Looking at the Design of Making-Based Coding Activities through the Lens of the ADDIE Model

Sofia Papavlasopoulou and Michail Giannakos and
Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Abstract
Making has received growing interest in formal and informal learning environments. However, there is an acute need to investigate and get a deep understanding of the characteristics of making-based coding activities for children and how to appropriately design them. Over three years, we conducted empirical studies to investigate children's learning experience during making-based coding workshops, in which children used a block-based programming environment (i.e., Scratch) and collaboratively created a socially meaningful artifact (i.e., a game). This chapter aims to illustrate and discuss the learning design, using the ADDIE instructional model, and lessons learned based on a making-based coding workshop in Norway, named Kodeløypa.

Keywords: Coding, Making, ADDIE Model, Instructional Design, Instructional Model, Children

1. Introduction

Instructional design (ID) is a systematic process of designing the instruction of a learning event in an efficient manner. The ID process consists of phases that aim to investigate and determine learning objectives; develop learning materials, strategies, and assessment tools for evaluation; and accommodate an environment that encompasses successful learning outcomes (Morrison et al., 2019). Different ID models exist, with many of them based on the generic ADDIE model (**A**nalysis, **D**esign, **D**evelopment, **I**mplementation, and **E**valuation), an instructional model that describes a step-by-step process that can be used by instructional designers and practitioners who want to plan and create educational training and learning events. It presents a dynamic and flexible tool that can be adapted and used in many different contexts and has been widely applied in various educational projects (Morrison et al., 2019). The model was developed in the 1990s by Reiser and Mollenda and has five phases: Analysis, Design, Development, Implementation, and Evaluation. These phases describe specific actions and clear instructions that are simple and easy to adopt, but at the same time are quite generic. It is also possible to use the ADDIE model as a framework for the development of educational products (Alley & Jansak, 2001) and to provide a systematic approach that can be integrated into learning strategies (Hall, 1997; Pribadi, 2009).

Using the ADDIE model for ID provides a basis to determine – depending on the course and the context that is applied each time – the learning objectives, develop the activities of a course, and evaluate the learners' progress and the effectiveness of the instruction. In the analysis phase, the starting point of the ADDIE model, specialists should investigate and have a clear view of what the learners already know, define the course's needs and characteristics, and develop instructional strategies. The next phase is design, which deals with the learning objectives, the content, the planning of the course, and the media selection. Drawing upon all the knowledge gained and the decisions made in the previous phases, in the development phase course content and learning materials are created, assembling the resources that were created in the previous phase. Depending on the course, the ADDIE implementation phase may include management issues, but it basically aims to put into action the plan decided in the previous phases, evaluate its effectiveness, and ensure that everything performs as planned. Lastly, the evaluation phase represents a process that can happen at any of the stages of the ID process and aims to get feedback for improvement of the instruction and the materials, and to confirm that the learning goals and objectives of the course are met. Overall, it is important that the process during all the phases is systematic and specific to achieve the course's goals.

The purpose of this chapter is to frame a making-based coding activity that takes place in an informal setting, using the ADDIE instructional model. Linking these activities with an existing model provides a systematic approach to design; this action can respond to a corresponding lack of improvement in learning practices and outcomes, and contribute to the design of meaningful learning experiences for specific needs and contexts. In addition, when instructors are in a mindset that allows them to think in such a way that they can structure their intuitive decisions, by interacting using a specific model and theory they can reflect, understand, and consequently make the design of the activity better.

2. The ADDIE Instructional Model and Its Application in Coding-Related Activities

The ADDIE model has been extensively used to meet the needs of learning events related to coding activities. It has also been used as a development process for materials and software tools related to learning. In their study, Novák et al. (2018) have used the ADDIE model for the design of educational materials, supporting the use of the Arduino platform, for teaching coding in high schools. The five phases of the model helped them to use a strategy for the development of the educational materials; through the analysis they recognized the tasks that it is appropriate to include in the materials. Then, the learning materials were divided into lesson guides. Those had proposals with tasks that the teachers can do, including worksheets, where the focus is primarily on students, and, depending on the topic, each time they included relevant questions. The ADDIE model has also been applied for the development of multimedia instructional material for robotics education (Liu et al., 2008). Such materials are designed to engage students through an adventure story in the assembly of a robot and the coding of its operations to complete the mission of the story. Aiming to support university students and teachers with Object-Oriented Programming (OOP) learning, Oliveira and Bonacin (2018) suggested the design and implementation of OOP learning tasks with digital modeling and fabrication. The ID is based on the ADDIE model, offering a systematic process in this challenging project of using such technologies in formal educational settings.

