NTNU – Trondheim
Norwegian University of
Science and Technology

Social Game-based Classroom Applications

Tintin Trong Hoang

Master of Science in Computer Science
Submission date: June 2015
Supervisor: Alf Inge Wang, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Problem Description

The goal of this project is to develop various prototypes for real-time social game-based classroom applications and run experiment with these applications in class. This project builds upon a previous master project on the same topic.

In this project, the students must study the state-of-the-art within game-based learning, and then develop applications for gamifying the classroom. The development is in HTML5/Java Script and using well known server technologies.

Abstract

The goal of this research was to explore how game modes that are based on reaction and completeness affect players enjoyment, learning, engagement, concentration, motivation, emotional involvement and social experience. It also discusses which one of the game modes had the greatest impact on these properties.

To explore these game modes, a functional prototype of a quiz game called "Quiz-a-tron" was developed. After testing the game modes on students the results where summarized and analyzed.

This thesis also briefly discuss the process of using HTML5, Node.js, Socket.IO, Express.js, JavaScript, and other state of the art technology for creating knowledge-based games.

Sammendrag

Målet med denne forskningen er å utforske hvordan ulike spillmoduses som er basert på reaksjon og kompletthet påvirker spillerens underholdning, læring, engasjement, konsentrasjon, motivasjon, følelsesliv og sosiale opplevelse. Det blir også diskutert hvilken av disse spillmodusene som har størt påvirkning på disse egenskapene.

For å undersøke disse spillmodusene ble en funksjonell prototype av et quiz-spill kalt "Quiz-a-tron" utviklet. Etter å ha testet disse spillmodusene på studenter ble resultatene oppsummert og analysert.

Det blir også kort diskutert prosessen med å bruke HTML5, Node.js, Socket.IO, Express.js, JavaScript, og annet nytt teknologi for å skape kunnskapsbaserte spill.

Preface

This thesis is the result of the subject TDT4900 - Computer and Information Science, Master Thesis at the Department of Computer and Information Science, under Faculty of Information Technology, Mathematics and Electrical Engineering, at the Norwegian University of Technology and Science

The project was carried out during the summer semester of 2015. The idea was brought up by my supervisor Alf Inge Wang, whom I would also like to thank for his help throughout the whole process.

It is assumed that the reader have a basic understanding of computer science.

Trondheim, 2015-06-19

Tintin Hoang

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

This chapter will give an introduction to the context, motivation, goals and structure for this project.

1.1 Context

For many years, games has been used as a learning tool for educating children. Empirical evidence (e.g., see Rosas (2003)) support that computer games has an positive effect on academic achievement, cognitive abilities, motivation towards learning, and improving attention and concentration. Additionally, research by Døvik and Hestad (2011) suggest that also university students benefit from integrating games as part of lectures. By using games, students are more motivated to participate in class, work harder on assignments and coming prepared to lectures. Additionally, it could stimulate a better learning environment by invoking emotions such as competitiveness and social bonding in the players.

One way of integrating games in a tradition classroom lecture is by having the students and the instructor play knowledge-based multiplayer games. An example of such a game is *Kahoot!*, which has frequently been integrated into the lectures held at NTNU and other Norwegian schools.

1.2 Motivation

The study of using knowledge-based games as an educational tool in a classroom setting could use more research in which methods are most effective at promoting learning outcome in students. Today, most knowledge-based games are quiz games that follow the tested and true formula of giving the players multiple choice questions and scoring performance by how fast they answered correctly.

By examining different ways of structuring a quiz game, in form of different game modes, it is possible to discover better ways to promote learning.

It is also interesting to create quiz games with state of the art technology, and assess how fitting they are for creating this kind of educational applications.

1.3 Project Goals

The project goals is to first create a functional quiz game prototype called *Quiz-a-tron*. It will be designed with modifiability, usability and platform independence in mind. This prototype will then be configured to different game modes and tested on groups of students. Based on data collected from forms and interview, it will be concluded if the newer game modes are more effective at learning than the old traditional.

There will also be a description of the development process where the experiences may be of use to other researchers.

1.4 Project Structure

The thesis is divided into six parts, ordered chronologically by where they naturally belong in the process.

1. **Introduction:** An introduction to the project context, project motivation, project goals, and project structure.

2. **Research Approach:** A description of the research questions, strategy, paradigms and data gathering methods used.
3. **Literature Study:** A short overview of the research that were studied and used to design the game.
4. **Design and Implementation:** Description of the game modes explored, software quality attributes that were focused on, a overview of the architecture and a walkthrough of the gameplay.
5. **Testing and Discussion:** A description of the testing strategy and testing participants. The results from the tests will be summarized and other information will also be discussed.
6. **Conclusion:** A conclusion of the project and further work.

Part II

Research Approach

Chapter 2

Research Questions and Methodology

This chapter will present the research questions, paradigm, strategy, and data generation methods.

2.1 Research Questions

From the project goals and motivation, the following research questions was formulated:

RQ1:
How do game modes that are based on reaction and completeness affect players enjoyment, learning, engagement, concentration, motivation, emotional involvement and social experience.

RQ2:
Which of the explored game modes has the biggest impact on players enjoyment, learning, engagement, concentration, motivation, emotional involvement and social experience.

RQ3:
How does a player's *player type* affect their preference in game mode.

RQ4:

How well fitted are HTML5, Node.js, Socket.IO, Express.js, JavaScript, and other state of the art technology for creating knowledge-based games (quiz game).

2.2 Research Paradigm

The philosophical paradigm followed in this project was *interpretivism*. Oates (2013) provide a definition on interpretivism as: "Interpretive research in IS and computing is concerned with understanding the social context of an information system: the social processes by which it is developed and construed by people and through which it influences, and is influenced by, its social setting." It is based on the opinion that there are fundamental difference between the subject matters of the natural and social sciences, which means that the methods of the natural sciences cannot be used in the social sciences. The goal of interpretivist research is to understand and interpret the meanings in human behaviour rather than to generalize and predict causes and effects. It is believed that there is impossible to completely remove all bias and no observations are unaffected by the how a researcher chooses to conceptualize them based on prior theory or previous experiences. An important thing to note is that the researcher always influence the study and its findings, making it very unlikely that another researcher will experience the same results.

2.3 Research Strategy

The research strategy chosen for this project was *Design and Creation*. This research strategy focuses on developing new IT products, also called *artifacts*. While there are many types of artifacts, this project focused on constructing a instantiation – a working system that demonstrates that constructs, models, methods, ideas, genres or theories can be implemented in a computer-based system (see March and Smith (1995)). The IT artifact is part of the contribution to knowledge, but also act as a tool to examine the effect of novel ideas.

2.4 Data Generation Methods

A data generation method is the means by which empirical (field) data or evidence is being produced. The data used for evaluating the game modes in this project will collected by using questionnaires and interviews.

2.4.1 Questionnaire

Questionnaire is a pre-defined set of questions assembled in a pre-determined order. Respondents are asked to answer to answer the questions, via multiple choice options, thus providing data that can be analysed and interpreted.

The questionnaires will be *self-administered*, which have the benefit of saving time and makes it possible for more people to be asked to participate. Another benefit is that the respondents are less likely to try to please the researcher by giving what they perceive to be the "correct" or "desired" answer. However, a important drawback is that it could lead to a lower response rate than if it was *researcher-administered* – since it is harder for people to refuse to complete the questionnaire when the researcher is standing in front of them.

The questions should be unambiguous, specific and objective. The questions in this project was formed as closed questions which forced the respondent to choose from a range of pre-defined answers. This makes the responses easier to compare and analyze, but with the risk that respondents answers quickly in a slapdash way without thinking much about their responses.

2.4.2 Interview

Interview is a particular kind of conversation between people where, at least at the beginning of the interview if not all the way through, the researcher controls both the agenda and the proceedings and will ask most of the questions. There will be held *unstructured interviews* after the players have tested the prototype. The goal is to let the players express thoughts and insight

that they would be hard to communicate through questionnaire responses alone.

Part III

Literature Study

Chapter 3

Research

This chapter will give a short overview of the main points from research articles, and how they can be used in Quiz-a-tron.

3.1 Player Types

Bartle (1996) is a co-creator of MUD (Multi-User Dungeon), a text-based adventure game that had the then-unique attribute of being able to be played alongside other human players. Following a debate about what people wanted out of a MUD, he consolidated the comments posted on the forum and developed a model which placed the players into four groups with different play style interests. These were called: Killers, Achievers, Explorers, and Socializers. Kyatric (2013) give the following summary:

- **Killers** like to provoke and cause drama and/or impose them over other players in the scope provided by the virtual world. Trolls, hackers, cheaters, and attention farmers belong in this category, along with the most ferocious and skillful PvP (player versus player) opponents.
- **Achievers** are competitive and enjoy beating difficult challenges whether they are set by the game or by themselves. The more challenging the goal, the most rewarded they tend to feel.
- **Explorers** like to explore the world - not just its geography but also

the finer details of the game mechanics. These players may end up knowing how the game works and behave better than the game creators themselves. They know all the mechanics, short-cuts, tricks, and glitches that there are to know in the game and thrive on discovering more.

