, Cogn. Sci. 29 (6) (2005) 1045–1060.

[104] Daniel C. Richardson, Rick Dale, Natasha Z. Kirkham, The art of conversation is coordination common ground and the coupling of eye movements during dialogue, Psychol. Sci. 18 (5) (2007) 407–413.

[105] Darren Gergle, Alan T. Clark, See what i'm saying?: using dyadic mobile eye tracking to study collaborative reference, in: Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work, ACM, 2011, pp. 435–444.

[106] Kshitij Sharma, Patrick Jermann, Pierre Dillenbourg, Displaying teachers gaze in a mooc: effects on students video navigation patterns, in: Design for Teaching and Learning in a Networked World, Springer, 2015, pp. 325–338.

[107] Sidney D'Mello, Andrew Olney, Claire Williams, Patrick Hays, Gaze tutor: a gaze-reactive intelligent tutoring system, Int. J. Hum. Comput. Stud. 70 (5) (2012) 377–398.

Paper 8:

Joint Emotional State of Children and Perceived Collaborative Experience in Coding Activities

Kshitij Sharma, Sofia Papavlasopoulou, and Michail N. Giannakos

In Proc. of Conference on Interaction Design and Children (IDC)