The different settings in which the ADDIE model is applied are reflected also in its use for developing different kinds of multimedia. One example is an adventure game to support students' understanding of basic programming in vocational high school (Hidayanto et al., 2017). Based on the ADDIE model's five stages, the authors created and evaluated their game with students, measuring their learning based on their understanding of programming, and evaluating the software and visual communication. Similarly, Salahli et al. (2017), following the ADDIE model, developed a mobile application for the Scratch programming environment, supporting secondary school students to enhance their programming skills. In the analysis stage of the model, the authors not only analyzed the affordances of the Scratch programming language, but also determined their target group of students. After the design and development phase, the students tested the mobile application in the implementation phase using pre-post skill tests. Based on the results, students from the experimental group who used the mobile application had a significant increase in their programming skills over those in the control group.

The ADDIE model has very often been modified in practice in compliance with the different learning settings that are applied. Wu (2014) proposed a seven-phase ID model based on ADDIE for educating game programmers. The goal is to create a model that is customized to the needs of stakeholders, curriculum developers, content designers, and others. In that case, the seven phases included "Definition" (providing a clear goal), planning and verification (to meet the industry's expectations), Design, Development, Implementation, and "Continuous Improvement" (continuous reevaluation and redesign of the instructional content to fulfill changing requirements).

The ADDIE model has been successfully associated with good-quality design; definition of clear objectives; appropriately designed materials, media, and content; a well-arranged workload for teachers and students; and evaluation connected to the desired outcomes. Thus, supporting the design of informal educational settings with ADDIE model principles can only benefit the presentation of a suitable environment, efficiently facilitating students' experience and learning.

3. Kodeløypa Making-Based Coding Workshops

"Kodeløypa" is a making-based coding program that consists of workshops that are designed and implemented at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. The activities of the workshop are based on the constructionist approach, following the main principles of *making*. The duration of the workshop is approximately four hours and it is conducted in a largely informal setting, as an out-of-school activity. Students from 8 to 17 years old are invited to attend the workshop, which takes place in specially designed rooms, where students work in groups and are introduced to coding and tinkering. Students engage in numerous activities, such as coding digital robots and interacting with them, and creating games using Scratch and the Arduino hardware platform. Digital robots are made from recycled materials and an Arduino is attached to each one. Scratch for Arduino (S4A) is an extension of Scratch that provides 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 (see Figure 1). During the workshop students work collaboratively in triads or dyads. The design of the workshop also allows students without (or with minimal) previous experience to attend. Instructors of the activities of the workshop are student assistants, who are responsible for supporting each one of the students' teams as needed. The workshop has two main sessions.

Interacting with the robots: During the first session, the students interact with the digital robots. First, one instructor welcomes the students and guides them to be seated, giving a brief overview of the workshop. Each team of students uses one robot. Then with the help of the instructors, students work with a worksheet that is placed on the desks. First, each of the students answers the questions on the worksheet about the exact place and number of sensors and lights on the robots. In addition, students take a tutorial that includes instructions with examples and pictures, similar to the robots they are using. Via the examples shown in the tutorial, students understand exactly how they can interact with the robots. The tasks include the accomplishment of a series of simple loops; those loops will make the robots interact with the environment and perform actions such as turning on a light when sensors detect that the light is below a certain threshold. The different parts of the robots cannot be changed by the students, but they can touch and play with them as they want. This section lasts between 45 and 90 minutes, depending on the team; when everyone finishes the tasks there is a break before the next session.

Creating games using Scratch: This is the main session of the workshop and focuses on the creative implementation of simple game-development concepts using Scratch. Students get another paper-based tutorial with examples and visualizations to help them ideate their own game. The tutorial has examples of possible loops that students could use to create their games, including simple text explanations of basic computational thinking concepts. First, the instructors encourage the students to concentrate on discussing ideas for their games and to come up with a draft paper storyboard in collaboration with their team members. Then, again working in teams, students develop their own game by designing and coding using Scratch. To accelerate the children's progress, they are given already existing game characters and easy loops. The instructors support the students while working on their projects, providing help whenever they ask for it. Sometimes, instructors introduce more complex programming concepts on an individual level, depending on the needs of their project. Students create their games step by step, by iteratively coding and testing them. At the end, after completing the games, all teams play each other's games. The duration of this session is approximately three hours.