- **Socializers** are often more interested in having relations with the other players than playing the game itself. They help to spread knowledge and a human feel, and are often involved in the community aspect of the game (by means of managing guilds or role-playing, for instance).

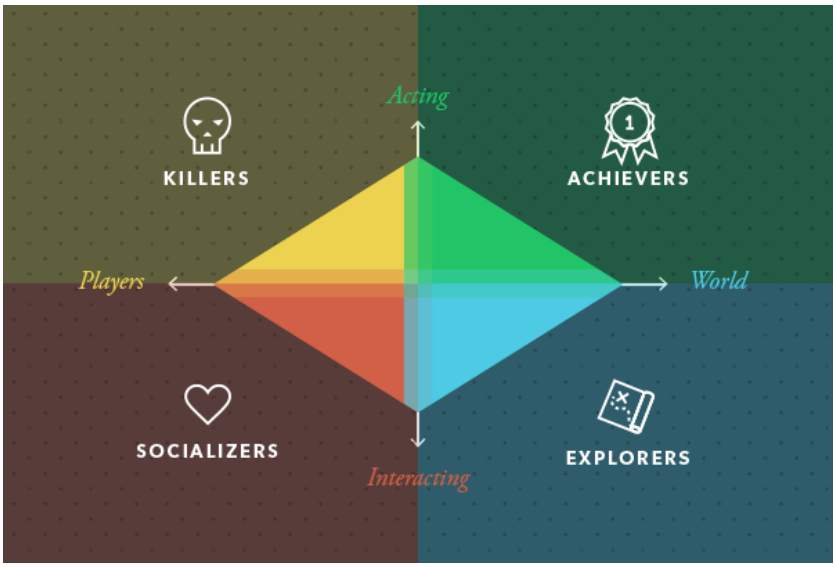

Figure 3.1: Bartle's player types

Bartle represented their relationship as shown in figure 3.1. The horizontal axis goes from an emphasis on players to an emphasis on the environment, and the vertical axis goes from acting with something to acting on something. The four extreme corners of the graph show the four typical playing preferences associated with each quadrant.

Killers and Achievers are mostly interested in acting on things or people, treating things and people as external objects. On the other hand, Explorers and Socializers prefered a deeper level of interacting with things or other people, focusing on internal qualities.

Similarly, Killers and Socializers wish to control how they are able to play dynamically with others in the game world, while Achievers and Explorers want to control their relationships with the developer-defined objects in and properties of the game world itself.

The players tended to belong to a primary player type, but could over time drift towards other types influenced by their mood, situation or goal in the game. The main purpose of Bartle's theory was to fine-tune specific mechanics in order to provide the game with content that was interesting for targeted player types.

While there has been much critisisim and debate about how accurate Bartle's player types carry over to other game genres, it may be a good tool to help examine the players behavior in Quiz-a-tron and customize it to better fit their wishes and needs.

3.2 Heuristics for designing instructional computer games

Malone (1980) provides heuristics or guidelines for designers of instructional computer games based on his intuition and former experience. The goal is to identify what essential characteristics makes a game fun and how this can be used in educational games. Malone explores three areas that makes an activity fun or rewarding for it's own sake rather than some external reward: challenge, fantasy and curiosity. These areas will be explained below.

3.2.1 Challenge

Malone performed a survey on grade school children about their preferences in computer games. The single feature of the games that correlated most strongly with preference was whether or not the game had a goal. It was also important that the attainment of the goal was uncertain and thereby gave the players a feeling of excitement and motivation.

Goals and challenges are captivating because they engage a person's self-esteem. Success in a computer game, like success in any challenging activity,

can make people feel better about themselves. The opposite side of this principle, is of course, that failure in a challenging activity can lower a person's self-esteem. Balancing the difficulty is of great importance since there is a point where challenge easily becomes discouraging rather than inviting.

One method to apply this principle to a game is simply have variable difficulty level so players have the opportunity to play at an appropriate level for their ability and skills. Malone describes many ways this can be accomplished. One way is to adjust the game difficulty either by letting the players manually select one they feel are fitting, or let the game automatically scale the difficulty based on their performance. Another approach is to have multiple level of goals. The player have some criteria they must fulfill to complete the game, but there are also optional bonus criteria for the players that would like more challenge. It is also possible to have score keeping and compare the results in a high score list. The players are then motivated to set their own goal, like beating their previous record or ranking higher than other players.

3.2.2 Fantasy

Computer games often become more interesting when they use fantasy as a context. In general, games that include fantasy show or evoke images of physical objects or social situations not actually present. One relatively easy way to try to increase the fun of learning is to take an existing curriculum and overlay it with a game in which the player progresses toward some fantasy goal, or avoids some fantasy catastrophe, depending only on whether the player's answers are right or wrong.

Malone distinguishes between extrinsic and intrinsic fantasy. In extrinsic fantasies, the fantasy depends on the use of the skill but not vice versa. For example, a computer "race car" game where students' cars move along a race track depending on how fast they answer arithmetic problems. The fantasy does not affect the skill used and the same fantasy could be used with completely different kinds of problems (e.g. how fast players can spell words).

In intrinsic fantasies, on the other hand, not only does the fantasy depend on the skill, but the skill also depends on the fantasy. For example, "Darts"

is a computer game where the students throw darts on balloons placed on a number line by inputting fractions. In this case the players are able to improve their skill (inputting correct fractions) by observing the fantasy (see if the arrows are hitting too short or too far). In other words, the elements of the skill being practiced are actually embedded in the fantasy being employed.

Malone believes that in general intrinsic fantasies are more interesting and instructional than extrinsic fantasies. He argues that they often:

1. Naturally indicate how the skills being taught can be used to accomplish a real world goal.
2. Employ relevant imagery which may provoke learner memory assimilation and recall.
3. Provide naturally occurring constructive feedback, since problems are naturally presented in terms of the elements found within the fantasy world.

Regarding the emotional aspects of fantasies, it is difficult to know what emotional needs people have and how these needs might be partially met by computer games. However, that computer games that embody emotionally-involving fantasies like war, destruction, and competition are likely to be more popular than those with less emotional fantasies.

3.2.3 Curiosity

Curiosity is the motivation to learn, independent of any goal seeking or fantasy-fulfillment. Based on the work of Berlyne (1965) and Piaget and Cook (1952), we can evoke a learner's curiosity by providing games that have environments with informational complexity. An optimally complex environment will be one where the learner knows enough to have expectations about what will happen, but where these expectations are sometimes unmet. It is also important that the environments are neither too complicated nor too simple with respect to the learner's existing knowledge. A right balance should be struck between being novel and surprising, but not completely incomprehensible.

Malone differentiates between two types of curiosity: sensory and cognitive. Sensory curiosity involves the attention attracting value of changes or patterns in the light, sound, or other sensory stimuli of an environment. Computer games can appeal to sensory curiosity by the use of audio and visual effects. Cognitive curiosity can be thought of as a desire to bring better "form" to one's knowledge structures. The way to engage learners' curiosity is to present just enough information to make their existing knowledge seem incomplete, inconsistent, or unparsimonious. The learners are then motivated to learn more in order to make their cognitive structures better-formed.

Based on Malone's research we want to design a quiz game that provide multi-level challenges. In the new game mode the players main goal should be to find at least one correct answer to a question, the more advanced should be motivated to find all the correct answers, and finally the best students should be compelled to compete on the leaderboard. While an intrinsic fantasy would be preferred, it is difficult to develop custom games for each lecture. The quiz will therefore possibly feature an extrinsic fantasy because one of the most important attributes is that it is multi-purpose. The questions should invoke the players curiosity, and motivate them to return to the curriculum when they find their answers incomplete.

3.3 Flow

Sweetser and Wyeth (2005) have written an article describing a concept called "flow". Flow is described as "... an experience so gratifying that people are willing to do it for its own sake, with little concern for what they will get out of it, even when it is difficult or dangerous". This concept has then been adapted to games to create a new model, "GameFlow", which consists of eight elements – concentration, challenge, skills, control, clear goals, feedback, immersion, and social interaction.

In summary, games must keep the player's concentration through a high work-load; but the tasks must be sufficiently challenging to be enjoyable. The player must be skilled enough to undertake the challenging tasks, the tasks must have clear goals so that the player can complete the tasks, and the player must receive feedback on progress towards completing the tasks. If the

player is sufficiently skilled and the tasks have clear goals and feedback, then he or she will feel a sense of control over the task. The resulting feeling for the player is total immersion or absorption in the game, which causes them to lose awareness of everyday life, concern for themselves, and alters their sense of time. The final element of player enjoyment, social interaction, does not map to the elements of flow, but is highly featured in the literature on user-experience in games. People play games to interact with other people, regardless of the task, and will even play games they do not like or even when they do not like games at all.

Sweetser's research points out the importance of having a game that immerse the players and help them concentrate. The quiz should try to minimize down-time where the players could easily be distracted by other things in the lecture hall and the user interface have to be easy to use so they can fully concentrate on the questions instead of the system.

Part IV

Design and Development

Chapter 4

Game Modes

This chapter will describe the game modes that were tested using Quiz-a-tron.

4.1 Reaction

Reaction is included as the game mode that will represent the status quo and serve as a baseline to compare the other game modes against. The players are presented with a question with only one correct answer. If the player submit a correct answer, they will receive a base score added with a bonus score. The bonus score is calculated by taking the base score and multiply it with the percentage of time left with two decimals precision.

```
score += questionBaseAward + questionBaseAward * (1 -
↪  (answer.time / question.time).toFixed(2));
```

Ex: A player submit an correct answer after spending 3 seconds on a 10 second limit question. The base score is 500. The player will receive 500 + 500 * (1 - (3 / 10)) = 850 points.

When a player has submitted an answer, all answers will be locked so the player is unable to change their submission.

This game mode has the advantage of being easy to understand and have been used for a long time in many games.

4.2 Completeness

Completeness is a novel game mode with the goal of encouraging the player to answer as complete as possible. The players are presented with a question that have at least one correct answer. If none of the the answers submitted are wrong, the player receive a score of the base score multiplied with the percentage how many correct answers they were able to identify. Additionally, by not submitting one of the incorrect answers the player will increase their multiplier with 1 for each question. However, if the player submit a incorrect answer, the player will get 0 points for the question and the multiplier will be reset. It is also possible to not answer anything. The player will then get 0 points, but will retain their multiplier.

```
if (wrongAnswers == 0) {
        score += (correctAnswers/maxAnswers).toFixed(2) *
        ↪  questionBaseAward * multiplier;
        multiplier += 1;
}
else {
        multiplier = 1;
}
```

Ex: A player submit two correct answers out of three correct answers. The base score is 500 and the player has answered the last question correctly, thus having a multiplier of 2. The player will receive 2/3 * 500 * 2 = 670 points.

The player are free to change their submission as many times as they like until the time runs out, at that point it will be graded.

The completeness game mode experiment with the players willingness to take risks on submitting answers they are not completely sure about in hope of gaining more points. The more questions a player answers correctly in a row, the more motivated he is to continue doing so to retain his multiplier. It also

simulates how tests are graded today, where it does not matter how long a student spent on a test, but how complete his answers are.

Chapter 5

Software Quality Attributes

This chapter will describe the software quality attributes that was kept in mind when designing Quiz-a-tron and reasoning of why they are important.

5.1 Modifiability

Modifiability deals with change and the cost in time or money of making a change, including the extent to which this modification affects other functions or quality attributes. There is a cost of preparing for change as well as a cost for making a change. This is an important attribute since the prototype will probably undergo rapid changes, both during initial development and if more game modes would be added in the future. The following modifiability tactics were used to prepare for changes.

Split Module
: The modification costs of a module with large capability is usually high. Refining the module into several smaller modules should reduce the average cost of future changes.

Increase Semantic Coherence
: If the responsibilites in a module do not serve the same purpose, they should be placed in different modules. The purpose of moving a responsibility from one module to another is to minimize the chance of

side effects affecting other responsibilities in the original module.

Encapsulate
Encapsulation introduces an explicit interface to a module. This interface includes an application programming interface (API) and its associated responsibilites. Encapsulation reduces the likelihood for changes from one module to propagates to other modules.

Restrict Dependencies
Restrict dependencies is tactic that restricts the modules that are given module interacts with or depends on and restricting access to only authorized modules.

Refactor
Refactor is a tactic used when two modules are affected by the same change because they are (at least partial) duplicates of each other. Code refactoring is a mainstay practice of agile development projects, as a cleanup step to make sure that teams have not produced duplicative or overly complex code; however, the concept applies to architectural elements as well. Common responsibilities (and the code that implements them) are "factored out" of the modules where they exist and assigned an appropriate home of their own.

Abstract Common Services
In the case where two modules provide not quite the same, but similar services, it may be cost-effective to implement the services just once in a more general (abstract) form. Any modification to the common service would then need to occur in just one place, reducing modification costs.

5.2 Usability

Usability is concerned with how easy it is for the user to accomplish a desired task and the kind of user support the system provides. Once a system is executing, usability is enhanced by giving the user feedback as to what the system is doing and by doing and by allowing the user to make appropriate responses. There are no obvious way to design the quiz game architecture to improve usability. Instead, there were used a technique called user-centered interaction design to evaluate the quiz game by testing it on users. Their

feedback was crucial in shaping a desired user interface and user experience. This attribute is important since the quiz will be played by students with varying skill in use of technology and should therefore be as helpful and intuitive as possible. Additionally, good usability help create immersion since the players can dedicate their full attention to the game instead of figuring out the system.

5.3 Platform Independence

Platform independence in software means that you can run some code with little or no modification on multiple platforms. This was achieved by using platform independent language like JavaScript and playing on a browser, which run on all operating systems. This attribute is important since the students will play the game on different devices with different hardware and software platforms.

Chapter 6

Underlying Technology

This chapter will describe the underlying technology in Quiz-a-tron.

6.1 Node.js

Node is the foundation for the back-end portion of the game. Node.js provides an asynchronous event-driven architecture and a non-blocking I/O API that optimizes an application's throughput and scalability. This solves concurrency challenges that often arise in many server-side programming languages. These traits makes it ideal for creating scalable web applications that operates in real-time.

By utilizing this framework, it is possible to accelerate the development time and opens the possibility of using the Express framework, and the Socket.IO library.

6.2 Express

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is designed for building single-page, multi-page, and hybrid web applications. Adding Express makes it much simpler to serve static files such as HTML,

CSS and JavaScript. Express is also used to adjust logging output, and create an easy-to-use environment for Socket.IO. While it originally was used to navigate between pages in the early iterations of the game, this responsibility has later been delegated to jQuery Mobile.

6.3 Socket.IO

Socket.IO is a JavaScript library for realtime web applications. It enables realtime, bi-directional communication between web clients and server. It has two parts: a client-side library that runs in the browser, and a server-side library for node.js. Both components have a nearly identical API. Like node.js, it is event-driven.

Socket.IO primarily uses the WebSocket protocol with polling as a fallback option, while providing the same interface. Although it can be used as simply a wrapper for WebSocket, it provides many more features, including broadcasting to multiple sockets, storing data associated with each client, and asynchronous I/O.

6.4 jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

6.5 jQuery Mobile

jQuery Mobile is a HTML5-based user interface system designed to make responsive web sites and apps that are accessible on all smartphone, tablet and desktop devices. It is built jQuery and jQuery UI foundation, and offers Ajax navigation with page transitions, touch events, and various widgets. Its

lightweight code is built with progressive enhancement, and has a flexible, easily themeable design.

6.6 ClassyCountdown

ClassyCountdown is a jQuery plugin that allows you to easily create clean circular countdowns by utilizing HTML5 canvas for rendering. This is based on the work of Marius Stanciu, but have been modified to fit this projects needs.

6.7 FastClick.js

FastClick is a simple, easy-to-use library with the purpose of eliminating the 300ms delay between a physical tap and the firing of a click event on mobile browsers. The delay caused by the browser waiting to see if you are actually performing a double tap. This library makes the quiz game generally more responsive, especially in the reaction game mode.

Chapter 7

Architecture

This chapter will give an overview of how the different modules fit together and explain their responsibilities. A class diagram of the architecture can be seen on figure 7.1.

7.1 index.js

This module is the first file being run at startup and is responsible for setting up a game server by using Express and Socket.IO modules.

7.2 quizgame.js

This module is handling all game logic and maintaining the game states of all game rooms.

7.3 server.js

This module is responsible for communication between the player or host and quizgame.