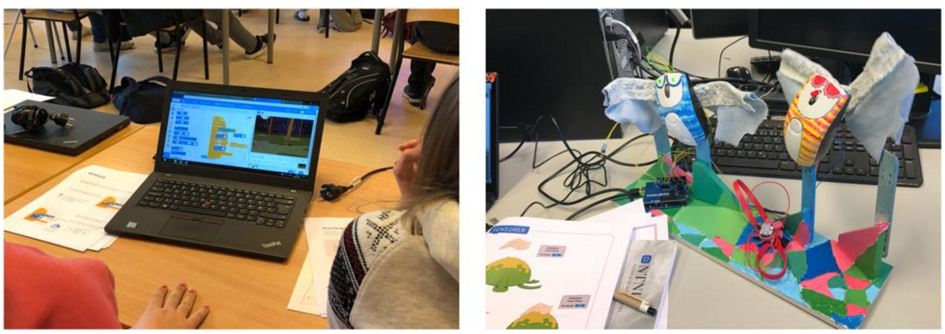

Figure 1: Children creating games using Scratch (left); interactive robots made from recycling materials (right).

4. Methodology

4.1 Focus Group

The study involved five participants: four instructors of Kodeløypa making-based coding workshops and one researcher who participated in focus group sessions discussing how to map those workshops to the instructional model (ADDIE). The researcher's role was to stimulate the brainstorming process and facilitate the sessions with her knowledge, assisting and guiding the instructors' discussions and thinking process. The instructors had a minimum of two years' experience in those workshops and were actively involved in both the instruction and the design decisions. In total two focus groups were conducted in order to finalize the description of the workshop based on the ADDIE model. During the first focus group session, the researcher presented general information about the existing instructional models, their benefits, and how they are applied, and then demonstrated a detailed description of the application of the ADDIE model in different settings. Then, everyone had a clear view of the ADDIE model, its phases, and an overview of how Kodeløypa making-based coding workshops should be investigated and approached in order to be mapped in the model. The aim was to brainstorm ideas and actions in the design of the workshops before and during their execution. Five posters, one representing each of the phases, were hanging on the wall. The task was to use Post-It notes and write down ideas, on an individual level at the beginning, reflecting on each of the five phases of the model.

At each phase of the ADDIE model, instructors spent 15-45 minutes brainstorming and writing down their ideas; then, they pinned the Post-Its on the respective poster and proceed to the next phase, repeating the same process. At the end of the session and when all the Post-Its were pinned on the posters, the researcher went through all of them and removed the non-relevant ones (if there were any), or asked for more explanations and wrote additional notes if needed. The next session was dedicated to discussing in detail all the ideas that were collected through the Post-It notes, ending up with the most important ones that would describe every aspect of the ADDIE model. Therefore, for each poster (representing the five phases of the ADDIE model) constructive discussions lasted for 30–45 minutes, until there was a general consensus among the participants on the ideas and decisions. The second focus group session lasted for approximately four hours. At the end the posters with all the ideas were collected by the researcher, who was responsible for organizing the results according to the categories corresponding to the five phases of the ADDIE model: Analysis, Design, Development, Implementation, and Evaluation.

4.2. ADDIE Instructional Model Applied to the Kodeløypa Making-Based Coding Workshops – Results

4.2.1. Analysis Phase

During the first phase of the ADDIE model, the focus is on analysis and identification of learners in order to determine the instructional goals and learning contexts. More precisely, it identifies the characteristics of the children (i.e., the learners in our case), their existing knowledge, their background and previous experience, as well as interests and attitudes. Having a clear view of the target audience is important, as it will guide the decisions in the next phases and also provide a realistic approach for the design. Thus, all the available information on the changes in Norwegian reality, including the plans of the Norwegian Government for the schools and educational system, were taken into consideration in how throughout the years Kodeløypa workshops have been evolving and have managed to adjust to circumstances. Three main categories of ideas emerged:

1) **The learners' background** is the main characteristic that all agreed was the most challenging, as it was very difficult to define in our workshops. In the Kodeløypa workshops participant children have various backgrounds, as there is an open call to local schools and no specific prerequisite knowledge from the children is targeted. The main goal is that all children get a general understanding of what coding is and participate in an enjoyable activity outside of the school context, by creating their own projects and collaborating with others. Therefore, the workshop has to be designed in a way that can be adapted to the needs of the children who are participating each time. The background of

the children may vary from having zero experience with coding to having a lot and being familiar with more advanced concepts for different reasons; for example, it depends on the school class (if it has coding as an elective subject, or a technology class) and each child's individual interest in coding, for instance trying to code at home or participating in local coding clubs. Consequently, the coding activity has to be adaptable and flexible. The workshop is thus designed for children without (or with minimal) previous experience in coding.

2) The primary target **age of the participants** is 10th grade; younger or older children can also participate, but each of the workshops should have a specific age group of children, carefully selected regarding age to have the same cognitive capacities. Concerning children's age, the design of the activity (interacting with robots and creating games) and the use of the Scratch programming language (suitable for all ages) provide flexibility and allow for the successful implementation of the workshop with participants from 8 to 17 years old.

As a conclusion to the previous two categories, children who are more knowledgeable in coding can create more advanced games, as the Scratch tool supports it. This is a very challenging process, which all the instructors admitted, because they have to have the experience and knowledge to support children in creating their games, from providing very basic to very advanced feedback. Therefore, they all concluded that they should be able to adapt each time depending on the group of children.

3) A third aspect that emerged was the **gender, motivations, and attitudes** of the children. Most of the time, girls are less exposed to coding than boys and have the impression that coding activities are not interesting for them; this is something that should be taken into consideration, and focus should be given to engaging them in such a way that they think it is not only for boys. Regarding children's motivations and attitudes, attention should be paid to the need to provide a very nice atmosphere during the activity, enhance children's interest in coding, and keep even the less-motivated children active in participating in the process.

4.2.2. Design Phase

In this phase, the most important aspect for the instructors was to define the learning objectives, but other aspects were also clarified by discussion. Kodeløypa workshops are designed to familiarize children with what coding is and to offer an easy way for them to be introduced to coding by creating their own projects through a pleasant, collaborative activity that lasts for approximately four hours. There are no lectures, but project-based learning methods are applied for high cognitive-level objectives. Instructors have the role of supporting the teams of children depending on their needs and on how they decide to approach the creation of their game, based on their decisions, efforts, and capabilities. Thus, each instructor tries to be in charge of observing two teams. Children working in teams are quite free to act on their own with the instructors as supporters.

The learning objectives of the workshop are implicit, and it turned out that they were never well defined. After the focus group the following learning goals emerged as expected outcomes from the workshop. The first two categories are connected to coding and problem solving, the third is related to collaboration, and the fourth to more general benefits and goals:

1) Learn basic coding skills:

- Learn basic computer science concepts (like loops and variables) and practices (like testing and debugging)
- Be able to create functional code by having an interacting "game"

Using the Scratch programming environment is a good choice, as the basic concepts and practices are well defined, but at the same time "hidden" behind colorful LEGO-like blocks used as commands for children to create their scripts.

2) Problem-solving skills in game creation and related actions to develop a solution that is new to them by designing and coding a program that meets a set of requirements:

- Investigate the parameters of the problem to guide their approach
- Split the problem into small components
- Generate ideas and alternatives (create their own approach, or explore several possible procedures that might be appropriate to the situation)
- Design a coherent solution
- Test the solution and iterate improvements to satisfy the requirements of the problem

3) Collaboration among the children during the process of creating something socially and personally meaningful:

- Decide on the topic that they will start to create
- Share their ideas freely and in a constructive way
- Plan what they have to do, when they will do it, and distribute roles and responsibilities if needed
- Discuss issues that occur and give feedback, with the goal to solve problems and be creative
- Make decisions in common for the design of the character and the story

4) General:

- Understand the functionality, possibility, and utility of coding environments
- Experience learning but also enjoyment
- Foster a sense of confidence
- Make coding more attractive to girls
- At the least get an understanding of how the creative process in technology happens in order for innovation to take place