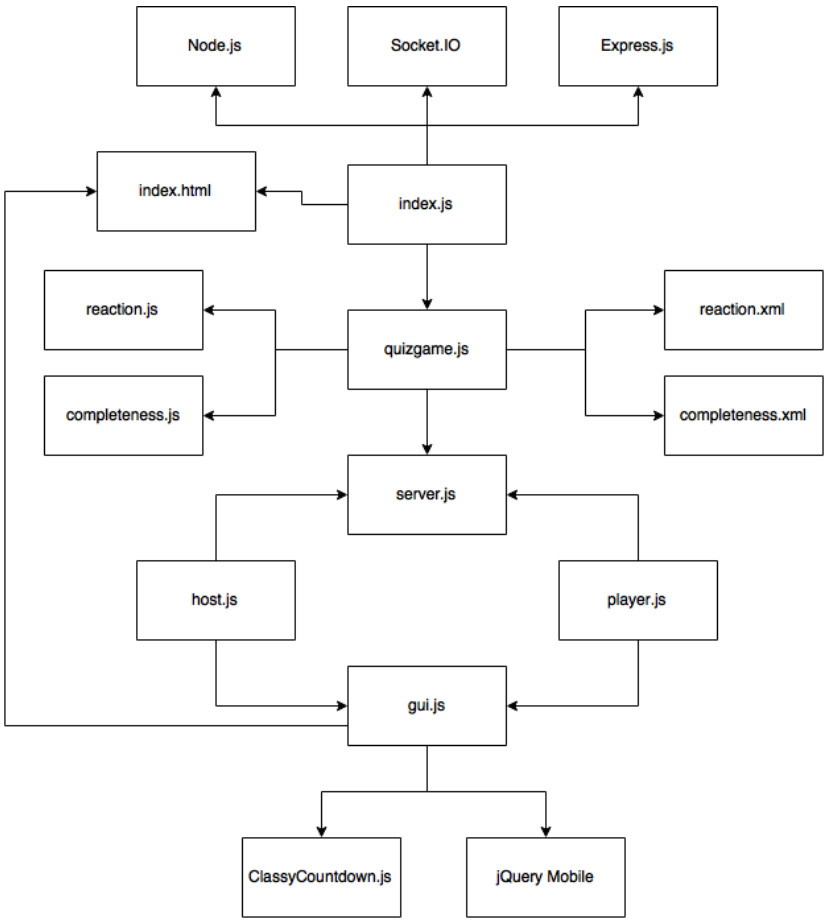

Figure 7.1: Class diagram of Quiz-a-tron

7.4 host.js

This module is responsible for maintaining information about the game such as GameID. It is using gui.js to display the host information.

7.5 player.js

This module maintain information such as playerID and calls gui.js for displaying information for the player.

7.6 gui.js

This module is responsible for functions like changing pages and formatting information so they can be displayed properly.

7.7 reaction.js

This module holds all game logic associated with the reaction game mode. This is for example the calculation method for computing score. It also holds the filepath to the questions for this game mode.

7.8 completeness.js

This is has the same functionality and responsibility as reaction.js, but for completeness game mode instead.

Chapter 8

Gameplay

The chapter will provide a walkthrough of the gameplay of Quiz-a-tron. The gameplay has been illustrated in the state diagram seen in figure 8.1.

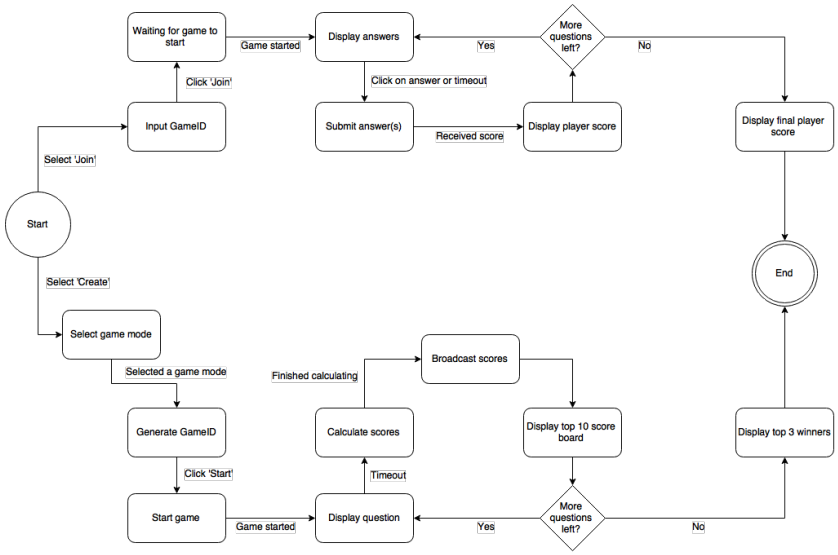

Figure 8.1: State diagram of Quiz-a-tron

When players first enter the website they are shown the intro screen (figure 8.2). At this point the user are able to choose between creating a game and acting as a host, or joining a game and acting as a player.

If chosing to become a host, the user is then prompted to select which game mode their game should use (figure 8.3). When selected, the host will then generate and display a GameID which the players use to join the game (figure 8.4).

If the user had chosen to be a player instead from the start screen, they are able to input nickname and a GameID to join (figure 8.5).

When the host has chosen to start the game, a question will be shown with a timer ticking down (figure 8.6). Meanwhile, the player will get some answers that they are able to choose from.

If the selected game mode was reaction, the players answers will be locked after the first answer has been selected (figure 8.7). On the other hand, if completeness was selected the player are free to change their decision until the time runs out (figure 8.8).

When the time has run out, the players answers will be sent to the host and a score will be calculated and returned. The host will display a score board of the top 10 players along with the solution of the question (figure 8.9). The players will see their own placement on their screen (figure 8.10). This loop of displaying of questions/answers and score and solution will repeat until all questions have been used. The host will then show the top 3 players as winners.

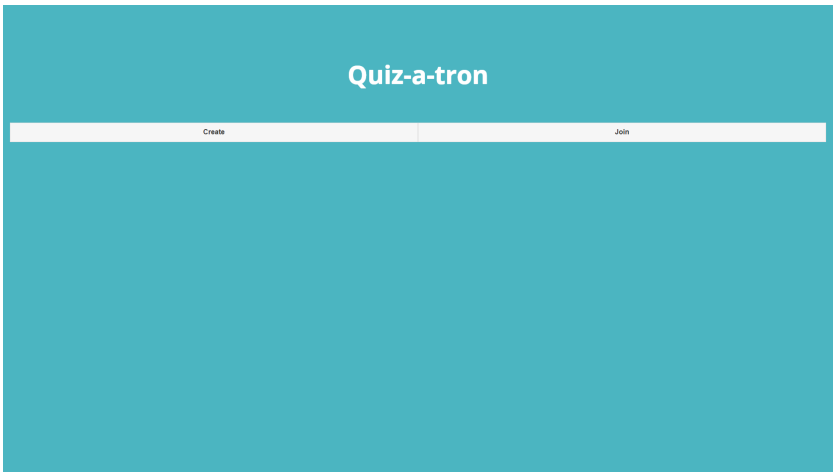

Figure 8.2: Start Screen

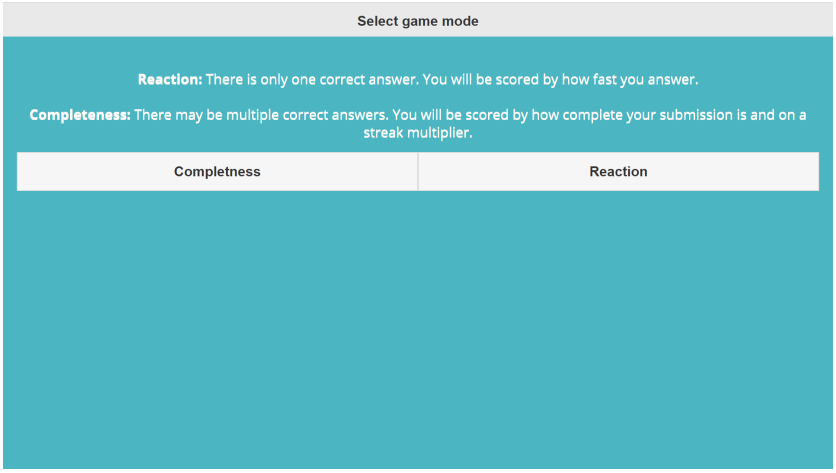

Figure 8.3: Host - Select Game Mode Screen

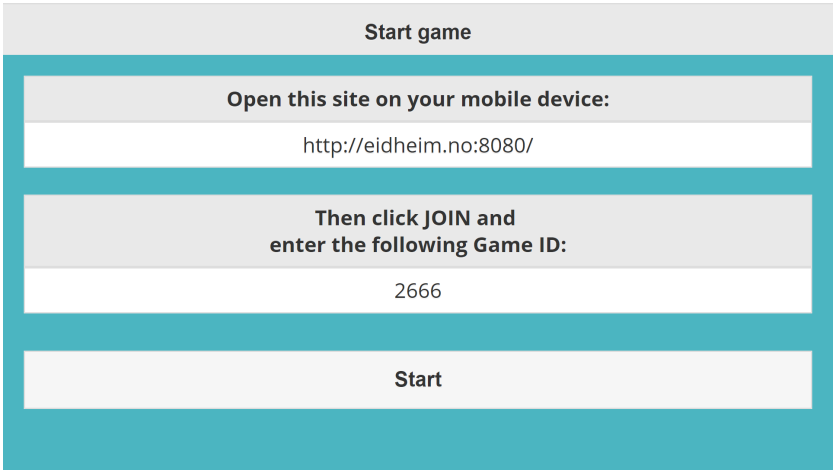

Figure 8.4: Host - Start Game Screen

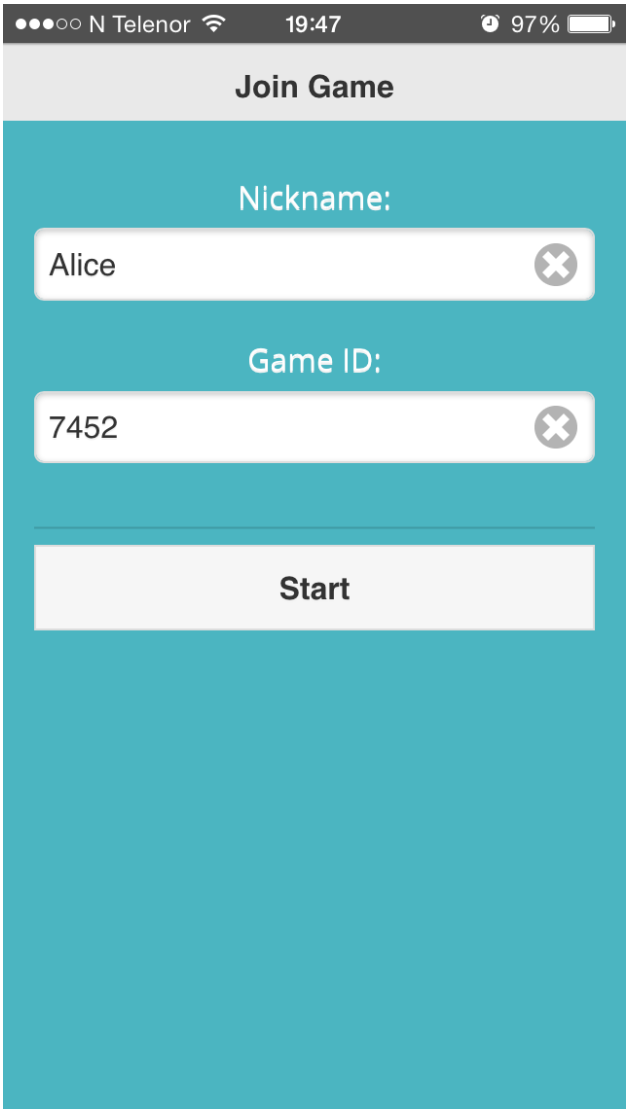

Figure 8.5: Player - Join Game Screen

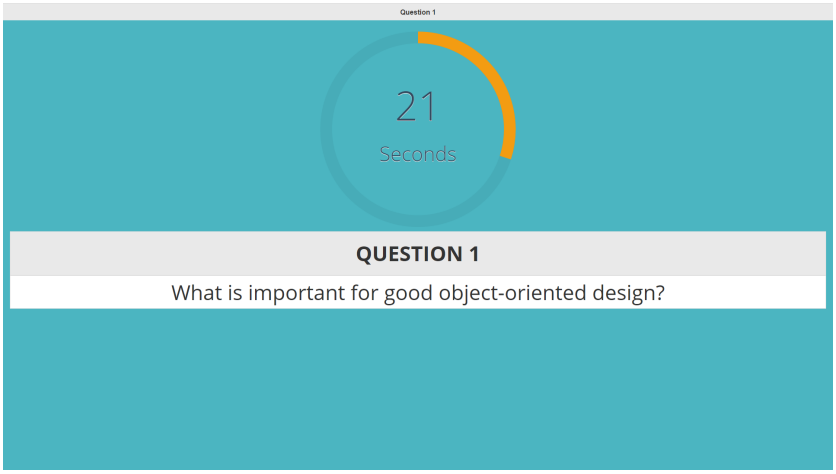

Figure 8.6: Host - Question Screen

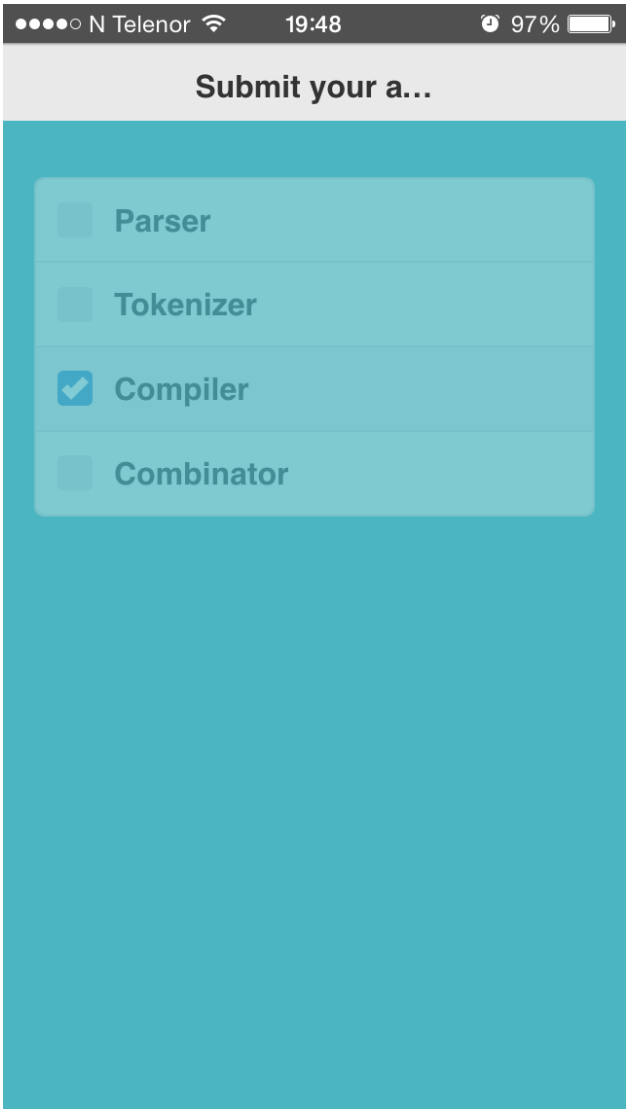

Figure 8.7: Player - Answer Screen for Reaction

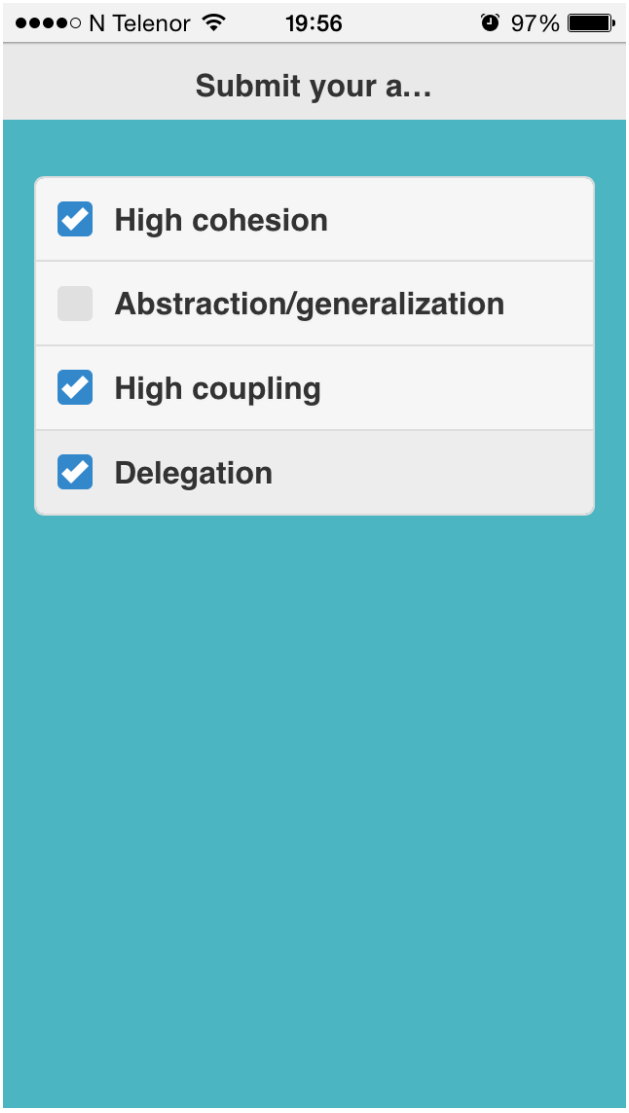

Figure 8.8: Player - Answer Screen for Completeness

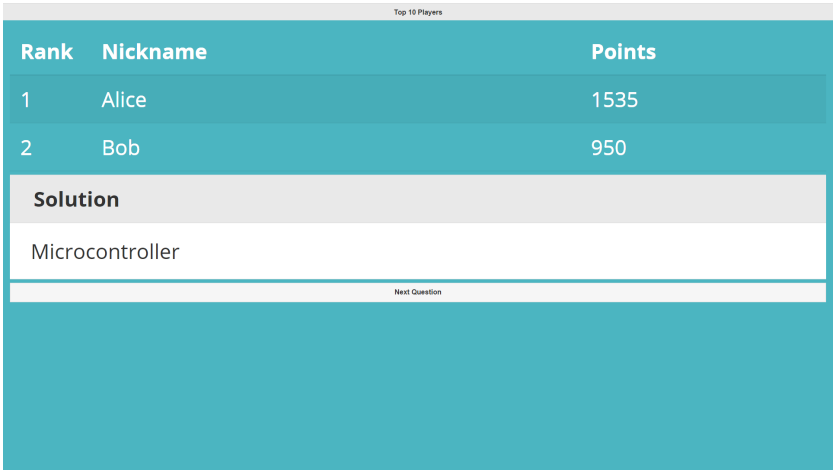

Figure 8.9: Host - Score Screen

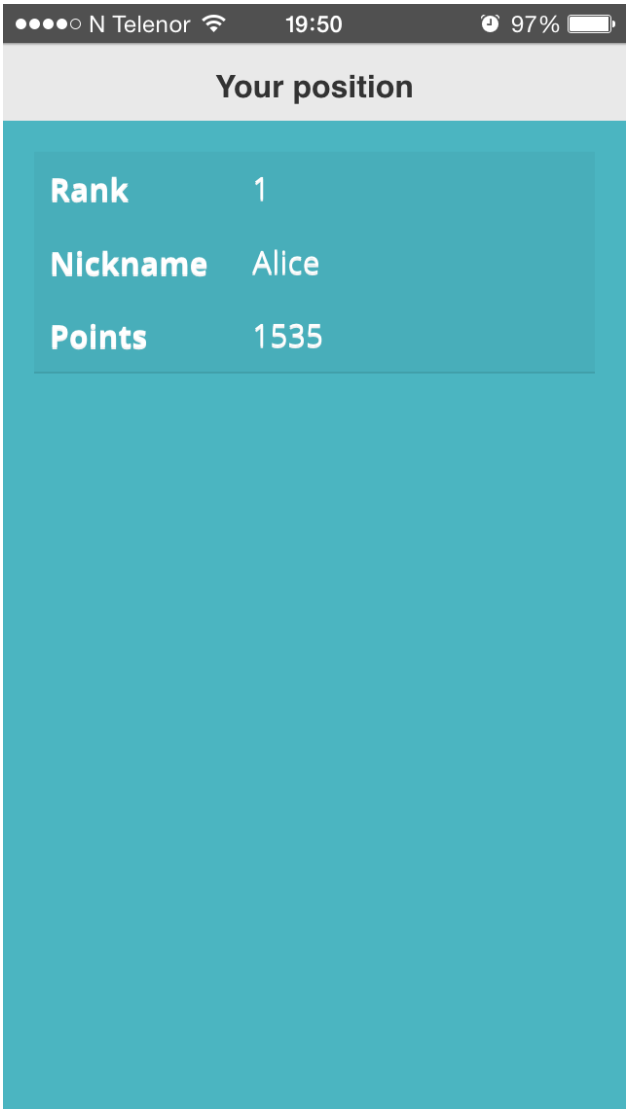

Figure 8.10: Player - Score Screen

Chapter 9

Development History

This chapter will describe the experience of developing the quiz game and the problems that arose.

9.1 First Iteration

Since the author had no prior experience with using technology such as Node.js, Socket.IO or Express.js, much development time was initially spent on learning the basic functionality, and developing small test applications. The first iteration of the quiz game was based on an article by Terpstra (2013), where he described his implementation of a multiplayer game called *Anagrammatix*. This was a game played with a large monitor and two smartphones. The two players where shown an anagram by the main screen and the goal was to be the first one to select the correct word on their smartphone.

Using this as a foundation, the code was studied and then modified to suit the needs for a quiz game. The quiz game architecture was based on three roles: server, host and client (player). The server was responsible for communication between the host and the player. The host was responsible for maintaining the game state and displaying the questions. The client displayed the answers and sent the selected answers back to the server. The client and host never communicated directly but had to go through the server

that relayed the information for them.

One problem arose when it was time to implement the different game modes as interchangeable modules. The game modes required implementation of more functionality and more communication between client and host. For each time the host and client wanted to send information to each other, it had to go through two links (client – server and server – host). If the player client wanted something sent to the server, it first would have to send the message to the server, have the server broadcast the message back to everyone in the room, and the host would then act on it. Additionally, it was difficult to have polymorphism for the game modes modules because of other architectural problems and combined made it hard to further build upon the current frame.

Another challenge was making sure information was being displayed correctly on all types of devices. At the beginning regular CSS and a layout consisting of nested divs was used, but soon proved to be very cumbersome. The author then discovered SS3 flexbox layout, which seemed to solve the problems. Development time was invested in learning how to use flexbox correctly, and it worked on some browsers, but many mobile devices had problems displaying it properly.

The was also used a jQuery plugin called ClassyCountdown to display the remaining time on questions. It was heavily modded to fit the needs of Quiz-a-tron. When the prototype changed, it was sometimes difficult to make it work with this plugin.

9.2 Second Iteration