4.2.3. Development Phase

In this phase, the outcome represents how the design will be put into action. Below are the two subjects that were discussed and appeared to be important to the instructors:

1) Together with the **project-based learning method**, the influential aspect of the pedagogical approach is the **"kindergarten approach to learning,"** with the spiral cycle of "imagine, create, play, share and reflect" which is a repeated process (Resnick, 2007). One element in this approach that was integrated is "inspiration," which is achieved through a warm-up activity of interacting with the robots and also showing participants similar examples of games. The children think and imagine what they want to create and then they try to make it real. When their games are at an appropriate level to be tested, they share them with the others, reflecting on their experience so far and getting new ideas to continue with their projects. The purpose is for the children to engage in the coding process

through exploration, iterations, using different concepts, and trying new elements, with the higher goal of creating the games they want.

2) The workshop, as described previously, is split into two sessions; it is a largely self-exploratory experience for the children, so the **learning materials** are worksheets and tutorials supporting this process. First, the worksheet is helpful for the children to interact with the robots. It includes questions regarding the position of the sensors, the light-emitting diode (LED) lights, the Arduino board. When it comes to the tutorials, two are needed, one for each session. For the first session, the tutorial helps with the control of the robots using S4A; and the other one, for the second session, aims to support children in the creation of the games and the use of Scratch. The robots tutorial has pictures that are similar to the robots children interact with and gives them simple examples of how to control them with the use of S4A. The second tutorial supports game creation and gives instructions for using Scratch. It starts with an introduction to the Scratch interface and the use of Scratch commands, beginning from the basics, for example explaining how to set the position of the characters, how to rotate elements, and also providing simple snippets of code for children to try out. Then, it gives examples of more and more complicated actions, like how to make the characters move, jump, and use collision detection or variables.

4.2.4. Implementation Phase

In this phase, the actual delivery of the instruction and the execution of the workshops were discussed. The ideas that emerged relate to what works well, the challenges the instructors are facing, and what they need to focus on in order to effectively and efficiently support the children's learning experience. Therefore, the following aspects appeared to be important:

1) Usually the children think they know more than they actually do, so **give them challenges** and **motivate them to use the tutorials**.
2) **Let the children decide their teams**. Friends collaborate better, as it is not easy to share ideas with someone you do not know.
3) **The robot part is a good starting point**; everyone participates without problems and uses the tutorial.
4) **Girls need more support and explanations** because they do not start the activities if they do not have sufficient understanding of what to do. Also, they read the tutorial more than the boys do.
5) **Starting to code is the most difficult part** and this is when the instructors should provide the most help and support to the teams. Also, force them to use the tutorial more.
6) In the case that someone in the teams knows more than others, an option is to motivate him/her to "teach" the other members, rather than having the attitude of creating everything on his/her own to show off; instead, let them all try to be **active participants**.
7) **Collaboration and discussion** are equally important to the other skills and should be enhanced.
8) **Playing each other's games** is a good motivation for all the children.

4.2.5 Evaluation Phase

In this phase the discussions during the focus group concluded with two main categories. The first refers to ways of assessing the children's learning experience in terms of instruction, how the workshop is designed, and how it is conducted, aiming to get feedback in an ongoing evaluation to improve the activities. The second refers to how to assess the children's learning experience in terms of the learning objectives, their engagement, attitudes, and behavior connected to the research objectives.

For the first category, after the end of the workshop the children are asked to fill in self-reflection cards individually, where they can anonymously and freely express their thinking about the process and the experience they had. A few questions help the children to elaborate: for example, what they liked most and what they did not like, what they would like to be added to the activity, what they think they have learned, and if they had fun. Also, at the end, a question asks them to write whatever they want and feel it will be useful to share.

For the second category, researchers are responsible for collecting qualitative and quantitative data using various data instruments, including:

- The code the children create in Scratch at different stages, approximately every hour, including the final version.
- Assistants take field notes, conducting structured observations to monitor actions like children's moments of frustration and examples of fun, as well as what kind of help they were getting from the instructors and when.
- Semi-structured interviews with the children at the end of the workshop. The interviews have the purpose of getting as much information as possible from the children on how they experience the making-based coding activity. The questions are related to what difficulties they face during the game creation experience and what is the easiest part, how collaboration is among the members of the team, what frustrates them, and what impresses them.
- Pre-post Scratch evaluation questions. In order to measure the learning gain from their participation in the workshop, the children have to fill in a pre-knowledge acquisition test consisting of coding questions with snippets of code in Scratch, increasing in difficulty, following instructors' suggestions on what the children can acquire from the workshop.
- Using eye-tracking glasses during all parts of the activity, the children's gaze is captured to give insights into their various cognitive mechanisms, predict their progress, and get deeper into their behavior.

All the above-mentioned data have as a higher goal to get a comprehensive view of children's learning experience, extract principles for the design of the workshop, and make further decisions.

5. Discussion

This chapter considers how a making-based coding activity, conducted in an informal setting, can be described, mapped, and benefit from an instructional model. In this case, the ADDIE ID model was used in order to discuss the design and development of a learning experience. During two focus group sessions with the instructors of the making-based coding workshop, we discussed the development of the workshop based on the model's five phases; after the sessions, the most important aspects were revealed, concerning what to think about in the design of coding workshops when applying the ADDIE model. In addition, we supported the fact that it is possible to fit an activity outside of formal settings into an instructional model that has not applied it before, and benefit from its systematic approach.

During the focus group, from the researcher's point of view, who was also the facilitator, it was difficult sometimes to guide the discussions. The instructors' final decisions, as Post-It notes and ideas, turned out to be more intuitive and not expressed properly. This is due to the fact that they were not familiarized with the model's five phases and what exactly is needed to be addressed in each of them. In particular, the discussions in the analysis and design phase were the most challenging; on the other hand, from all the five phases, the one with the most effective discussions was the implementation phase. When the workshop was initially designed, the focus was to familiarize the students with coding, show them the possibilities of a programming environment, help them become aware that they can be creators rather than simply consumers of technology, and overall give them an idea about computer science. This makes it difficult to determine exactly the identity of the learners, because the characteristics of the possible participants of the workshop are very broad, regarding both age and background. One solution is to design flexible and adaptable activities (Papavlasopoulou et al., 2019). However, based on their experience, in the analysis phase of the ADDIE model the instructors managed to focus on the most prominent characteristics of the learners, like age, background, gender, motivation, attitudes, and so on. In their study, Ozdilek and Robeck (2009), analyzing the responses of instructional designers in various areas of education, found that the analysis step of the ADDIE model was the most challenging and that most of the attention is given to learner characteristics compared to other elements. One of the important aspects that was shown is the importance of designing an enjoyable activity, which is in line with similar workshops (Norouzi et al., 2019).

Regarding the learning objectives that were specified in the design phase, they were apparently in the instructors' minds, but it was difficult for them to express and explain what they were thinking about exactly before taking part in the focus group. After the discussions, the learning goals were clear and well defined by everyone, placing them on common ground; it is apparent that the focus is not only on learning coding but also on the overall experience. Furthermore, the learning materials have to be in line with the learning goals and support the smooth execution of the workshop (Liu et al., 2008); their development has to be carefully and strongly connected with the design of the workshop. Novák et al. (2018) used the five phases of the ADDIE model as a strategy to develop educational materials for the use of the Arduino platform. During the discussions for the implementation phase, it was obvious that instructors were more active and efficient, without getting much help from the researcher to explain and guide them; discussing the execution of the workshops, needs, and challenges was something more natural to them. However, the implementation phase needs constant revision based also on the results of the evaluation, which requires an appropriate approach. For example, researchers and instructors should agree on the evaluation strategy, and then the researchers should communicate the results to the instructors. In this way they will introduce a teaching approach and design decisions, with the higher goal of creating a beneficial learning experience for the children.

In general, despite some challenges, the instructors found the ADDIE model really interesting and very helpful for understanding, framing, and advancing the design of the Kodeløypa making-based workshops. In the focus groups, important aspects were revealed of what to consider as the main characteristics in order to develop a similar workshop, indicating how it has to be approached: for example, have clear learning objectives; consider the most important aspects of the learners' identity; during the activities act accordingly, like motivating them to use the tutorials more, to get help, and to support boys' and girls' teams depending on their needs and capabilities. In addition, some of the discussions and ideas described have not been explicitly implemented in the Kodeløypa workshops in their current state, but examining the use of the instructional model and how it is implemented gave instructors the opportunity to think about future improvements and plans, like deciding on the use of appropriate evaluation techniques and how to implement the workshops more effectively, with the correct choices based on the circumstances and the characteristics of the learners. Reflections on the basic structure of the workshop also helped the instructors to see things more clearly and develop more ideas, despite the fact that these were not expressed or were explicit from the beginning of the focus group. This indicates that the ID process gives the instructors the opportunity to think about and understand the steps and the process of how to follow a specific model and theory, consequently leading to a better design of the learning activity and helping them to become better (Khalil & Elkhider, 2016).

Future work should focus on adjusting the model, such as adding and reassuming the phases of the model, depending on the needs of the learning activity that has to be developed. This will allow a better-designed experience for the learners and customize the needs of the instructors or other stakeholders who are interested each time. This study is limited in that the ADDIE model is used in one case; we suggest the need for confirmation in other similar cases, which will show evidence and contribute to the use of the ADDIE design model to inform, guide, and lead to successful educational experiences (Smith & Boling, 2009). However, we should maintain a critical point of view, and not forget the limitations of the model and the fact that it has been criticized as not always being very effective (Bichelmeyer, 2004).

Acknowledgments

This work is supported by 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) and the "Learn to Machine Learn" (LearnML) project, under the Erasmus+ Strategic Partnership program (Project Number: 2019-1-MT01-KA201-051220).

REFERENCES

Alley, L.R., & Jansak, K.E. (2001). The ten keys to quality assurance and assessment in online learning. *Journal of Interactive Instruction Development*, 13(3), 3–18.

Bichelmeyer, B. (2004). The ADDIE model – A metaphor for the lack of clarity in the field of IDT. AECT 2004 IDT Futures group presentations IDT Record.

Hall, B. (1997). *The Web-Based Training Cookbook with CDRom*. Hoboken, NJ: John Wiley & Sons.

Hidayanto, D.R., Rahman, E.F., & Kusnendar, J. (2017). The application of ADDIE model in developing adventure game-based multimedia learning to improve students' understanding of basic programming. *2017 3rd International Conference on Science in Information Technology (ICSITec)*, pp. 307–312. IEEE.

Khalil, M.K., & Elkhider, I.A. (2016). Applying learning theories and instructional design models for effective instruction. *Advances in Physiology Education*, 40(2), 147–156.

Liu, E.Z.F., Kou, C.H., Lin, C.H., Cheng, S.S., & Chen, W.T. (2008). Developing multimedia instructional material for robotics education. *WSEAS Transactions on Communications*, 7(11), 1102–1111.

Morrison, G.R., Ross, S.J., Morrison, J.R., & Kalman, H.K. (2019). *Designing Effective Instruction*. Chichester: John Wiley & Sons, 480 p..

Norouzi, B., Kinnula, M., & Iivari, N. (2019). Interaction order and historical body shaping children's making projects—a literature review. *Multimodal Technologies and Interaction*, 3(4), 71.

Novák, M., Kalová, J., & Pech, J. (2018). Use of the Arduino platform in teaching programming. *2018 IV International Conference on Information Technologies in Engineering Education (Inforino)*, pp. 1–4. IEEE.

Oliveira, G.A.S., & Bonacin, R. (2018). A method for teaching object-oriented programming with digital modeling. *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, pp. 233–237. IEEE.

Ozdilek, Z., & Robeck, E. (2009). Operational priorities of instructional designers analyzed within the steps of the Addie instructional design model. *Procedia-Social and Behavioral Sciences*, 1(1), 2046–2050.

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.

Pribadi, B.A. (2009). Model desain sistem pembelajaran. *Jakarta: Dian Rakyat*, 35.

Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. *In Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, pp. 1-6.

Salahli, M.A., Yildirim, E., Gasimzadeh, T., Alasgarova, F., & Guliyev, A. (2017). One mobile application for the development of programming skills of secondary school students. *Procedia Computer Science*, 120, 502–508.

Smith, K.M., & Boling, E. (2009). What do we make of design? Design as a concept in educational technology. *Educational Technology*, 49(4), 3–17.

Wu, P. (2014). A game programming instructional design model. *Journal of Computing Sciences in Colleges*, 29(6), 57–67.