The author, now with more experience and insight in the potential problems, eventually decided to rewrite the code for the quiz game from scratch. The architectural elements was still the same as the previous one, but this time all game logic for all game instances was maintained by the server. The host did nothing else than displaying the information it got from the server. The communication was now much easier to handle as it removed a middle man and the client could just send answer data directly to the server.

The displaying of information was now done by using jQuery Mobile. This

meant that time again had to be spent on learning how to use this library.

Part V

Testing and Discussion

Chapter 10

User Testing Strategy

This chapter will outline the testing procedure and explain the reasoning behind the participants being chosen and the customization of quiz questions.

10.1 Testing Procedure

The test participants will be chosen randomly from campus Gløshaugen at NTNU and asked to volunteer in a study of classroom applications. Half of the participants will be playing the *reaction* game mode first, and the other half will begin with the *completeness* mode. After finishing the game, they will be asked to fill out the appropriate form for their game mode. These forms ask about how they feel the game mode help support the studied properties such as fun, motivation, engagement, learning etc. The reasoning behind this method is to reviewed each game mode individually and reduce bias from a game mode affecting the perception of the other. In other words, the participants will give a "pure" first impression.

The participants will then play the other game mode. After finishing this mode they will fill out a comparison form about which game mode they felt were most successful in supporting the studied properties.

Next, the participants will be given a form to measure usability of the system. These results will help establish if the design for usability in the quiz game

has been successful and remove the doubt that poor usability affecting the results from the previous forms.

The author will also observe how the participants are using the system. At the conclusion of the test, the author will then ask them about behaviour they have observed and ask about feedback. This way the author can discuss thoughts and opinions from the participants that are not easily inferences from the forms alone.

All forms can be found in their entirety in the appendix.

10.2 Testing Participants

Ideally, the testing would have been performed on students in various lectures from different disciplines. However, because the development of the quiz game took longer than expected, most classes were already finished when it was ready. Instead, the testing was done on 30 engineering students and the questions were based on first year curriculum that is mandatory for students that study scientific subjects on NTNU. The quiz game was therefore serving the purpose as a refresher tool instead of a lecture summary. Testing concluded after 30 participants because data saturation had been reached led to little new information being gained.

Chapter 11

User Testing Result

This chapter will present the summarized results from the distributed forms.

11.1 Reaction Form

The average age for the participants was 22.7 with a 73.3% male and 26.7% female distribution.

Statement	Strongly disagree	Disagree	Neither agree or disagree	Agree	Strongly agree
I have fun [...]	0%	0%	6.7%	80.0%	13.3%
I am engaged [...]	0%	0%	6.7%	80.0%	13.3%
I am learning [...]	0%	0%	20.0%	66.6%	13.3%
I am concentrating [...]	0%	0%	6.7%	40.0%	53.3%
I am motivated to learn [...]	0%	0%	46.7%	53.3%	0%
I am emotionally involved [...]	0%	0%	6.7%	66.7%	26.6%
I have a good social experience [...]	0%	0%	6.7%	40.0%	53.3%
I often play games in my spare time.	0%	26.7%	6.7%	13.3%	53.3%

Table 11.1: Results from reaction form

11.2 Completeness Form

The average age for the participants was 22.3 with a 56.3% male and 43.7% female distribution.

When I play games I am defined by a focus on...	Percentage
... winning rank, and direct peer-to-peer competition	33.3%
... attaining status and achieving preset goals quickly and/or completely	26.7%
... socializing and a drive to develop a network of friends and contacts	13.3%
... exploring and a drive to discover the unknown	26.6%

Table 11.2: Results from player types in reaction form

Statement	Strongly disagree	Disagree	Neither agree or disagree	Agree	Strongly agree
I have fun [...]	0%	0%	13.3%	66.7%	20.0%
I am engaged [...]	0%	0%	6.7%	60.0%	33.3%
I am learning [...]	0%	0%	13.3%	80.0%	6.7%
I am concentrating [...]	0%	0%	13.3%	46.7%	40.0%
I am motivated to learn [...]	0%	6.7%	33.3%	46.7%	13.3%
I am emotionally involved [...]	0%	0%	40.0%	40.0%	20.0%
I have a good social experience [...]	0%	6.7%	20.0%	46.7%	26.7%
I often play games in my spare time.	6.7%	13.3%	6.7%	20.0%	53.3%

Table 11.3: Results from completeness form

When I play games I am defined by a focus on...	Percentage
... winning rank, and direct peer-to-peer competition	33.3%
... attaining status and achieving preset goals quickly and/or completely	33.3%
... socializing and a drive to develop a network of friends and contacts	20.0%
... exploring and a drive to discover the unknown	13.3%

Table 11.4: Results from player types in completeness form

11.3 Comparison Form

Statement	Reaction	Completeness
I had the most fun [...]	33.3%	66.7%
I was most engaged [...]	50.0%	50.0%
I learned most when playing [...]	16.7%	83.3%
I was most concentrated [...]	40.0%	60.0%
I was most motivated [...]	23.3%	76.7%
I was most emotionally involved [...]	46.7%	53.3%
I have a good social experience [...]	16.7%	83.3%

Table 11.5: Results from comparison form

11.4 SUS Form

Statement	Strongly disagree	Disagree	Neither agree or disagree	Agree	Strongly agree
I think that I would like to use this system frequently.	0%	10.0%	23.3%	56.7%	10.0%
I found the system unnecessarily complex.	56.7%	30.0%	10.0%	3.3%	0%
I thought the system was easy to use.	0%	3.3%	0%	56.7%	40.0%
I think that I would need the support of a technical person to be able to use this system.	66.7%	23.3%	6.7%	3.3%	0%
I found the various functions in this system were well integrated.	0%	6.7%	20.0%	63.3%	10.0%
I thought there was too much inconsistency in this system.	30.0%	66.7%	3.3%	0%	0%
I would imagine that most people would learn to use this system very quickly.	0%	0%	0%	53.3%	46.7%
I found the system very cumbersome to use.	40.0%	46.7%	13.3%	0%	0%
I felt very confident using the system.	3.3%	0%	30.0%	46.7%	20.0%
I needed to learn a lot of things before I could get going with this system.	63.3%	30.0%	3.3%	3.3%	0%

Table 11.6: Results from SUS form

Brooke (1986) explains how to calculate SUS score. To calculate the SUS score, first sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For items 1, 3, 5, 7, and 9 the score contribution is the scale position minus 1. For items 2, 4, 6, 8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of SU. SUS scores have a range of 0 to 100.

$$S_1 = \frac{3*4+17*3+7*2+3*1+0*0}{30} \approx 2.7$$

$$S_2 = \frac{17*4+9*3+3*2+1*1+0*0}{30} \approx 3.4$$

$$S_3 = \frac{12*4+17*3+0*2+1*1+0*0}{30} \approx 3.3$$

$$S_4 = \frac{20*4+7*3+2*2+1*1+0*0}{30} \approx 3.6$$

$$S_5 = \frac{3*4+19*3+6*2+2*1+0*0}{30} \approx 2.8$$

$$S_6 = \frac{9*4+20*3+1*2+0*1+0*0}{30} \approx 3.3$$

$$S_7 = \frac{14*4+16*3+0*2+0*1+0*0}{30} \approx 3.5$$

$$S_8 = \frac{12*4+14*3+4*2+0*1+0*0}{30} \approx 3.5$$

$$S_9 = \frac{6*4+14*3+9*2+0*1+0*0}{30} \approx 2.8$$

$$S_10 = \frac{19*4+9*3+1*2+1*1+0*0}{30} \approx 3.5$$

$$S_Score = (S_1 + S_2 + ... + S_10) * 2.5 = 81$$

Sauro (2011) states that the average SUS score is 68. The usability of this quiz game would place it in the 85th percentile and graded A- as seen in figure 11.4. Regarding sample sizes Sauro (2013) say that: "Five is often a magic number for early-phase usability studies. Confidence intervals will be rather wide, but the average SUS score will be surprisingly stable." With this in mind, the SUS score being presented is propably fairly accurate as it has a sample size of 30.

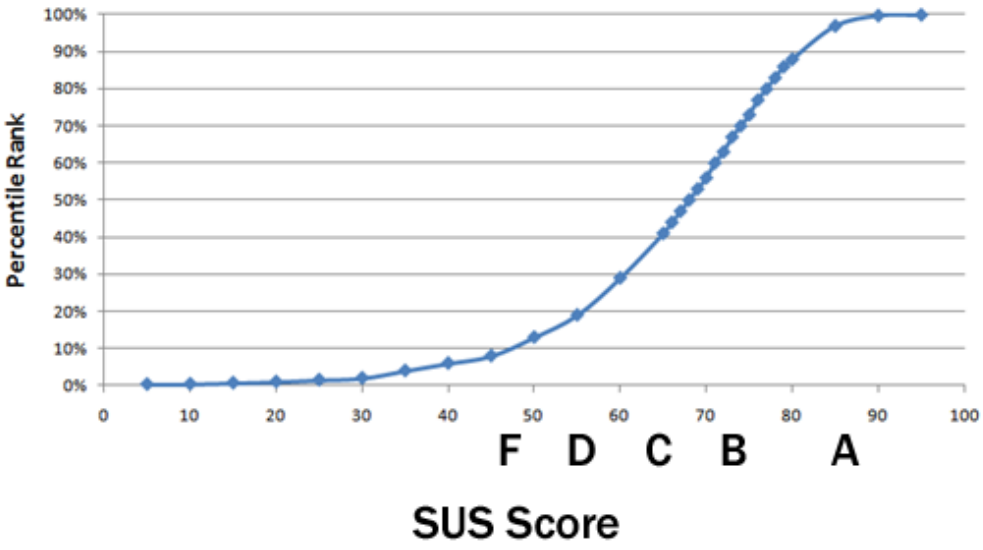

Figure 11.1: Raw SUS scores as percentiles and grades.

Chapter 12

Discussion

Because score is calculated by how fast you answer correctly in reaction, most players submitted their answer within the first 3 seconds, even when given a time limit of 30. Answering quickly was the top priority, and players often skimmed the text, resulting in them picking the first answer that seemed right. Shortly afterwards they often became frustrated when they realized they had hastily picked an incorrect answer.

The reasoning behind this rush was that you had to answer quickly to be able to climb the score board and if you took too long, you got so few points that you could not win anyways. When the players was given questions that required mental calculation, most picked a random answer without giving any effort in solving it. This of course have a negative effect on learning.

Players that answered quickly had a lot of "dead time", where they were simply waiting for the others to finish. This affected their immersion and learning. They said that they did not really think about the question after answering since it was not possible to undo your submission. The only learning they got was therefore the reveal of the solution after each question.

Interestingly, this "dead time" was helpful in promoting social bonding. When they knew they had gotten a question wrong, they often had a outburst of frustration and they laughed together with their friends. This helped increase the engagement in the game.

Players also noted that this game mode was a bit superficial and shallow. It

was hard to be motivated to learn when the questions was not given much time investment. You either knew the correct answer of the top of your head, or you guessed at the first one that seemed right.

On the other hand, the completeness game mode was more punishing for wrong answers and players second guessed themselves more often. In this game mode you had to be careful with the answers, since a wrong one could jeopardize all the correct answers you would have scored otherwise.

There was an analogy where reaction was compared to a sprint, while completeness was a marathon. Many liked the slow pace of the game mode, since they got more time to recall what they had learned.

It was noted that the level of frustration was much lower in this game mode. Players felt that it was easier to show your skill in the subject instead of being tested on how fast you could read. The motivation was also higher since players that did not know much about the subject could stick to the answers that were obvious, while others was able to show off their knowledge by correctly identify all the right answers.

Many players also said that it was more engaging since the comeback potential was higher. Despite having few points early in the game, a player could easily climb the score board if they did not lose their multiplier.

Completeness was however less social during the questions. Since you could change your answers until the time ran out, the most competitive players had little incentive to discuss the question with their neighbors.

Most players identified themselves as Killers or Achievers. There were however, no direct link between player type and preference of game mode. There could still be potential for customizing these game modes to better cater for the intended player types.

Part VI

Conclusion

Chapter 13

Conclusion

In this chapter we will return to the research questions asked in section 2.1, and summarize the conclusions.

13.1 RQ1

Chapter 12-13 show that players enjoyment, learning, engagement, concentration, motivation, emotional involvement and social experience differ between the game modes reaction and completeness that was explored. The rules of the game mode and how the scoring is calculated will heavily influence the players experience of these properties.

13.2 RQ2

The game mode that had the biggest impact on these mentioned properties is the completeness game mode. While many players had arguments for which areas the game modes excelled at it was generally felt that completeness was a better way of promoting learning.

13.3 RQ3

The player types seem to have little effect on their overall experience of the game modes. There could still be potential for customizing these game modes to better cater for the intended player types.

13.4 RQ4

As described in chapter 7-10, HTML5, Node.js, Socket.IO, Express.js, JavaScript, and other state of the art technology are very fitting for creating knowledge-based games. The problems that arouse during the development process was not caused by the choice of technology, but rather the author's inexperience with using them.

Chapter 14

Further Work

This chapter will outline some possible ideas for further work.

14.1 Explore Other Game Modes

During this project only one novel game mode was explored. With the current state of Quiz-a-tron, it would be very easy to incorporate other game modes and test if they are more fitted for promoting learning.

14.2 Incorporate A Fantasy

This version of Quiz-a-tron does not use any fantasy. It could have improved the user experience and enjoyment of the game if there had been some kind of fantasy theme put over the scoring system. Instead of climbing numbers on a score board, the players performance could for example be visualized as horses on a racing track.

14.3 Make The Scoring of Completeness More Elegant

Many test participants felt that the scoring system of completeness game mode was confusing, especially the bonus multiplier. It was hard to predict and min-max your score and they would rather have something easier to grasp.

14.4 Player Interaction

Some test participants wished for more player interaction. They suggested adding power ups to the game that could either help yourself or hinder your opponents.

Bibliography

Bartle, R. A. (1996). Hearts, clubs, diamonds, spades: Players who suit muds.

Berlyne, D. E. (1965). *Structure and direction in thinking*. John Wiley & Sons, Inc.

Brooke, J. (1986). Sus - a quick and dirty usability scale.

Døvik, K. and Hestad, J. A. (2011). Lecture quiz 3.0. Master's thesis.

Kyatric (2013). Bartle's taxonomy of player types (and why it doesn't apply to everything). `http://gamedevelopment.tutsplus.com/articles/bartles-taxonomy-of-player-types-and-why-it-doesnt-apply-to-everything--gamedev`

Malone, T. W. (1980). What makes things fun to learn? heuristics for designing instructional computer games.

March, S. T. and Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15:251–266.

Oates, B. J. (2013). *Researching Information Systems and Computing*. SAGE Publications Ltd, London, 1st edition.

Piaget, J. and Cook, M. (1952). *The origins of intelligence in children*. International Universities Press.

Rosas, Ricardo, e. a. (2003). Beyond nintendo: design and assessment of educational video games for first and second grade students. *Computers and Education*, 40:71–94.

Sauro, J. (2011). Sustisfied? *Little-Known System Usability Scale Facts. User Experience Magazine*, 10(3).

Sauro, J. (2013). 10 things to know about the system usability scale (sus). http://people.dsv.su.se/~hercules/papers/Textsumsummary.html.

Sweetser, P. and Wyeth, P. (2005). Gameflow: A model for evaluating player enjoyment in games.

Terpstra, E. (2013). Building multiplayer games with node.js and socket.io.

Part VII

Appendices

Reaction form

* Required

1. **Age** *

2. **Gender** *
   Mark only one oval.
   - ◯ Male
   - ◯ Female

3. **I have fun playing the reaction game mode.** *
   Mark only one oval.
   - ◯ Strongly disagree
   - ◯ Disagree
   - ◯ Neither agree or disagree
   - ◯ Agree
   - ◯ Strongly agree

4. **I am engaged when playing the reaction game mode.** *
   Mark only one oval.
   - ◯ Strongly disagree
   - ◯ Disagree
   - ◯ Neither agree or disagree
   - ◯ Agree
   - ◯ Strongly agree

5. **I am learning when playing the reaction game mode.** *
   Mark only one oval.
   - ◯ Strongly disagree
   - ◯ Disagree
   - ◯ Neither agree or disagree
   - ◯ Agree
   - ◯ Strongly agree

6. **I am concentrating when playing the reaction game mode.** *
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

7. **I am motivated to learn when playing the reaction game mode.** *
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

8. **I am emotionally involved when playing the reaction game mode.** *
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

9. **I have a good social experience playing the reaction game mode.** *
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

10. **I often play games in my spare time.** *
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

11. **When I play games I am defined by** *

Mark only one oval.

- A focus on winning rank, and direct peer-to-peer competition
- A focus on attaining status and achieving preset goals quicky and/or completly
- A focus on socializing and a drive to develop a network of friends and contacts
- A focus on exploring and a drive to discover the unknown

Completeness form

* Required

1. **Age** *

2. **Gender** *
   Mark only one oval.
   - Male
   - Female

3. **I have fun playing the completeness game mode.** *
   Mark only one oval.
   - Strongly disagree
   - Disagree
   - Neither agree or disagree
   - Agree
   - Strongly agree

4. **I am engaged when playing the completeness game mode.** *
   Mark only one oval.
   - Strongly disagree
   - Disagree
   - Neither agree or disagree
   - Agree
   - Strongly agree

5. **I am learning when playing the completeness game mode.** *
   Mark only one oval.
   - Strongly disagree
   - Disagree
   - Neither agree or disagree
   - Agree
   - Strongly agree

6. **I am concentrating when playing the completeness game mode.** *
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

7. **I am motivated to learn when playing the completeness game mode.** *
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

8. **I am emotionally involved when playing the completeness game mode.** *
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

9. **I have a good social experience playing the completeness game mode.** *
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

10. **I often play games in my spare time.** *
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

11. **When I play games I am defined by** *

Mark only one oval.

- A focus on winning rank, and direct peer-to-peer competition
- A focus on attaining status and achieving preset goals quicky and/or completly
- A focus on socializing and a drive to develop a network of friends and contacts
- A focus on exploring and a drive to discover the unknown

Quiz-a-tron Comparison

* Required

1. **I had the most fun when playing** *
   Mark only one oval.
   - ◯ Reaction
   - ◯ Completness

2. **I was most engaged when playing** *
   Mark only one oval.
   - ◯ Reaction
   - ◯ Completness

3. **I learned most when playing** *
   Mark only one oval.
   - ◯ Reaction
   - ◯ Completness

4. **I was most concentrated when playing** *
   Mark only one oval.
   - ◯ Reaction
   - ◯ Completness

5. **I was most motivated when playing** *
   Mark only one oval.
   - ◯ Reaction
   - ◯ Completeness

6. **I was most emotionally involved when playing** *
   Mark only one oval.
   - ◯ Reaction
   - ◯ Completness

7. **I had the best social experience when playing** *
   Mark only one oval.
   - ◯ Reaction
   - ◯ Completeness

System Usability Scale (SUS)

1. **I think that I would like to use this system frequently.**
   Mark only one oval.

   - ◯ Strongly disagree
   - ◯ Disagree
   - ◯ Neither agree or disagree
   - ◯ Agree
   - ◯ Strongly agree

2. **I found the system unnecessarily complex.**
   Mark only one oval.

   - ◯ Strongly disagree
   - ◯ Disagree
   - ◯ Neither agree or disagree
   - ◯ Agree
   - ◯ Strongly agree

3. **I thought the system was easy to use.**
   Mark only one oval.

   - ◯ Strongly disagree
   - ◯ Disagree
   - ◯ Neither agree or disagree
   - ◯ Agree
   - ◯ Strongly agree

4. **I think that I would need the support of a technical person to be able to use this system.**
   Mark only one oval.

   - ◯ Strongly disagree
   - ◯ Disagree
   - ◯ Neither agree or disagree
   - ◯ Agree
   - ◯ Strongly agree

5. **I found the various functions in this system were well integrated.**
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

6. **I thought there was too much inconsistency in this system.**
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

7. **I would imagine that most people would learn to use this system very quickly.**
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

8. **I found the system very cumbersome to use.**
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

9. **I felt very confident using the system.**
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

10. **I needed to learn a lot of things before I could get going with this system.**
Mark only one oval.

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree
