



NTNU – Trondheim
Norwegian University of
Science and Technology

KnowledgeWar 2.0

A Pervasive Knowledge War Game

Stian Forberg

Jon Kjetil Øye

Master of Science in Informatics

Submission date: November 2012

Supervisor: Alf Inge Wang, IDI

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

Abstract

This thesis set out to study how a pervasive multiplayer lightweight Role-Playing Game could be used within higher education in order to motivate and engage students. The primary research goal was to study the educational effect of this type of game, while a secondary research goal was to experiment with HTML5 and JavaScript to build a cross-platform friendly solution.

The game was tested in two user experiments on voluntary students having taken the course Software Architecture at the Norwegian University of Science and Technology. Data from these user experiments were collected through a custom-made quantitative questionnaire and a System Usability Scale, as well as informal group discussions with the participants of the experiments.

The results indicate that students found the game to be both fun and educational. However, results further indicate that the game need more of both gameplay and game world content, in order to “disguise” its educational content well enough to make students want to play the game regularly.

The thesis concludes that more gameplay elements needs to be added to the game in order to fully reach its potential as both a pervasive game and a RPG game. By doing so the results indicate that the game has the potential to become a successful educational aid in higher education.

Keywords: Games, Game-based Learning, Pervasive, Role-Playing Games, RPG

Acknowledgements

This thesis is the result of the course *IT3901 - Informatics Postgraduate Thesis: System Engineering and Human-Computer Interaction* at the Department of Computer and Information Science under Faculty of Information Technology, Mathematics and Electrical Engineering, at the Norwegian University of Science and Technology.

We would like to thank our supervisor professor Alf Inge Wang for continued support, guidance and feedback during the past year. His ambition served as a motivation for us throughout the project duration.

Our gratitude to Wireless Trondheim for providing us with access to their GeoPos system for locational data to use in the game.

Our praise also goes to Scott Johnson for allowing us to use images from his “56 geeks” project for graphics in our implementation.

And finally our thanks to everyone who participated in our prototype testing, and otherwise aided us in the making of this thesis.

Trondheim, 30th of November, 2012.

Jon Kjetil Øye

Stian Forberg

Table of Contents

I	Introduction	1
1	Introduction	3
1.1	Project Motivation	3
1.2	Project Context	4
1.3	Problem Description	4
1.4	Reader's Guide	4
II	Research Method	7
2	Research Question	9
2.1	Educational	9
2.2	Technical	9
3	Research Method	11
3.1	Literature Study/Pre-study	11
3.1.1	Sources	11
3.2	Software Engineering Research Methods	12
3.2.1	Our Method of Choice	13
3.3	User Experiment	13
3.3.1	Questionnaire	14
3.3.2	Group Discussion and Feedback	14
3.4	Ethical Issues	14
III	Pre-Study	17
4	Earlier Work	19
4.1	An Evaluation of a Mobile Game Concept for Lectures	19
4.2	Age of Computers	20
4.3	KnowledgeWar	20
4.4	Educational MMORPG - World of Wisdom	21
4.5	Pervasive Gaming in the Everyday World	21
4.6	Capture the Flag: Mixed-Reality Social Gaming with Smart Phones . .	23
4.7	Using a PDA for Mobile Learning	23
4.8	Campusquest.net	24
4.9	App Framework by Amendor	24

4.10	Buzz! The Schools Quiz	26
5	Mobile Technologies and Platforms	29
5.1	Mobile Application Development	29
5.1.1	Native App	29
5.1.2	Web App	29
5.1.3	Platform-specific Web App	30
5.1.4	Hybrid App	30
5.2	Web Technologies	30
5.2.1	HTML5	30
5.2.2	CSS3	31
5.2.3	JavaScript	32
5.2.4	WebSocket	32
5.3	Platforms	32
5.3.1	Google Android	32
5.3.2	Apple iOS	33
5.3.3	Microsoft Windows Phone	34
5.3.4	Other: Symbian, BlackBerry OS, Bada, webOS	34
5.4	Cross-Platform Development	34
5.4.1	PhoneGap	35
5.4.2	MoSync	35
5.4.3	Rhodes	36
5.4.4	appMobi	36
5.4.5	Titanium Mobile	36
5.4.6	Sencha Touch	37
5.4.7	jQuery Mobile	37
6	Server Technologies	39
6.1	Jetty	39
6.2	nodeJS	39
6.3	jWebSocket	40
7	Game Studies	41
7.1	Genres	41
7.1.1	Simulation	41
7.1.2	Strategy	42
7.1.3	Action	43
7.1.4	Adventure games and Role-playing games (RPG)	43
7.1.5	Fighting	44
7.2	Flow Theory	45
7.3	Games and Education	46
7.4	Multiplayer Game Balancing	46
8	Prestudy Summary	47
8.1	Earlier Work	47
8.2	Mobile Technology	48
8.3	Server Technology	48

8.4	Game Studies	49
IV	Own Contribution	51
9	Concept Sketch	53
9.1	Influences and Inspiration	53
9.2	Early Sketches	55
10	Requirement Specification	57
10.1	Non-Functional Requirements	57
10.2	Functional Requirements	57
10.3	Technical Requirements	58
10.4	Summary	59
11	Design	61
11.1	Initial Game Concept Design	61
11.1.1	Character Creation	61
11.1.2	Gameplay	63
11.2	Paper Prototype	65
11.2.1	Test Subjects	65
11.2.2	Findings	65
11.2.3	System Usability Scale (SUS)	66
11.3	Game Concept Redesign	68
12	Implementation	71
12.1	The Client	71
12.1.1	PhoneGap	71
12.1.2	WebSocket	72
12.1.3	jQuery Mobile	73
12.1.4	Leaflet and CloudMade	74
12.1.5	Client Dependencies	77
12.1.6	The Finished Client	78
12.1.7	Admin Client	84
12.1.8	Summary	85
12.2	The Server	86
12.2.1	Server Logic	86
12.2.2	Statistics	88
12.2.3	Client Location Update	89
12.2.4	Database	90
13	User Testing	93
13.1	Acquiring Test Subjects	93
13.2	Execution	93
13.2.1	Presentation	94
13.2.2	Test Phase	94
13.2.3	Questionnaire and Feedback	94

13.3	Test of Software Prototype 1 (SP1)	94
13.3.1	The Feedback	95
13.4	Tweaking the Prototype	96
13.5	Test of Software Prototype 2 (SP2)	97
13.5.1	The Feedback	98
13.6	Summary	99
14	Experiment Results	101
14.1	System Usability Scale (SUS) Results	101
14.1.1	Comparison of the Paper Prototype and the Software Prototypes	101
14.2	KnowledgeWar 2.0 Questionnaire Results	104
14.2.1	Scoring	104
14.2.2	Participants and Devices	104
14.2.3	Learning	104
14.2.4	Gameplay	107
14.2.5	Technical	107
14.2.6	Social & Mobility	109
14.2.7	Experiment	109
15	Evaluation and Discussion	111
15.1	Non-Functional Requirements	111
15.1.1	NFR1 - Usability	111
15.1.2	NFR2 - Fun	112
15.1.3	NFR3 - Engagement	113
15.1.4	NFR4 - Immersion	114
15.1.5	NFR5 - Educational	114
15.2	Technical Requirements	115
15.2.1	TR1 - Roaming	115
15.2.2	TR2 - Location	116
15.2.3	TR3 - Persistence	117
15.2.4	TR4 - Platform	117
16	Conclusion	119
16.1	Educational Research Questions	119
16.1.1	Research Question 1	119
16.1.2	Research Question 2	120
16.1.3	Research Question 3	120
16.2	Technical Research Questions	121
16.2.1	Research Question 4	121
16.2.2	Research Question 5	122
16.2.3	Research Question 6	122
16.3	Project Evaluation	123
16.3.1	Pre-study	123
16.3.2	Software Engineering Method	124
16.3.3	User Experiment	124
16.3.4	The Research	125

17 Further work	127
17.1 System Completion	127
17.2 Potential Features	128
17.2.1 Gameplay	128
17.2.2 Technical	131
17.3 Future Dreams	131
V Appendix	133
A Acronyms	135
B Paper Prototype	137
C Scott Johnson’s 56 Geeks	147
D Weapons and Armor	149
E Experiment Instructions	153
F KnowledgeWar 2.0 Questionnaire	155
G System Usability Scale	159
H Developer’s Guide	161
H.1 Client/AdminClient Development Setup	161
H.1.1 Project Setup	161
H.1.2 Framework Updates	162
H.1.3 Other Platforms	162
H.2 Server Development Setup	162
H.2.1 Project Setup	162
H.2.2 Wireless Trondheim	163
H.2.3 Running the Server	163

List of Tables

10.1	Non-functional requirements	57
10.2	Functional requirements	58
10.3	Technical requirements	59
11.1	Starting attributes for the Computer Geek class	62
11.2	Starting attributes for the Robot Geek class	62
11.3	Starting attributes for the Lab Geek class	62
11.4	Starting attributes for the Math Geek class	63
11.5	SUS Scores for the Paper Prototype	67
11.6	Attributes comparison of the redesigned character classes	69
11.7	Starting character equipment	69
11.8	All available armor	70
11.9	All available weapons	70
14.1	SUS scores for the Software Prototype	102
14.2	KnowledgeWar 2.0 Questionnaire: Positive statements (higher is better)	105
14.3	KnowledgeWar 2.0 Questionnaire: Negative statements (lower is better)	106
14.4	KnowledgeWar 2.0 Questionnaire: Additional technical statements . . .	108
14.5	KnowledgeWar 2.0 Questionnaire: Additional experiment statement . .	110
15.1	Related questions for Usability from the SUS form and questionnaire .	112
15.2	Related questions for Fun from the questionnaire	112
15.3	Related questions for Engagement from the questionnaire	113
15.4	Related questions for Immersion from the questionnaire	114
15.5	Related questions for Educational from the questionnaire	115
15.6	Related questions for Roaming from the questionnaire	116
15.7	Related questions for Location from the questionnaire	116
15.8	Related questions for Platform from the questionnaire	118
15.9	Android version usage share	118

List of Figures

4.1	Screenshot from the game Lecture Quiz	19
4.2	Screenshot from the game Age of Computers	20
4.3	Screenshot from the game KnowledgeWar	21
4.4	Screenshot from the game World of Wisdom	22
4.5	Screenshot from the game SupaFly	22
4.6	Images from the Capture the Flag game	23
4.7	Screenshot from a proposed RPG game for PDAs	24
4.8	Screenshot of the game Campus Quest	25
4.9	Screenshots from the Amendor quiz app	25
4.10	Screenshots from Buzz! The Schools Quiz	26
4.11	Photo from a trial of Buzz! The Schools Quiz in a British primary school	27
5.1	HTML5Test.com: Android, iOS and Windows Phone Score Chart	31
6.1	jWebSocket plug-in infrastructure	40
7.1	Microsoft Flight gameplay screenshot	42
7.2	Starcraft gameplay screenshot	42
7.3	Red Dead Redemption gameplay screenshot	43
7.4	Final Fantasy VII gameplay screenshot	44
7.5	Street Fighter X Tekken gameplay screenshot	45
9.1	Screenshot from two generations of Pokémon games	53
9.2	Screenshot from Street Fighter (left) and Tekken (right)	54
9.3	Illustration of how fog-of-war might be implemented in an RPG	54
9.4	Early concept sketch from brainstorming session	55
11.1	Initial Characters	62
11.2	Paper prototyping in action	65
12.1	KnowledgeWar 2.0: Screenshot of the map	76
12.2	Client dependencies	77
12.3	The Different view states of the client	78
12.4	KnowledgeWar 2.0: Screenshot of the main menu	79
12.5	KnowledgeWar 2.0: Screenshots of the registration screens	79
12.6	KnowledgeWar 2.0: Screenshots of the character profile	80
12.7	KnowledgeWar 2.0: Screenshots of the map	81
12.8	KnowledgeWar 2.0: Screenshots of picking up and equipping an item	81

12.9 KnowledgeWar 2.0: Screenshots of the attacklist and an incoming challenge	82
12.10The Different view states of a battle	82
12.11KnowledgeWar 2.0: Screenshots of the battle screen	83
12.12KnowledgeWar 2.0: Screenshots of a quiz question	84
12.13KnowledgeWar 2.0: Screenshots of the battle report, question review and battle reward	84
12.14The Admin Client: Screenshot of the message system and map	85
12.15Server Class diagram	87
12.16Statistics pie chart	89
12.17ER model	91
13.1 SP2 test in action	98
B.1 Paper Prototype: Main menu and the recover account screens	137
B.2 Paper Prototype: Step 1 & 3 of the register new account screens	138
B.3 Paper Prototype: Step 2 of the register new account screens	139
B.4 Paper Prototype: The different pop-ups related to the registration process	140
B.5 Paper Prototype: The highscore table	140
B.6 Paper Prototype: The initial profiles of the four character classes	141
B.7 Paper Prototype: The map screen, treasure chest pickup screen and the pop-up info-boxes related to them	142
B.8 Paper Prototype: Initial battle screen and battle screen after some rounds	143
B.9 Paper Prototype: The quiz question screen when battling	143
B.10 Paper Prototype: The different pop-ups related to the battle screen	144
B.11 Paper Prototype: The battle report, question review and battle reward screens	145
C.1 Scott Johnson's 56 Geeks	148
D.1 KnowledgeWar 2.0 Weapons	150
D.2 KnowledgeWar 2.0 Armor	151

Part I

Introduction

Chapter 1

Introduction

Since the release of the Apple iPhone in 2007, the sales and use of smartphones have become a widespread phenomenon. Today smartphones are a common carry-on, not only for the middle-aged business man, but also for kids all the way down to primary school. Alongside recent years increase in popularity of tablets [12], mobile pervasive computing has become an everyday activity. This has also caused a large new market to appear in the development of application for a large range of mobile devices, with the biggest platforms being Google's Android, Apple's iOS, Microsoft's Windows Phone and Nokia's Symbian [13]. With popular games like Angry Birds¹ setting the standard, the smartphone has become a leader in the world of mobile and casual gaming. The rapidly increasing performance of the devices allow for more and more advanced graphics and gameplay. Now they even support full scale 3D multi-player Role-Playing Games (RPGs)² remarkably similar to the holy grail of RPGs World of Warcraft³. Both iOS and Android offer a number of educational applications in their application stores. These are, however, mostly aimed at children for use outside of school, teaching them simple mathematics or the alphabet.

1.1 Project Motivation

Aimed towards higher education, there are but a few contributions in the segment of mobile gaming and learning. Amendör's quiz apps for students taking Examen philosophicum or Information Technology Introduction⁴ and the quiz web app Campus Quest are some examples. These are, however, not made specifically to be integrated and used as an educational aid. There are examples of traditional video games being used in teaching higher education. Wang & Wu presents three main ways of how video games can be integrated into a university course [25]. First, a game can be used to replace normal exercises in a course, thus trying to motivate the students to put more effort into the exercises while providing the lecturer the opportunity to monitor student progress. Second, a game can be integrated by using it within lectures to promote student participation and motivation during lectures. Third, game development can be

¹<http://www.rovio.com/index.php?page=angry-birds> (16.11.12)

²<http://www.dvidearts.com/games.php?gid=9> (16.11.12)

³<http://eu.battle.net/wow/en/> (16.11.12)

⁴<https://play.google.com/store/apps/developer?id=Amendör> (16.11.12)

used in computer science (CS) or software engineering (SE) courses to teach CS and SE practices and skills. These ways of integration have all been tried out at the Norwegian University of Science and Technology (NTNU). Lecture Quiz (LQ) [24] and Age of Computers (AoC) [18] are good examples of the former two ways of implementation. Lecture Quiz is a lecture integrated quiz game while Age of Computers has been used instead of regular exercises in the course “Computer Fundamentals” since fall 2003. Both games showed promising results as students using them claimed both increased learning and motivation compared to regular lectures or exercises. Game development projects have also been used at NTNU in the course “Software Architecture” since 2008 as part of the exercises, teaching the students how to plan and implement patterns in software development. Course staff registered an increased interest and motivation for the exercise after switching to the game development project [25].

1.2 Project Context

The KnowledgeWar 2.0 game project is a part of the Lecture Games research project at the Department of Computer and Information Science (IDI) at the Norwegian University of Science and Technology (NTNU). The Lecture Games project aims to explore how to use games in higher education to provide variation in teaching and new ways of promoting learning through interaction between teacher and students. It also aims to increase interaction between fellow students [22].

A previous version of KnowledgeWar was developed by Sveinung Kval Bakken for his master’s thesis in June 2010. The KnowledgeWar 2.0 project will build on the experiences of the previous work and hopefully also increase the games potential to give students a new fun and engaging platform for learning.

1.3 Problem Description

This project focuses on the development and testing of a new type of learning game played on smartphones. The game will be a lightweight multiplayer RPG where players can challenge other players in the same physical location to fight in knowledge battles. The game can be expanded to utilise other features of the smartphone as well as other gameplay elements from pervasive games. The game consists of a server hosting the game and client software where players connects to the server and play the game. The target platform is a cross-platform friendly HTML5 and JavaScript solution and will be tested on the Android platform.

The first phase of the project is to develop a prototype of this game including both the game server and game clients. The second phase is to test the game with players and review the user experience.

1.4 Reader’s Guide

This thesis consists of three main parts, a pre-study part, a system design and implementation part, and a part about usability experiment and evaluation. The pre-study part present some earlier work that is found to be relevant for this research project

and review some potential technologies for both the server and client software. A look into the art of game design in context of educational gaming is performed, before the design of a game concept together with a system design that supports the concept. A paper prototype test was performed to verify and improve the game concept before the implementation phase is initiated. The implementation phase then goes through the technical implementation of the software and how the major technical challenges have been handled. A user experiment is then performed with the developed prototype and the collected data is presented. Lastly an evaluation is presented and some conclusions are drawn before some suggestions for further work.

For those especially interested in the *technical* aspect of this thesis might find the following parts useful:

Chapter 5: Mobile Technologies and Platforms

Chapter 6: Server Technologies

Chapter 10: Requirement Specification

Chapter 11: Design

Chapter 12: Implementation

Chapter 15: Evaluation and Discussion

Chapter 17: Further Work

For those especially interested in the *educational gaming* aspect of this thesis might find the following parts useful:

Chapter 4: Earlier Work

Chapter 7: Game Studies

Section 11: Design

Chapter 14: Experiment Results

Chapter 15: Evaluation and Discussion

Chapter 16: Conclusion

Chapter 17: Further Work

Finally, for those especially interested in the *user experiment* part of this thesis might find the following parts useful:

Part II: Research Method

Section 11.2: Paper Prototype

Chapter 13: User Testing

Chapter 14: Experiment Results

Chapter 15: Evaluation and Discussion

Chapter 16: Conclusion

Part II
Research Method

Chapter 2

Research Question

This research project aims to discover how to motivate students to do more than just the bare minimum required to be eligible for the final exam. It is not desired to provide an alternative to the physical act of going to a lecture, which may in turn lead to reduced motivation for attending lectures [21]. The research will have to shed light on how to make the mobile application engaging in such a way that it is not seen as extra work or a replacement for lectures, but rather a fun way to learn outside of the lecture hall as well as be engaged to learn even more about the subject at hand.

The project will also experiment with new technology, by the means of using HTML5 to develop a cross-platform friendly solution. By using HTML5, the project will determine how well it serves as a platform for mobile game development and if it is suitable for the goal to make a fun and engaging educational game⁵.

2.1 Educational

RQ1 - How can a smartphone application be used to make the curriculum of a university course more engaging for students?

RQ2 - How can game design techniques be used so that the application promotes and rewards further use?

RQ3 - To what extent can a pervasive educational game help to engage and motivate students and increase their learning experience?

2.2 Technical

RQ4 - How mature is the HTML5 platform for use on smartphones?

RQ5 - How well does the current HTML5 performance suit mobile pervasive games?

RQ6 - How is it possible to make a fun and engaging game that works well across all major platforms without having to write two sets of code?

⁵Parts of this introduction are taken from a report, delivered in the course IT3010 - Research methods in informatics, authored by Stian Forberg

Chapter 3

Research Method

To reach the answers to the research questions defined in Chapter 2, a four-part pre-study will first be performed to look into the areas: earlier/similar work, mobile technologies, server technologies and game studies. Second, two iterations of a software engineering method will be performed, where the development of a paper prototype is done, testing it on fellow students, making improvements and then go on to develop a software prototype. Third, a user test of the software prototype will be performed in a real world setting with students at NTNU. From this user experiment, data will be collected through logs from the server software, one or more questionnaires and discussions or informal interviews with participants of the experiment. This data together with experiences from the pre-study and implementation of the software, will form the basis on which the research questions are answered⁶.

3.1 Literature Study/Pre-study

A literature study will be performed to get acquainted with previous work done in the field of educational games and mobile pervasive games. The study will focus on four main areas of interest, namely; earlier/similar work, mobile technology, server technology and game studies. This study will then provide a foundation for the research to stand on and place the study in the proper context.

3.1.1 Sources

In addition to sources of literature recommended by the thesis supervisor Alf Inge Wang, a variety of online resources will be used to find literature relevant for the pre-study. Such resources include, but are not limited to:

- ACM Digital Library⁷
- IEEE Xplore Digital Library⁸

⁶Parts of this chapter is taken from a report, delivered in the course IT3010 - Research methods in informatics, authored by Jon Kjetil Øye and Stian Forberg

⁷<http://dl.acm.org/> (16.11.2012)

⁸<http://ieeexplore.ieee.org/Xplore/home.jsp> (16.11.2012)

- ISI Web of Knowledge⁹
- Google Scholar¹⁰

If a piece of literature is found to be relevant, the sources of the text will also be considered and evaluated.

3.2 Software Engineering Research Methods

A lot of research have been made in the area of software engineering. People are rapidly building tools, models, procedures and technology. But sadly a lot of projects claiming to be research projects are simply development projects, as they fail to thoroughly or realistically test, measure and validate the topic that is being studied. Basili et al. [3] defines software research as follows:

“Research involves gaining understanding about how and why a certain type of tool might be useful and by validating that a tool has certain effects by carefully designing an experiment to measure the properties or to compare it with alternatives.”

In order to differentiate between research activities from development activities, Basili presents three paradigms of conducting software engineering research:

1. The scientific method:

Consists of observing the world, propose a model or a theory of behaviour, measure and analyse, validate hypotheses of the model or theory and if possible repeat the procedure.

This method is, in the context of software engineering, most suitable for when a researcher wants to understand something about a certain process, product, people or environment. The method tries to extract a form of model from the world which can be used to describe an underlying phenomena. Further, Basili presents two variations of the scientific method:

- (a) The engineering method:

Consists of observing existing solutions, propose better solutions, build/develop, measure and analyse, and repeat the process until no more improvements appear possible.

This is an evolutionary approach that assumes there already exist models to evaluate and makes modifications to the models or aspects of the models in order to improve whatever is being studied.

- (b) The empirical method:

Consist of proposing a model, develop statistical or qualitative methods, apply the model to case studies, measure and analyse, validate the model and repeat the procedure.

⁹<http://apps.webofknowledge.com> (16.11.2012)

¹⁰<http://scholar.google.no/> (16.11.2012)

This approach is a revolutionary improvement oriented approach that begins with proposing a new model, and attempts to study the effects of this new model as it is being put to use. The crucial aspect of this approach is the measure and analyse part, as simply building a tool, or proposing a model is not research. There must be a clear way of evaluating whether the new model or tool gives any new advantages over what is currently available.

2. The mathematical method:

Consists of proposing a formal theory or set of axioms, develop a theory, derive results and if possible compare with empirical observations.

This approach is an analytical and deductive model which does not necessarily rely on experimental data for validation, but instead provides an analytical framework for building models and understanding their boundaries by manipulating the model itself.

3.2.1 Our Method of Choice

This study will follow a variation of what Basili defined as the engineering method. Since this research study is a one year master's thesis, there will be no time to reach the goal of "no more improvements appear possible". The research project will instead perform two major iterations. Based on the previous work by Sveinung Kval Bakken on the KnowledgeWar project, a new and expanded game design will be proposed. Next, a paper prototype will be built and used to run usability tests on students in order to get feedback and inspiration for further improvements. A final game design will be made based on this feedback before planing and building a software prototype using HTML5 and JavaScript. The software prototype will then be used to run user experiments.

3.3 User Experiment

Experimentation alone is of no value if there is no underlying framework where experimental results can be interpreted [3]. One of the main properties of a pervasive game is that it promotes gameplay to be place-independent and that it integrates with the physical world. This makes evaluating a pervasive application quite difficult for a researcher, as the researcher has a very limited ability to study; when, where and how the player choose to play the game. In order to compensate for this, data collection will be done in three ways. First, the system will incorporate a logging tool to keep track of user activities which can be reviewed at a later stage. Second, users of the system will take part in a quantitative questionnaire after having used the system for some time. Third, informal interviews or group discussions will be performed with a selection of users. This selection can be made either randomly or by the researchers which can pick specific users they find interesting based on, e.g., logged activity.

The game will be tested on voluntary students from a selected course at NTNU. This will be done in cooperation with the course lecturer so that the game can be implemented to match at least some subsections of the course curriculum.

3.3.1 Questionnaire

After using the prototype for some time during the experiment phase, the participants will be asked to fill out two questionnaires. The first questionnaire will be custom-made by the developers to collect relevant data in different categories as learning/educational profit, gameplay and design, technical performance and the use of mobile/pervasive elements. The second questionnaire will be a System Usability Scale, which is a generic questionnaire of ten questions that results in a indication of usability as a number between 0 and 100 [4].

A System Usability Scale (SUS) form is always presented as a five-level Likert scale. Participants responding to a SUS form will choose the level of agreement to a statements ranging from Strongly Disagree, through Neutral, to Strongly Agree. If Strongly Disagree is answered on any of the odd numbered statements, or Strongly Agree is answered on any of the even numbered statements, the SUS score for that one answer on that statement will be 0. If the answer to any of the even numbered statements is Strongly Agree, and likewise, Strongly Disagree to any of the even numbered statements, the SUS score for that particular question will be 10.

3.3.2 Group Discussion and Feedback

If there is a large number of participants in the user experiment, the participants will be split into smaller groups for separate discussions. The purpose of these groups is to get a discussion going between the participants on subjects such as; how they played the game, overall quality of the design and gameplay, to what extent they learned anything and areas that are in need of improvement. It would be preferable if the lecturer would be a part of at least one of these focus groups to spark a discussion that would take in and compare both the lecturer and student point of view.

3.4 Ethical Issues

When doing research that involves other participants than the researcher himself, the researcher have a responsibility in relation to the participants. Oates discusses how the participants should be treated and mentions five specific points that make up, “The rights of participants”: Right not to participate, to withdraw, to give informed consent, to anonymity and to confidentiality [19].

In our game, a player is supposed to be able to challenge other players in his or her vicinity to a knowledge battle. This can be done by a simple query to the game server, which then provides the player with a list of other players in his vicinity, thus keeping the specific location of players private. Alternatively the player could be limited to only see or be able to challenge other players that are within a certain distance.

As the prototype application will be logging when, where and who is using it, users need to be made aware that being a participant in the testing phase will require some usage data to be collected. This does not mean that highly personal data will be collected and stored, but if some personal data is needed it can easily be anonymised in such a way that participants are protected. All data collected will also have to be stored confidentially. The previously mentioned questionnaire should also be formed

in such a way that if the respondents do not wish to answer a particular question, they will not be forced to answer it.

Part III
Pre-Study

Chapter 4

Earlier Work

In this section a few selected articles on mobile-, pervasive-, educational- and serious games are presented. They are highlighted based on the relevance for this thesis and for their findings.

4.1 An Evaluation of a Mobile Game Concept for Lectures

Lecture Quiz is a quiz game which can be used within lectures in higher education, in order to enhance student participation during lectures and give some variation in the way lectures can be taught [23]. The game aims to test, motivate and engage the students while using the existing equipment and infrastructure in the lecture halls at the university. The game was split into a game server that was run on the lecturer's computer and shown using a projector, and a game client which run on Java 2 Micro Edition. This allowed students to use either their mobile phones or their laptops for playing the game. Their research results indicated that the students found the game engaging, and perceived to learn more using the game. Students who had participated in the trials of the game said that they would be more likely to show up for lectures if games like Lecture Quiz was used regularly. Their findings further indicate that more research need to be done in order to establish if regular use of the game would actually give any educational benefits.

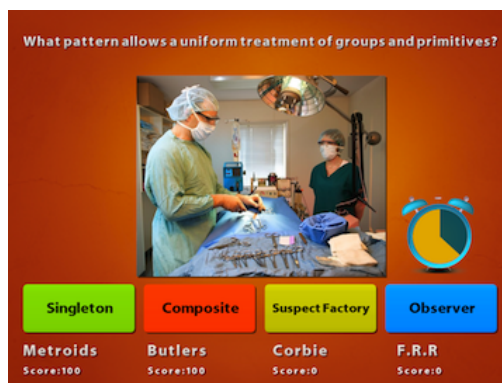


Figure 4.1: Screenshot from the game Lecture Quiz

4.2 Age of Computers

Age of Computers (AoC) is an online multiplayer game used instead of traditional exercises in the course “Computer Fundamentals” at NTNU [18]. It was made in order to cope with a large inhomogeneous class and to provide the students with a new flexible learning environment. To support cross-platform functionality, the game is implemented as dynamic web-pages which the player accessed through a browser. AoC was well received by the students as 67% of them rated the game as the top learning effect in the course, where only 16% voted for the lectures and 14% for the textbook. The game has been improved several times and is currently in a third version. It is still used as the main exercise environment for the “Computer Fundamentals” course.

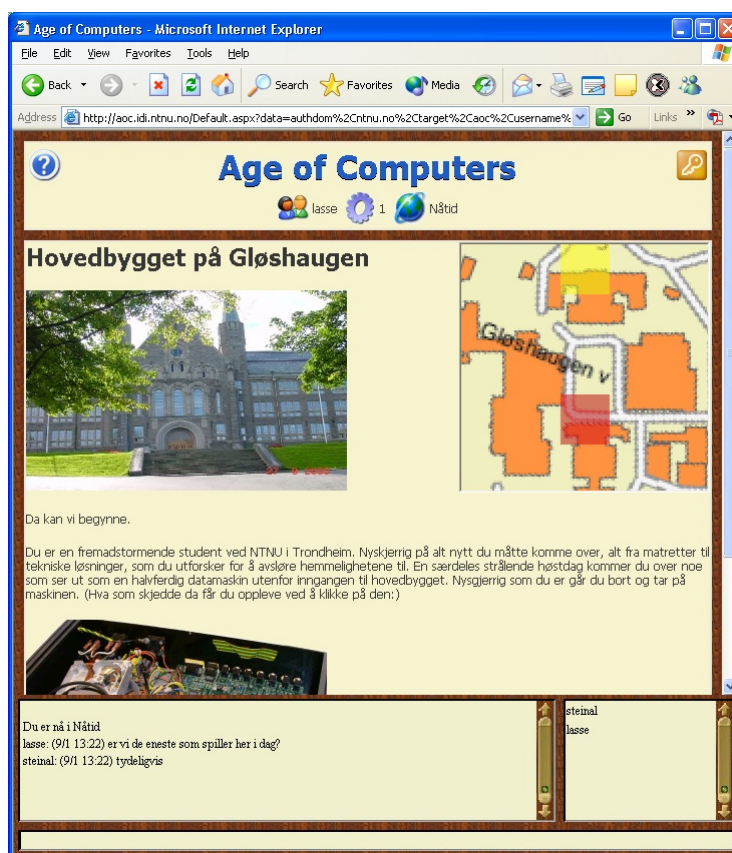


Figure 4.2: Screenshot from the game Age of Computers

4.3 KnowledgeWar

Sveinung Kval Bakken presents the experiences from implementing a quiz-like application for the iPhone [15]. The game allows the user to challenge people in your physical location to a quiz battle. On the technical aspect both the Android and the iOS platform was considered, but iOS was ultimately chosen because of the maturity and extensiveness of the platform. On the educational side, the application was hands-on

tested by a small number of subjects that also answered a questionnaire after use. The test subjects were positive to both the concept and the application itself. The thesis describes some further work that can be done in the form of new game play elements, graphical enhancements and group play. As these elements could greatly enhance the game's ability to engage and motivate players.

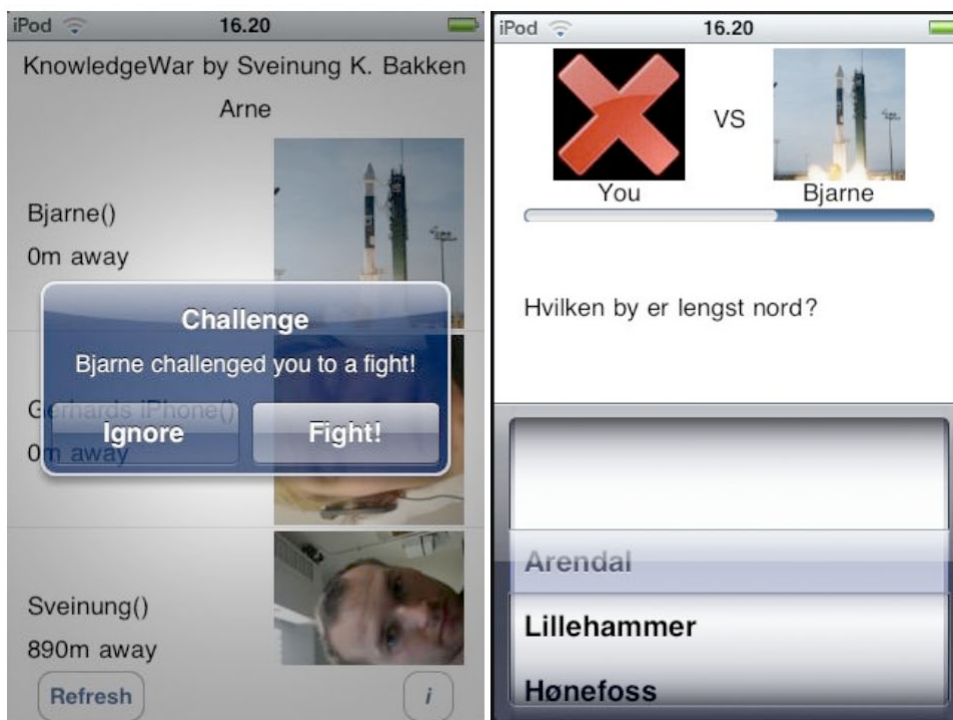


Figure 4.3: Screenshot from the game KnowledgeWar

4.4 Educational MMORPG - World of Wisdom

World of Wisdom is an attempt to implement an educational Massively Multiplayer Online Role-Playing Game (MMORPG) using game design theory, game features and flow theory [26]. The goals of the project was to make a game that could be used in lectures in higher education to replace traditional paper exercises. The project also provides tool sets that can be used to create new games for different courses without the need for programming. One of the key problems identified in the article is how to choose and apply relevant theory to support the design of an educational MMORPG. The article concludes that in order to reach a set of criteria or framework for how to design an educational MMORPG, much more research and experiences are needed.

4.5 Pervasive Gaming in the Everyday World

“Supafly” is an everyday-world pervasive game and a virtual soap opera [14]. In the game, players can create a character and socialize with other players through their character. The paper presents an empirical study that examines to what extent the



Figure 4.4: Screenshot from the game World of Wisdom

vision of pervasive gaming is becoming a reality in the context of Supafly. Though not a lecture game, the study reveals some of the challenges a researcher can meet when studying a pervasive game. It also reveals that more research needs to be done regarding a pervasive game's ability to be played any time and any where. Their findings lead to a series of questions such as; "Where do people play the game?", "In what situations do people choose to enter the game?" and "Do people play alone or when they get together?", as it turned out that most of their users played the game in the comfort of their own home.

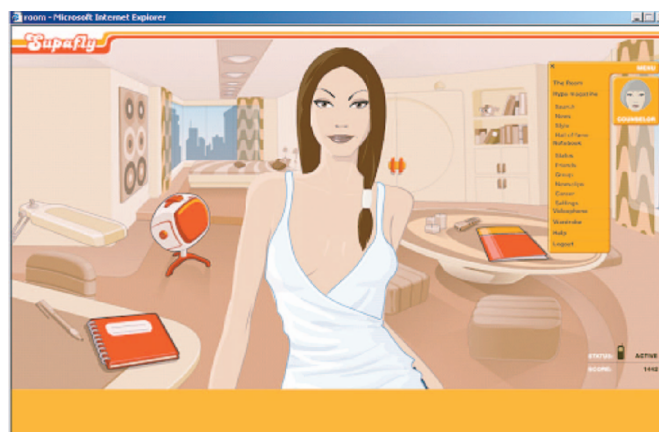


Figure 4.5: Screenshot from the game Supafly

4.6 Capture the Flag: Mixed-Reality Social Gaming with Smart Phones

Capture the Flag: Mixed-Reality Social Gaming with Smart Phones describes an approach to implement a Capture The Flag (CTF) type pervasive game involving both real-world and virtual world elements, and how an experiment with the game was carried out [6]. The study consisted of four phases. First, the participants were introduced to the CTF system. Second, they answered questions about the perceived level of challenges. Third, the participants played the game. Finally, they gave feedback in the form of answers to more questions. Their findings include some communication problems due to the inconvenience of writing a text-message with a mobile phone while on the run, but also that team play worked quite well when the Guide (the player who views the virtual world from a computer and communicates what he sees to the other players on his team, see top left of Figure 4.6) and the Knight (a player in the field, looking for the flag, see middle of Figure 4.6) trusted each other. User response revealed that the players felt the gameplay was highly innovative. It was a unique experience to combine both real-world players as well as virtual-world players.



Figure 4.6: Images from the Capture the Flag game

4.7 Using a PDA for Mobile Learning

Using a PDA for Mobile Learning proposes an RPG model for a learning environment, which is considered to provide a good learning space that supports social interactions at a higher level, progression by incremental tasks, feedback to the player and a reward system [17]. It describes what types of tasks a user can complete (e.g. reading a

paragraph, answering a question with the correct answer), and the amount of reward in the form of “experience points” the user can get for completing these tasks. The user can then use these experience points to purchase aids in the form of displaying the correct answer to a question or removing one wrong answer from the alternatives of a question.



Figure 4.7: Screenshot from a proposed RPG game for PDAs

4.8 Campusquest.net

Campus Quest¹¹ was created by Net Directories Inc. while working with schools and colleges. Campus Quest allows the users themselves to create the learning experience. A faculty member can define how and when a game should be played, and since the game is web-based it is available to use on most smartphones of today. It does not necessarily have to be used to learn the curriculum, but can also be utilised for student orientation, campus tours as well as classroom learning. The game has many other applications. It is a game based quiz application primarily for use with smartphones, which allows an administrator to set up a game, questions and define alternatives for the questions. Students can then group up in teams or play alone. The players answer the questions the administrator sets up and is then ranked by their performance in terms of speed and correct answers. The administrator can view a pseudo-live scoreboard updated at a frequency of choice.

4.9 App Framework by Amendor

In autumn 2011 the company Amendor¹² released two quiz oriented apps aimed at student taking either Examen philosophicum or Information Technology Introduction¹³.

¹¹<http://www.campusquest.net/> (06.11.12)

¹²<http://amendor.com/> (16.11.12)

¹³<https://play.google.com/store/apps/developer?id=Amendor> (16.11.12)

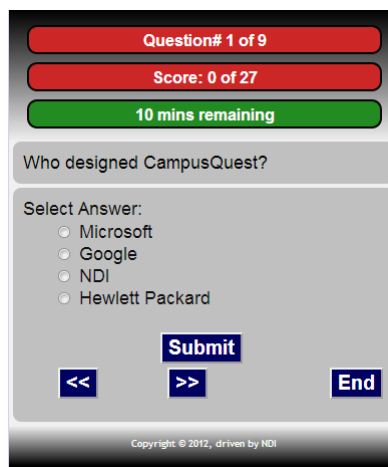


Figure 4.8: Screenshot of the game Campus Quest

Both apps is made with their own app framework which can be adapted and associated with different educational resources. The company has its roots from a former student organization at NTNU called “Teknisk Studentforlag” which focused on delivering services “by students for students”. The company has since then expanded and are now focusing on delivering digital educational resources for the Norwegian educational sector, supply of web solutions using Drupal¹⁴ and publishing printed educational resources.



Figure 4.9: Screenshots from the Amendord quiz app

Their quiz app framework aims to motivate students to learn by applying simple gamification where the students earn points and advance through different knowledge levels. Apps made with the framework can be compiled for both iPhone and Android thus covering the majority of smartphones used by Norwegian students. The apps made for Examen philosophicum and Information Technology Introduction have been made

¹⁴<http://drupal.org/> (20.11.12)

based on the curriculum at the University of Oslo (UiO) and Norwegian University of Science and Technology (NTNU), and have been targeted at students practising for their final exam.

4.10 Buzz! The Schools Quiz

In 2008 Buzz! The Schools Quiz was released for the PlayStation 2 console as an educational edition of the popular Buzz! series of quiz games, developed by Relentless Software¹⁵. The game is based on the British National Curriculum for Key Stage 2, which covers 7-11 year old kids in primary school. The game contains over 5000 questions from subjects as English, Geography, History, Maths, Modern Foreign Languages, Religious Education and Science.

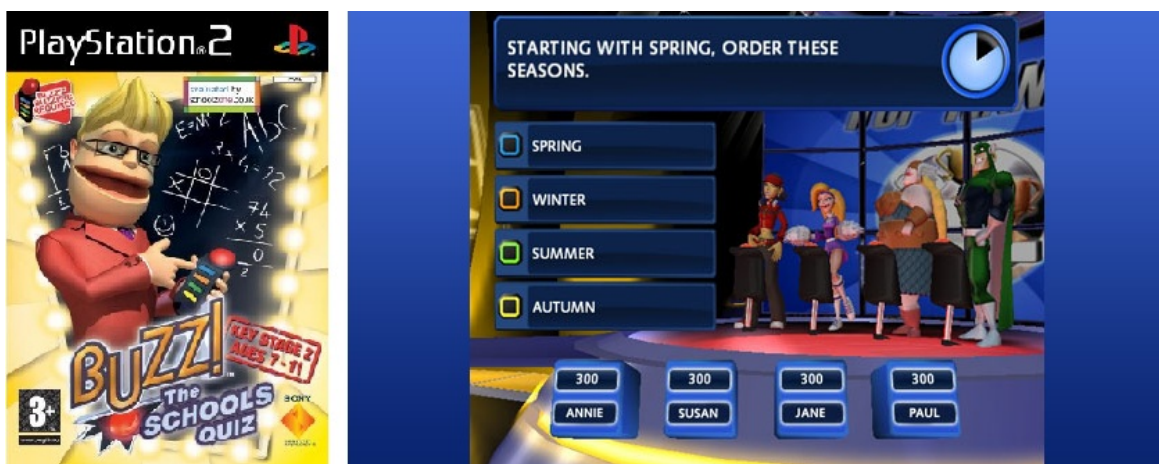


Figure 4.10: Screenshots from Buzz! The Schools Quiz

The game includes several different game modes and a teacher feature, based on feedback from teachers, in order to make the game more flexible and give the teacher an opportunity to direct how the game is played. A trial run of the game was performed ahead of the release, and the over-all feedback on the game was described as overwhelmingly positive. Co-founder of Relentless Software, Andrew Eades, said in an interview with Eurogamer¹⁶: “The whole experience of taking Buzz into Schools has been brilliant. The teachers have been really supportive and have given us great feedback. We have reports of children staying in during their breaks to keep playing Buzz.”

¹⁵<http://www.relentless.co.uk/games/buzz-the-schools-quiz/> (15.11.12)

¹⁶<http://www.eurogamer.net/articles/buzz-the-schools-quiz-interview> (15.11.12)

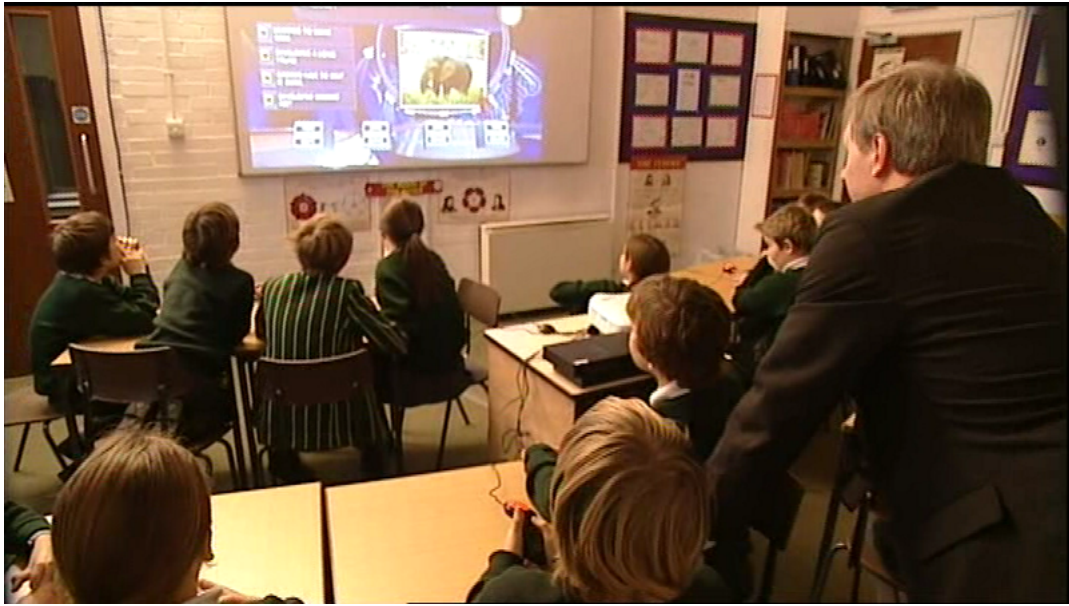


Figure 4.11: Photo from a trial of Buzz! The Schools Quiz in a British primary school

Chapter 5

Mobile Technologies and Platforms

In this chapter some of the current and relevant platforms, technologies and trends are presented in association with the development of mobile applications. A selection of technologies, tools and frameworks will be based on the findings presented to assist in the development of a software prototype.

5.1 Mobile Application Development

When developing mobile applications or services for today's smartphones, developers have several paths they can follow. Four main development types of applications are presented here and compared.

5.1.1 Native App

A native app is an application that has been developed specifically to run on the architecture of a specific type of mobile device (e.g. Android, iOS, Windows Phone). A native app can be downloaded, purchased and upgraded through the device's central distribution portal (e.g. App Store for iOS devices or Play Store for Android devices). Native apps are usually written in a programming language specified by the platform's Software Development Kit (SDK).

5.1.2 Web App

A web app is basically a web page that has been specifically designed to be viewed through a mobile web browser, sometimes imitating the look-and-feel of a native app. A web app has, unlike native apps, no access to a device's Application Programming Interface (API). Web apps can therefore not access the camera, sensors, file system or other device specific services and technologies (HTML5, however, mitigates some of these short comings with new functionality like geolocation, see Section 5.2.1). Developing a web app means lower cost of development, fast time to market and easy cross-platform development through HTML. However, web apps also suffer from poor performance, latency issues, require an Internet connection and cannot be sold or distributed through central distribution channels.

5.1.3 Platform-specific Web App

A platform-specific web app is, basically, a web app that has been wrapped into a native app, thus making it possible to distribute through the device's central distribution portal. This is possible as most platform SDKs offer a browser object that allows for displaying web content. A platform-specific web app still has no access to a device's API but can, as opposed to a normal web app, be made to run in "offline-mode" showing web content stored locally on the device (though it might offer a limited service).

5.1.4 Hybrid App

A hybrid app, takes the advantages of both web and native technology. In a hybrid app at least some of the main user interface is usually made in HTML, CSS and JavaScript, which makes it easier to make cross-platform applications. The user interface is then, like the web app, displayed through the SDK's browser object. However, device specific services and technology, has to either be tailored for each device or accessed through a common third-party interface like PhoneGap¹⁷.

5.2 Web Technologies

In this section, some relevant web technologies will be reviewed for what they have to offer when it comes to mobile development.

5.2.1 HTML5

HTML5 is the fifth revision of the Hyper Text Markup Language (HTML) standard and is the core language for presenting content on the web. At time of writing, HTML5's official status is still a "Working Draft" and is only scheduled to reach the status of "W3C Recommendation" in 2014¹⁸. Because of this, support for HTML5 in browsers is still highly variable especially when considering mobile browsers. Yet, HTML5 has been embraced by both web and mobile developers and have rapidly gained approval in both communities. For mobile developers, HTML5 provides an excellent and easy way of doing cross-platform development. Either through pure web apps or by using HTML5 to build a cross-platform user interface that can be implemented in applications for multiple platforms.

HTML5 tries to improve interoperability and to reduce development costs by defining precise rules on how HTML elements should be handled, and how to recover from errors¹⁹. The fifth revision brings several new functions and possibilities for developers. New functions include how to embed audio and video through the introduction of the audio and video tags, present graphics on the new canvas and also introduce new services for storing data client-side, as well as support for geolocation. It also presents new elements like section, header, nav, footer, article, aside and figure allowing for better structuring of the layout of web pages.

¹⁷<http://phonegap.com/> (19.11.12)

¹⁸<http://www.w3.org/2011/02/htmlwg-pr.html> (31.05.12)

¹⁹<http://www.w3schools.com/html5/default.asp> (31.05.12)

The maturity and widespread support for HTML5 has come along way the past two years in both desktop and mobile browsers. This can be seen from the score timeline on html5test.com²⁰, a website that measure the visiting browser’s support for the HTML5 specification and calculates a score between 0 and 500. Figure 5.1 shows the score timeline for the native browsers of Android, iOS and Windows Phone²¹. The chart shows how the scores has gradually increased for each new version of the different platform operating systems.

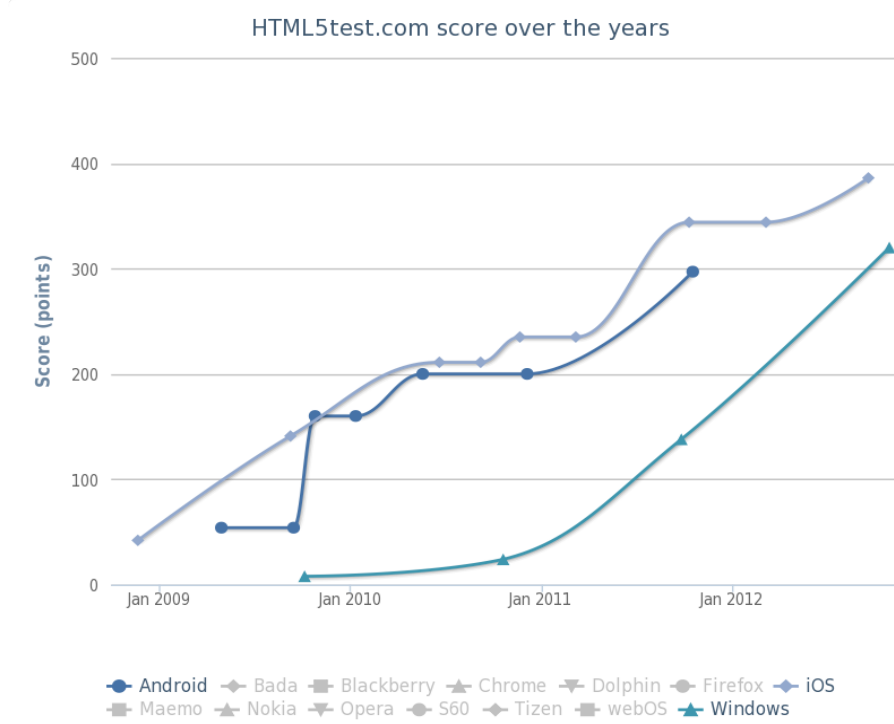


Figure 5.1: HTML5Test.com: Android, iOS and Windows Phone Score Chart

5.2.2 CSS3

HTML alone was not intended to be used for visualising web pages, but for holding the content of the page. Thus, the W3C consortium introduced the Cascading Style Sheets (CSS) language for describing presentation semantics of HTML documents. CSS does this by declaring rules that affect, among other things, elements such as layout, colours and fonts. This enable developers to use CSS to specify multiple presentation formats for the same content. In a mobile development setting, this can be used to deal with the wide variety of devices, platforms and browsers that the users might be using. In the latest version, CSS3, a new rule was introduced called “media queries”. The rule allows the selection of different views based on characteristics of the device the content is displayed on (usually the width of the screen). This rule has become a cornerstone in what is called Responsive Web Design, where the focus is designing self adapting web content to provide an optimal viewing experience across a wide range of devices.

²⁰<http://html5test.com> (21.11.12)

²¹<http://html5test.com/results/mobile.html> (21.11.12)

5.2.3 JavaScript

JavaScript is a general-purpose, multi-paradigm programming language developed by Netscape. It follows a C-like syntax and has (despite the name) nothing to do with the Java language. JavaScript has grown to become *the* script language of the web and can be used to validate input, communicate with the servers and add all kinds of functionality to HTML documents. To run JavaScript code a JavaScript engine is needed, and the wide use and applicability of the language has caused JavaScript engines to become standard in all major web browsers. Because of the language's versatility, it is one of the most common language used for making tool sets, libraries and frameworks to be used in web development for both mobile and standard platforms.

5.2.4 WebSocket

A very promising part of the HTML5 initiative is the WebSocket Protocol²² which enables two-way communication between a client and a remote host through a single socket. The protocol consists of two parts; a “handshake” and the data transfer. After the handshake has been established, the websocket can be used to push requests and data from both ends of the connection. This opens up for development of advanced browser-based applications. When using the WebSocket API, a developer has to handle the simple events; `onopen`, `onclose`, `onerror` and `onmessage`. This simplifies development, because the developer does not have to worry about the finer mechanics of network programming, and can instead focus on building functionality.

5.3 Platforms

For a mobile developer, there are many different platforms that has to be considered. The two major platforms are Google's Android and Apple's iOS, which together have a 89,9% market share of world wide sales in the third quarter of 2012 [10]. However, there are also other mobile operating systems to consider, especially the up and coming Microsoft Windows Phone 8 that was released in October 2012. In this section, we will review these major platforms and look at how development of applications for each respective platform can be conducted.

5.3.1 Google Android

Android is a software stack for mobile devices that includes a Linux-based operating system, middleware and some key applications. While primarily designed for touch based devices as smartphones and tablets, Android can be adapted to be run on an astounding range of devices including laptops, netbooks, smartbooks, ebook readers, portable media players and set-top boxes.

²²<http://tools.ietf.org/html/rfc6455> (16.11.12)

Development

The Android SDK²³ provides the tools and libraries necessary to begin developing applications that run on Android-powered devices. The Android SDK includes a set of development tools including a debugger, libraries, documentation, sample code and tutorials, as well as the Android Emulator. The SDK is split into an SDK Manager and an Android Virtual Device (AVD) Manager. The SDK Manager lets the developer choose to install different packages of tools and platform versions, while the AVD Manager allows the developer set up several AVDs with varying specifications. This makes it possible to install, run and test applications on a range of different Android versions, screen sizes and SD card sizes. The emulator also allows for data, like phone or SMS interrupts and GPS readings, to be sent to it through a telnet connection. Using the OpenIntents SensorSimulator²⁴, developers can also simulate and play with a range of sensors like accelerometer, compass, temperature, light and proximity.

For Android development, the applications model and controller are mainly coded in Java, while the user interface is structurally set up in XML files and consists of Layout and View elements. Android applications are compiled to run on Google's own Dalvik Virtual Machine where each application runs on its own separate Virtual Machine. An important aspect of Android development is to deal with the diversity of devices running on different screen resolutions, hardware, sensors and platform versions.

5.3.2 Apple iOS

iOS is Apple's mobile operating system that runs on all their mobile iFamily devices including the iPhone, iPod and iPad, as well as on the second generation Apple TV. With the iPhone's release in 2007, many say that Apple sparked the rapid growth and widespread use of smartphones and has since been a market leader and a trendsetter.

Development

iOS Development requires a Macintosh computer with the iOS SDK and the Xcode IDE which together delivers a comprehensive collection of tools for building iOS applications including the iOS Simulator, graphical debugger, performance data instruments and an Interface Builder. While the SDK and Xcode IDE are free of charge, it only allows the developer to run applications on the iOS Simulator. To be able to run applications on a iOS device, developers need to join the iOS Developer Program which has an annual cost of \$99²⁵. This subscription allows the developer to submit the application for distribution through the App Store.

Applications for the iOS platform are mainly written in Objective-C, allowing for some content to be written in C or C++, using the Xcode IDE. Xcode project can be specifically targeted to a specific iFamily device, or run full-screen on the iPhone/iPod and in a selected resolution for the iPad. The user interface of an iOS application is built by using the Cocoa Touch framework, which follows a Model-View-Controller (MVC) architecture.

²³<http://developer.android.com> (19.11.12)

²⁴<http://www.openintents.org/en/node/6> (19.11.12)

²⁵<https://developer.apple.com/programs/ios/> (31.05.12)

5.3.3 Microsoft Windows Phone

Windows Phone is Microsoft's new mobile operating system that succeeds their previous Windows Mobile platform. In February 2011, Microsoft and Nokia announced plans to form a broad strategic partnership, where Nokia declared they would adopt Windows Phone as their primary smartphone platform [5]. This has lead Nokia's own Symbian-based smartphones, and the Symbian platform itself, to a gradual decrease in market share, and subsequently to an increased market share for Windows Phone[11]. With the Windows Phone platform, Microsoft introduced the new Metro UI design which focus heavily on creating a clean and simple interface that makes it easy to read and use while on the go.

On October 29th 2012, Microsoft officially launched the newest version of their mobile platform Windows Phone 8 (WP8) (also known as Apollo). With WP8, Microsoft replaced the former Windows CE based architecture of Windows Phone 7 (WP7) with an architecture based on the Windows NT kernel. This allows for easy porting of applications between WP8 and Microsoft's new PC operating system Windows 8 which was released only three days before WP8 on October 26nd.

Development

To develop applications for Windows Phone a Windows computer and the Windows Phone SDK is needed. The SDK is bundled with Visual Studio Express, Expression Blend for Visual Studio, the Windows Phone Emulator, Silverlight SDK, Windows Phone SDK extensions for Microsoft XNA Game Studio, and a Microsoft Advertising SDK. Instead of inventing a new language or framework for their Windows Phone platform, Microsoft simply adapted their existing frameworks. Thus applications are written in C# and Visual Basic using the .NET Framework. Applications can be made using the Silverlight or XNA Frameworks or in a combination of the two which both uses the core .NET framework. Silverlight is mainly used for development of business and regular 2D applications, using the Extensible Application Markup Language (XAML) for creating and structuring the user interface. On the other hand, XNA is used mostly for creating games, allowing for development of loop-based games as well as 3D games using the XNA game engine [16].

5.3.4 Other: Symbian, BlackBerry OS, Bada, webOS

In addition to the major smartphone platforms described, there are several other platforms to consider, Symbian being the biggest contender amongst them. However, most of the other platforms (BlackBerry, Bada, webOS) have never had any real market share in Norway. With Windows Phone taking over for Symbian as Nokias main smartphone platform, Symbian will not be considered as a potential target platform for this project.

5.4 Cross-Platform Development

For a mobile application to become truly successful, it has to cater to more than a single platform. However, the process of porting an application between platforms is costly. Therefore quicker and easier development of cross-platform applications is

highly sought after. To cater to these needs, a range of tools and SDKs have been developed allowing development of apps using a single code base that can be compiled and run on multiple platforms. This section presents short reviews of a selection of these tools, APIs and SDKs in order to evaluate their compatibility with our project and goals.

5.4.1 PhoneGap

PhoneGap²⁶ is an open-source development framework that allows for the development of cross-platform mobile apps using web technologies while giving the developer access to native functionality through their JavaScript API. It allows for a single code base, written in HTML, CSS and JavaScript, to be compiled and deployed to all the major platforms (iOS, Android and Windows Phone), as well as some of the less used platforms Blackberry OS, webOS, Symbian and Bada. The application content is not converted to native UI components but rather wrapped in native code and displayed using it's target platform browser object (e.g. WebView on Android). One of the most powerful features of PhoneGap is its plug-in architecture. It allows the developer to write native code and make a plug-in which makes the native code accessible from the HTML/JavaScript user interface. PhoneGap offers a build service called PhoneGap Build that lets developers upload their application source and get apps ready for all the supported platforms. This allows developers to build their applications for all supported platforms without the need of downloading and installing all necessary SDKs and compile the application themselves.

5.4.2 MoSync

The MoSync SDK²⁷ makes it possible for the developer to develop mobile applications for all major mobile platforms from a single code base. It runs on both Microsoft Windows and Apple OS X and includes an Eclipse-based IDE. Applications can be coded in HTML5/JavaScript, C/C++, or a combination of the two to make hybrid applications. MoSync is open-source and free for private use and currently supports the iOS, Android, Windows Phone, Windows Mobile, Symbian, Blackberry, JavaME and Moblin platforms. MoSync supports both the use of PhoneGap to wrap the application into native code, and the possibility to create fully native UI through what they call their Wormhole JavaScript technology. The MoSync SDK also provides a development tool for rapid development called MoSync Reload²⁸, which allows the developer to quickly and easily deploy the application to several devices and/or emulators. It consists of three parts; a client application that is installed on the devices and/or emulators, a server running on the developers PC/Mac/Linux machine and a HTML interface also running on the development machine.

²⁶<http://phonegap.com/> (19.11.12)

²⁷<http://www.mosync.com/sdk> (24.10.12)

²⁸<http://www.mosync.com/documentation/manualpages/mosync-reload> (24.10.12)

5.4.3 Rhodes

Rhodes²⁹ is a cross-platform mobile development framework written in Ruby, that uses a Model-View-Controller approach. Applications are mainly written in Ruby but some development can be done by using HTML and JavaScript. Rhodes can then compile the application into fully native applications for several platforms including iOS, Windows Mobile, BlackBerry, Symbian and Android. The Rhodes framework mainly targets development of enterprise applications where the user interface mainly consists of standard input/output components.

5.4.4 appMobi

The appMobi³⁰ framework focuses on delivering products that supports the shift from proprietary native tools and services over to the industry standard that is HTML5, CSS3 and JavaScript. appMobi does this by delivering a set of tools to the developer as jqMobi, PhoneGap XDK, Impact GameDev XDK and the appMobi XDK. jqMobi is a rewritten version of the jQuery JavaScript library that simplifies HTML document traversing, event handling, animating and Ajax interaction, especially for mobile HTML5 on iOS and Android devices. The PhoneGap XDK adds to the existing PhoneGap API by offering multi-device simulation and debugging, both local and remote on-device testing, as well as a cloud-based build service (subscription required) that builds both for iOS, Android and the open web. The Impact GameDev XDK is a development tool pack sold by appMobi that delivers among other things an accelerated canvas for 2D games, the Impact game engine for HTML5 JavaScript games and a prepaid subscription to their cloud services. The appMobi XDK, much like the PhoneGap XDK, offers multi-device simulation and debugging, both local and remote on-device testing, as well as a cloud-based build service that builds both for iOS, Android and the open web. The appMobi XDK, as opposed to the PhoneGap XDK, builds the application into native applications instead of just wrapping the HTML and showing it through a browser object.

5.4.5 Titanium Mobile

The Titanium Mobile SDK³¹, delivered by Appcelerator, is a framework where development is done through their JavaScript based framework and Eclipse based IDE. The framework allows for development of both hybrid or mobile web apps. Appcelerator's main focus with the Titanium SDK has been to make a framework that builds native applications for the iOS and Android platforms that performs and behaves as though they were written in their native languages.

²⁹<http://rhomobile.com/products/rhodes/> (31.05.12)

³⁰<http://www.appmobi.com/> (31.05.12)

³¹<http://www.appcelerator.com/platform/titanium-sdk> (31.05.12)

5.4.6 Sencha Touch

Sencha Touch³² is a JavaScript Framework for building mobile web apps that look and feel like native apps for iPhone, Android and BlackBerry touch devices. As a pure web app it lacks the ability to access device components. However, the Sencha SDK tools counter this by giving the developers the opportunity to wrap the web app in a native shell, for both Android and iOS. The framework embeds a Model-View-Controller (MVC) structure and includes libraries for both storing data and building user interfaces.

5.4.7 jQuery Mobile

The jQuery Mobile³³ (JQM) Framework is, much like Sencha Touch, a touch optimised framework for building cross-platform HTML5 user interfaces for all the major smartphone and tablet platforms. JQM is built on the larger jQuery framework made for development of desktop browser applications. JQM is split into two major components, namely an Ajax navigation system that delivers animated page transitions and the jQuery UI that provides a set of core UI widgets and styles everything by using CSS.

³²<http://www.sencha.com/products/touch> (31.05.12)

³³<http://jquerymobile.com/> (19.11.12)

Chapter 6

Server Technologies

Several technologies were evaluated as candidates for use in the implementation for the server-side of the software prototype. Criteria for selecting a technology include the possibility to maintain a reliable connection between the server and the client, as well as ease of development. The developers can not be tasked to “re-invent” the wheel, and therefore need a solution that already offers some basic features to build upon. This chapter presents some candidates for evaluation.

6.1 Jetty

The Jetty³⁴ project which started in 1995 is based on Java and is a HTTP server and Java Servlet container. Jetty is capable of serving both static and dynamic content as it supports the WebSocket protocol among other integrations. The Jetty components are open source and available for commercial use and distribution and is currently used by many large and popular projects like the Spring Framework³⁵ and Maven³⁶ among others. Jetty is easily integrable with the Eclipse IDE. Setting up a project only requires the developer to set up a new Jetty project, define an index-file, and then the development can be done directly in the index.html-file.

6.2 nodeJS

nodeJS³⁷, originally created in 2009, is a simple, fast and scalable solution for web-servers that is built on Chrome’s JavaScript runtime. The scalability of nodeJS is one of the things that has made it the favourable technology to use for many different web-pages. NodeJs is currently used by several large companies including LinkedIn³⁸ and Microsoft³⁹ among others. Development for nodeJS is done using JavaScript, which is a very mature and well documented language. nodeJS uses http-requests to send mes-

³⁴<http://jetty.codehaus.org/jetty/> (29.11.2012)

³⁵<http://www.springsource.org/> (16.11.2012)

³⁶<http://maven.apache.org/> (16.11.2012)

³⁷<http://nodejs.org/> (16.11.2012)

³⁸<http://www.linkedin.com/> (16.11.2012)

³⁹<http://www.microsoft.com/> (16.11.2012)

sages, but can be extended to support WebSockets through Node Packaged Modules (NPM). There are many ready-to-use modules that are freely available.

6.3 jWebSocket

jWebSocket⁴⁰ is an open source solution for real time applications. Among other functionalities, it offers a reliable TCP connection between a server and a client. jWebSocket uses tokens with information on the form of JavaScript Object Notation (JSON), that is processed on the server side or client side. JSON is a lightweight data-interchange format⁴¹ which is language independent and human readable and writeable. jWebSocket is based around the concept of plug-ins. The framework comes with a System plug-in that gives basic functionality for connectivity. To extend with custom functionality, a Java class is required that extends the TokenPlugIn from the jWebSocket framework. This class then includes basic methods for when a client connects or disconnects and a method to process an incoming token. Incoming tokens are passed through the Plug-In Chain on the server and processed every time that particular token is picked up. This means that it is possible to have multiple plug-ins that process the same token differently. This is shown in Figure 6.1⁴².

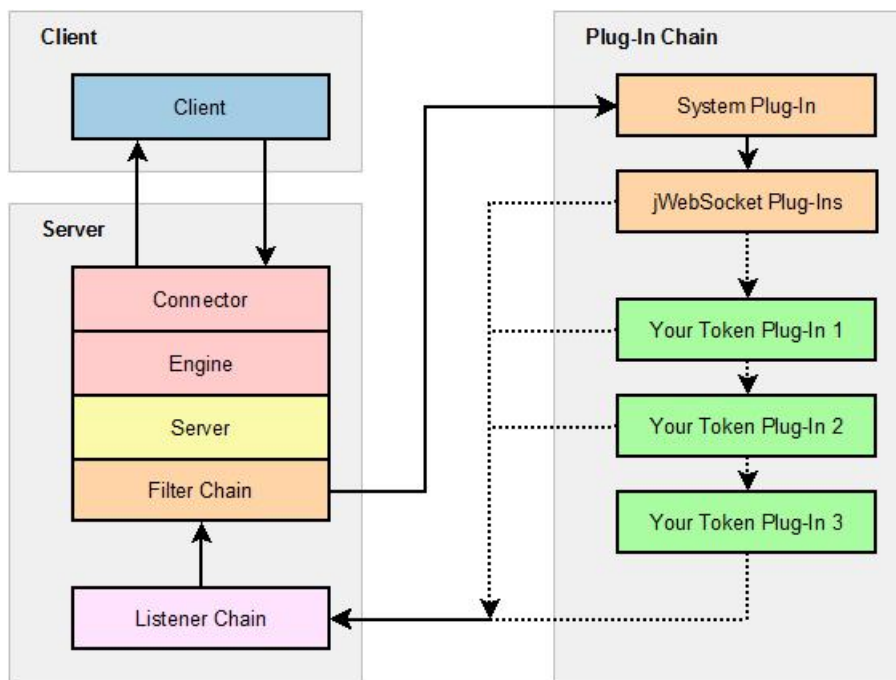


Figure 6.1: jWebSocket plug-in infrastructure

⁴⁰<http://jwebsocket.org/> (16.11.2012)

⁴¹<http://www.json.org/> (16.11.2012)

⁴²http://jwebsocket.org/infrastructure/is_pluginchain.htm (16.11.2012)

Chapter 7

Game Studies

Different aspects of gaming is studied in this section. Choosing the right genre, including elements that make the game fun and balanced is crucial for a game to be successful. The game also has to incorporate the right amount of learning elements so that the user is not only playing the game to have fun, but is also learning.

7.1 Genres

Games are different from other media by how the user interacts with it, rather than just watching passively. Apperley defines how genres are categorizations on how the user interacts with the game and presents a genre study elaborating on a video game genres like simulation, strategy, action and role-playing [2]. The following section presents and discusses Apperley's findings and elaborates on genres that might suit this research project's needs.

7.1.1 Simulation

This genre consists of games that try to emulate situations and make them as close to the real world as possible. Playing simulation based games can make the user better at performing the equivalent real world tasks. Games include driving or flying simulators, for example Microsoft Flight⁴³ (see Figure 7.1) or the Gran Turismo series⁴⁴, sports games like Fifa (also known as Fifa Football or Soccer)⁴⁵ or the real-life simulator The Sims⁴⁶.

⁴³<https://microsoftflight.com/> (16.11.2012)

⁴⁴<http://www.gran-turismo.com/> (16.11.2012)

⁴⁵<http://www.ea.com/fifa> (16.11.2012)

⁴⁶<http://thesims.com/> (16.11.2012)



Figure 7.1: Microsoft Flight gameplay screenshot

7.1.2 Strategy

Strategy based games are usually played with the camera angle in a top-down isometric view, and can be grouped into two categories: Real-time Strategy (RTS) and Turn-based Strategy (TBS). Examples of RTS games include StarCraft⁴⁷ (see Figure 7.2) while examples of TBS games include Chess and Sid Meier's Civilization⁴⁸. In RTS games, the action occurs in real time which requires the player to make split second decisions in the heat of the battle, whereas in TBS games, the player is able to make informed and wise choices properly thought through before setting them in to action. Games range from planning and building a city and making its citizens happy, to building a base and troops to eradicate the enemy.



Figure 7.2: Starcraft gameplay screenshot

⁴⁷<http://eu.blizzard.com/games/sc/> (16.11.2012)

⁴⁸<http://www.civilization.com/> (16.11.2012)

7.1.3 Action

Action games allow the player to control a character or a group of characters usually referred to as “avatars”. The genre consists of two major subgenres: First-person Shooter (FPS) and Third-person games. While the FPS genre includes games like Quake⁴⁹ and Half-Life 2⁵⁰, the Third-person game genre includes the more adventurous Red Dead Redemption⁵¹ (see Figure 7.3) and the Tomb Raider series⁵². FPS games are played from the perspective (through the eyes) of the protagonist, usually with a gun in hand, while Third-person games places the camera a short distance behind the players’ avatar. Gameplay often consist of the player running through a certain level or area in the game, in which, the game presents dangerous tasks, obstacles or enemies the player has to get past in some way.



Figure 7.3: Red Dead Redemption gameplay screenshot

7.1.4 Adventure games and Role-playing games (RPG)

The role-playing genre for video games originated from Dungeons & Dragons (D&D). D&D is a fantasy pen-and-paper RPG played by having one designated “Dungeon Master” who decides how different parts of the game will act out while the players makes choices along the way, rolling one or several dices to determine the success of their choice. RPGs are usually set in a period resembling the Middle Ages and incorporates sword fighting, horse riding and the use of magic to defeat foes along the way to success. The RPG genre is not limited to one setting [1]. The most popular setting for Massively Multiplayer Online Role-Playing Games (MMORPG), and to this

⁴⁹<http://www.idsoftware.com/games/quake/quake> (16.11.2012)

⁵⁰<http://www.half-life2.com/> (16.11.2012)

⁵¹<http://www.rockstargames.com/reddeadredemption/> (16.11.2012)

⁵²<http://www.tombraider.com/> (16.11.2012)

extent RPGs in general are: Fantasy/Medieval, Futuristic, Post Apocalyptic, Outer Space and Contemporary.

Adventure games are somewhat similar to RPGs, but differ in the way the player advances in the game. Adventure games usually do not include features such as fighting and killing, but rather presents the player with difficult puzzles and challenges. Popular MMORPG's include World of Warcraft⁵³ and EVE Online⁵⁴, popular single-player RPG's include the Final Fantasy series⁵⁵ (see Figure 7.4) and Baldur's Gate⁵⁶, while popular adventure games include The Longest Journey⁵⁷ and The Secret of Monkey Island⁵⁸.



Figure 7.4: Final Fantasy VII gameplay screenshot

7.1.5 Fighting

Apperley does not describe the fighting genre of video games [2]. It is, however, a prevalent genre for video games. Fighting genre games are action oriented, but usually limited to the one-versus-one setting of fighting your opponent. An example of an early fighting game is Heavyweight Champ, made by Sega for arcade machines which came out in 1976. As the years have passed and the genre has grown, a multitude of fighting game genres have been created. A fighting game includes a diverse range of characters to choose from. Usually all of them have different fighting styles which are almost always based on real world martial arts or fighting styles. The more successful fighting game franchises from the 1990s and 2000s include the Street Fighter series, the Virtua Fighter series⁵⁹, the Tekken series⁶⁰, and the Dead or Alive series⁶¹ (all which have inspired a large amount of spin-off games and movies). Figure 7.5 shows the game

⁵³<http://eu.battle.net/wow/en/> (16.11.2012)

⁵⁴<http://www.eveonline.com/> (16.11.2012)

⁵⁵<http://eu.square-enix.com/games> (16.11.2012)

⁵⁶<http://www.baldursgate.com/> (16.11.2012)

⁵⁷<http://www.longestjourney.com/> (16.11.2012)

⁵⁸<http://www.lucasarts.com/games/monkeyisland/> (16.11.2012)

⁵⁹<http://www.sega.com/games/virtua-fighter-5/> (16.11.2012)

⁶⁰<http://www.tekken.com/> (16.11.2012)

⁶¹<http://teamninja-studio.com/doa5/uk/top.html> (16.11.2012)

“Street Fighter X Tekken”, which is a crossover fighting game. Here characters from the Street Fighter universe meet characters from the Tekken universe for an ultimate showdown. Crossover games are common in the fighting genre. They may include only one character from another game, or the entire cast of two or more games. Popular crossover games include the Super Smash Bros. Brawl⁶² and the upcoming Playstation All-Stars Battle Royale⁶³.

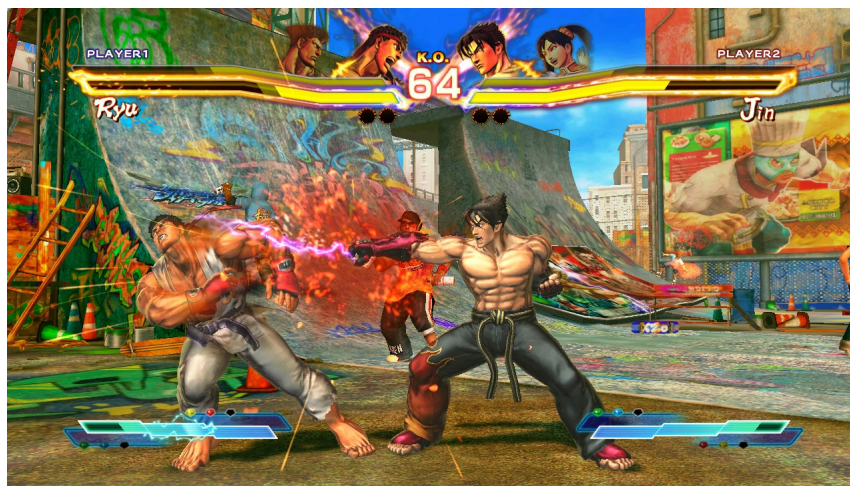


Figure 7.5: Street Fighter X Tekken gameplay screenshot

7.2 Flow Theory

The flow theory is an important aspect that should be considered when dealing with a situation that involves deep concentration.

“Flow is the mental state of operation in which a person in an activity is fully immersed in a feeling of energized focus, full involvement, and success in the process of the activity.” [7]

This can be used to describe a person immersed in an activity that requires deep concentration, like playing video games. To get a player to be in a state of flow requires following a golden mean; the player cannot become bored because the challenges are too easy, nor can the player become frustrated because the challenges are too hard. It has also been suggested that being in a state of flow can be used to increase motivation, which in turn can be used to increase the learning effect. It must be noted that the same research also shows that one of the most important parts of learning is reflection. While in a state of flow, players rarely exercise reflection on the learning part of the game [20].

There is also presented a solution to this:

“Reflection must appear to the learner as one of the many in-game goals that drive the game-play.”

⁶²<http://www.smashbros.com/> (16.11.2012)

⁶³<http://www.playstationallstarsbattleroyale.com/> (16.11.2012)

This means that if the reflection part of the learning is disguised as an element the player *must* complete in order to advance to the next level of game-play, the player can be “forced” to reflect upon the learning being done. Although, as mentioned, one must also take care not to make this element too boring or challenging so that the player can maintain focus and the state of flow.

7.3 Games and Education

To be able to reach a state of flow, as mentioned in the previous section, the learner first has to be motivated. Getting learners motivated is not easily done, especially if the task the learner is about to perform is difficult. James Paul Gee proposes that there are two options available when faced with trying to get learners to do difficult tasks [8]: Forcing them to do assignments or tempt them with some sort of profit. The article goes on presenting a “baker’s dozen” of examples on principles of learning that are built into good video games. These principles include that the learner should be faced with problems that are well designed and that the problems are just challenging enough. The learner should have to repeat similar types of exercises until they are nearly automatic. Information should only be available when relevant and important to the task. Learners should also be able to practice their skills in context, as well as be able to fit the experience into an overall larger system, this will help give the experience a meaning. The article concludes that few of these principles are implemented in educational games, whether it be in school, for businesses or workplace learning.

7.4 Multiplayer Game Balancing

Although this project will focus on education by repeated learning, no one likes to lose a competitive game unfairly because the opponent has some kind of unfair advantage, is higher level than you, or wields better equipment. Making a multiplayer game balanced is not an easy task. A game should be fun for a wide variety of players, and alternative playing styles should exist so that the game can suit the individual player. The different playing style options should provide the player with an alternative way of playing the game, but the game designer also have to make sure that one style of playing will not dominate another. Habgood and Overmars has written a book about the making of games and proposes a few alternative ways of balancing a game [9]. First, give the players a very similar set of character classes to choose from. This is less desirable, as it gives no real alternative playing style. Second, the players can be provided with characters that incorporates different strengths and weaknesses. Third, it is possible to weigh in choices so that the players have to choose between different trade-offs. Finally, an implementation of a cyclic relationship can provide a richer type of gameplay. The only way to be able to balance a game is according to the chapter in the book by Habgood and Overmars, to continue is to test the game repeatedly.

Chapter 8

Prestudy Summary

The prestudy part of this thesis will here be viewed in light of the problem description from Section 1.3 and the research questions from Section 2.1 and Section 2.2, in order to make decisions on what technology to be used, which game elements to implement and what to keep in mind when conducting our research.

8.1 Earlier Work

Through the first KnowledgeWar project, Sveinung Kval Bakken showed some promising results in terms of student interest in a pervasive knowledge game, along with suggestions for further work. A main area of improvement suggested from the first project was increased use of gameplay elements. The game was implemented using very little graphical elements and gameplay consisted only of answering quiz questions. A suggested solution was to set the gameplay into a virtual world RPG context and create a more compelling visual experience.

The LectureQuiz and CampusQuest projects serves as good sources on how to conduct quality quiz based learning. Especially LectureQuiz has shown good improvements and is currently in a third version. Study of the LectureQuiz game indicated that students found the game engaging they and perceived to learn more by playing the game. The LectureQuiz research further validates the use of a quiz game to be highly suitable for learning in higher education through not only the software's content but by engaging the students and increasing their attention level. A multiplayer educational game can bring forth the social aspect of learning as well. When playing such a game, players might not want to lose against fellow students which increases both attention level and focus among the students. Students might end up discussing the answers and transfer knowledge between themselves to learn more and do better in an educational game.

Amendor's quiz app framework serves as an example of how one can create a cross platform solution that has the same look-and-feel on both the iOS and Android platforms. Their app framework also show how points and unlocking levels together with connection to social medias can be used to motivate the user to learn more.

The "World of Wisdom" and "Using a PDA for mobile learning" projects show how learning can be set in an RPG setting, where the player's progress is rewarded by experience points toward its character. They further show how the use of flow theory

can captivate the user into wanting to play the game more than just to learn the course curriculum.

Through the study of the pervasive virtual soap opera “SupaFly”, Kalle Jegers and Mikael Wiberg identified several challenges a researcher may meet while studying a pervasive game. As pervasive games are made to be played any time and anywhere conducting observations of real world usage of the game is quite tricky. The study further suggests strategies as system logs, qualitative questionnaire and focus group interviews, in order to compensate for the lack of opportunity to study the game in real life.

Through the implementation of a mixed-reality capture the flag game, the study showed that players found it both innovative and fun when the game mixed real-world players and virtual-world players.

Though not aimed at higher education, Buzz! The Schools Quiz serves as a great example of how a successful commercial video game have been turned into an educational game. The game shows how a fun and engaging game also can have a more hidden educational side.

8.2 Mobile Technology

The easiest way of making a cross-platform mobile application is to make a web application that can be used through a mobile web browser. This eliminates most issues with development of a native application across the different mobile operating systems (iOS, Android, Windows Phone). However, a web application has some restrictions such as limited performance, speed and latency, but most notably, that it cannot access device specific services and technologies. A hybrid application solves this problem by letting developers create cross-platform user interfaces using web technology (HTML, CSS and JavaScript) while device specific development can be done either separately or through a common third-party API. There are several third-party solutions to choose from, but one of the more popular and widely used APIs is PhoneGap. PhoneGap is a complete open source solution which additionally provides a build service that can build projects for all its supported platforms, which covers all the major operating systems, free of charge for public applications.

8.3 Server Technology

To be able to keep a persistent connection between the client and the server, an implementation of WebSockets is required. All three of the technologies considered offered a solution for WebSocket implementation. Jetty and jWebSocket are both based on Java, while nodeJS is based on JavaScript. They are all well implemented and documented, and can boast of a very broad range of users. Using a database with Java requires a simple JDBC driver which is included with the Java SDK. A database for a JavaScript implementation can be implemented using the light-weight SQLite Relational Database Management System.

8.4 Game Studies

The desired effect of a learning game is that users continue to use the game even though they are not required to. The game has to offer the user more than just repeated questions from the player's course curriculum. Even though Kval Bakken mentioned Role-Playing Game (RPG) as a possible genre for this type of game, other genres were evaluated to be absolutely certain the right setting is chosen before development is started. When evaluating the different types of genres it became clear that there were multiple genres that could fit. The prototype will also make use of elements from multiplayer games, this will introduce some latency during gameplay and thus the "Action" or "Real-time Strategy" genres are not ideal choices for the prototype. However, to make the user want to come back and play the game, persistent game elements can be introduced. This will give the player an incentive to come back to the game world. persistent game elements are characteristics of the RPG genre, and this can be an ideal choice for the learning game prototype. The player chooses an avatar to represent him or her in the game. The avatar comes with some basic attributes that governs the amount of health points, the strength of the attacks and so on. The player can also earn Experience Points (XP) by winning battles. With enough XP, the player advances to the next level, adding bonuses to the basic attributes and making the character stronger. In this way the user is rewarded for playing the game, not just because of the benefit of learning a curriculum's topics, but also because of the element of competition and the excitement of becoming stronger and facing more difficult challenges.

To promote further use of a learning game, it should incorporate elements of such that the user can achieve a state of flow. If done correctly, the user would want to play the game because he or she thinks the game in itself is fun, not just because it has a specific task of learning a course's curriculum. Reflection will have to be a part of the game, if the students who will be using the game is going to be able to learn. As explored in Section 7.3, there exists a study on what elements a learning game should incorporate in order to appeal to a learner as well as make it a good experience.

Finally, a multiplayer game have to be balanced so that players do not feel like they are in a hopeless situation and that the opponent is favorably stronger by the design of the game.

Part IV
Own Contribution

Chapter 9

Concept Sketch

The game, like its predecessor, is to be centred on quiz battles fought between the players. The game should further implement RPG design techniques to engage the player and create immersion. Each player will be associated with a character that can evolve by gaining experience and equipment through battles and exploring a virtual game world.

9.1 Influences and Inspiration

During the initial discussions of how we pictured the game as well as features we wanted to implement, it was easy to draw upon known existing games which implemented a similar feature. For the battle itself we pictured the opponents facing each other with some basic information such as name, level and health-bar next to their avatar. Here, the inspiration was mostly based on battle and fighting games like Pokémon⁶⁴, Tekken⁶⁵ and Street Fighter⁶⁶ (see Figures 9.1 and 9.2).



Figure 9.1: Screenshot from two generations of Pokémon games

⁶⁴<http://www.pokemon.com/> (19.11.12)

⁶⁵<http://www.tekken.com/> (19.11.2012)

⁶⁶<http://www.streetfighter.com/> (19.11.12)



Figure 9.2: Screenshot from Street Fighter (left) and Tekken (right)

As the game was to be set in a virtual game world and incorporate place dependent gameplay of some sort, the player will need an illustration or map of this world, which display the player in relation to his surroundings. An immediate idea of implementing a fog-of-war type of map was presented, where only a limited area surrounding the player is visible. Undiscovered areas are completely black while earlier explored areas that are not in the players immediate surroundings are hidden by fog. This gives some extra excitement to the game as the player has to explore to find other players, enemies, locations and items. Figure 9.3 shows an illustration of how an implementation of fog-of-war has been done in a RPG game using RPG Maker XP⁶⁷.



Figure 9.3: Illustration of how fog-of-war might be implemented in an RPG

⁶⁷<http://www.rpgmakerweb.com/product/rpg-maker-xp> (14.11.12)

9.2 Early Sketches

While brainstorming about the concept we made some simple wireframes, as can be seen in Figure 9.4. These sketches helped get the ball rolling as we moved on to establish the initial game design and some requirements of the software prototype of the KnowledgeWar 2.0 game.

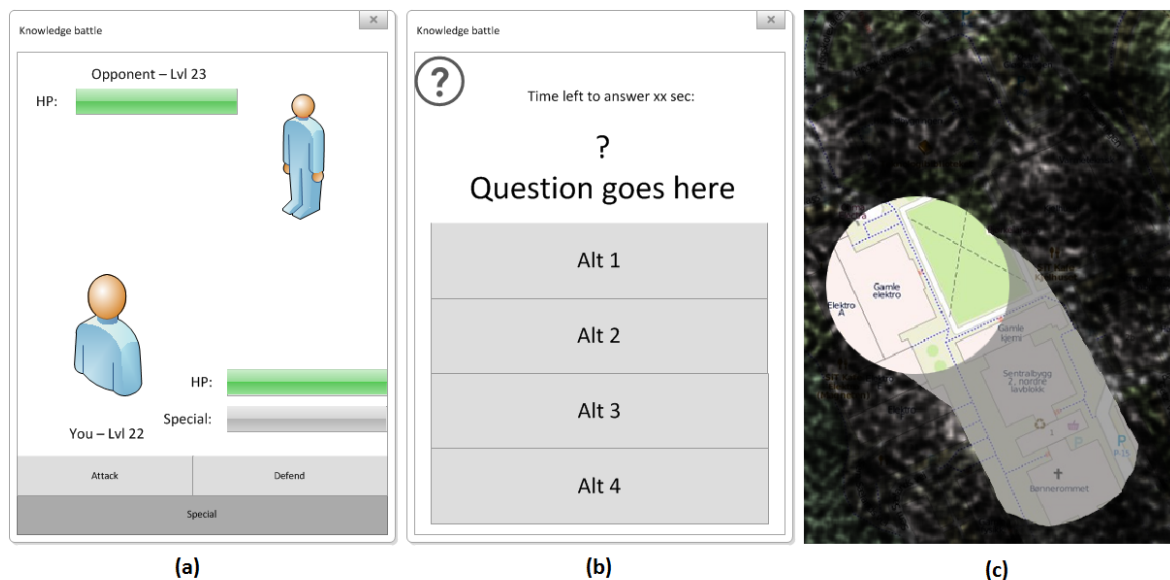


Figure 9.4: Early concept sketch from brainstorming session

The wireframes illustrate how we imagine the game screen would look while in a combat, see Figure 9.4 (a). It shows the portrait of the player's avatar, along with a bar to represent the amount of Health Points (HP) left, as well as a bar to indicate how much "Special Resources" the player has available. The buttons at the bottom of the screen are interactive in order to perform actions. The Special button is not available as the player does not have a full bar of "Special Resources". The wireframe also shows the opponent the player is fighting against. The player can only see the opponent's health bar with remaining HP. This is to hide how much of the special resources the opponent has available.

The quiz question screen will have to be as simple as possible, Figure 9.4 (b). It is planned to only show how much time the player has left to answer the question, along with the question itself and four buttons that represent the four alternatives the player is presented with.

The map screen, Figure 9.4 (c), shows the intended implementation of fog of war. The black area represents an area the player has yet to visit and which he cannot see. The grey area represents an area in which the player has visited, but is not updated live as the player is not currently there. The colored area of the map represents the area the player is currently located. This area will be updated live to reflect other players who move about and enter the area of vision for the player.

Chapter 10

Requirement Specification

This chapter outlines the functional, non-functional and technical requirements, which were established based on the main concept of the game, as well as the findings from the pre-study phase of this thesis.

10.1 Non-Functional Requirements

We believe that for an educational game to be successful, it has to primarily be a game, and second most be educational. This is supported by the findings presented in the game studies section, and lead us to define the non-functional requirements shown in Table 10.1.

Table 10.1: Non-functional requirements

ID	Name	Priority	Description
NFR1	Usability	H	The game should be intuitive and easy to use especially for people not familiar with the RPG genre.
NFR2	Fun	H	The game should be perceived as fun to play.
NFR3	Engagement	H	The game should be engaging and create motivation for the players.
NFR4	Immersion	H	The game should deliver gameplay that immerses the player in the game to further motivate the player.
NFR5	Educational	H	The game should foremost be able to help the players to memorize existing knowledge, and second, be able to give them new knowledge.

10.2 Functional Requirements

The requirements shown in Table 10.2 form the basis for the users interaction with the client software, as well as establish a framework of underlying services which the server has to provide for the client.

Table 10.2: Functional requirements

ID	Name	Priority	Description
FR1	Register	H	A user must be able to register an account that will associate him/herself and a device with a character in the game.
FR2	Log in/out	H	A user must be able to log in and out of the game.
FR3	Position	H	A user must be able to see his/her position on a map.
FR4	Fog-of-war	M	Only a part of the map surrounding the user shall be fully visible. Unexplored areas shall be black while earlier explored areas shall be covered by “fog”.
FR5	Discover	M	A user must be able to discover places, items and people using the map.
FR6	Interact	M	A user must be able to interact with special places on the map (e.g. Non-Playable Characters (NPCs) or Shops).
FR7	Pick up	M	A user must be able to pick up and choose whether or not to equip items he/she finds.
FR8	Challenge	H	A user must be able to challenge other adjacent players to a quiz battle. Players receiving challenges must be able to accept or decline challenges.
FR9	Battle	H	A user must be able to fight battles by choosing actions for his/her character and answering quiz questions.
FR10	Battle report	H	Users must be able to review their answers at the end of a battle.
FR11	Score	M	A user must be able to see his/her place on a scoreboard relative to other players.

10.3 Technical Requirements

Table 10.3 define some technical requirements for the system, which is found to be crucial in order to deliver the functional requirements defined in Section 10.2 and for delivering a reliable and fun user experience.

Table 10.3: Technical requirements

ID	Name	Priority	Description
TR1	Roaming	H	The system must be able to handle clients roaming as they move about on campus, keeping the connection to the server alive and reconnect if connection is lost.
TR2	Location	H	The system must keep an updated list of online players and their positions.
TR3	Persistence	H	The game world and character statistics must be persistent. Data should not be lost if the client loses its connection to the server, or if the user logs off and on again.
TR4	Platform	M	The game should work on 90 % of all Android devices

10.4 Summary

The requirements presented in this Chapter are meant to be used as a guide and goals for what we hope to accomplish with the software prototype. However, please note the time constraints of the project and amount of work necessary to be able to complete some of these tasks. It is not likely that we will have the opportunity to complete them all.

Chapter 11

Design

The design process for the prototype was run in two iterations. The initial design from the first iteration is a result of the developers' brainstorming, and the findings of the pre-study in Part III. The second design iteration follows after the feedback from the paper prototype tests.

11.1 Initial Game Concept Design

In order to provide an immersive and fun setting for the game, we created a humoristic take on the life and setting of being a university student and (most likely) being a geek of some sort.

11.1.1 Character Creation

To be able to create characters that fit well into the university setting of NTNU, Scott Johnson was contacted and received permission to use his "56 Geeks" project (see Appendix C), to provide a suitable set of avatars to choose from.

Classes

The game allows the players to choose one of four character classes, each with slightly different character attributes. The final avatars and classes chosen were an attempt to make the player identify themselves with the character in the game. See Figure 11.1 for the character images used for the prototype.

The four different classes available for players to choose include Computer Geek (a), Robot Geek (b), Lab Geek (c) and Math Geek (d). Each class has their own set of attribute points and different starting weapons and armor.

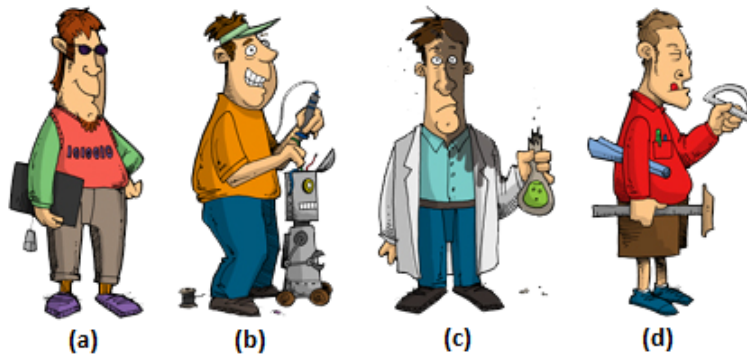


Figure 11.1: Initial Characters

Table 11.1: Starting attributes for the Computer Geek class

Health	150
Stamina	5
Strength	10
Intelligence	5
Agility	20
Weapon	Dirty Sticky Keyboard
Armor	All-weather Jacket

Table 11.2: Starting attributes for the Robot Geek class

Health	150
Stamina	5
Strength	20
Intelligence	5
Agility	10
Weapon	Flathead Screwdriver
Armor	All-weather Jacket

Table 11.3: Starting attributes for the Lab Geek class

Health	150
Stamina	5
Strength	5
Intelligence	20
Agility	10
Weapon	Test tube
Armor	Labcoat

Table 11.4: Starting attributes for the Math Geek class

Health	150
Stamina	5
Strength	10
Intelligence	20
Agility	5
Weapon	Pencil
Armor	All-weather Jacket

Character Attributes

Inspired by high-profile computer RPG's like World of Warcraft⁶⁸ and the Fallout series⁶⁹, character attributes for each class was worked out. Primary attribute points are multiplied by two and added to the base attack damage, while secondary attribute points are multiplied by 0.5 and added to the base attack damage.

- **Stamina**
Attribute which dictates the amount of Health Points (HP) a player has. The amount of attribute points in Stamina is multiplied by a constant and added to the base HP of the class.
- **Strength**
Primary attribute for attack strength for the Robot Geek class.
Secondary attribute for attack strength for the Computer Geek and Math Geek classes.
- **Intelligence**
Primary attribute for attack strength for the Lab Geek and Math geek classes.
- **Agility**
Primary attribute for attack strength for the Computer Geek class.
Secondary attribute for attack strength for the Robot Geek and Lab Geek classes.

Additional Attribute Points

Players are given an extra set of attribute points upon character creation to distribute as they see fit. This to encourage strengthening the character's primary and secondary attributes as well as giving the player a chance to customize the character to their liking. Upon increasing level, the players are provided with even more additional attribute points to distribute as the players want.

11.1.2 Gameplay

The game consists of three gameplay elements:

⁶⁸<http://eu.battle.net/wow/en/> (16.11.2012)

⁶⁹<http://fallout.bethsoft.com/> (16.11.2012)

Discovery

In discovery mode the player will be able to walk around the NTNU campus while seeing their character on the screen. While walking around the player will be able to discover items and other players currently playing the game. It is also possible to find treasure chests located around campus. These contain weapons and armor which can be equipped by a character, making it stronger

Battle

A battle consists of consecutive battle rounds. The battle round comprise two stages, where on stage 1 the players choose an action, on stage 2, players answer a quiz question. This continues until a player's Health Points (HP) has been depleted or a player forfeits the battle.

On the first stage, the player can choose one of three actions. Attack, Defend and Special Attack (where the Special Attack is only available once the player has accumulated enough special resources). The Attack actions gives damage to the opposing player. The Defend action makes sure that you receive no damage at all and will build up your Special meter more quickly. Finally the Special Attack action will unleash a powerful and devastating attack causing more damage than the normal standard attack. When both players have chosen an action, the battle round is taken in to stage 2. On this stage, both clients receive the same question and the same four alternatives. The alternatives are, however, randomized to prevent the player from knowing which alternative to answer the moment the player sees the question. Along with the questions and alternatives, the players are presented with a time limit for answering the question. When both players have chosen their answer, or the time has run out for one or both players, the round is finished. The outcome of the round is calculated at the end of each round, as well as a check to see if the battle is won by either of the players. A player wins if the opponents health points are depleted or the opponent decides to forfeit. If a player chooses to forfeit their opponent is notified and declared the winner, the round is finished as if a player had lost in a regular manner.

Battle Report

After the battle is finished players are taken in to a game screen and the battle report is displayed. Here, players have the opportunity to review all the questions that were presented during the battle rounds. The questions are displayed as a list where each question is presented as a button, which, when clicked on, takes the player to the same question game screen as they saw in the actual battle. However, the stats on this game screen is frozen so that the player can review their answer to the question, how much time was used to answer the question, and whether or not the given answer was the correct one. This type of after battle report is included based on Section 7.2 and how reflection must be a part of a learner's experience. Allowing players to reflect on their answer is a possible way for them to learn the right answer to the quiz questions and, in turn, their course's curriculum.

11.2 Paper Prototype

Based on the requirements presented in Section 10.2 and the game design in Section 11.1, a paper prototype was made in order to do some quick and dirty usability testing, and to get feedback from students which are in the potential target demographic. Pictures of the paper prototype can be found in Appendix B. The testing was done using a standard procedure⁷⁰. One played the part of computer, or in this instance smartphone, and was in charge of changing the screenshots based on the users input. The other was tasked with observing the user and take notes and pictures.

11.2.1 Test Subjects

The background of the test subjects was not very broad, but still gives a representable result for the target of the project. All seven that were tested were students of higher education. Five of them were students of computer science, one was studying to become a teacher and the last was a student of European studies. Four were males and three were females. All were in their mid-twenties. All of the computer science students had been a part of a paper prototyping at some point, while it was the first time for the two others.



Figure 11.2: Paper prototyping in action

11.2.2 Findings

Testing showed elaborate patterns to which pointed out some areas of concern.

- *Course name/code*
Several of the students did not understand why which courses they were taking was relevant, and did not necessarily know the correct name or code of all their courses.
- *Character class*
Test participants found it hard to make a choice in character class when registering as they at that point did not fully understand what and how the different

⁷⁰http://www.paperprototyping.com/what_howtotest.html (16.11.2012)

classes was related and affected the game. This lead most of them to just select a class almost at random.

- *Character attributes*

Character attributes had been named using traditional names from the RPG genre, and was meant to affect the character classes differently. Testing showed that users that were familiar with playing RPGs roughly understood their effect, while users not familiar with the genre did not.

- *Additional attribute points*

When tasked with placing additional attribute points for their characters, several of the students just ended up putting all their extra points on a single attribute, seemingly at random as they did not understand what the different attributes meant.

- *The map*

Almost all of the test participants expected the map to be interactive and tried clicking directly on items and players shown on the map. This functionality was not planned by the developers, who intended all actions from the user to be performed by buttons on the control panel, while the map just worked as a display.

- *Battle*

Test participants did not quite understand how the battle mechanics worked, and seemed to think actions was turn-based.

- *Battle report*

Some of the test participants did not understand that the battle report allowed them to revisit the questions from the battle to see where they made mistakes.

11.2.3 System Usability Scale (SUS)

The main reason the paper prototype was made was to get feedback on the game design, but the test participants also answered a SUS questionnaire (see Appendix G). On a System Usability Scale, the final score is an average of all the answers aggregated and applied a scoring method to. The scores are determined by taking the odd numbered questions scale position minus 1. For the even numbered questions, the score is 5 minus the scale positions of these. Taking the total sum of all the scores and multiplying by 2.5 gives the overall Usability Scale score. The results for the paper prototype shows an average score of 75.4 (as seen in Table 11.5). As the average score for System Usability trends towards 70 points, the score for the paper prototype can considered to be well above average⁷¹. Despite the prototype not being perfect, the test revealed that the score and general feedback from the test subjects gave a positive outlook on the project.

⁷¹<http://www.measuringusability.com/sus.php> (16.11.2012)

Table 11.5: SUS Scores for the Paper Prototype

SID	Statement	PP
S01	I think that I would like to use this system frequently	5.71
S02	I found the system unnecessarily complex	7.14
S03	I thought the system was easy to use	6.79
S04	I think that I would need the support of a technical person to be able to use this system application	8.93
S05	I found the various functions in this system were well integrated	6.79
S06	I thought there was too much inconsistency in this system	8.57
S07	I would imagine that most people would learn to use this system very quickly	8.57
S08	I found the system very cumbersome to use	7.50
S09	I felt very confident using the system	6.79
S10	I needed to learn a lot of things before I could get going with this system	8.57
–	Total Average Score	75.36

11.3 Game Concept Redesign

The paper prototype identified some key areas that needed to be changed. When working with the design of an application, it is easy to become blind with which features work and which that does not make sense to someone who has not seen the application before. The paper prototype gives the developers great feedback on which features actually makes sense when viewing the application for the first time.

Changes made to the findings from the paper prototype:

- *Course name/code*
A dropdown list was added which then listed the courses available, with both name and the course code. This replaced a normal text input box with an auto-complete function.
- *Character class*
The university themed classes were dropped in favour of more generalized classes. See Table 11.6
- *Character avatar*
Section of avatars no longer associated with the character classes. All of Scott Johnson's "56 Geeks" are now available for selection.
- *Character attributes*
As the naming of the attributes was shown to be confusing, they were renamed and redesigned to affect a character's action skill directly.
- *Additional attribute points*
Removed completely from the registration process. Attributes increase based on character class when leveling up.
- *The map*
The map was made interactive so that players may actually click elements and perform actions.
- *Battle*
The actions a player can perform during a battle was redesigned. Three types of attack along with the special attack was introduced. This meant that the damage a player does to its opponent is dictated by which attack is chosen, what weapon the player has currently equipped and what armor the opponent has equipped. The new attack types are named "Quick", "Normal" and "Heavy".
- *Battle report*
No changes made, but the review question buttons needs to clearly show that they are interactive buttons.

In addition, other changes were made to accommodate for smaller screen sizes (different Android devices).

- *Profile screen*

The profile screen was split in to two, clearing up the screen, which makes a better overview.

Table 11.6: Attributes comparison of the redesigned character classes

Class Name	Health	Attack	Defence	Special	Description
Average Joe	120	10	5	30	Average in every way.
Attacker	100	15	5	30	The best defence is a good offence.
Defender	100	10	10	30	The best defence is good runningshoes.
Specialist	100	10	5	50	Special in every way.

Table 11.7: Starting character equipment

Armor	Weapon
Green T-shirt	Green Bic Pen

Battle

Some changes to the concept of a battle round was made when redesigning.

First, on stage 1, the player can choose one of four actions. The mechanics behind attacks in combat were changed so that there are three types of (normal) attack and a special attack. The different attacks were named Quick Attack, Normal Attack and Heavy Attack. They do different damage based on the attributes of a character as well as weapon equipped by the character and armor equipped by the opponent. The Normal Attack does 100% of the character's base attack damage added together with the base attack damage range of the weapon the character is currently wielding. Quick Attack does 80% of the Normal Attack's damage while Heavy Attack does 120%. Using the Quick Attack will give the player a time bonus when answering the question. Similarly, using the Heavy Attack gives the player a time penalty. This is to give players that are insecure in the topic at hand some more time to think about the question and still be able to win a round. Likewise, a player well versed in a topic, and confident that the player can answer any question in a rapid fashion can choose a stronger attack. This will allow players that know the subject well to choose stronger attacks to end the battle more quickly. When both players have chosen an action, they are taken in to stage 2 of the battle. When they have both answered the question, or the time has run out, the round is finished. The battle is finished if either of the players forfeits the battle, or one of the players Health Points are depleted.

Battle report

As in the initial design, the list of buttons are kept. A list of buttons is more visibly clickable when displayed on a smartphone screen, unlike when vieweing a prototype concept art on a piece of paper.

Weapons and Armor

A whole range of weapons and armor was designed in order to be able to increase a character's potential damaging and defensive capabilities. In Table 11.8 the numbers represent the amount of extra armor added to the character's defense. In Table 11.9, the numbers represent the range of damage that a weapon can do. The distribution of weapons and armor are divided into tiers, where each tier has a level range of two levels. Full icon sets of the weapons and armor can be seen in Appendix D.

Table 11.8: All available armor

Tier	Name	Green	Blue	Red	Black	Level
1	T-shirt	1	2	3	5	1-2
2	Hoodie	6	7	8	10	3-4
3	Coat	11	12	13	15	5-6
4	Marius Sweater	16	17	18	20	7-8
5	All-Weather Jacket	21	22	23	25	9-10
6	Kevlar	26	27	28	30	11-12
7	Stormtrooper	31	32	33	35	13-14
8	Nanosuit	36	37	38	40	15-16

Table 11.9: All available weapons

Tier	Name	Green	Blue	Red	Black	Level
1	Bic Pen	5-7	6-8	7-9	9-11	1-2
2	Marker	10-12	11-13	12-14	14-16	3-4
3	Pin	15-17	16-18	17-19	19-21	5-6
4	Keyboard	20-22	21-23	22-24	24-26	7-8
5	Scissors	25-27	26-28	27-29	29-31	9-10
6	Bat	30-32	31-33	32-34	34-36	11-12
7	Acid	35-37	36-38	37-39	39-41	13-14
8	Tomahawk	40-45	45-50	50-55	60-75	15-16

Multiplayer Game Balancing

A function for matching every combination of class, weapon and armor up against each other was made to see and test the balance of the elements in the game. This made it possible to test the designed weapons and armor, make adjustments and test again. Classes were made similar, as the goal is not to invent a whole new type of multiplayer game, but to test a multiplayer game in an educational setting. Functional testing was also conducted, where the developers registered each their own character and tested the combat in a real setting.

Chapter 12

Implementation

The implementation is split into two parts, a server with a database and a client. The client-side application will be where the game is played and interacted with by the users, and will be implemented as a HTML5, CSS3 and JavaScript combination. The target platform is Android, however there is a possibility to implement it for multiple platforms. The server side will handle all the logic and communication needs for the client, following the requirements specified in Chapter 10. The chosen implementation uses the jWebSocket framework and Java version 1.6

12.1 The Client

After reviewing the different possibilities presented in Chapter 5 for developing a cross-platform friendly application, the decision was made to make the client in HTML5, CSS and JavaScript. To avoid the restrictions a pure webapp has when it comes to accessing device specific functionality, it was decided to make a hybrid app by wrapping the client in a native environment using PhoneGap. The app will be targeted towards Android devices. This is mainly for ease of development and testing, as neither of the developers has an Apple Mac computer nor access to any iOS devices. To enable communication with the jWebSocket server, an implementation of websocket that is compatible for use with PhoneGap will be used. To assist in making a clean and easy to use user interface we have selected to use the jQuery Mobile framework. For the implementation of a map, the Leaflet JavaScript library will be used and will display map data delivered by CloudMade.

12.1.1 PhoneGap

Being the most widely used open source framework for cross-platform HTML5 development, the choice to use PhoneGap was an easy one. The framework is well documented and discussed both on the official website as well as on several other development forums and websites.

Here is an example of the coupling of an Android application to the index.html page, through extending the PhoneGap class DroidGap.

```
public class KnowledgeWarClient extends DroidGap {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/index.html");
    }
}
```

The PhoneGap framework is further used for a range of tasks as; retrieving a Universal Unique Identifier (UUID) for the device, play audio files, vibrate and listening for device events like deviceready, pause, resume and hardware button presses. Here is an example of using PhoneGap's media object to playback audio files.

```
function playMusic(){
    if(!music)
        music = new Media("/android_asset/www/technogeek.mp3",
            onSuccess, onError);
    }
    music.play();
}
```

12.1.2 WebSocket

At the time of writing (and development) the PhoneGap framework does not yet support WebSockets. So a third party websocket implementation had to be found. As the decision was made to use the jWebSocket server, their jWebSocket JavaScript clients was first examined but turned out to not be compliant with PhoneGap. jWebSocket has also developed and released an Android client, but the client is developed in Java as a native Android application, which means a separate JavaScript interface would have to be developed before it could be used from the client software. After some further research, a working open source WebSocket implementation for PhoneGap with a WebSocket Factory and JavaScript interface was found⁷². As all communication with the jWebSocket server is token based, all the client has to do is to parse the incoming messages using JSON and pass the message on to a token handler.

To use the library the connection between the JavaScript interface and the Java WebSocket source code must be set in the PhoneGap project.

```
import com.strumsoft.websocket.phonegap.WebSocketFactory;

public class KnowledgeWarClient extends DroidGap {
    /** Called when the activity is first created. */
    @Override
```

⁷²<https://github.com/anismiles/websocket-android-phonegap> (28.11.12)

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    super.loadUrl("file:///android_asset/www/index.html");

    appView.addJavascriptInterface(new WebSocketFactory(appView),
        "WebSocketFactory");
}
}
```

Then the client can use the JavaScript library to connect to the jWebSocket server and override the websockets onopen, onmessage, onerror and onclose functions and handle the events accordingly.

```
var socket;
function connect (){
    socket = new WebSocket(
        'ws://129.241.222.111:8787/jWebSocket/jWebSocket');

    // write a log entry after the connection is established.
    socket.onopen = function() {
        console.log('DEBUG: Connected');
    };

    // process a message pushed from server.
    socket.onmessage = function(msg) {
        token = processMsg(msg);
        processToken(token);
    };

    // alerts that an error has occurred.
    socket.onerror = function(msg) {
        alert('Error: '+JSON.stringify(processMsg(msg)));
    };

    // write a log entry after the connection is closed.
    socket.onclose = function() {
        console.log('DEBUG: Disconnected');
    };
}
```

12.1.3 jQuery Mobile

In order to build a cross-platform friendly user interface in HTML5 that looks and feels more like a native application, the jQuery Mobile (JQM) framework was used.

Here is an example of a standard composition of a JQM page. The page is defined with a header with a title, an empty “div” that is to contain the map and a footer with some action buttons.

```
<div data-theme="b" data-role="page" id="mapScreen">
  <div data-role="header" data-position="fixed" data-tap-toggle="false">
    <h1>This is the map screen</h1>
  </div>
  <div data-role="content">
    <div id="map"></div>
  </div>
  <div data-role="footer" data-position="fixed" data-tap-toggle="false">
    <button id="mapAttackBtn">Attack</button>
    <button id="mapPickupBtn">Pick up</button>
    <button id="mapProfileBtn">Profile</button>
  </div>
</div>
```

12.1.4 Leaflet and CloudMade

In the search for a flexible and mobile friendly JavaScript Library for implementing the map, the developers discovered Leaflet⁷³. Leaflet is a lightweight open source JavaScript library made for use with map imagery from CloudMade⁷⁴ which is build upon the OpenStreetMap⁷⁵ map data. The library, though lightweight, offers an API that leaves nothing unwanted, delivers a good performance and user experience and together with CloudMade fits perfectly in the visual theme set by jQuery Mobile. Here is an example of initializing a new map showing Gløshaugen with a marker and placing it in the map “div” defined earlier with JQM.

```
// initialize the map on the "map" div
var theMap = new L.Map('map');

// create a CloudMade tile layer
var cloudmade = new L.TileLayer(
  'http://{s}.tile.cloudmade.com/
  62c1e9b59ce842dd9efa47741a637e7e/62728/256/{z}/{x}/{y}.png',
  {maxZoom: 18}
);

// create a latitude and longitude position holder.
var gloshaugen = new L.LatLng(63.417694, 10.403838);

// add the CloudMade layer to the map set the view to a given
// center and zoom
```

⁷³<http://leafletjs.com/> (26.11.12)

⁷⁴<http://cloudmade.com/> (26.11.12)

⁷⁵<http://www.openstreetmap.org/> (26.11.12)

```
theMap.setView(gloshaugen, 16).addLayer(cloudmade);

// create a marker in the given location and add it to the map
var marker = new L.Marker(gloshaugen);
theMap.addLayer(marker);

// attach a given HTML content to the marker and immediately open it
marker.bindPopup("<label>This is Gløshaugen</label>").openPopup();
```

The Leaflet library also provides good tools for drawing vector graphics onto the map. The example show how to draw some circles around the players current location which signifies the player's visual and action range in the game. Further GeoJSON⁷⁶ is used to draw a fog-of-war beyond the players visual circle, thus hiding the rest of the map.

```
//Set style options for the visual circle
var visualCircleOptions = {
  color: 'black',
  weight: 2,
  fill: false
};
//Set style options for the attack circle
var attackCircleOptions = {
  color: 'green',
  weight: 2,
  fillColor: '#0C3',
  fillOpacity: 0.25
};

//Create circle objects and specify location, radius and style options
visualCircle = new L.Circle(myPos, 50, visualCircleOptions);
attackCircle = new L.Circle(myPos, 25, attackCircleOptions);

//Draw the circles on the map
theMap.addLayer(visualCircle);
theMap.addLayer(attackCircle);

//Set style options and calculate the geometry of the visual circle
fogOfWar = {
  "type": "Feature",
  "properties": {
    "style": {
      weight: 1,
      color: "#000",
      opacity: 1,
      fillColor: "#000",
      fillOpacity: 1
```

⁷⁶<http://www.geojson.org/> (26.11.12)

```

    }
  },
  "geometry": {
    "type": "Polygon",
    "coordinates": calcFogOfWar(myPos)
  }
};

//Define a geoJSON layer to draw the fog-of-war on and add it to the map
geoJsonLayer = new L.geoJson(null, {
  style: function (feature) {
    return feature.properties && feature.properties.style;
  },
  onEachFeature: onEachFeature
}).addTo(theMap);

//Draw the fog of war on the geoJSON layer
geoJsonLayer.addData(fogOfWar);

```

With some added styling, and logic for handling the communication with the server and drawing of enemies and item chests within the visual range, the map screen ended up looking as presented in Figure 12.1.

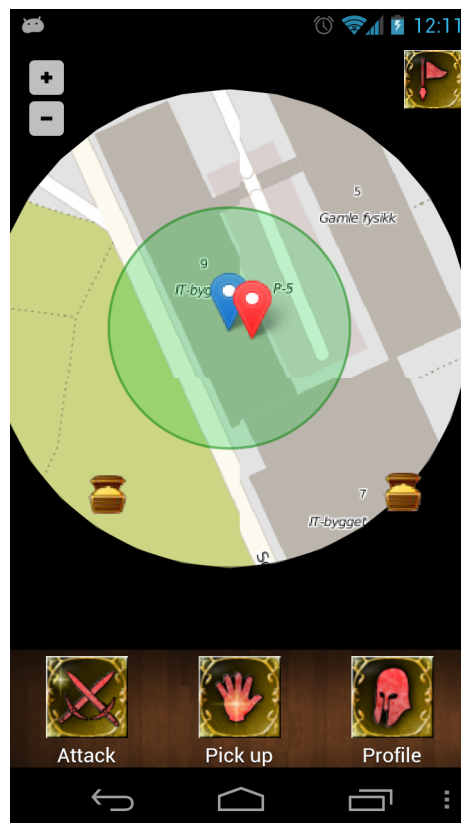


Figure 12.1: KnowledgeWar 2.0: Screenshot of the map

12.1.5 Client Dependencies

The client is composed of and dependent on several JavaScript and CSS libraries as can be seen in Figure 12.2.

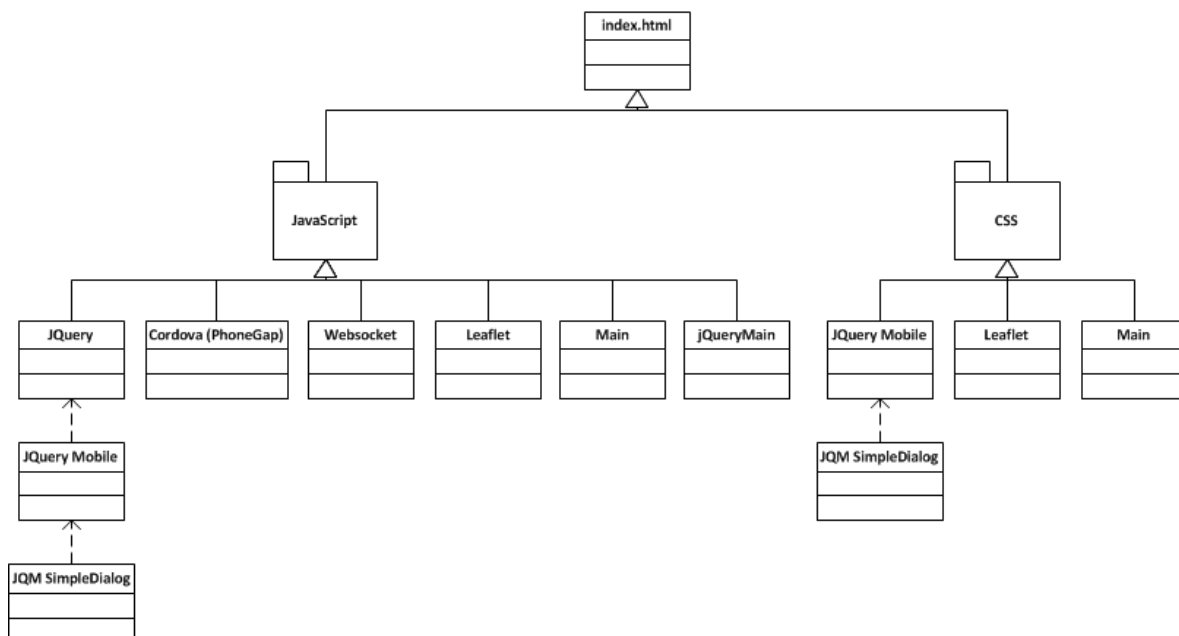


Figure 12.2: Client dependencies

The jQuery, jQuery Mobile and JQM SimpleDialog are dependent of each other and together forms the backbone of the user interface and is dependent of the jQuery Mobile and JQM SimpleDialog CSS files. Cordova is the PhoneGap library that bridges the gap between the device and the HTML and JavaScript. Websocket is the JavaScript interface for creating and communicating through a WebSocket. Leaflet is the map library which allows us to create and show the map data from CloudMade and is dependent on the Leaflet CSS to style its components. The Main and jQueryMain JavaScript files contains the main parts of logic for the client. The Main class handles the connection to device specific interaction through the PhoneGap framework. This consist of, among other things, access to the device's UUID, playback of audio files, vibration and device specific events. The Main class is also responsible for creating and opening a WebSocket connection to the server, parsing incoming messages into tokens and triggering the correct response. After connecting to the server it performs a handshake with the server and presents its UUID which is then used by the server to identify which device and user is connecting. The Main class is also responsible for handling the registration of a new user, login of an existing user, as well as holding a local copy of the player object it receives from the server. The jQueryMain class responsibility is handling all user input and activities and building all the different views and pages from the data received from the server.

12.1.6 The Finished Client

Here the finished client with screenshots and some state diagrams is presented to show how the functional requirements specified in Chapter 10 was solved. Figure 12.3 shows the clients different views and how they relate to each other.

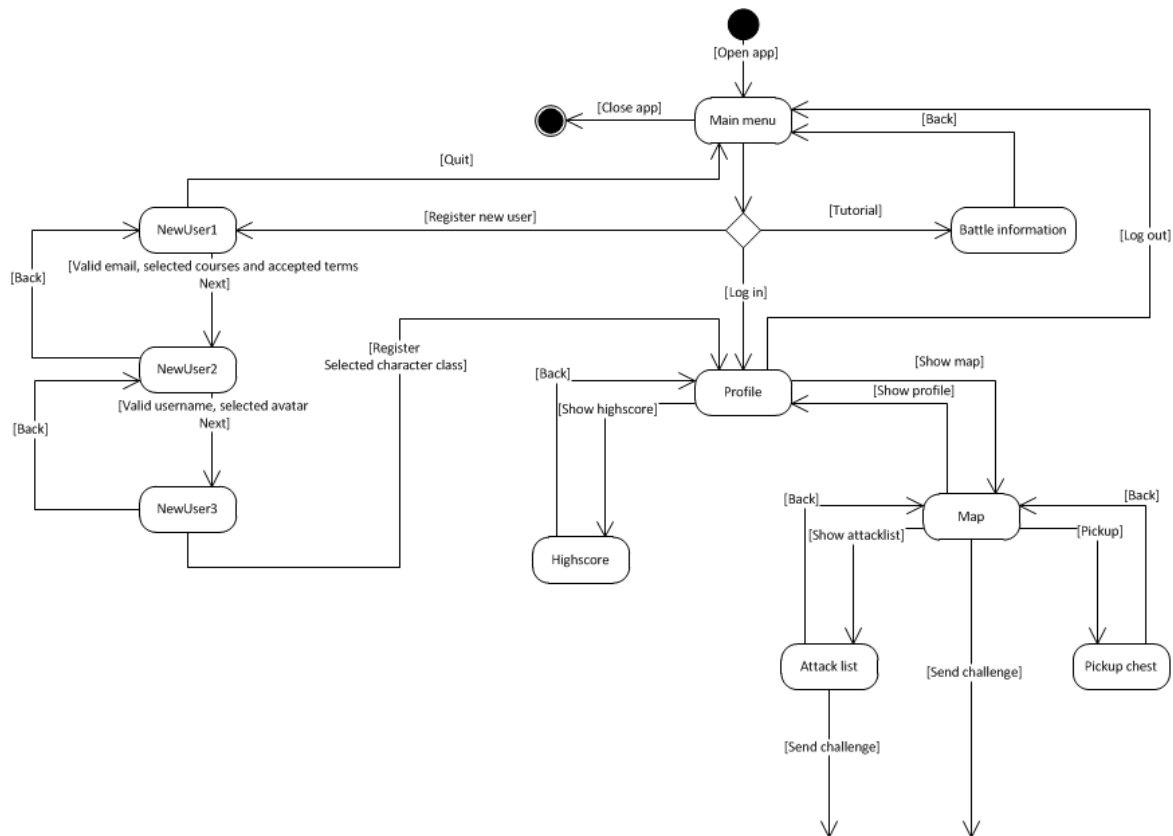


Figure 12.3: The Different view states of the client

The main menu From the main menu, see Figure 12.4, the user has the opportunity to log in (FR2), register a new character (FR1), read the tutorial and to turn background music on and off.

Registration To fulfil the functional requirement FR1, a three stage registration process was set up as seen in Figure 12.5. In step 1, the user has to input a valid email address, select one or more courses from the list and agree to the applications terms and conditions. In step 2, the user has to select a name for his/her character that is not already registered by someone else and select an avatar from one of Scott Johnson's 56 geeks which can be seen in Appendix C. In the last step of registration, the player has to assign a character class for his/her character from one of the four available classes. When completing the registration the information is sent to the server which sets up the correct database entries and returns a player object which the client can hold locally for displaying the profile view.

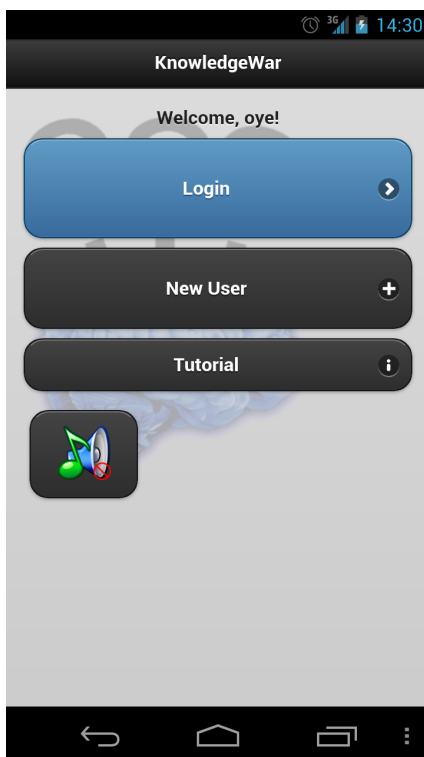


Figure 12.4: KnowledgeWar 2.0: Screenshot of the main menu

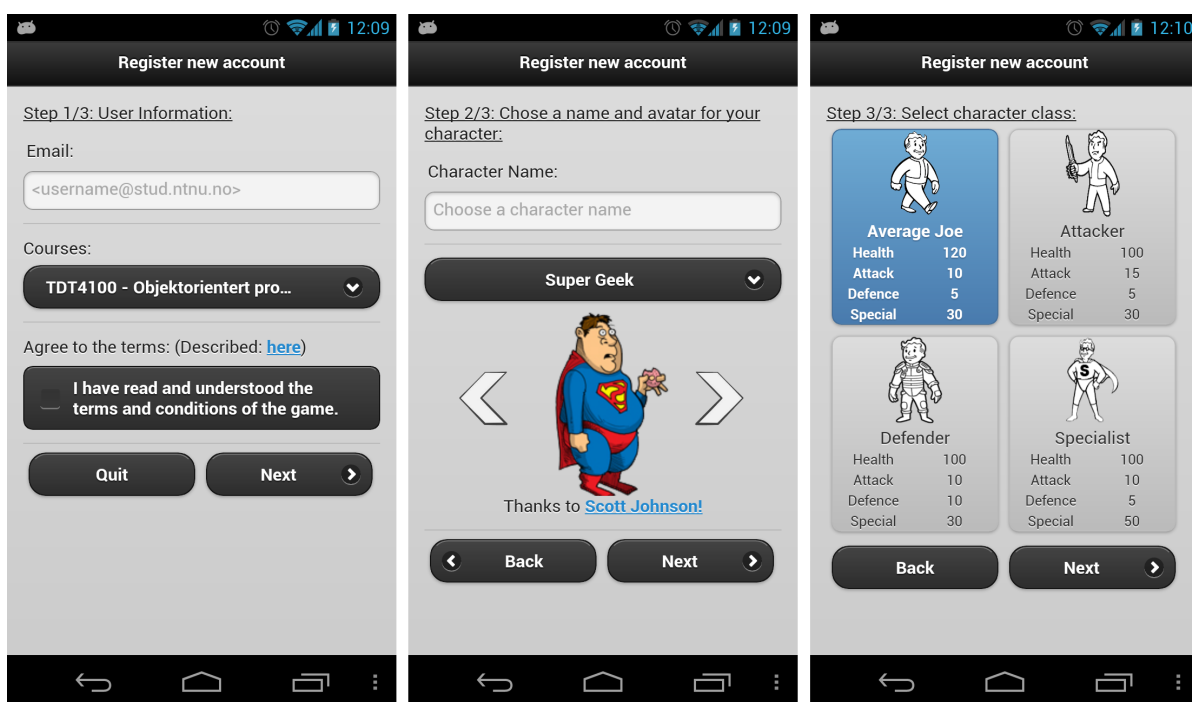


Figure 12.5: KnowledgeWar 2.0: Screenshots of the registration screens

Profile After logging in or completing the registration process, the user is taken directly to his/her profile as seen in Figure 12.6. From the character tab the player is shown all information about its characters level, class, experience, health, character

attributes and also some simple statistics on number of wins, loss and forfeits. The equipment tab shows the players current weapon and armor with their attributes. From the profile screen the user can either log out (FR2), go to the highscore table (FR11) or go to the map.

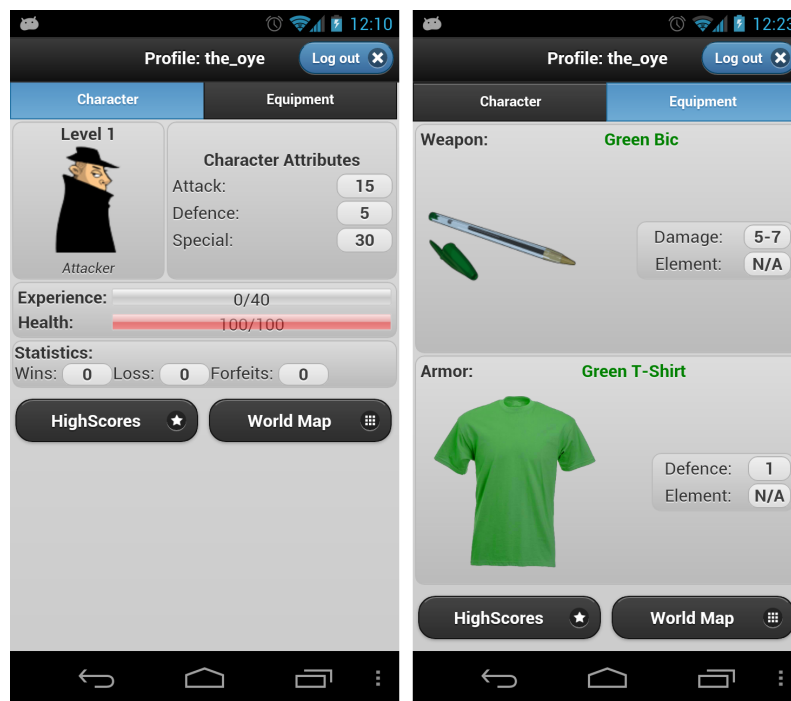


Figure 12.6: KnowledgeWar 2.0: Screenshots of the character profile

The map To implement FR3, FR4 and FR5, the map shown in Figure 12.7 was implemented. From the map, the user can see his/her own position marked with a blue marker. Enemies within the visual range will be displayed using an identical, but red marker and items that can be picked up is shown on the map as a small treasure chest. The player is displayed in the center of a green circle, which represents the player's action range, i.e. the range that chests and enemies have to be within for the player to interact with them. From the map screen the player can choose to display the “legend” panel which describes the different markers and their meaning. The player can also choose to go back to the profile page, try and pick up a chest or go to the attack list.

Treasure Chests Figure 12.8 shows screenshots of what happens when a player picks up a chest. The player can view the found item and will get a comparison of the item's attributes up against his or her currently equipped item of the same type. Higher attributes are marked with green font, lower attributes with red font and if they are the same the font is blue. The player can then choose to equip and replace its current weapon or armor with this new item or to ignore the new item and continue playing (FR7).

Challenge From the attacklist, the player can send out challenges. Opponents that are not within attack range or is of a too high or too low level is greyed out and can not

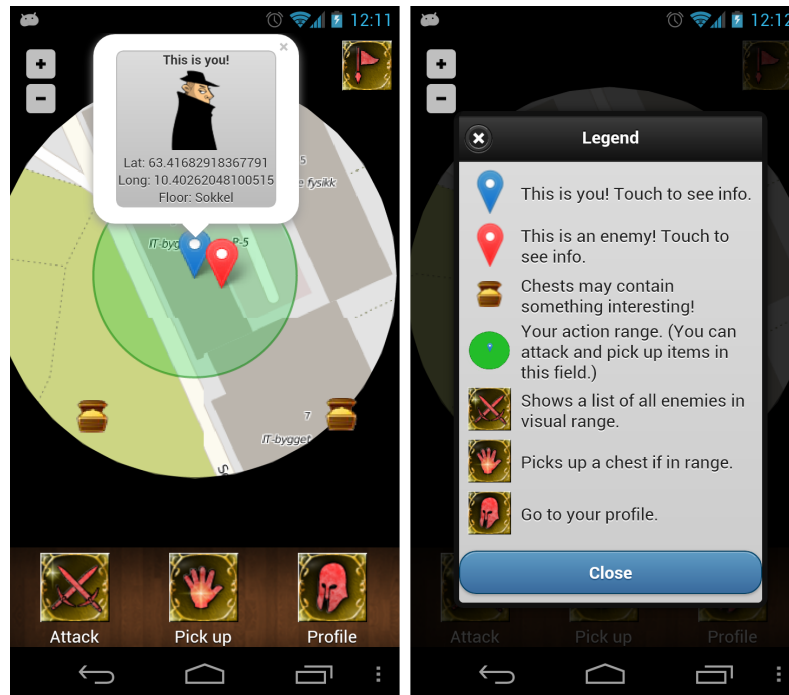


Figure 12.7: KnowledgeWar 2.0: Screenshots of the map

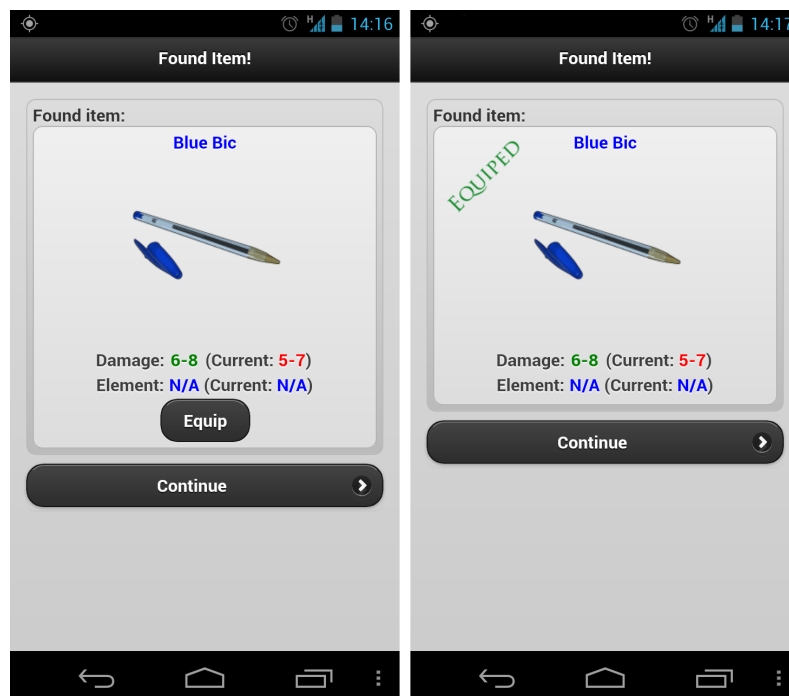


Figure 12.8: KnowledgeWar 2.0: Screenshots of picking up and equipping an item

be challenged. When someone receives a challenge, his or her smartphone will vibrate and make an “Aaaaarrghh!” sound effect. The display will show a popup dialog with the attackers name, level and avatar. The player can then choose whether he wants to initiate a battle with this player or not (FR8). When a player receives a challenge or a

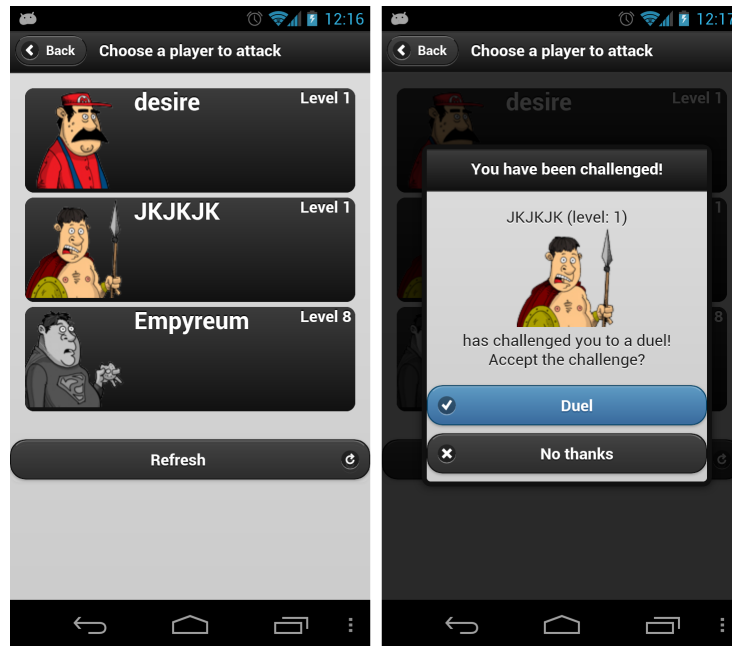


Figure 12.9: KnowledgeWar 2.0: Screenshots of the attacklist and an incoming challenge

response to a challenge he/she has sent out, the player can be in any of the “logged-in” view states. Figure 12.10 shows how the client takes the player through the different view states of a battle when a challenge is received.

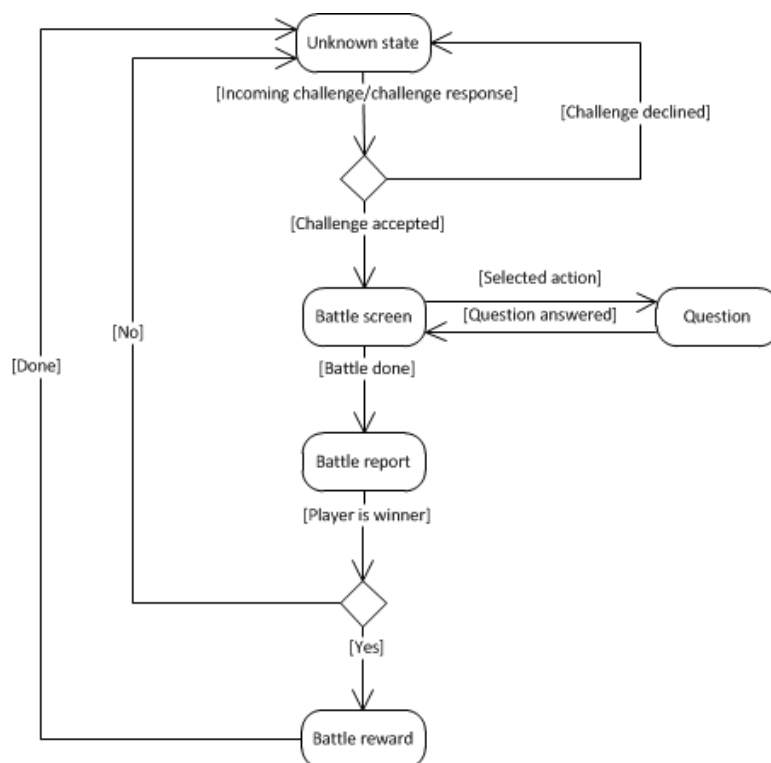


Figure 12.10: The Different view states of a battle

Battle After accepting a challenge, both players are taken to the battle screen as seen on the first screenshot in Figure 12.11. If the player has not battled before he or she can find some basic information on how to play from the question mark bubble. The game mechanics is as described in Section 11.1, and follows a series of choosing an attack action and answering quiz questions until one of the players has no more health or forfeits (FR9). The player can choose between four different types of attack, namely “Quick”, “Normal”, “Heavy” or “Special” (if the player has accumulated enough special resources).

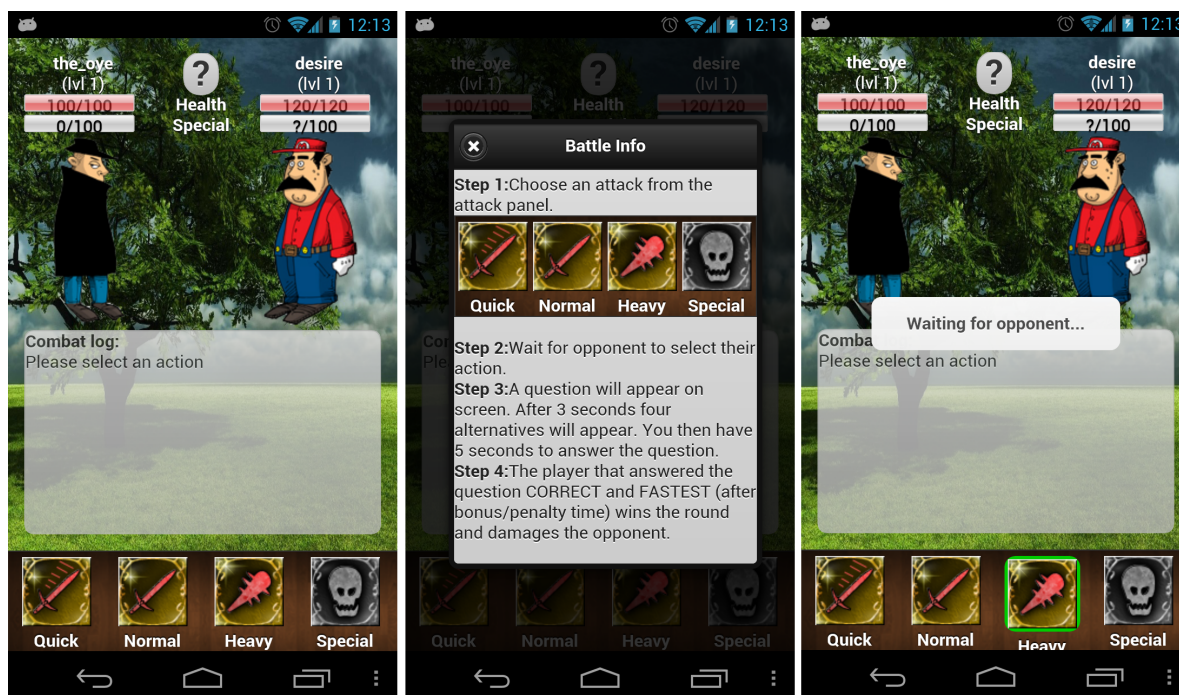


Figure 12.11: KnowledgeWar 2.0: Screenshots of the battle screen

Questions After both players has chosen an attack type, they are presented with a question as shown in the first picture of Figure 12.12. A counter will count down from 3 seconds before the alternatives are shown as in the second picture. After both players has answered the question the avatars of both players collide and a brief fight animation is shown.

Battle report and reward When a battle is over, both players are taken to the battle report screen, see Figure 12.13. From there, each player gets the opportunity to review all the questions from the battle, and see their answer time and whether they answered correctly or not (FR10). The player that won the battle will after the battle report screen be taken to the battle reward screen where the amount of experience gained and a possible item reward is displayed. The player can then, as with a normal pick up, choose whether he or she wants to equip and replace their current weapon/armor with this item.

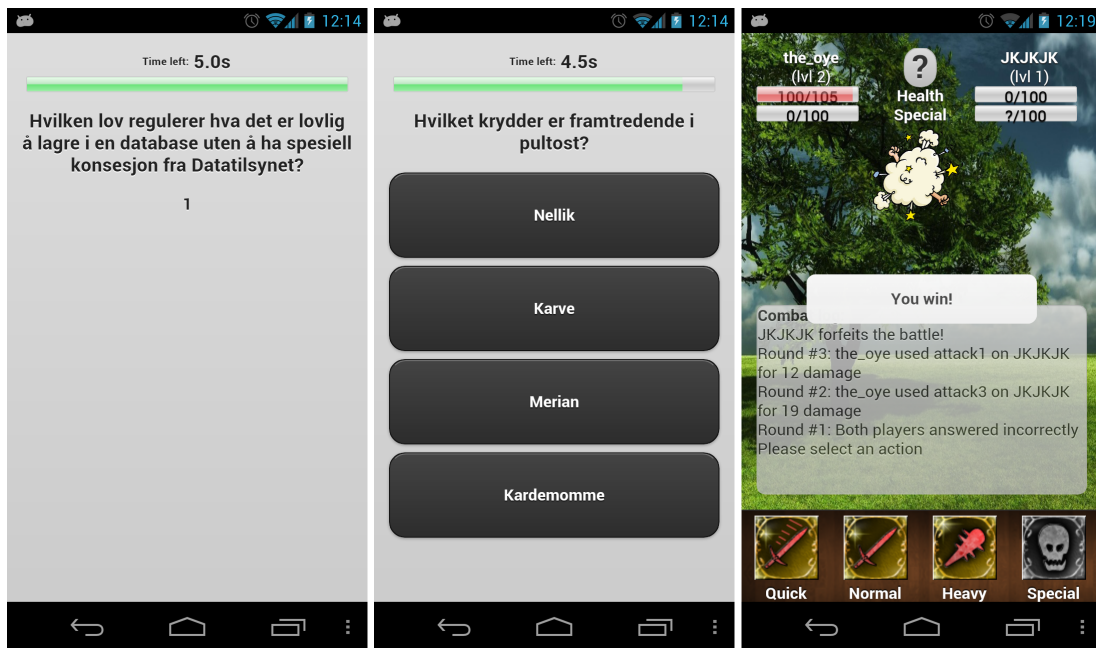


Figure 12.12: KnowledgeWar 2.0: Screenshots of a quiz question

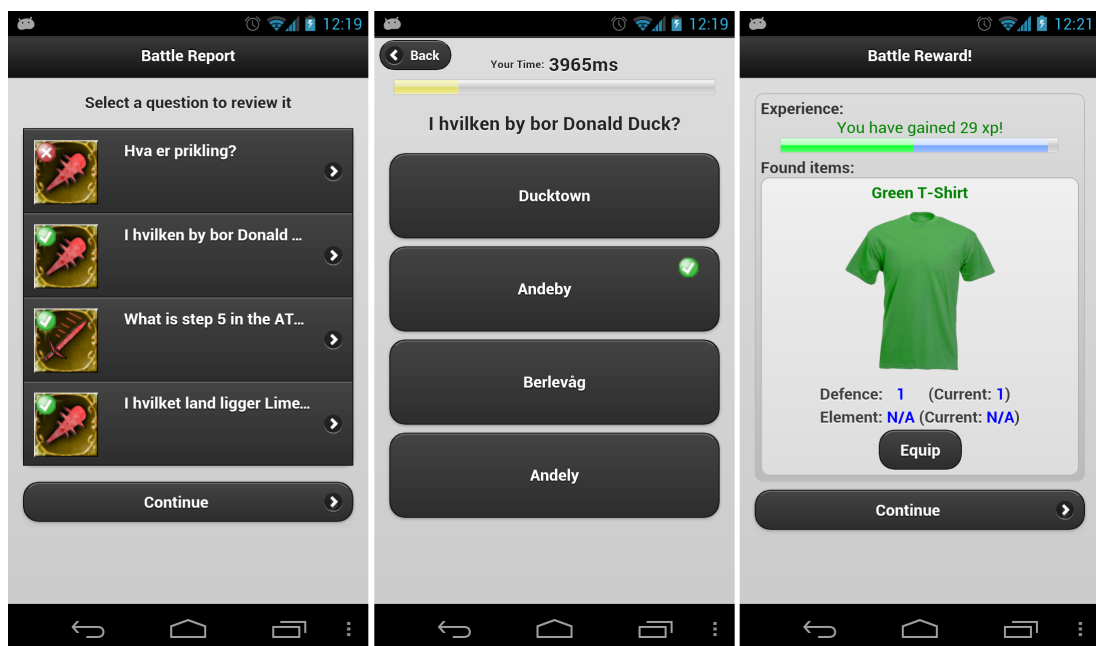


Figure 12.13: KnowledgeWar 2.0: Screenshots of the battle report, question review and battle reward

12.1.7 Admin Client

As Jegers & Wiberg explains, there are several difficulties in observing how players of a pervasive game choose to play the game [14]. This inspired the developers to make a modified “admin” client in addition to the normal client. This was done with the user experiments in mind, to allow the test personnel to monitor the participants movements and actions during the experiment. The client work much like the normal one, except

that it shows the entire map with all treasure chests and players with circles marking their visual and action ranges. The admin client also does not report its location to the server and is as a result invisible for the other players. Additionally a simple message system was built on top of the standard System Plugin of the jWebSocket server which allows for sending messages either to a single client, or broadcast a message to all clients connected to the server. Screenshots of the admin client can be seen in Figure 12.14.

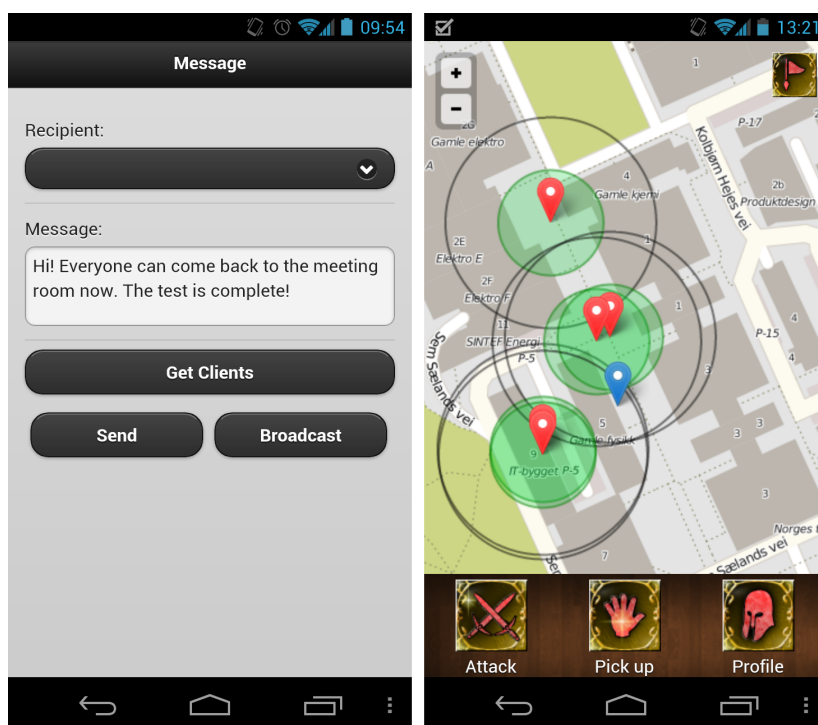


Figure 12.14: The Admin Client: Screenshot of the message system and map

12.1.8 Summary

With the finished client most of the functional requirements established in Chapter 10 has been successfully implemented. Only requirement FR6, “Interact” where special areas with shops or NPCs that the user could interact with was not implemented. This was due to time constraints and that a choice was made to rather improve other functionalities like roaming, location and persistence, in order to give a more stable gameplay experience. FR4, “Fog-of-War” was also at first discarded but was later discovered to be partially quick to implement. The hiding of all undiscovered areas with a black cover was easily done using a GeoJSON layer on the Leaflet map. The second part of the requirement, shading (fog) of earlier discovered areas was however a bit more tricky and was not implemented. Further the developers had plenty of ideas of additional content to implement, that would help bring the game to a much higher level in terms of playability. Some of these ideas will be presented later in Chapter 17.

12.2 The Server

A powerful server is needed to realise the project. It has to be able to offer logical calculations of everything from battles to handling of positions at the same time for a multitude of users. It also has to offer a reliable communications channel to send data between the client and server. The server must store information about clients that are logged on and must write the appropriate data to the database as soon as the user goes offline. To minimise the amount of database accesses the server stores information about disconnected users for a period, before writing it to the database as if the user logged off in a usual manner. jWebSocket Server proved to be just what was needed. It offers reliable communication between two pieces of software as well as an easy way to handle packets of information in JSON format. It is based on open source technology, and is written in Java and JavaScript. Java is a matured and well documented language which allows for easier development.

12.2.1 Server Logic

To keep record of all players while they are logged in and actively playing the game, according to TR2 (see Chapter 10), all players are stored in a list object where their status is kept for as long as they are online. This list of players is implemented as a singleton, to ensure that only one instance of the object exists at any one time. This makes it possible for all classes to access the list without having to instantiate a new object. The Class diagram for the server is shown in Figure 12.15.

jWebSocket

The goal of the server-client communication was the possibility to maintain a full-duplex communication channel over a single TCP-connection. Along with jWebSocket implementing WebSockets in Java, the ease of development when using the framework made it an easy choice to use for the server-side logic.

Some basic methods are included with a java class that extends the TokenPlugIn class from the jWebSocket framework:

```
public void connectorStarted(WebSocketConnector aConnector) { }

public void connectorStopped(WebSocketConnector aConnector,
    CloseReason aCloseReason) { }

public void connectorStopped(WebSocketConnector aConnector,
    CloseReason aCloseReason) { }

public void engineStarted(WebSocketEngine aEngine) { }

public void engineStopped(WebSocketEngine aEngine) { }

public synchronized void processToken(PlugInResponse aResponse,
    WebSocketConnector aConnector, Token reqToken) { }
```

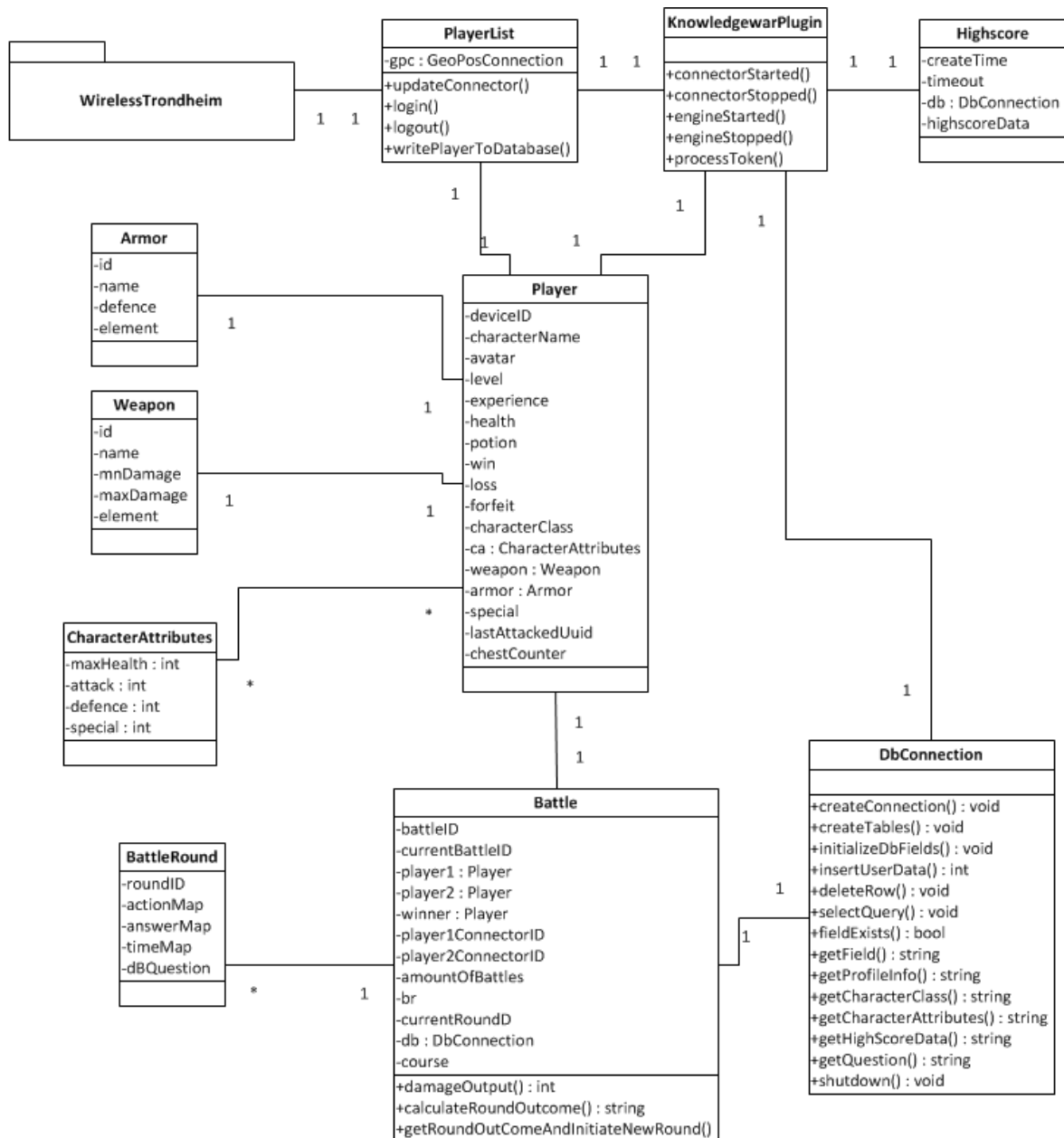


Figure 12.15: Server Class diagram

The “connectorStarted” method is called each time a client establishes a connection to the server. Likewise, the “connectorStopped” method is called when a client loses connection. It is in the “processToken” method that the incoming custom tokens are interpreted and dealt with. The tokens are passed between the server and the client at almost all times during the lifetime of the application. A token usually contains situational information about what is happening around a client, but it may also contain information about what is happening in a battle or positional updates for clients.

Client handshake and registration

The first token is sent from the client to the server when the user opens up the client app for the first time. The client sends the unique ID of the device to the server to check if it is already listed in the database. If it is, the device automatically logs in, if not the user will be required to register a new user. When a new user is registered, it is given standard values for weapon and armor as well as initiated to level 1 with no prior experience points. This is then written to the database from the server, and a standardised token with values needed for the client-side profile screen is sent to the client. During the register process several tokens are exchanged. The fields that allow for user input (e-mail and username) are validated both for security and to see if they are already in use by an existing user.

Highscore

A feature for viewing highscores for registered players is implemented. The highscore contains information on players' level, amount of experience, wins, losses and total number of forfeits. When a client asks for the highscore, the server checks to see if there exists cached highscore data and if it is expired. If the data does not exist or is expired the highscore list is obtained from the database, sorted and cached locally before it is sent to the client.

Challenges and Battles

To be able to challenge other players to a quiz battle, a client can challenge players either directly from the map or from the attack list. Selecting a player from the list will issue a challenge to that player. This token will pass right through the server and to the player that is to be challenged. The challenged player then chooses whether to accept or decline the challenge. A response is formed and sent back to the server, which then notifies the original challenger. If the challenge was accepted the server sets up a new battle between the two players and sends a battle initiation token to both clients. The battle game loop is then started where the server waits for both players to have selected an attack, sends a question token to both clients, waits for an answer from both clients, calculates the result and sends a round outcome token to the clients. The loop then continues until a player has no more health or chooses to forfeit the match.

12.2.2 Statistics

To be able to keep track of which questions that are asked while players are in a battle, a method was made to keep updated statistics about questions asked, which answers were given, and the average answer time for the question.

The method used was to update a XML-file with the information every time a question was set while players were in a battle. Whenever an admin wants to see the statistics, an independent parser is used to parse through the XML and draw pie-charts as can be seen in Figure 12.16.

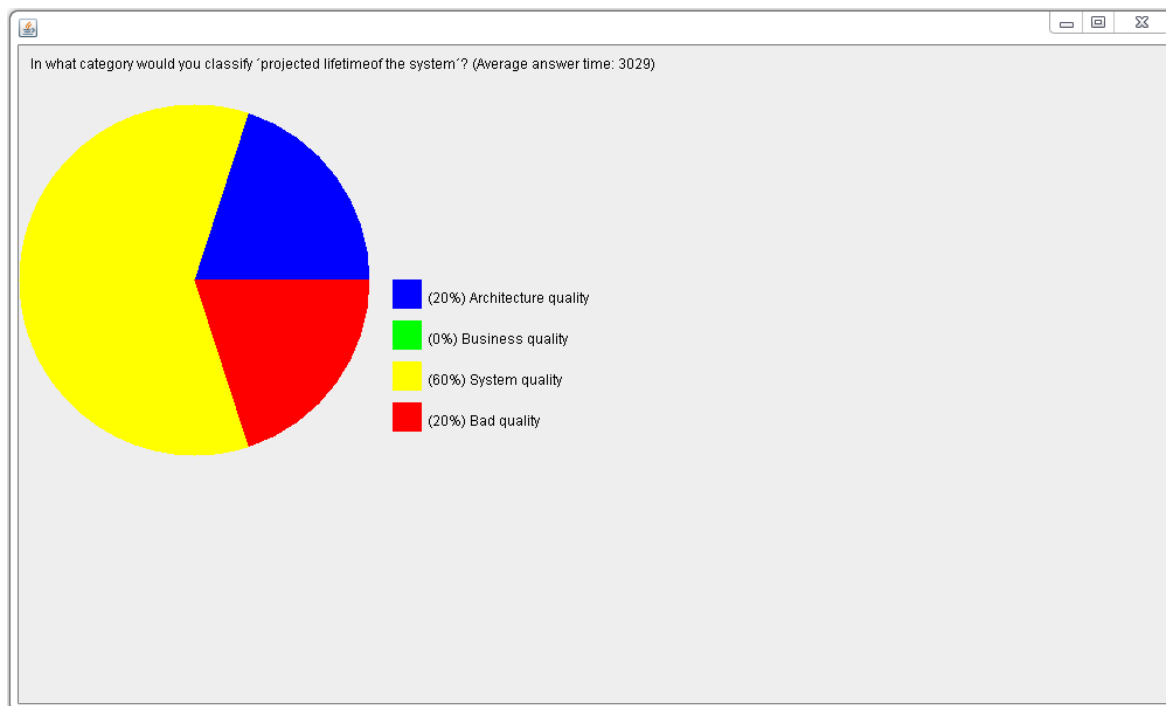


Figure 12.16: Statistics pie chart

12.2.3 Client Location Update

Wireless Trondheim⁷⁷ offers a solution for gathering locational information using only the IPv4-address of a user. The user can through their system be localised to what campus, building and floor they are on. It also provides information on the latitude and longitude of the user as well as a confidence factor for this position. The system is however limited as it only support a limited range of IP-addresses provided by the wireless networks of either eduroam⁷⁸, NTNU or Wireless Trondheim. The developers were also given restriction on how often they were allowed to check an IP-address for its position. A maximum of five positions per second. This led to the development of an algorithm that updated the position of each user based on how many users that are online at a given time, down to a maximum of 1 update each 200ms. For each user that uses WiFi for positioning, this means that an updated position is kept on the server at all times while the user is active and logged in. The position for all the users are distributed once every five seconds. This means that players should receive an updated position for other players, and itself at least once every five seconds. Another possible drawback of using the Wireless Trondheim solution is that it takes some time for the smartphone to change from one Wireless Access Point to another when roaming. This results in the user being “stuck” at one location while actually moving and the location update can be perceived as both slow and inaccurate. The system therefore is best suited as a secondary fall-back solution for indoor play, and works great in a “play-where-you-are” setting like in a lecture hall during the break.

It is also possible to use the smartphone’s telecommunication network (3G/EDGE)

⁷⁷<http://tradlosetrondheim.no/> (16.11.2012)

⁷⁸<https://www.eduroam.no/> (16.11.2012)

for communication with the server, if it has a SIM card installed. When using 3G the IP-address of the client will not be compatible with the system of Wireless Trondheim, and the client will have to fetch its own position and update the server. The server will then update its position register which is used when sending positional updates to all the clients. If the position is based solely on the telecommunication network neither the accuracy or refresh rate of the position is suitable for a pervasive game like KnowledgeWar 2.0, as gameplay revolves around the player interacting with its surroundings. Most smartphones are however equipped with GPS which gives both highly accurate positions and immediate updates as the player moves. GPS is however dependent on the device being outdoors in order to get sufficient satellite coverage for the service to work.

A combination of the two solutions is used in the implementation of the software prototype of KnowledgeWar 2.0, allowing both the use of WiFi, 3G/EDGE and GPS. It is however encouraged to use 3G/EDGE and GPS for outdoors play and WiFi for indoors play, to give the most stable and persistent gameplay experience.

12.2.4 Database

A database is required to store information, so that it is possible to maintain persistence for recurring users and make statistics about how users are playing the game (TR3, Chapter 10). As the server software is made using Java, Apache Derby⁷⁹ is chosen as the Relational Database Management System (RDBMS), as it gives easy access and integration with the server software. Derby is small, and includes the base engine and the embedded JDBC (Java Database Connectivity) driver. It also employs an easy to use and humanly readable Structured Query Language (SQL).

To prevent the database from being flooded with updates, all player objects are stored on the server for a small amount of time after the user has logged out or disconnected, before they are written to the database. The ER-model for the implemented database solution is show in Figure 12.17.

⁷⁹<http://db.apache.org/derby/>

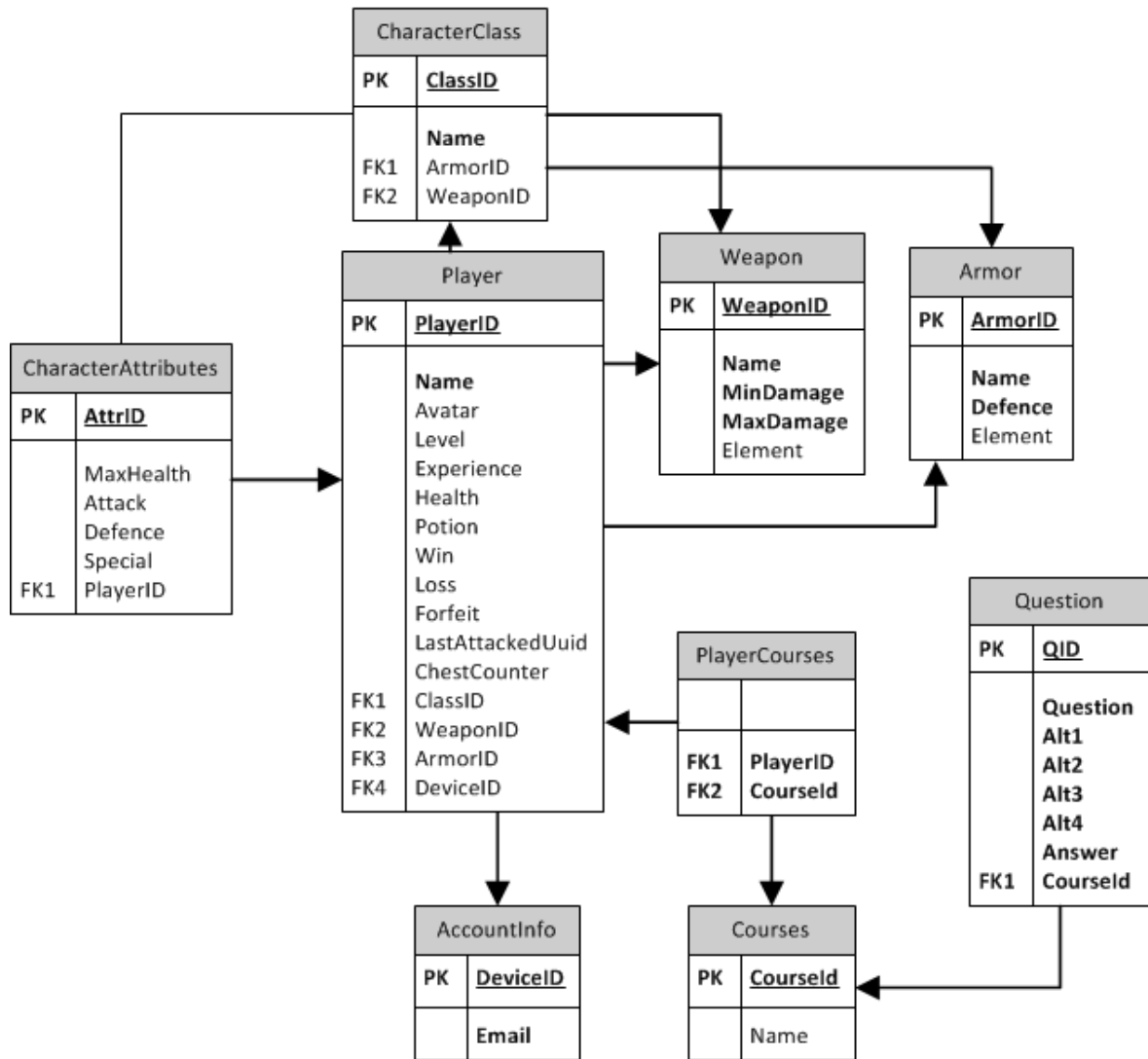


Figure 12.17: ER model

Chapter 13

User Testing

From testing the software prototype, the WiFi-based indoor positioning system was found to deliver an unstable performance. This led to the decision to perform an outdoor test using GPS to give the most accurate and frequent position updates, and using the mobile telecommunication networks (3G/EDGE) for communication and data transfer between the clients and the server. From testing this was experienced as the most stable way of playing the game and subsequently gave a better gameplay experience.

13.1 Acquiring Test Subjects

An online sign up sheet was set up using Doodle⁸⁰, where interested parties could show their interest and select their suitable timeslots for attending a user experiment. To reach potential test subjects, emails were sent to current and earlier students that had taken the course in Software Architecture and encouraged them to join the experiment. It was declared that there would be a small prize for the player with the highest score at the end of the experiment in order to further increase interest and motivation for them to join. The prize was provided by our supervisor and was a 16GB USB flash drive with a value of approximately NOK200.

13.2 Execution

Test personnel during the experiment consisted of Jon Kjetil Øye and Stian Forberg, who held the roles of experiment leads, developers, thesis writers and observers. The thesis supervisor Alf Inge Wang was also present and served as an external observer, advisor, project owner and course lecturer. Two user tests of the software prototype was performed and held one week apart. The execution of both experiments was split into four major parts: a presentation, testing, questionnaires and a discussion and feedback session. Between the first and second user test some quick and simple tweaking of the prototype was performed, based on the feedback from the participants of the first test. In this section the four phases of the experiment is presented and explained. Further

⁸⁰<http://www.doodle.com/> (16.11.12)

a deeper run down of the execution and feedback session of the first test, the tweaks made to the prototype and the execution and feedback of the second test is presented.

13.2.1 Presentation

The experiments was initiated with a short presentation where the concept of the game was presented together with some screenshots of the game in order to take the participants through the basics. To further motivate the participants a scoring system was presented where they would get points for each chest they pick up, each different player they battled and three points for every level they advance to. The player who had the most points at the end of the test session would be declared the winner and receive the prize.

13.2.2 Test Phase

Before being sent out each participant was handed some simple instructions (see Appendix E). These instructions tell the participant to first register a character, turn on GPS and turn off WiFi on their smartphone, close the application and walk to a specified unique location before taking up the application and start playing. This way all participant was evenly spread out on campus. They then had to use the game's map to navigate and locate other players to challenge. For monitoring purposes developers had made a modified client. This "admin" client, presented in Section 12.1.7, allowed the test personnel to see the entire map with all treasure chests and player locations with circles marking their visual and action ranges.

13.2.3 Questionnaire and Feedback

After the test phase everyone went back to the meeting room where the initial presentation had been held. The subjects then had to fill out two questionnaires. The first was a custom made questionnaire which can be found in Appendix F and the second was a standard SUS questionnaire and can be found in Appendix G. After everyone had filled out the questionnaires, the final scores were announced and the winner was presented with the prize. The test personnel then sat down with the participants and initiated a group discussion about how they experienced the game while playing, what functionality worked well and not so well and ideas for additional functionality which they thought might add to either the educational effect or gameplay experience.

13.3 Test of Software Prototype 1 (SP1)

For the first software prototype test, twelve individuals had signed up using the Doodle form, but only six showed up to the presentation. After the presentation the participants started downloading and installing the client to their smartphones. One of the participants did not have his own Android smartphone and paired up with one of the other participants. The two would perform the test phase together and switch who was controlling the smartphone. A second participant had problems registering an account in the game once he had installed the client. He did not have the possibility to use his

phone's data traffic service and was using WiFi, but the client seemed not to be able to connect to the server. While the test personnel were trying to assist and figure out the problem, the other participants started to play the game and battle each other. This was not the intention but as the experiment leads had forgotten to hand out the instruction set, the participants were told to finish any battle they were in and then move out to their starting positions. As the participants moved out, the admin client was brought up to give an overview of campus and ability to observe when the players logged on and started to play. After having tried three different devices it turned out that the participant with registration issues was using the guest WiFi network which has limitations on traffic and open ports for communication. As this was discovered the test phase had been going on for about 30 minutes, and one of the other participants came back as his battery had died. He connected a charger and the two stayed inside and battled each other over WiFi, until the end of the test phase and the other participants came back.

13.3.1 The Feedback

After the test phase was done and everyone had finished filling out the questionnaires the test leads sparked a discussion on how the participants had experienced the game.

The first response was that their fingers were frozen and that their battery had been drained during the test. This was to be expected as both the 3G and GPS services has a high power consumption. Together with the fact that players were walking around with the device screen on during the length of the test also severely drained the battery.

The participants further expressed frustration over the quiz alternative answer buttons. Most of the players complained that the buttons were unresponsive when pressed and that when the buttons responded the countdown still continued a bit after they had selected an answer. This made them feel that they did not get the correct answer time and was experienced as quite frustrating.

Participants also found battles to be long and tedious. This was also evident from the server logs which show that battles lasted from 15 to 23 rounds. This is about twice as long as was first intended. As the game also had a rather small database of questions, players also complained that they got the same questions multiple times, not only in one battle but sometimes several times in a row. As battles were so long and tedious, none of the participants managed to reach a level where they would find any new and interesting weapons or armor.

Some of the participants found the map to be a bit confusing, and wanted either the map to turn as they walked around campus or a compass to help them navigate. Some also wanted the map to automatically centre itself on their location as they moved. Others wanted a bigger visual range or an arrow indicating a direction where they might find an enemy or a treasure chest. As there were few participants walking around and only three treasure chest available for pick up at all time spread out on the campus, players experienced a lack of things to do in the game.

As the discussion turned towards the concept and gameplay, the participants showed great interest and enthusiasm for the potential of the game. Several of the participants presented ideas of gameplay elements as shops and a form of in game currency that could be used to buy upgrades for their character. The shops would also serve as a

physical place where one might find other players to battle. Others wanted to be able to fight bigger bosses or maybe team up and battle two vs. two players. One participant proposed an interception feature, where if two players had the same chest within their action range and one of the players attempt to pick it up, the other player would be warned and given the opportunity to initiate a battle with the player, in order to stop him and take the chest for himself.

One of the ideas presented stood out from the rest as it had nothing to do with the gameplay itself but rather addressed the educational aspect of the game. A participant suggested to have a reference to more information about a specific question or a topic from the either the battle report or question review screen. This reference could simply be a Wikipedia⁸¹ article, a PDF, a web page made by course staff or just a reference to pages in the course curriculum where one might read up on the topic.

13.4 Tweaking the Prototype

Based on the feedback of the first software prototype test, some simple steps and tweaks were identified in order to mitigate some of the issues the test subjects experienced. These tweaks would hopefully to be able to minimize issues that broke the game flow and give the participants a better gameplay experience. An increased gameplay experience could then help to further increase the educational effect and motivation of the game. The identified issues and the steps that were taken to mitigate them will here be briefly presented and explained.

User interface response time

Several of the participants complained about lack of or severely reduced response when they tapped on one of the alternatives of a question. This was an issue the developers also had experienced but after having played and tested the game over a longer period the developers had become accustomed to the way the user interface reacted. Researching the matter using the jQueryMobile forum, and other non-official forums and development blogs, the developers discovered that there was an approximately 300ms delay when using the “click” event for registering interaction with buttons or elements on mobile browsers⁸². The issue was then solved by switching to jQueryMobile’s “tap” event, which is triggered much earlier and greatly improved the response throughout the user interface.

Answer time

Participants of the first software prototype test expressed some frustration as they did not have enough time to read some of the longer questions before the alternatives arrived. Furthermore when the alternatives themselves were more than a single word, they did not have time to read them all while trying to remember the theory. This lead them to just guessing wildly before the time ran out. This is not very pedagogical, as one of the main goals of the game is to help the students to memorize existing

⁸¹<http://en.wikipedia.org/wiki/Wikipedia> (16.11.12)

⁸²https://developers.google.com/mobile/articles/fast_buttons (05.11.12)

knowledge by giving them some mental “pegs” for this knowledge. To resolve this issue the question-only read time was increased from 3 to 5 seconds, giving the students better time to read the question and start a thought process. Furthermore the answer time after the alternatives arrive was increased from 5 to 10 seconds to allow the students time enough to read all the alternatives, think and eliminate alternatives before answering. This increased answer time further lead to increased bonus/penalty time of the “Quick” and “Heavy” attack types from 0.5 to 1 second.

Character progression

From the feedback session it became clear that participants thought battles were too long, taking up to 15-23 questions before someone was victorious. Together with a limited set of questions in the database this meant questions was repeated often and battles became long and tedious. As the test phase was limited to approximately one hour, long battles also meant that players did not have time to fully experience the game. The players did not have enough time to get to a high enough level to experience anything but the lower tier of weapons and armor available. This was partially a result of the initial intention to have a longer running test phase and the gameplay had been designed thereafter. Some simple steps was taken to solve this. First, the attack attribute was increased for all character classes. This attribute is used as a base to calculate the amount of damage that is done on the opponent when a player win a round, which in turn will help decrease the number of rounds/questions in a battle. Second, the number of wins/experience needed to level up was reduced to exactly two wins per level. This greatly increase the rate at which players advance in the game. Third, the coupling of weapons and armor tiers was decreased to follow single character levels in stead of doubles as described in Section 11.3, which together with quicker character progression will enable the participants to have a greater gameplay experience.

13.5 Test of Software Prototype 2 (SP2)

In order to test whether the tweaks made since the first software prototype test had any real effect, a second user experiment was set up. For this test, participants could sign up for three different timeslots. This was done because of feedback from people who wanted to participate in the first test but did not have the possibility to join on the suggested timeslot. In total six people signed up but at most four of them were available on the same timeslot. In order to get an extra active player during the test phase one of the test leads would participate and thus also get some hands on experience of how the game works in a real-life setting. After the short presentation, the instruction sets were handed out and the participants started to download and install the client. Two of the participants did not have the possibility to use mobile data traffic during the test, and would have to use WiFi for communication with the server. As the WiFi-only players moved about the campus they had to stay rather close to buildings in order to get a sufficiently stable and strong WiFi signal. Roaming about campus also caused their client to frequently lose its connection to the server as they moved from one WiFi access point to the next. Sometimes losing connection while in a battle would

cause the game to freeze for both players, who then had to restart the client software which caused the players to lose some of their character progress. As some participants were using 3G for communication and others WiFi, there was some significant latency differences between players. This caused some difference in when the players received the quiz question and outcome of a battle round. This caused participants to believe the latency had an effect on their answer time as they would hear the sound effects from their opponent's smartphone before they got it themselves. WiFi players were also limited in their ability to move about and discover treasure chests, and caused players to cluster around the same area and alternate on challenging each other, which can be seen in Figure 13.1. With the extended answer and bonus/penalty time it soon became obvious to the players that using the quick attack was the safest and most effective attack strategy, and using special when available.



Figure 13.1: SP2 test in action

13.5.1 The Feedback

From his observations as a player during the test phase Jon Kjetil experienced the participants as having fun playing the game and battling each other. However, the main area of discussion both during and after the test phase was the unstable performance of playing the game over a WiFi connection.

WiFi players complained that they experienced a lot of delay as they had to wait for the client to re-establish the connection to the server quite often, and that they had to move over large distances before they got a new and updated position on the map. One participant complained that he had lost a newly picked up weapon. The client had frozen during battle because it had lost the WiFi signal, which forced both players to have to close and re-open the client.

A participant remarked that the client software did not support landscape mode. The participant had a smartphone with a slide-out keyboard which made it difficult for him during the registration process as he had to turn the phone to be able to read his own text input.

One participant commented on the randomness of questions as he had gotten the same question three times in a row. Others had issues with the map and wanted a compass to show which way they was heading. One participant explained that he had to zoom out and in again in order for the map to display properly, which the developers have discovered is a bug specific for Android version 2.3.

Some ideas to more gameplay elements also arise, as some wanted three-way battles, more items or single question creeps/monsters they could encounter which would attack the player.

Even though the test phase had been agreed upon to last 45 minutes, the participants continued playing for an additional 15 minutes before walking back to the meeting room. This might be an indication that the game has become more engaging and fun to play.

13.6 Summary

From the battle logs from the second software prototype test the number of questions/rounds per battle ended up ranging from 8-15. Which is a more suitable length and which is supported by the fact that none of the test subjects complained that they found the battles boring, with the exception of getting the same question multiple times in a row. Even though there were more technical difficulties during the second prototype test (as a result of half the participants having used WiFi), there seemed to be a higher level of enthusiasm and enjoyment of playing the game than when testing the first software prototype.

As there were only six participants on the first software prototype test and only four participants for the second software prototype test, the limited basis of data might have had an affect on the ending result. With such few participants the individual answers have a greater affect on the outcome of the questionnaire statements. Because of this, the scores calculated from the questionnaires are not generalisable, but will indicate the tendency of the outcome.

Chapter 14

Experiment Results

This chapter displays and discuss the results of the System Usability Scale (see Appendix G) and the KnowledgeWar 2.0 Questionnaire (see Appendix F). When interpreting the results, it is important to keep in mind the background of the participants. For example, the experience the participant has with similar types of software, how much experience they have with a smart device prior to testing, if they are familiar with video games in general, and especially the Role-Playing Game genre chosen for this game. The thoroughness of the introduction prior to the testing session, as well as age and field of interests, may also affect how participants answer the questionnaires.

14.1 System Usability Scale (SUS) Results

Following the testing of the Paper Prototype (PP) outlined in Section 11.2 and the testing of the Software Prototypes (SP1 and SP2) outlined in Section 13.3 and 13.5, all participants filled in a form determining the System Usability Score for the game. Table 14.1 shows the results of the seven SUS forms for the Paper Prototype, and a total of 10 forms for the tests of the software prototypes.

The average SUS score is about 68, but to get in the top 10% of SUS scores, it has to be above 80.3. If the application receives a score this high, it is also more likely that a user will recommend it to those close to him or her. Jeff Sauro of [measuringusability.com](http://www.measuringusability.com)⁸³ provides statistics and usability consulting, and has worked out statistics on the interpretation of SUS results. The scores in Table 14.1 have been determined according to standard SUS scoring methods, explained in Section 3.3.1 (where a score of 0 is the lowest possible and 10 is the highest possible).

14.1.1 Comparison of the Paper Prototype and the Software Prototypes

From the Paper Prototype (PP) to Software Prototype 1 (SP1)

When the Software Prototype was finished, the SUS scores from the Paper Prototype saw an increase for all statements except S01 and S06. The increase in scores are attributed to the feedback and changes made from the Paper Prototype to the

⁸³<http://www.measuringusability.com/sus.php> (08.11.2012)

Table 14.1: SUS scores for the Software Prototype

SID	Statement	PP	SP1	SP2	SP Avg
S01	I think that I would like to use this system frequently	5.71	4.58	7.50	5.75
S02	I found the system unnecessarily complex	7.14	8.75	9.38	9.00
S03	I thought the system was easy to use	6.79	7.92	8.13	8.00
S04	I think that I would need the support of a technical person to be able to use this system application	8.93	9.17	10.0	9.50
S05	I found the various functions in this system were well integrated	6.79	7.50	8.13	7.75
S06	I thought there was too much inconsistency in this system	8.57	7.08	8.13	7.50
S07	I would imagine that most people would learn to use this system very quickly	8.57	8.75	8.13	8.50
S08	I found the system very cumbersome to use	7.50	7.92	8.13	8.00
S09	I felt very confident using the system	6.79	7.08	7.50	7.25
S10	I needed to learn a lot of things before I could get going with this system	8.57	9.58	8.75	9.25
–	Total Average Score	75.36	78.33	83.75	80.50

implementation of the prototype software, which are outlined in Section 11.3. The increase on S02 (7.14 to 8.75) and S03 (6.79 to 7.92) correlates to the simplification of the registration process. Players were provided with a list of courses to choose from instead of having to input their course code manually along with the generalization of the character classes and removal of extra attribute points made it easier to register and get into the game.

As the map was now interactive players did not get confused when attempting to tap on it, which in turn also helped increase the scores for system usability. The changes in battle mechanics also made for a less confusing experience for the players and can be linked to the fact that S07 (8.57 to 8.75), S08 (7.5 to 7.92), S09 (6.79 to 7.08) and S10 (8.57 to 9.58) all saw an increase in scores. The decrease in scores for S01 (5.71 to 4.58) and S06 (8.57 to 7.08) can be explained by that the testers of the Paper Prototype was more forgiving when trying out something they knew was still in an early development phase. When faced with the finished Software Prototype, the testers might not have the same mentality as when the Paper Prototype was tested and may have expected a more polished product.

From Software Prototype 1 (SP1) to Software Prototype 2 (SP2)

With the exception of the decreasing scores on statements S07 and S10, the changes made to form Software Prototype 2, which are outlined in Section 13.4, indicates a dramatic increase in scores. The major increase on S01 (4.58 to 7.5) and the increases on S02 (8.75 to 9.38), S03 (7.92 to 8.13), S04 (9.17 to 10.0), S05 (7.5 to 8.13) and S08

(7.92 to 8.13) are attributed to the tweaks made on the GUI response and the question answering time given to the players. These small tweaks made the game overall more polished and responsive and in turn made it less cumbersome to use. The increased answering time and the easier character progression made the game more fun and in turn made the test participants more confident when using the system.

An external observer that participated in testing both software prototypes, noted that there was a difference in how the users were introduced to the prototypes. Software Prototype 1 had a longer and more thorough walk-through and description of the application and gameplay prior to letting the test participants try it out for themselves. This could in turn justify the decrease in scores on S07 (8.75 to 8.13) and S10 (9.58 to 8.75). However, the dramatic positive increase in scores shows that for each session of tweaking the software prototype the scores would increase. The first tweaking session would see the most dramatic increase (as is shown) while subsequent tweaking sessions would see an increase, albeit not as dramatic as the first one.

From the Paper Prototype to the total average score for the Software Prototype (PP to SP Avg)

Looking at the big picture, the scores saw an increase for all statements except S06 and S07 from the test of the Paper Prototype to the final Software Prototype test. The decrease in S07 might be due to that the test participants were not given an intro to how the game would be played during the Paper Prototype. While when the Software Prototype was tested the test participants were given an extensive intro to the game. When the test participants were told how the game works without being able to explore it on their own first they might have viewed the intro as useful, but not something they would have learned on their own. As previously mentioned, testing a software prototype might leave testers anticipating a completely finished product and thus being more aware of flaws in the system than if they were testing something on paper. All in all, the total average SUS score has increased from the Paper Prototype to the collected Software Prototype tests with over 5 points, from 75.36 to 80.50.

14.2 KnowledgeWar 2.0 Questionnaire Results

The KnowledgeWar 2.0 questionnaire, which can be found in Appendix F, was split into six main topics: Participants and Devices, Learning, Gameplay, Technical, Social & Mobility, and Experiment. This section will present and compare the results gathered from the questionnaires of the first and second software prototype test. For each category of the questionnaire, a selection of statements will be discussed before the category as a whole is summed up.

14.2.1 Scoring

The sections Learning, Gameplay, Technical and Social & Mobility consists of short statements, like “The game was fun”. The subject then has to select an answer using a likert scale ranging from “Strongly Disagree” to “Strongly Agree” like the SUS questionnaire. Each alternative is scored from 1 (lowest) to 5 (highest). This was done to keep the two questionnaires as similar as possible in structure, so the participants could quickly go from one to the other. The selected scoring system means that if the average score is above 3.0, the participants are leaning towards agreeing, while a score below 3.0 indicates that subjects are leaning towards disagreeing with the statement.

Some statements have been formulated in a negative fashion, like “The game was hard to understand”. The results will be displayed in two separate tables, one displaying the positive statements (Table 14.2) and one displaying the negative statements (Table 14.3). A few statements in the Technical and Experiment sections has a slightly different scale and have been put in separate tables (Table 14.4 and Table 14.5). This separation of statements have been done in order to simplify the comparison and evaluation of the results of the software prototype tests. Keeping in mind that for positive statements, a high score is better, while for negative statements a low score is better.

14.2.2 Participants and Devices

As previously stated in Section 13.1, most of the participants of the software prototype tests was recruited from the Software Architecture course. Participants were also specifically asked to bring their own Android smartphones. This unfortunately resulted in a set of test subjects with a narrow distribution across age, gender and study program, as well as level of skill with smartphones and role-playing games.

The average participant was a 25 year old male studying for a masters degree in either Computer Science or Informatics with intermediate knowledge and skill with RPGs and is a professional smartphone user. The most common device was a Samsung Galaxy S2 with Android version 4.04.

14.2.3 Learning

The learning section of the questionnaire contained seven statements regarding the educational profit/effect of KnowledgeWar 2.0 and its ability to motivate the players to learn more. These statements will hopefully assist in the evaluation of the educational effect and potential of the game. For the learning section statement Q9, Q10 and Q15 will be discussed separately before a summary of the section.

Table 14.2: KnowledgeWar 2.0 Questionnaire: Positive statements (higher is better)

QID	Statement	Avg. Score		
		SP1	SP2	Total
Learning				
Q9	My knowledge from NTNU courses came to use	3.00	3.75	3.25
Q11	The game stimulated me to learn more	2.50	4.25	3.08
Q12	The game makes course theory more fun	4.13	5.00	4.42
Q13	I would like to take courses that use a game like KnowledgeWar 2.0	4.13	4.75	4.33
Q14	I was motivated to learn more in order to do better in the game	3.38	4.50	3.75
Q15	I reviewed my answers after a battle	2.13	2.00	2.08
Gameplay				
Q16	The game was fun	3.75	4.50	4.00
Q17	I would like to play this game even if I am not required to	3.00	4.25	3.42
Q18	I got relevant feedback on how to play the game	3.63	4.50	3.92
Q19	I was motivated to move around and discover	3.50	4.00	3.67
Technical				
Q23	I liked the look-and-feel of the game	3.63	4.25	3.83
Social & Mobility				
Q29	It was fun to battle against fellow students	4.25	4.75	4.42
Q31	I liked the fact that I have to move around to play the game	3.75	3.75	3.75
Q34	Playing against fellow students made me concentrate more	3.75	4.50	4.00
Q35	I do not like to lose against fellow students	4.38	4.75	4.50

Q9: My knowledge from NTNU courses came to use

The question database used in the game only partially consisted of questions from the curriculum of the Software Architecture course, while the rest of the database consisted of basic trivia questions. Most of the participants had taken the Software Architecture course at some point in their studies. As seen in Table 14.2, there is an increase in Q9 (3.00 to 3.75) from SP1 to SP2. This indicates that the participants, at least to some extent, experienced the educational aspect of the game. The increase itself we would like to merit to the extension of answer time, which allowed players to think in stead of having to select an answer in haste.

Q10: Answering questions was stressful

From the first feedback session it was obvious that the participants found it very stressful to answer the quiz questions. As previously mentioned, this resulted in an extension of time for both reading and answering a question. Table 14.3 show a score decrease for Q10 (4.13 to 3.25) from SP1 to SP2, which might be a result of these

Table 14.3: KnowledgeWar 2.0 Questionnaire: Negative statements (lower is better)

QID	Statement	Avg. Score		
		SP1	SP2	Total
Learning				
Q10	Answering questions was stressful	4.13	3.25	3.84
Gameplay				
Q20	The game was not engaging	2.38	1.50	2.08
Q21	The game was hard to understand	2.00	2.00	2.00
Q22	The game would be more enjoyable with more game-play elements	3.88	4.00	3.92
Technical				
Q24	The game had poor performance	3.63	3.75	3.67
Q25	I would like a more graphically advanced game	2.13	2.75	2.33
Q26	Technical difficulties distracted me from the game	3.38	4.00	3.58
Social & Mobility				
Q30	I would rather play against a computer controlled opponent	2.13	1.25	1.83
Q32	I would rather play a traditional non-location specific game	2.50	2.75	2.58
Q33	There were too few elements to interact with	4.25	3.50	4.00

modifications.

Q15: I reviewed my answers after a battle

In order to embed a subtle educational aspect, as described in Section 7.2, where the players can reflect upon the questions and their answers, the battle report was implemented. From the results of the questionnaire it seems as though the participants did not see any use of reviewing their questions. From Table 14.2 the results shows that the score for statement Q15 (2.13 to 2.00) is low for both SP1 and SP2. A reason for this might be that the participants of this experiment had no real incentive, nor time, to learn and improve, but was rather focused on finding treasure chests and other players to battle. As that was what they were given points for.

Learning Summary

Through the modifications made between the first and second software prototype, the developers hoped to increase the gameplay experience of the players and make the game more fun to play. Despite the fact that they generally experienced more delay and inaccurate positioning, the participants from testing SP2 seemed to have more fun playing the game than participants who tested SP1. Increased gameplay experience seems to also affect the learning experience. The questionnaire results show a clear improvement overall but especially on statements Q11 (2.50 to 4.25), Q12 (4.13 to 5.00), Q13 (4.13 to 4.75) and Q14 (3.38 to 4.50), which gives a great outlook on the games educational ability.

14.2.4 Gameplay

The gameplay section of the questionnaire contained statements which focus on the players' perception of the gameplay experience and whether they found the game engaging, immersive, fun and easy to play. A main factor that influenced the gameplay experience during the testing of SP1 was the slow character progression together with long and tedious battles. For the second software prototype test, gameplay got influenced by the technical issues that resulted from players roaming on a WiFi connection. For this section Q19 and Q20 will be discussed separately before a summary of the section.

Q19: I was motivated to move around and discover

For the participants of the prototype tests there were two main motivations for moving around and discovering. The first, to find other players to challenge, and the second, to find treasure chests which were scattered around campus. Repeated feedback from the participants of both user experiments was the request for places of interest, like shops, as well as computer controlled monsters or bosses to fight. This was also something the developers had been hoping to have time to implement (see FR5 in Section 10), but had to be dropped as a result of time constraints.

Q22: The game would be more enjoyable with more gameplay elements

This statement was added and counted as a negative statement, in order to get feedback on whether players were happy with, or needed more, gameplay elements. The generally high score of Q22 (3.88 to 4.00) from testing both SP1 and SP2 indicate that the players still desire more gameplay. The implementation of FR5 would probably have greatly affected the outcome of this statement and the gameplay as a whole. Giving the player a more diverse and fun gameplay experience.

Gameplay Summary

Despite some technical difficulties during both the first and second software prototype test, the results show a score increase for most of the statements in this section. This is particularly evident in statement Q16 (3.75 to 4.50), Q17 (3.00 to 4.25), and Q20 (2.38 to 1.50), which might be a result of the tweaks made on the software prototype. Further the results of Q18 (3.63 to 4.50) and Q20 (2.38 to 1.50) together with the feedback from the participants indicates that they found the game easy to understand and play. Q19 (3.50 to 4.00) shows a slight increase in motivation to move around and discover. However, the results of Q22 (3.88 to 4.00) and the feedback from participants clearly indicates that the game needs more pervasive gameplay content to motivate the user to explore and play the game regularly.

14.2.5 Technical

The technical section of the KnowledgeWar 2.0 questionnaire contained statements that focus on the perceived performance and graphical user interface of the client software. Q27 and Q28 had a different scale than the other statements and is presented in Table

14.4. The table presents the results as percentage of participants who answered each alternative. One participant, however, chose not to answer these questions as he only played the game while sitting indoors.

Table 14.4: KnowledgeWar 2.0 Questionnaire: Additional technical statements

QID	Statement	Alternatives				
		Never	Once or twice	25% of the time	50% of the time	Throughout the game
Technical						
Q27 (SP1)	My position was correct.	0%	0%	28.6%	57.1%	14.3%
Q27 (SP2)	My position was correct.	0%	50.0%	25.0%	25.0%	0%
Q28 (SP1)	I experienced delay.	0%	28.6%	14.3%	28.6%	28.6%
Q28 (SP2)	I experienced delay.	0%	0%	25.0%	75.0%	0%

The technical section of the questionnaire is generally the worst scoring section for both SP1 and SP2. Technical difficulties during the test of SP1 were mainly aimed at the slightly sluggish user interface, and one user who had problems connecting and registering an account. As stated in Section 13.5, half the participants testing SP2 did not have the possibility to perform the test using their mobile data traffic service and had to use WiFi, which was the main source of technical difficulties during the test of SP2.

Q27: My position was correct

Participants of the first software prototype test were playing the game using 3G for communication and GPS for position tracking. As can be seen from Table 14.4, the results of Q27 (SP1) indicate that the participants experienced accurate positions most of the time during the experiment. Participants playing over a WiFi connection, during the second software prototype test, experienced positioning to be slow and inaccurate. This is supported by the results of Q27 (SP2), where 50% of the participants only experienced an accurate position “once or twice”.

Q28: I experienced delay

For Q28 (SP1), the results show that an even distribution of participants experienced delay ranging from “once or twice” to “throughout the game”. For Q28 (SP2) the result are more concentrated around experiencing delay “50% of the time”. For this statement, “delay” may have been interpreted in different ways. Participants who tested SP1 experienced a delay in the user interface, which resulted in a slow response when players tried to answer the questions. As SP1 participants were using the mobile telecommunication networks (3G/EDGE), players may also have experienced variations in the network coverage while roaming, which would greatly affect the round trip time of tokens sent to and from the server. The latter would also be true for the participants of SP2 who used 3G and GPS during play. The participants using WiFi would experience a lower round trip time if they had a stable WiFi connection. They would, however, experience delays in the form of having to wait for the client to reconnect as the smartphone switch from one wireless access point to the next while roaming.

Technical Summary

The high score of Q23 (3.63 to 4.25) together with the low score of Q25 (2.13 to 2.75) generally indicates that the GUI of the client served well enough for this type of game. Though the high scores of Q24 (3.63 to 3.75) and Q26 (3.38 to 4.00), indicates that technical difficulties were still a major problem for the game. As this was a prototype implementation, some performance issues was to be expected. For a full scale implementation, more time and effort will have to be put towards using software architecture techniques. As discussed, the results of Q27 and Q28 might be seen in direct association with the participants using either WiFi or 3G and GPS when playing the game.

14.2.6 Social & Mobility

The Social & Mobility section of the questionnaire consists of statements mainly focused on the participants experience of playing the game against fellow students, and how well they enjoyed the pervasive aspect.

Social & Mobility Summary

From the feedback for the participants and observations made by the test personnel during the user experiments, it seemed evident that the participants thought playing against fellow students was both fun and very engaging. This is further supported by the scores of the Q29, Q30, Q34, and Q35 statements of the questionnaire. Q29 (4.25 to 4.75) and Q30 (2.13 to 1.25) indicates that players enjoy playing against their fellow students more than they would playing alone or against a computer controlled player. The results of Q34 (3.75 to 4.50) and Q35 (4.38 to 4.75), further indicates that players are more engaged and motivated to perform well and win because they do not like to lose against their fellow students. The pervasive elements of the game also seems to be well received amongst the students, which is indicated by the scores of Q31 (3.75 for both SP1 and SP2) and Q32 (2.50 to 2.75). From the feedback of the user experiment of SP1, one of the participants also stated that “My idea of fun with this game is the running around, gathering items, gaining experience, and have a lot of fights.”. Still, the high score of Q33 (4.25 to 3.50) indicates that a lot more content need to be implemented to fully embrace the pervasive potential of the KnowledgeWar game.

14.2.7 Experiment

In the experiment section the participants were asked how often the client software froze or crashed on a scale from “0” to “greater than 9”. The results from testing SP1 and SP2 are presented in Table 14.5 and is shown as a percentage of the participants answering each alternative.

Table 14.5: KnowledgeWar 2.0 Questionnaire: Additional experiment statement

QID	Statement	Alternatives				
		0	1-2	3-5	6-9	>9
Experiment						
Q36 (SP1)	How often did the client crash?	62.5%	12.5%	25%	0%	0%
Q36 (SP2)	How often did the client crash?	25%	25%	25%	25%	0%

The results of Q36 (SP1) supports the experience of the developers, that the game is more stable being played outdoors using 3G and GPS, as 62.5% of the participants of testing SP1 reported zero crashes. The test results from testing SP2 presents a more wide distribution ranging from “0” to “6-9” crashes. Which might be the result of half of the participants using 3G and GPS, and the other half using WiFi.

The last field of the questionnaire was a section asking for further comments, where participants could suggest other improvements and thought about the game. These comments have been included in the presentation of feedback after each of the software prototype tests, and consisted mainly of suggestions for additional gameplay elements.

Chapter 15

Evaluation and Discussion

In this chapter the Non-Functional Requirements and Technical Requirements from Chapter 10 will be presented and discussed up against the findings in Chapter 14. All the relevant statements from tables in Chapter 14 will be included in this chapter for ease of access for the reader. Positive and negative statements are marked to distinguish between a good score and a bad score. For a positive statement, a higher score is better. For a negative statement, a lower score is better.

15.1 Non-Functional Requirements

This section summarises the non-functional requirements, as they are an important aspect for the enjoyment of a game. It is, however, hard to measure for example what fun is or how engaging a game can be. This means that the results will have to mainly be based on the discretion of the test participants.

15.1.1 NFR1 - Usability

“The game should be intuitive and easy to use especially for people not familiar with the RPG genre”

As is indicated in Section 14.1, and shown in Table 15.1, the total average SUS score for the combined Software Prototypes is 80.5 points. A SUS score of over 80.3 points places the prototype in the statistical top 10% of scores⁸⁴. From the questionnaire results in Table 15.1, the scores for Q23 (3.83/5.00) gave a positive outlook on how the participants of the tests liked the look-and-feel of the game. While statements Q24 (3.66/5.00) and Q26 (3.58/5.00), which are negative statements where a higher score is worse, gave room for improvements. Especially, some of the participants reported that during the testing period their battery was drained along with their fingers freezing in the cold outside. This is, unfortunately, somewhat out of our hands. However, as it is mentioned in Section 14.1, there would have been many positive side-effects, if there had been time for additional iterations of development, feedback and redesign. This could improve usability, as well as other identified issues (except for the weather). With this being said, all of the test participants, including those who were not as familiar

⁸⁴<http://www.measuringusability.com/sus.php> (08.11.2012)

with the RPG genre, were able to play the game. Despite the mixed feedback from test participants, the testing of the learning aspect of game was still achieved. The results from Table 15.1, especially Q18 (3.92/5.00), indicates that the test participants did indeed get relevant feedback on how to play the game, which is a good indicator that people who are not familiar with the RPG genre still can find the game intuitive and easy to use.

Table 15.1: Related questions for Usability from the SUS form and questionnaire

QID	Statement	Avg. Score		
		SP1	SP2	Total
–	System Usability Scale scores	78.33	83.75	80.50
KnowledgeWar 2.0 Questionnaire Positive Statements				
Q18	I got relevant feedback on how to play the game	3.63	4.50	3.92
Q23	I liked the look-and-feel of the game	3.63	4.25	3.83
KnowledgeWar 2.0 Questionnaire Negative Statements				
Q21	The game was hard to understand	2.00	2.00	2.00
Q24	The game had poor performance	3.63	3.75	3.67
Q26	Technical difficulties distracted me from the game	3.38	4.00	3.58

15.1.2 NFR2 - Fun

“The game should be perceived as fun to play”

The game has already been deemed usable, and as it was indicated from the increase in scores from SP1 to SP2: the more usable it is, the more fun it can be to use. As it is indicated previously, there is room to improve usability still. Having a look at Q12, Q16 and Q29, in Table 15.2, gives the idea that the testers rated the elements of the game that were supposed to represent fun as high as from 4.00 out of 5.00 (80%) to 4.42 out of 5.00 (88.4%). Test participants reported that battling against fellow students gave the game an extra element of fun and challenge. In order not to look bad, the players had to focus more on the curriculum in order not to lose. This was reported to be both fun and somewhat stressful. The second software prototype test indicated that a quicker character progression, along with other tweaks, gave the players more to care about in the game, and they in turn reported the game to be more fun. The scores in Table 15.2 gives room for improvements, but overall indicates that the game was perceived as fun to play.

Table 15.2: Related questions for Fun from the questionnaire

QID	Statement	Avg. Score		
		SP1	SP2	Total
Positive Statements				
Q12	The game makes course theory more fun	4.13	5.00	4.42
Q16	The game was fun	3.75	4.50	4.00
Q29	It was fun to battle against fellow students	4.25	4.75	4.42

15.1.3 NFR3 - Engagement

“The game should be engaging and create motivation for the players”

Following the results from the questionnaires, the test participants viewed the game as something they would like to use in combination with courses at NTNU. They felt motivated to learn more in order to perform better in the game and even said they would use the game even though they were not required to. The questionnaire contained the negative statement Q20, participants rated it as 2.08 out of 5.00 which means that they found it mostly engaging. The participants also rated Q21 to 2.00 out of 5.00, which means the game was not hard to understand either. Most would like more gameplay elements to interact with in the future (Q22), as it was reported from both the questionnaire and feedback given after the experiment. However, the participants found the game enjoyable as it was. Playing a game against fellow students creates an interesting side-effect not easily achieved by designing a game in itself: A possibility to lose face when encountering friends on the battlefield reportedly made the test participants concentrate more in order to not lose the battle. The scores from the positive statements in Table 15.3, are well above the middle score, which creates a positive result for the game’s engaging qualities. Q20 and Q21 are below the middle score, which is also a positive result. The results from Table 15.3 tells us that the players found the game engaging and a motivational factor in order to learn more.

Table 15.3: Related questions for Engagement from the questionnaire

QID	Statement	Avg. Score		
		SP1	SP2	Total
Positive Statements				
Q13	I would like to take courses that use a game like KnowledgeWar 2.0	4.13	4.75	4.33
Q14	I was motivated to learn more in order to do better in the game	3.38	4.50	3.75
Q17	I would like to play this game even if I am not required to	3.00	4.25	3.42
Q19	I was motivated to move around and discover	3.50	4.00	3.67
Q34	Playing against fellow students made me concentrate more	3.75	4.50	4.00
Q35	I do not like to lose against fellow students	4.38	4.75	4.50
Negative Statements				
Q20	The game was not engaging	2.38	1.50	2.08
Q21	The game was hard to understand	2.00	2.00	2.00
Q22	The game would be more enjoyable with more gameplay elements	3.88	4.00	3.92

15.1.4 NFR4 - Immersion

“The game should deliver gameplay that immerses the player in the game to further motivate the player”

There were some technical difficulties during the test phases, as indicated by Q26 in Table 15.4. However, taking a look at Q23 and Q25, test participants liked the look-and-feel of the game, even though most wanted a more graphically advanced game. Players felt motivated to move around campus to discover treasure chests and other players. It was reported that the game would be more enjoyable with more gameplay elements, and if there had been time and resources to implement this, the game could be made even more immersive.

Table 15.4: Related questions for Immersion from the questionnaire

QID	Statement	Avg. Score		
		SP1	SP2	Total
Positive Statements				
Q23	I liked the look-and-feel of the game	3.63	4.25	3.83
Q31	I liked the fact that I have to move around to play the game	3.75	3.75	3.75
Negative Statements				
Q22	The game would be more enjoyable with more gameplay elements	3.88	4.00	3.92
Q25	I would like a more graphically advanced game	2.13	2.75	2.33
Q26	Technical difficulties distracted me from the game	3.38	4.00	3.58

15.1.5 NFR5 - Educational

“The game should foremost be able to help the players to memorize existing knowledge, and second, be able to give them new knowledge.”

The database of questions acquired for testing was provided by the project supervisor - Alf Inge Wang. The database consisted mostly of quiz-type questions from his lectures on Software Architecture. This also meant that to be able to get a result for the learning aspect of the game, the test participants had to have taken this course. Unfortunately, because of this, the user experiment presents a hypothetical situation where the test participants have taken this course in the past. Therefore it can be a bit hard to evaluate the educational aspect of the game.

It was attempted to avoid making the players stressed when answering a quiz question in battle. In Software Prototype 1 the players were given only 3 seconds to read the question and 5 seconds to choose an alternative. This resulted in players giving high scores on Q10 for SP1(4.13/5.00), see Table 15.5. The tweak for Software Prototype 2, which gave players 5 seconds to read the question and a full 10 seconds to choose an alternative resulted in players giving lower scores for Q10 on SP2 (3.25/5.00). The element of stress is, however, not completely gone. Players still have to answer the quiz question faster than their opponent. Despite this, Q10 gives an overall adequate

results, and although being above the middle score had a positive development from SP1 to SP2.

The reflection part of learning shows a negative result in Q15. It was believed that a list of buttons would be more clearly visible on a software prototype and that this would be enough for the users to understand what it was and use it. Feedback from the participants revealed that they felt they had no real incentive to review questions, as it was not rewarded in any way. The users were also more keen on finding the right answer through trial and error, rather than find the course's curriculum and studying the answer. This might be because the test was run in a fictional setting, where all participants had gone through the course in an earlier semester, and didn't actually need to study for an exam at the time of testing. One test participant suggested that a reference link to relevant information might be provided directly in the battle report so that users might acquire new knowledge about the quiz question directly in the game without having to switch focus. Regardless of the negative feedback on Q15 and Q10, the results for NFR5 shows a significant increase from SP1 to SP2 in all the statements in Table 15.5, except Q15. The specific feedback about incorporating course curriculum directly in the game indicates that there is some further work to be done in order motivate players to reflect upon, and be able to gain new knowledge. However, the results show that a quiz-type application for repeating knowledge makes course theory more fun and easy to memorize.

Table 15.5: Related questions for Educational from the questionnaire

QID	Statement	Avg. Score		
		SP1	SP2	Total
Positive Statements				
Q9	My knowledge from NTNU courses came to use	3.00	3.75	3.25
Q11	The game stimulated me to learn more	2.50	4.25	3.08
Q12	The game makes course theory more fun	4.13	5.00	4.42
Q15	I reviewed my answers after a battle	2.13	2.00	2.08
Negative Statements				
Q10	Answering questions was stressful	4.13	3.25	3.84

15.2 Technical Requirements

This section summarises the technical requirements from Chapter 10. They are a measurable form of requirements that are aimed mainly at the server software, but also the system as a whole.

15.2.1 TR1 - Roaming

“The system must be able to handle clients roaming as they move about on campus, keeping the connection to the server alive and reconnect if connection is lost”

The client software is set up to attempt to reconnect to the server if it loses its connection. Although test participants experience some delay, they were able, and motivated

to move around and discover. When walking outside the range of an Access Point (AP) test participants reported that they were disconnected for a short period before getting reconnected again. It was reported that this happened both during the Discovery and Battle gameplay elements. One user also reported that he had disconnected during a battle, at which time the client also had frozen to the point where he had to restart the client. This also caused his opponent's client to freeze up and had to be restarted. Both players lost all statistics for previous battles, as well as any weapons, armor, experience and level gained since last login. This, however, only happened once out of the 31 total battles that were fought during the tests. Even though error's like this might happen, the prototype testing was still successful. There is, however, a need to put in more time and effort in to using software architecture techniques to rule out these kinds of errors.

Table 15.6: Related questions for Roaming from the questionnaire

QID	Statement	Avg. Score		
		SP1	SP2	Total
Positive Statements				
Q19	I was motivated to move around and discover	3.50	4.00	3.67
Negative Statements				
Q32	I would rather play a traditional non-location specific game	2.50	2.75	2.58

15.2.2 TR2 - Location

“The system must keep an updated list of online players and their positions”

The server queries the Wireless Trondheim system as often as possible with the restriction given to the developers. The server for KnowledgeWar 2.0 was not allowed to query for more than 5 positions per second. The algorithm mentioned in Section 12.2.3 was developed, and as long as a user was logged in and not in between switching from one AP to another, the user received an updated position from the Wireless Trondheim system. If a player uses 3G or EDGE, the client itself queries for the location and updates the server with its position. This means that if a user is logged in, and the client has a connection to the server, a position is retrieved and kept at the server.

Table 15.7: Related questions for Location from the questionnaire

QID	Statement	Never	Once or twice	25% of the time	50% of the time	Throughout the game
Q27 (SP1)	My position was correct	0%	0%	28.6%	57.1%	14.3%
Q27 (SP2)	My position was correct	0%	50.0%	25.0%	25.0%	0%
Q28 (SP1)	I experienced delay	0%	28.6%	14.3%	28.6%	28.6%
Q28 (SP2)	I experienced delay	0%	0%	25.0%	75.0%	0%

15.2.3 TR3 - Persistence

“The game world and character statistics must be persistent. Data should not be lost if the client loses its connection to the server, or if the user logs off and on again”

As long as the server is turned on and running, the game world exists on campus regardless of whether any users are logged on or not. The game world is set up to primarily exist on university campus, as this is where the treasure chests are found. It is, however, possible to play outside of this area as well if the users wishes to do so. As a persistent connection between the client and the server was kept at all time, the server would instantly be aware if a client would, for any reason, disconnect. The server would, in the case of a disconnecting client, store the data for the character associated with the client. The data is stored on the server in the case that the client is only roaming. If the player has actually logged off or moved outside an area where there is no connection, the data is eventually stored in the database. As mentioned in Section 15.2.1, during testing, there was one instance of an error that broke persistence for two players. This was a result of the fact that they were roaming while in a battle, combined with an unfortunate client freeze up. This error is only related to if players disconnect completely while using WiFi during a battle. Some additional work needs to be done in order to maintain a fully persistent world, especially if more advanced virtual world elements are to be included.

15.2.4 TR4 - Platform

“The game should work on 90% of all Android devices”

The client software has been tested and works on a wide variety of Android devices. It has been tested on versions ranging all the way from version 2.1 - “Eclair” to 4.1 - “Jelly Bean”. The Jelly Bean update had a peculiar bug, if the user zooms in or out too fast, the application crashes. This is however specific to Android version 4.1 and Leaflet version 0.4. As seen in Table 15.9, the Android version 4.1 usage share is only 2.7%, and was only used by the developers during the testing. There was also a bug with version 2.3, when the user accesses the map for the first time, it would only show a grey screen instead of the map. This could, however, easily be solved by zooming in or out once.

To achieve the best possible performance in the game, a newer version of Android is better. Version 4.0 and up handles JavaScript, HTML5 and CSS3.0 better than the older versions. Taking a look at the Usage share for Android versions as of 1st of November 2012 on Table 15.9⁸⁵, the total sum of android versions the KnowledgeWar 2.0 client works on exceeds 99%. Test participants also reported that the game had an overall adequate performance, but some crashes occurred.

⁸⁵<http://developer.android.com/about/dashboards/index.html> (08.11.2012)

Table 15.8: Related questions for Platform from the questionnaire

QID	Statement	Avg. Score		
		SP1	SP2	Total
Negative Statements				
Q24	The game had poor performance	3.63	3.75	3.67

Table 15.9: Android version usage share

Version	Version name	%
2.1	Eclair	3.1%
2.2	Froyo	12%
2.3	Gingerbread	54.2%
3.x.x	Honeycomb	1.8%
4.0	Ice Cream Sandwich	25.8%
4.1	Jelly Bean	2.7%
	Total	99.6%

Chapter 16

Conclusion

This chapter summarises the projects research findings up against the research questions that were formulated in Chapter 2. Further we reflect upon how well we have followed the selected research method presented in Chapter 3.

16.1 Educational Research Questions

This section summarises the educational aspects of the research questions in light of the findings of the pre-study combined with observations, results and feedback from the user experiments.

16.1.1 Research Question 1

“How can a smartphone application be used to make the curriculum of a university course more engaging for students?”

Over the course of studying the current field of games and higher education (presented in Chapter 4), it was discovered that there are not that many examples of smartphone applications being used as we had expected. Furthermore the existing applications focus more on the learning aspect rather than on creating fun and enjoyable games. From the evaluated software, we discovered that quiz questions are a good way of repeating knowledge. Quiz type applications have already been tested, with success, in Lecture Quiz, KnowledgeWar, Campus Quest, Amendor’s App Framework and Buzz! The Schools Quiz. However, we wanted to make a game that took educational gaming to another level by drawing upon well known game genres and the possibilities of modern smartphones, to make a fun and engaging pervasive game.

To make the game accessible, multiple technologies were evaluated and discussed in Chapter 5. Seeing as we wanted the game to be accessible to most, if not all, students without having to purchase or borrow an additional device, it was decided to use HTML5 and JavaScript for development. This made the game cross-platform friendly and easily adaptable for all smartphone platforms, not just the tested Android platform.

A multitude of game genres were evaluated in Chapter 7. In the end we chose a combination of the Role-playing Game genre for persistent world and accessible game

design elements, together with the perspective of the fighting game genre during battles. These genres provided a volume of easily implementable game design techniques that would make our solution not only help students learn, but make it look, feel and act as a game.

16.1.2 Research Question 2

“How can game design techniques be used so that the application promotes and rewards further use?”

In our implementation we included several game design techniques in order to stimulate, promote and reward further use. Especially the RPG genre was interesting because it includes a multitude of game elements suitable for this purpose. Chapter 11 describes the game design process for what type of techniques and elements we included in the game. These elements include the creation of a character representing the player in the virtual world and equipment in the form of weapons and armor which help the character defend itself in battles. Furthermore a point system for experience, health and a form of mana (usually used for casting spells) which in our case is dubbed “Special Resources” and is used for a special type of attack. For the scenario where two players meet up and challenge each other to a fight, we chose to take the players into a fighting game style of combat. In this way, players cannot be disturbed by others, and are able to be totally immersed in the game and focus on the quiz questions they have to answer in order to win. The combat also features animations and sound effects to make the game even more engaging.

When creating a character in the game, the player can choose an amusing avatar (courtesy of Scott Johnson’s “56 Geeks” project) whereupon the character chosen gets a set of attribute points unique to their class. These attributes dictate how strong the attack damage, defence, and special attack of the character is. The different attributes have been aptly named “Attack”, “Defence”, and “Special”. Together with these attributes, the weapon and armor decide the actual damage a character receives. A weapon has a damage range that is added with the Attack attribute, while a piece of armor has a defence amount that is added to the Defence attribute. The game also includes virtual world elements that can only be seen while using the game. Players were given an interactive map with fog of war in order to stimulate them to move from their current location to find an opponent to challenge. Treasure chests containing weapons or armor are spread around the virtual campus map for players to discover. The items a player can get from a chest is greater than that which can be won from a battle, giving players an incentive to discover the map and collect treasure chests.

Together these elements create the basis of a virtual game world set in the context of being some sort of geek taking higher education. This game world can further be extended to contain a great amount gameplay elements for the player to immerse him/herself in.

16.1.3 Research Question 3

“To what extent can a pervasive educational game help to engage and motivate students and increase their learning experience?”

In order to answer this research question we performed a user experiment with a software prototype based on the game design presented in Chapter 11. To collect data from this user experiment we; made observations, had the participants of the experiment fill out a custom questionnaire (see Appendix F) and a SUS questionnaire (see Appendix G) and had informal group discussions with the participants.

The main goal of our project was to allow students to repeat and better memorise their knowledge, as well as motivating them to learn more. Section 14.2 is dedicated to presenting and evaluating the result set of the custom questionnaire. Further, Chapter 15 evaluates the results and feedback up against the goals of this project. All together, our evaluation indicates that the game was both fun and engaging, as well as a motivational factor to learn more. Participants reported that it was fun to battle against fellow students and that they got engaged and concentrated more as they did not want to lose. Participants claimed that they would like to take courses that used a game like KnowledgeWar 2.0, that the game made course theory more fun and that it motivated them to learn more. However, participants expressed a need for more gameplay, things to do in the game, and that the software prototype had some performance issues. The overall result does, however, give a great outlook on the possibilities a pervasive light-weight RPG game can help to engage and motivate students of higher education.

16.2 Technical Research Questions

This section looks at the technical research questions in light of findings of the pre-study, experience from implementing the prototypes as well as observations, feedback and results of the user experiments.

16.2.1 Research Question 4

“How mature is the HTML5 platform for use on smartphones?”

Section 5.2.1 of the pre-study presented how the support for HTML5 in mobile browsers has evolved over the last few years. Each new release of any of the major mobile platforms show an increased support for the new web standard. However, as presented earlier in Table 15.9, over 50% of Android devices are still running version 2.3 of the OS. In terms of the scores presented by `html5test.com`, Android 2.3 scores 200/500 which is significantly lower than Android 4.0 (or newer) with a score at 297/500.

In our implementation, we encountered lack off HTML5 support when we tried to use the new audio tag for playback of sound effects. Even though the audio tag itself is recognised and supported by the browser object (at least from Android version 2.3), the browser does not necessarily have the codecs needed to play, e.g., an MP3 file. Thus, PhoneGap’s media object was used instead in order to provide the best cross-version and platform support.

Even though HTML5 can be said to still be “under construction”, increased support for the web standard through the different cross-platform frameworks are rapidly being developed. This has been evident throughout the implementation phase of this project, as all frameworks used have had at least one major version update, while others have

had as much as eight updates (PhoneGap version 1.6.0 to 2.2.0⁸⁶). Even the Android platform have had two major updates to the operating system (Android 4.1 & 4.2 Jellybean⁸⁷) during the lifetime of this project.

16.2.2 Research Question 5

“How well does the current HTML5 performance suit mobile pervasive games?”

Even though HTML5 provides quicker development of cross-platform friendly user interfaces, it does not deliver as good a performance as the native user interfaces of Android or iOS devices. Based on the feedback from the participants of the first software prototype test, it became clear that the user interface had performance issues. At first it was thought that this was due to the HTML user interface itself. However, as presented in Section 13.4, this turned out to be a result of how events was triggered on user input. The issue was easily fixable and gave a big increase in response and performance of the user interface. Given more time, there are probably several similar tweaks that could have been identified and performed in order to increase the performance of the client even more.

An important aspect of a pervasive game is its ability to blend into the player’s everyday life. In today’s society there is no better way of doing this then by using the smartphone that has in an ever increasing way, become a natural carry-on for most young adults. The variety of device types, sizes, performance and platforms is rapidly increasing and is a big obstacle to overcome, if a game such as KnowledgeWar 2.0 is to reach the heterogeneous mass of people taking higher education.

The results of the questionnaire presented in Chapter 14.2 indicate that the participants liked the look-and-feel of the game and did not need more advanced graphics in order to have fun playing the game. This despite the fact that both of the software prototypes had some performance issues. Using a third party API like PhoneGap bridges the gap between the HTML5/JavaScript user interface and the device’s native possibilities. Altogether this leads to the conclusion that HTML5 was both easy, cheap and suitable for our purpose of making a light-weight cross-platform pervasive RPG game.

16.2.3 Research Question 6

“How is it possible to make a fun and engaging game that works well across all major platforms without having to write two sets of code?”

The software prototype was developed to be cross-platform friendly. However, as communication with the server was done through a websocket connection, a native websocket solution with a JavaScript websocket API was needed. For the software prototype to be truly cross-platform, it would have to have separate native websocket

⁸⁶<http://phonegap.com/download> (24.11.12)

⁸⁷<http://developer.android.com/about/versions/jelly-bean.html> (24.11.12)

implementation for each platform (Android, iOS, Windows Phone), with a shared common JavaScript API. Otherwise large parts of the code would need to be tailored or re-written for each platform.

As we did not have the possibility to develop and test the software prototype for more than the Android platform, it is hard to draw any clear conclusions on the cross-platform performance of the software prototype. However, the server software is completely ignorant of the device platform the user is on, as it simply accepts websocket connections and communicates with clients through generic JSON tokens. The client software itself should only need simple modifications in order to be compiled for other platforms, as it would need a websocket implementation compatible with the target platform.

The results from the gameplay section of the KnowledgeWar 2.0 Questionnaire in Section 14.2.4 and the evaluation of NFR3 (Engagement) in Section 15.1.3, gives the clear indication that participants found the game to be very engaging, as they did not want to lose when playing against fellow students. This leads in heightened focus and motivation during the quiz battle which in turn might lead to an increased motivation to learn and retain knowledge. Furthermore, the evaluation of NFR2 (Fun) in Section 15.1.2 together with the feedback received from the participants presents how they found the game fun to play, though they also request more gameplay elements and things to do in the game. Though lack of gameplay itself is not positive, the enthusiasm shown by the participants as we discussed additional features that could be added to the game clearly shows that KnowledgeWar 2.0 has a good potential of becoming a well-liked educational aid among university students at NTNU.

16.3 Project Evaluation

Chapter 3 described our planned research methodology for this project. The method defined a three part procedure consisting of; a pre-study, a software engineering method and a user experiment. We will in this section review our planned procedure up against the work done and presented in this thesis.

16.3.1 Pre-study

Our pre-study is presented in Part III of this thesis and consists of four main areas of study; Earlier Work, Mobile Technology, Server Technology and Game Studies. For each category we have presented the findings we found to be relevant to the problem description of the thesis. In earlier work we present research and development projects that has focus on either educational games, pervasive games, use of a smartphone or a combination of the three in relation to higher education. For the mobile technology part of the pre-study, we researched various development strategies and found and tested out a series of cross-platform development framework. The findings from this study helped us to select a combination of tools and frameworks to work with for the development of the client-side of the software prototype. In the server technology part of the pre-study, we researched different server software that allowed for implementation of bi-directional communication between the server and the client and that provided basic functionality and a quick start to prototype the server-side of the software prototype. Last, in the

game study section we looked into the many different and popular genres of commercial games which helped us to find game design techniques to use in our own game design in order to make a game that would be received as both fun and engaging by the students at NTNU.

16.3.2 Software Engineering Method

In Section 3.2 we present the different approaches to performing software research, presented by Basili et al. [3]. We further proclaimed that we would follow the engineering method which; consists of observing existing solutions, propose better solutions, build/develop, measure and analyse, and repeat the process until no more improvements appear possible. As a result of the time constraints of the thesis we declared that we would perform two major iterations.

Observing existing solutions was performed as a part of the earlier work (see Chapter 4) of the pre-study of this thesis. In addition to studying similar projects, the earlier work section presents and evaluates the previous implementation of KnowledgeWar. Based on the findings of the pre-study we proposed a better solution which is presented in Chapters 9, 10 and 11. From the initial game design, we created a paper prototype which we used to perform user test on students and had them fill out a SUS questionnaire to measure usability (see Section 11.2).

Based on the feedback from the paper prototype test we started a new iteration of the engineering method. A new and improved game design was proposed (see Section 11.3) and set some requirements for the software prototype (see Section 10) before we started the implementation phase outlined in Chapter 12 This resulted in the first software prototype (SP1). SP1 was tested on voluntary students in a user experiment as described in Chapter 13. As we had the time and possibility we decided to make a final third iteration of the engineering method. Based on the results and feedback from testing SP1 we identified some critical issues that negatively affected the gameplay. We performed some tweaking of the software prototype to mitigate these issues which resulted in the second software prototype (SP2) and performed a second user experiment.

16.3.3 User Experiment

When we presented our method for performing the user experiments in Section 3.3 we had envisioned a larger user experiment where perhaps the greater part of the students of a selected course would take part. Instead, as previously stated, two smaller user experiments were performed. Though, this lead to fewer participants and points of reference for the results, this gave us the possibility to make an extra development iteration to further enhance the software prototype. This turned the already good results of SP1 into great results for SP2, which can be seen in Chapters 14 and 15. The fact that there were rather few participants consequently makes the results not generalizable. However, it made the user experiments easier to perform and monitor the participants as they played the game. With fewer participants the group discussions were also able to include and take in everyone's perspective. However, before a final implementation of the game a large scale user experiment should be performed, to

create more generalizable results and to be certain that all implemented features are working as intended.

16.3.4 The Research

Our take on educational games, in order to make it fun and engaging to learn, was to add plenty of game design techniques on top of a learning based quiz head-to-head battle between students. In addition, the game was made as a location-based pervasive game. In our pervasive game, players have to move around in the real world in order to move their character in the game. This leads to an innovative take on learning games, where the user cannot be seated in one place while playing the game. The players have to move around in order to find treasure chests to be able to perform better in combat (aside from actually knowing the course curriculum), or to find opponents to challenge for a knowledge duel.

The research done in this thesis indicates that the market today does not contain many examples of fun and engaging smartphone based learning games that incorporate more than just basic game design techniques.

This is further supported by cognitive researcher Daphne Bavelier, who in her TED⁸⁸ talk; “Your brain on video games”⁸⁹, presents her research on how fast-paced video games can positively affect the brain. On the question of why the market is not flooded by games specifically designed to take advantage of these positive effects, she presents the following equation:

“There are brain scientists like me that are beginning to understand what are the good ingredients in games to promote positive effects, and that’s what I’m going to call the broccoli side of the equation. There is an entertainment software industry which is extremely deft at coming up with appealing products that you can’t resist. That’s the chocolate side of the equation. The issue is we need to put the two together, and it’s a little bit like with food. Who really wants to eat chocolate-covered broccoli? None of you. And you probably have had that feeling, right, picking up an education game and sort of feeling, hmm, you know, it’s not really fun, it’s not really engaging. So what we need is really a new brand of chocolate, a brand of chocolate that is irresistible, that you really want to play, but that has all the ingredients, the good ingredients that are extracted from the broccoli that you can’t recognize but are still working on your brains.”

This supports our conclusion from the feedback and results of the user experiments. The test participants were thrilled by the idea of a pervasive learning game, but want even more gameplay elements included in the game.

⁸⁸<http://www.ted.com/> (29.11.12)

⁸⁹http://www.ted.com/talks/daphne_bavelier_your_brain_on_video_games.html (29.11.12)

Chapter 17

Further work

This chapter presents some remaining work, along with some suggested further work, to finalize the project before being ready to be used in an educational setting. The items listed under system completion is a result of suggestions from the feedback and discussion sessions following the testing of the software prototypes (outlined in Chapter 13) in combination with discussion between the developers and the supervisor of this project. This chapter also lists a volume of potential features, beyond those who have been implemented in the prototype resulting from this project. These potential features are a result of features discovered by the developers during development. These are features that might add another element of fun to the game, and would also result in a more enjoyable game and learning experience for the players. The reason that this chapter exists is because of time constraints of the project, and that the project as a whole wants to test if a lightweight RPG pervasive game works for educational purposes, not to make the perfect educational game.

17.1 System Completion

Course theory reference

One problem that was encountered during testing of the software prototype was that the test participants did not use the functionality for reviewing questions after a battle. The evaluation of the feedback from Section 15.1.5 indicates that the players had no incentive to use the functionality. In Section 7.2 we defined that reflection must appear as an in-game goal that drives the gameplay in an educational game. An idea that was given to us during a feedback session was the incorporation of course curriculum directly into the game. This would give the players the opportunity to learn the theory behind a question that was asked during a battle, without having to switch focus away from the game. Relevant information from the course curriculum could be presented as a simple popup screen when the user clicks on a reference link next to the question on the battle report screen. When the course theory is available directly within the game, the motivation for students to reflect upon their answer while playing the game can increase. Another simple way to create motivation for players to review their questions is to offer a reward for doing so. A reward could consist of extra experience points or in-game currency.

True cross-platform

To be able to offer students a game they can play on a device they already own, the game has to be made truly cross-platform. As the game is made using HTML5 and CSS3 it is possible to package the game for other OS types. It is also possible to make the game run fully in the browser on a device. The advantage of packaging the game in native code is because of the possibility to distribute the game over normal distribution channels for the platform (Google Play for Android, AppStore for iOS), as well as reducing the data traffic required for the end users. To be able to truly support all of the potential users device's, the game must be made to run on Android, iOS as well as Windows Phone 8.

Live player tracking

The system from Wireless Trondheim is sadly not adequate for live tracking of players. The transition when moving from indoors to outdoors and back has to become more smooth in order for the players not to experience delay when switching between access points or generally moving around. A possible solution use a form of sensor fusion, where one would combine the use of location from WiFi, GPS and 3G/EDGE. This would have to be implemented so that it is transparent to the user.

Scalable server

The server must be scalable so that it is possible to support the use of the game across multiple courses. The jWebSocket implementation supports the use of multiple plugins, as well as clustered multiple servers that allows for an almost unlimited amount of client connections.

Extensive testing

Large scale testing to see the result of many players playing the game at the same time. This is needed in order to rule out any major malfunctions in the software, as well as to make sure that all features that are implemented works as intended. It is also required in order to be certain that all the gameplay elements are working properly. A large scale testing will uncover flaws in the balance of the game.

17.2 Potential Features

Various features for the game were both planned ahead of development, as well as figured out during development. Unfortunately, due to time constraints, not all features made it in to the prototype. These features are presented in this section as potential features that would contribute positively to the game.

17.2.1 Gameplay

Potential gameplay features that would add more to the game beyond repeating curriculum in a quiz battle.

Lore or Story mode

A compelling story mode which takes the player through various quests with rewards for completing different stages could further increase motivation and immersion.

Various obtainable achievements or vanity rewards

Players could be presented with achievements for completing certain in-game goals like fighting a certain amount of battles or opening an extravagant amount of treasure chests. It would bring a whole new level of competition to the game as well as increase the replay value of the game.

Persistent Character Status

Persistent health, special resources and status ailments would mean that players would have to retrieve a health potion or some sort of food to increase their health points after a battle. They might even have to seek out a cure for their status ailments before daring to battle again.

Currency

A possibility to earn an in-game currency, in order to be able to buy new weapons, armor or other helpful items.

Places of Interest

Including different places of interest for the players to explore might also encourage exploration and increase immersion in the game. Examples of such places might be:

- *Special locations*
Battle arenas or special zones with exceptional functionality that could give better rewards for winning a combat, or even reward more XP.
Players could also be given a temporary boost to combat skills after attending a lecture.
- *Shops*
To buy weapon, armor or magical potions in order to make your character stronger.
- *Non-Playable Characters (NPCs)*
NPCs can teach the players about something in their curriculum, or serve pointers to where one might find the answers.
- *Creeps*
Creeps that might serve as a computer controlled opponent when no other players are around, or that gives fast or easy questions for a possibility for the players to keep repeating a particular part of a course's curriculum.

Alternative game modes

The game can incorporate different game modes, which can be based on time of day, week, month or even year.

These might include:

- *Zombie Tuesday*
Who survives the longest when an educational zombie outbreak occurs? Educational zombies does not require a steady supply of fresh brains to survive, they consume knowledge. You have to run and hide, but if you get caught, do not answer quiz questions wrong or you will get infected.
- *Team based battle*
In this mode, players could team up to battle against each other. Team members' answers could be combined in order to create combo attacks. This mode could also include a last man standing mode where a wrong answer would eliminate one team member. The surviving team would be the winner.
- *Boss battles*
Players could team up or go alone to take down the professor for your university course. Players meeting a boss would see a more challenging and difficult battle than a regular battle against other students. The boss, in the form of either a virtual world boss or the very real professor would have small advantages that would make the battle more interesting.

Crafting mini-game

Mini-quiz game where the player has to collect pieces (for example metal or leather scraps) for use in crafting new armor pieces or weapons. To craft equipment, the player would have to seek out a special location, like a forge, where the actual crafting would be done. The player would initiate the crafting mini-game, in which he has to answer quiz questions correctly consecutively in order to achieve the best possible result.

Full armor set

The armor in the game could consist of more than just one piece. Where the player could mix and match different pieces to create a set that would match the style of playing. Many contemporary RPGs like World of Warcraft⁹⁰ and Diablo⁹¹ give players that equip a full set of armor pieces with the same name a set bonus. A similar type of bonus could be applied in KnowledgeWar 2.0 for players who have collected a set. The bonuses could apply to the character's attributes or to the quiz question answering time.

Character inventory

Players should be able to store items they found during their adventures on campus in their backpacks. The backpack could hold weapons, armor, potions and other items.

⁹⁰eu.battle.net/wow/en/

⁹¹<http://eu.battle.net/d3/en/>

17.2.2 Technical

Potential technical features that would make the game easier to administrate for lecturers, course staff and game administrators.

Question availability

All quiz questions are now available at the same time. A feature to make it easier for lecturers would be the possibility to add all questions at the start of the semester and defining when these questions should become available in accordance with how far in to the curriculum the lectures have come.

Server configuration software

A configuration panel to make it possible for game administrators to edit game rules or details, as well as adding additional quiz questions while the game is live. With this in place, there would be no need to shut down the whole system while initiating for example a new game mode. This panel would show players that are currently logged on, where they are on a map and which battles that are going on at the moment. During this thesis, we implemented an administrator client for use on a smart device similar to the regular client (see Section 12.1.7). It would be more manageable if it was usable from a computer with a larger screen. It would also have to support multiple administrators at the same time.

Lecture administrator panel

A dedicated administrator panel for lecturers and those responsible for the course in order to view automated statistics about the students taking their course. This panel would give the lecturers the possibility to see which part of the curriculum the students are doing well in, as well as seeing if there are parts of the curriculum that needs a more thorough review. It would also add the possibility to add quiz questions for the lecturer's course, in addition to defining when those questions should become available.

17.3 Future Dreams

This section presents our wildest dreams, where resources are unlimited and the possibilities are endless.

World domination

We would very much like to see this end up being used in every course of every year for universities around the world. There would be no problems finding opponents as every student would be a potential player. The game could even be something that was discussed during or be a part of lectures.

Gestures

Instead of using the map to challenge opponents and picking up treasures, we imagine that it could be possible to implement gestures to perform certain actions. Instead of pushing a button when wanting to attack the opponent in battle, the player could move the smartphone in a *slashing* manner to signal what type of attack that the player wishes to perform. Similarly, a player would have to wave the smartphone a certain way to be able to pick up a treasure or perform purchases at the local tavern.

Augmented reality

Instead of just having a interactive map, we envision the possibility to use augmented reality as a way to increase immersion. Using the smartphone camera we could transform the campus environment in to a real live battlezone and even make the players wear visible virtual weapons and armor. There could even be virtual world road-signs or message boards only viewable through the augmented reality. To take this one step further, it might even be possible to make use of Google's new Project Glass for easy accessible and wearable computing.

Near Field Communication (NFC)

NFC is an up-and-coming technology. The technology already exists, but is not used for anything in particular at the moment. In the future, it is set to be used for small transactions or the exchanging of business cards. NFC tags are physical stickers with information stored in them. Smartphones with NFC capabilities can read these stickers and do something according to the information on the NFC tag. This can be used to further enhance the immersion of a game by providing real world physical locations for players to visit.

Part V
Appendix

Appendix A

Acronyms

3G	3rd Generation of mobile telecommunications technology
ACM	Association for Computing Machinery
AOC	Age of Computers
AP	Access Point
API	Application Programming Interface
AVD	Android Virtual Device
CS	Computer Science
CSS	Cascading Style Sheets
CTF	Capture The Flag
D&D	Dungeons & Dragons
EDGE	Enhanced Data rates for GSM Evolution
ER	Entity-Relationship
FPS	First Person Shooter
FR	Functional Requirement
GB	Gigabyte
GPS	Global Positioning System
GUI	Graphical User Interface
HP	Health Points
HTML	HyperText Markup Language
ID	Identification Data
IDE	Integrated Development Environment
IDI	Department of Computer and Information Science (at NTNU)
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
JDBC	Java Database Connectivity
JQM	jQuery Mobile
JSON	JavaScript Object Notation
LQ	LectureQuiz
MMORPG	Massively Multiplayer Online Role-Playing Game
MP3	MPEG-2 Audio Layer III
MPEG	Moving Picture Experts Group
MVC	Model View Controller
NFC	Near Field Communication

NFR	Non-Functional Requirement
NOK	Norwegian krone
NPC	Non-Playable Character
NTNU	Norwegian University of Science and Technology
OS	Operating System
PC	Personal Computer
PDA	Portable Digital Assistant
PDF	Portable Document Format
PP	Paper Prototype
QID	Questionnaire statement Identifier
RDBMS	Relational Database Management System
RPG	Role-Playing Game
RTS	Real-Time Strategy
SD	Secure Digital
SDK	Software Development Kit
SE	Software Engineering
SID	System usability scale Identifier
SIM	First Person Shooter
SMS	Short Message Service
SP	Software Prototype
SQL	Structured Query Language
SUS	System Usability Scale
TBS	Turn Based Strategy
TCP	Transmission Control Protocol
TR	Technical Requirement
UI	User Interface
UiO	University of Oslo
USB	Universal Serial Bus
UUID	Universal Unique Identifier
W3C	World Wide Web Consortium
WP7/WP8	Windows Phone 7 / Windows Phone 8
XAML	Extensible Application Markup Language
XML	Extensible Markup Language
XP	Experience Points

Appendix B

Paper Prototype

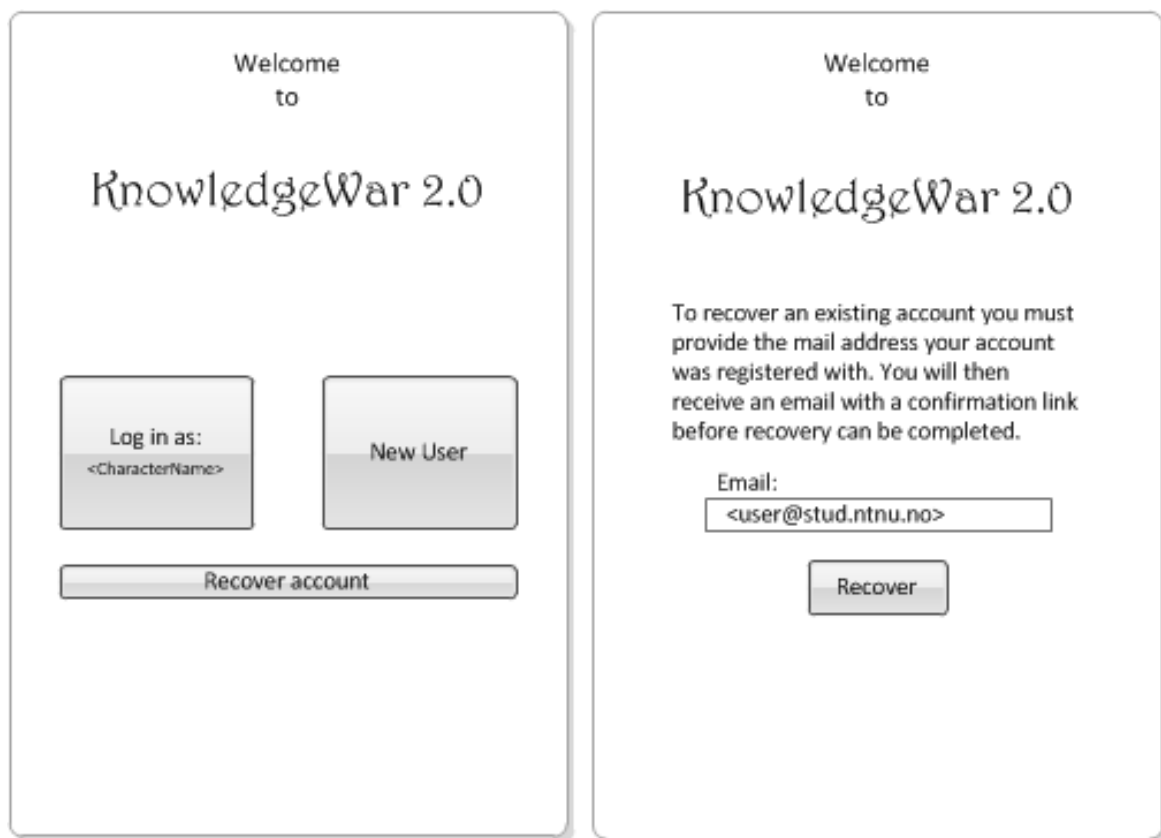


Figure B.1: Paper Prototype: Main menu and the recover account screens

The image shows two panels of a paper prototype for a registration process. Both panels are titled "Register new account".

Step 1/3: User Information:

- Character Name: *
- Email: *
- Courses:
- I have read and understood the terms and conditions of this game ([link](#))
- Buttons: "Quit" and "Next" with a right-pointing arrow.

Step 3/3: Place additional attributes:

Attributes left:

	Stamina	<input type="text" value="5"/>	<input type="text" value="+"/>
	Strength	<input type="text" value="20"/>	<input type="text" value="+"/>
	Intelligence	<input type="text" value="5"/>	<input type="text" value="+"/>
	Agility	<input type="text" value="10"/>	<input type="text" value="+"/>

Buttons: "Back" with a left-pointing arrow and "Complete" with a right-pointing arrow.

Figure B.2: Paper Prototype: Step 1 & 3 of the register new account screens



Figure B.3: Paper Prototype: Step 2 of the register new account screens



Figure B.4: Paper Prototype: The different pop-ups related to the registration process

highscore Back

Rank	Name	Level	W/L Ratio	L/W Ratio
1.	Navn 1	Level 26	20/5	4.0
2.	Navn 2	Level 24	18/5	3.6
3.	Navn 3	Level 23	17/5	3.4
4.	Navn 4	Level 22	16/5	3.2
5.	Navn 5	Level 20	17/6	2.8
6.	Navn 6	Level 19	16/6	2.6
7.	Navn 7	Level 18	12/5	2.4
8.	Navn 8	Level 17	10/5	2.0
9.	Navn 9	Level 15	6/5	1.2
10.	Navn 10	Level 13	10/12	0.8
15.	Ditt navn	Level 8	5/50	0.1

Figure B.5: Paper Prototype: The highscore table

The figure displays four character profile cards, each with a unique class and associated attributes. Each card includes a name field, class, level, experience bar, health bar, attributes table, equipment list, and navigation buttons for Highscores and World Map.

Character Profile 1: Computer Geek

Name: <YourCharacterName>
 Class: Computer Geek
 Level: 1
 Experience: 0/50
 Health: 150/150
 Attributes:
 Stamina: 5
 Strength: 10
 Intelligence: 5
 Agility: 20
 Equipment:
 Weapon: Dirty Sticky Keyboard
 Armor: All-Weather Jacket

Character Profile 2: Robot Geek

Name: <YourCharacterName>
 Class: Robot Geek
 Level: 1
 Experience: 0/50
 Health: 150/150
 Attributes:
 Stamina: 5
 Strength: 20
 Intelligence: 5
 Agility: 10
 Equipment:
 Weapon: Flathead Screwdriver
 Armor: All-Weather Jacket

Character Profile 3: Lab Geek

Name: <YourCharacterName>
 Class: Lab Geek
 Level: 1
 Experience: 0/50
 Health: 150/150
 Attributes:
 Stamina: 5
 Strength: 5
 Intelligence: 20
 Agility: 10
 Equipment:
 Weapon: Test Tube
 Armor: Labcoat

Character Profile 4: Math Geek

Name: <YourCharacterName>
 Class: Math Geek
 Level: 1
 Experience: 0/50
 Health: 150/150
 Attributes:
 Stamina: 5
 Strength: 10
 Intelligence: 20
 Agility: 5
 Equipment:
 Weapon: Pencil
 Armor: All-Weather Jacket

Figure B.6: Paper Prototype: The initial profiles of the four character classes

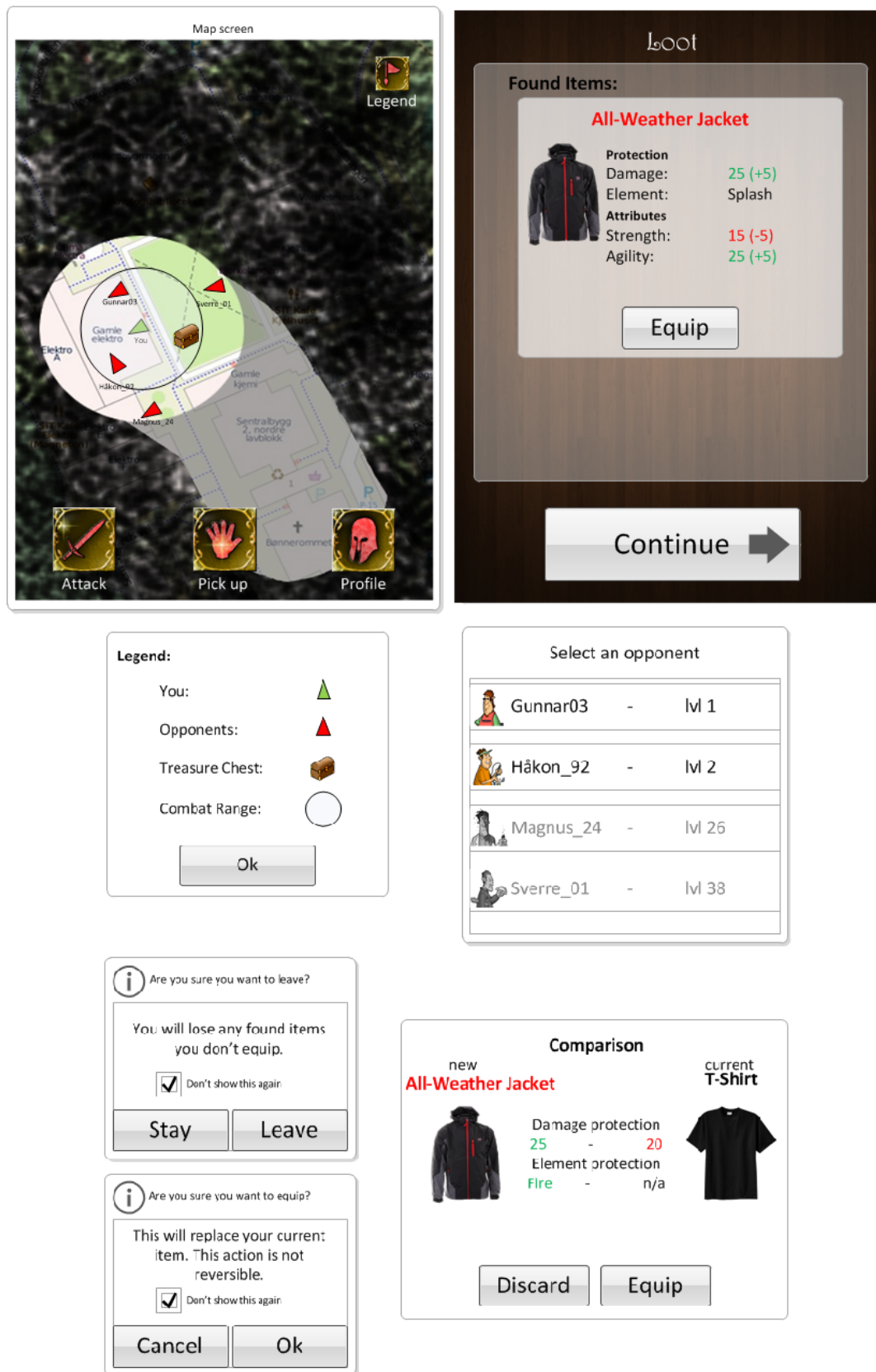


Figure B.7: Paper Prototype: The map screen, treasure chest pickup screen and the pop-up info-boxes related to them

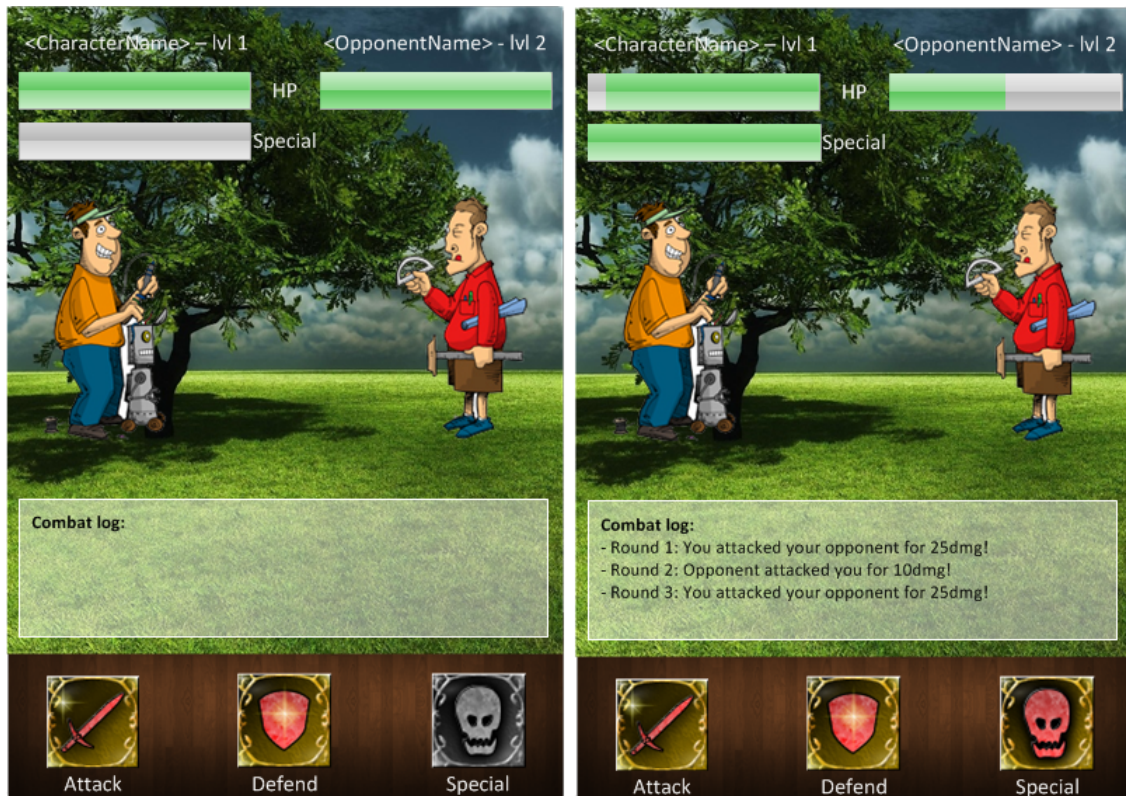


Figure B.8: Paper Prototype: Initial battle screen and battle screen after some rounds



Figure B.9: Paper Prototype: The quiz question screen when battling

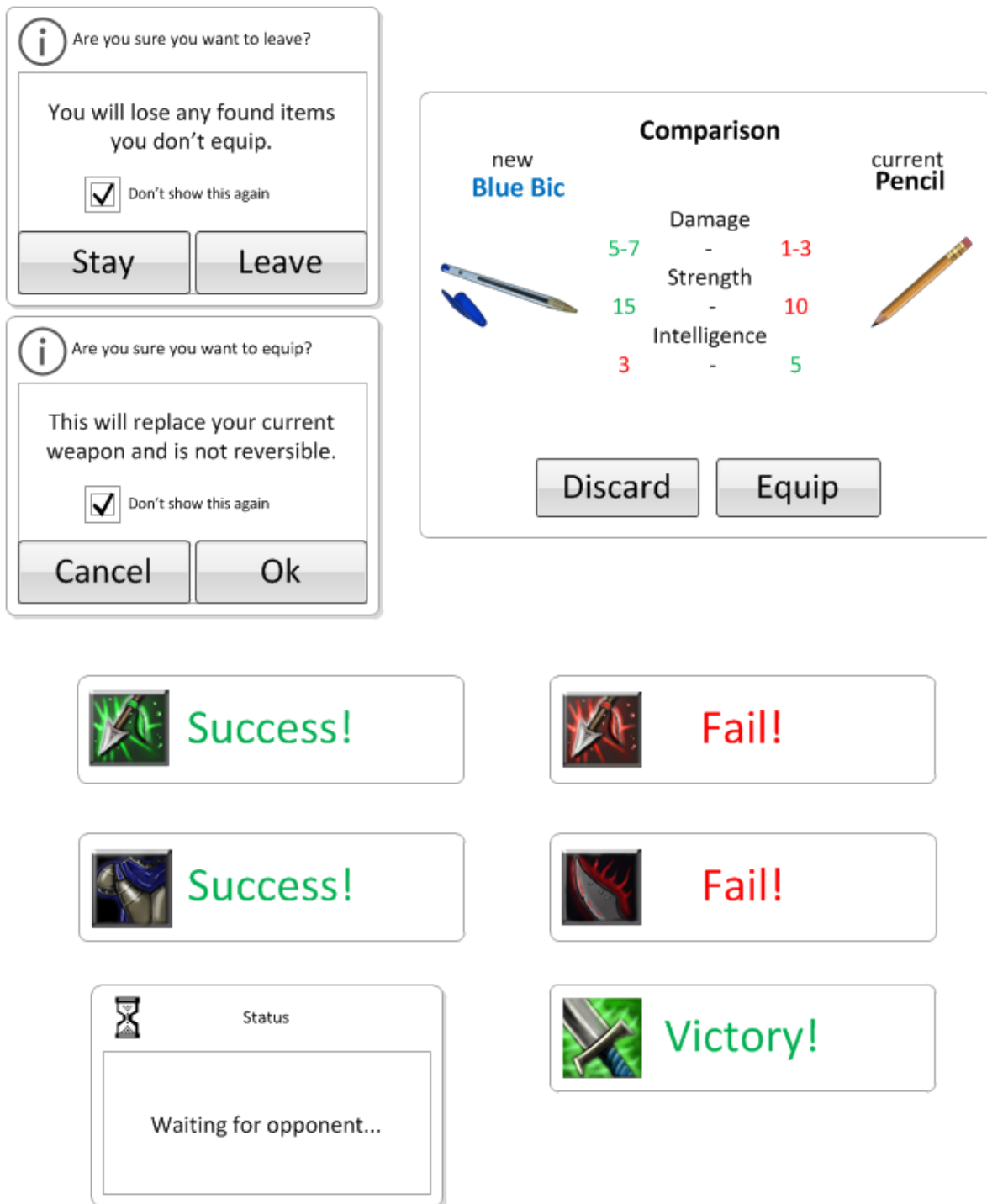


Figure B.10: Paper Prototype: The different pop-ups related to the battle screen



Figure B.11: Paper Prototype: The battle report, question review and battle reward screens

Appendix C

Scott Johnson's 56 Geeks



The 56 Geeks, by Scott Johnson
www.myextralife.com - 2007

Figure C.1: Scott Johnson's 56 Geeks

Appendix D

Weapons and Armor

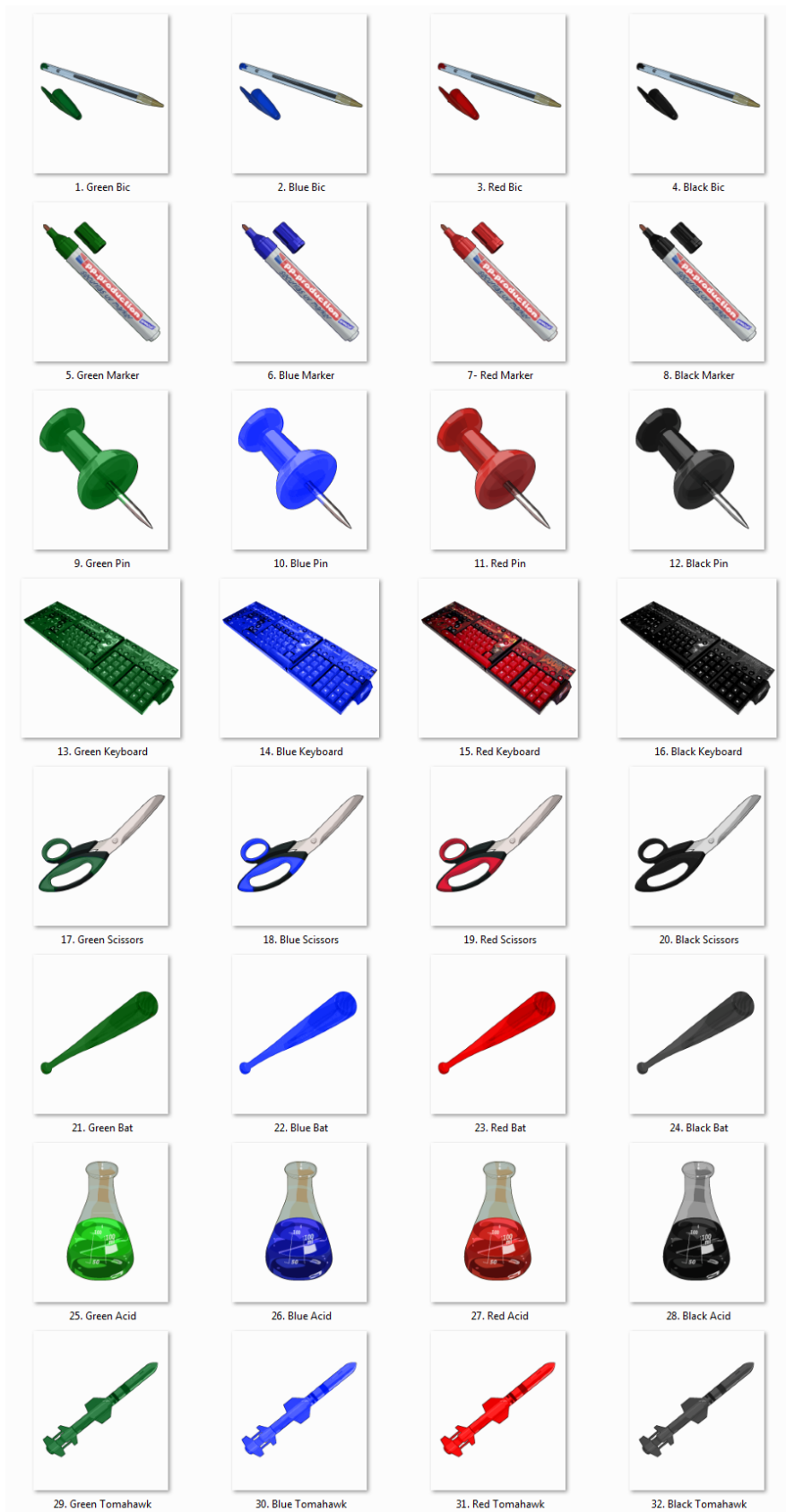


Figure D.1: KnowledgeWar 2.0 Weapons



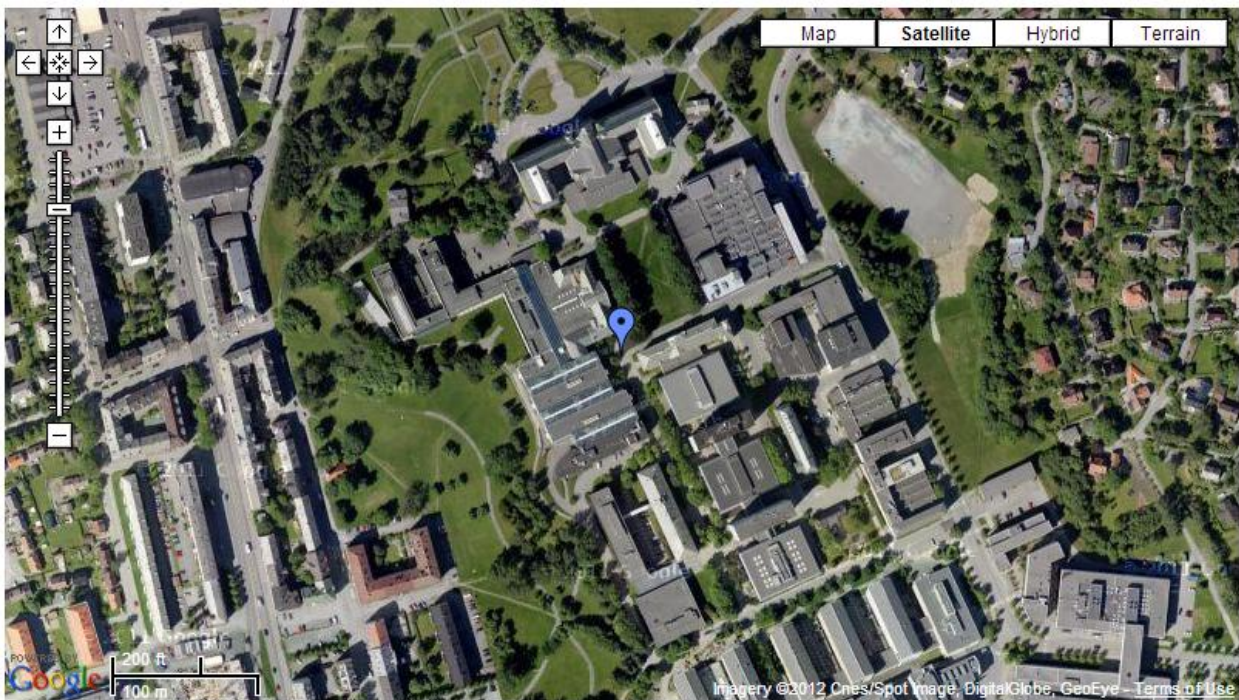
Figure D.2: KnowledgeWar 2.0 Armor

Appendix E

Experiment Instructions

Instructions

1. Launch app and register an account
2. Log out and close app
3. Turn on GPS
4. Turn off WiFi
5. Walk to: Entrance of EI-bygget



6. Launch the game and start playing!

PS. If you want to play indoors turn on WiFi after entering building (remember to turn off WiFi when exiting the building).

Appendix F

KnowledgeWar 2.0 Questionnaire

KnowledgeWar 2.0 Questionnaire

About you

1. Age: _____
2. Gender: female male
3. Program of study: _____
4. My level of skill with role-play games is...:
- | | | | | |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Novice | Beginner | Intermediate | Professional | Expert |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
5. My level of skill with a smartphone is...:
- | | | | | |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Novice | Beginner | Intermediate | Professional | Expert |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

About your device

6. a. Brand: _____ b. Model: _____
7. What connection type did you use?: WiFi 3G GPRS/EDGE
8. Did you have GPS activated?: Yes No

Learning

- | | Strongly disagree | Disagree | Neutral | Agree | Strongly agree |
|--|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 9. My knowledge from NTNU courses came to use | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 10. Answering questions was stressful | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 11. The game stimulated me to learn more | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 12. The game makes course theory more fun | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 13. I would like to take courses that use a game like KnowledgeWar 2.0 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 14. I was motivated to learn more in order to do better in the game | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 15. I reviewed my answers after a battle | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Gameplay

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
16. The game was fun	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17. I would like to play this game even if I'm not required to	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18. I got relevant feedback on how to play the game	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19. I was motivated to move around and discover	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20. The game was not engaging	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
21. The game was hard to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
22. The game would be more enjoyable with more gameplay elements	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Technical

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
23. I liked the look-and-feel of the game	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
24. The game had poor performance	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
25. I would like a more graphically advanced game	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
26. Technical difficulties distracted me from the game	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
27. My position was correct...:					
Never	Once or twice	25% of the time	Half of the time	Throughout the game	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
28. I experienced delay...:					
Never	Once or twice	25% of the time	Half of the time	Throughout the game	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Social & Mobility

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
29. It was fun to battle against fellow students	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
30. I would rather play against a computer controlled opponent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
31. I liked the fact that I have to move around to play the game	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
32. I would rather play a traditional non-location specific game	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
33. There were too few elements to interact with	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

34. Playing against fellow students made me concentrate more
35. I do not like to lose against fellow students

Experiment

- | | 0 | 1-2 | 3-5 | 6-9 | >9 |
|--|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 36. How many times did the client crash? | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 37. Did any other/specific problems occur? | | | | | |

38. Further comments? / Thoughts on the game? / Ideas for things to add to the game?

Appendix G

System Usability Scale

System Usability Scale

	Strongly disagree								Strongly agree
1. I think that I would like to use this system frequently	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	1	2	3	4	5				
2. I found the system unnecessarily complex	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	1	2	3	4	5				
3. I thought the system was easy to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	1	2	3	4	5				
4. I think that I would need the support of a technical person to be able to use this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	1	2	3	4	5				
5. I found the various functions in this system were well integrated	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	1	2	3	4	5				
6. I thought there was too much inconsistency in this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	1	2	3	4	5				
7. I would imagine that most people would learn to use this system very quickly	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	1	2	3	4	5				
8. I found the system very cumbersome to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	1	2	3	4	5				
9. I felt very confident using the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	1	2	3	4	5				
10. I needed to learn a lot of things before I could get going with this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
	1	2	3	4	5				

Appendix H

Developer's Guide

H.1 Client/AdminClient Development Setup

In order to examine the source code or continue development of the client and/or admin client software, it is recommended to have a look at the guidelines presented in this section.

H.1.1 Project Setup

1. Install the Android SDK and an IDE with Android development support (recommended Eclipse with ADT plugin).
 - (a) The ADT Bundle⁹² contain everything needed to start Android development. The provided URL also contain additional guidelines for various alternatives for making your computer ready for Android development.
2. If your IDE supports the Eclipse project format the Client.zip should be possible to be directly imported into your workspace. For the Eclipse IDE do the following:
 - (a) Go to File ->Import. A dialog will appear.
 - (b) Select General ->Existing Projects into Workspace.
 - (c) Click the radio button next to "Select archive file", click the Browse button and use the following dialog to locate and select the Client.zip
 - (d) The project name will appear in the box below, already checked. Click Finish to perform the import

If your IDE has been correctly setup with the Android SDK the project is ready for deployment to either the Android Emulator or a connected Android device. All HTML, JavaScript and CSS source code is located in the folder "assets\www" in the project.

⁹²<http://developer.android.com/sdk/index.html> (28.11.12)

H.1.2 Framework Updates

For further development of the client or admin client software, it is recommended to look for updates for the following used frameworks:

- PhoneGap⁹³
- jQuery Mobile⁹⁴
- Leaflet⁹⁵

Additionally it be a good idea to look into an alternative WebSocket implementation as the currently used implementation⁹⁶ might no longer be up to date.

H.1.3 Other Platforms

If the client and/or admin client is to be adapted for a different platform the following steps might be useful:

1. Setup your computer for PhoneGap development for the selected platform⁹⁷.
2. From the Client.zip project archive you are interested in the: “Client\assets\www” folder, to copy it into your new PhoneGap project. The folder contains the index.html together with all the CSS, JavaScript, frameworks, images and audio which together make up the client software.
3. Find a suitable native WebSocket implementation for the selected platform with a JavaScript API and replace the current WebSocket implementation.

H.2 Server Development Setup

H.2.1 Project Setup

1. Install JDK 1.6 (or higher)⁹⁸.
2. Install the Eclipse IDE⁹⁹.
3. Import jWebSocketServer.zip directly to your workspace:
 - (a) Go to File ->Import. A dialog will appear.
 - (b) Select General ->Existing Projects into Workspace.
 - (c) Click the radio button next to “Select archive file”, click the Browse button and use the following dialog to locate and select the jWebSocketServer.zip
 - (d) The project name will appear in the box below, already checked. Click Finish to perform the import

⁹³<http://docs.phonegap.com/> (28.11.12)

⁹⁴<http://jquerymobile.com/download/> (28.11.12)

⁹⁵<http://leafletjs.com/download.html> (28.11.12)

⁹⁶<https://github.com/anismiles/websocket-android-phonegap> (28.11.12)

⁹⁷http://docs.phonegap.com/en/2.2.0/guide_getting-started_index.md.html (28.11.12)

⁹⁸<http://www.oracle.com/technetwork/java/javase/downloads/index.html> (28.11.2012)

⁹⁹<http://www.eclipse.org/>(28.11.2012)

H.2.2 Wireless Trondheim

1. Visit Wireless Trondheim
2. Tell them you need access to GeoPos
3. Wireless Trondheim will give you a username, password and a file called “gpos-SpinOff.keystore”. Define these in the GeoPosConnection.java.

H.2.3 Running the Server

1. If it is the first time the server will be run, the variable “createTables” in the DbConnection.java class needs to be set to “true”. This will set up the database to be ready for use. If this is not the first time the server will be run, the variable will have to be set to “false”.
2. Right click on the project and choose “Run as” -> “Java Application”.
3. If prompted to choose a specific Java Application, find “JWebSocketServer - org.jwebsocket.console” in the list. Click “OK”.

Bibliography

- [1] Leigh Achterbosch, Robyn Pierce, and Gregory Simmons. Massively multiplayer online role-playing games: the past, present, and future. *Comput. Entertain.*, 5:9:1–9:33, March 2008.
- [2] Thomas H. Apperley. Genre and game studies: toward a critical approach to video game genres. *Simul. Gaming*, 37:6–23, March 2006.
- [3] Victor R. Basili. The experimental paradigm in software engineering. In *Proceedings of the International Workshop on Experimental Software Engineering Issues: Critical Assessment and Future Directions*, pages 3–12, London, UK, UK, 1993. Springer-Verlag.
- [4] J. Brooke. Sus - a quick and dirty usability scale. *Usability Evaluation in Industry*, 1996.
- [5] Microsoft News Center. Nokia and microsoft announce plans for a broad strategic partnership to build a new global mobile ecosystem. <http://www.microsoft.com/presspass/press/2011/feb11/02-11partnership.msp>. Last Accessed: 31.05.12.
- [6] Adrian David Cheok, Anuroop Sreekumar, Cao Lei, and Le Nam Thang. Capture the flag: Mixed-reality social gaming with smart phones. *IEEE Pervasive Computing*, 5:62–69, 2006.
- [7] M. Csikszentmihályi. *Flow: The Psychology of Optimal Experience*. P. S. Series. HarperCollins, 2008.
- [8] James Paul Gee. *Good video games and good learning*. Peter Lang, 2007.
- [9] Jacob Habgood and Mark Overmars. Game design: Balance in multiplayer games. In *The Game Maker's Apprentice*, pages 211–222. Apress, 2006.
- [10] IDC. Android marks fourth anniversary since launch with 75.0in third quarter, according to idc. <http://www.idc.com/getdoc.jsp?containerId=prUS23771812>. Last Accessed: 06.11.12.
- [11] Gartner Inc. Gartner says android to command nearly half of worldwide smart-phone operating system market by year-end 2012. <http://www.gartner.com/it/page.jsp?id=1622614>. Last Accessed: 31.05.12.

-
- [12] Gartner Inc. Gartner says apple ios to dominate the media tablet market through 2015, owning more than half of it for the next three years. <http://www.gartner.com/it/page.jsp?id=1626414>. Last Accessed: 30.05.12.
- [13] Gartner Inc. Gartner says sales of mobile devices grew 5.6 percent in third quarter of 2011; smartphone sales increased 42 percent. <http://www.gartner.com/it/page.jsp?id=1848514>. Last Accessed: 30.05.12.
- [14] Kalle Jegers and Mikael Wiberg. Pervasive gaming in the everyday world. *IEEE Pervasive Computing*, 2006.
- [15] Sveinung Kval Bakken. KnowledgeWar. Master's thesis, Norwegian University of Science and Technology, 2010.
- [16] Henry Lee and Eugene Chuvyrov. *Beginning Windows Phone 7 Development*. Apress, Berkely, CA, USA, 2nd edition, 2011.
- [17] M.J. McAlister and Peng Hui Xie. Using a pda for mobile learning. In *Wireless and Mobile Technologies in Education, 2005. WMTE 2005. IEEE International Workshop on*, page 3 pp., nov. 2005.
- [18] Lasse Natvig, Steinar Line, and Asbjørn Djupdal. Age of computers: An innovative combination of history and computer game elements for teaching computer fundamentals. In *FIE 2004: Proceedings of the 2004 Frontiers in Education Conference*, 2004.
- [19] B.J. Oates. *Researching information systems and computing*. SAGE, 2006.
- [20] Brad Paras and Jim Bizzocchi. *Game, motivation, and effective learning: An integrated model for educational game design*, volume 33, pages 1–7. Citeseer, 2005.
- [21] S. J Shannon. Why don't students attend lectures and what can be done about it through using ipod nanos? *Who*, 2006. Refereed paper. Includes bibliographical references.
- [22] A.I. Wang, Bian Wu, and S.K. Bakken. Experiences from implementing a face-to-face educational game for iphone/ipod touch. In *Games Innovations Conference (ICE-GIC), 2010 International IEEE Consumer Electronics Society's*, pages 1–8, dec. 2010.
- [23] Alf Inge Wang, Terje Øfsdahl, and Ole Kristian Mørch-Storstein. An evaluation of a mobile game concept for lectures. In *Proceedings of the 2008 21st Conference on Software Engineering Education and Training - Volume 00: IEEE Computer Society*, 2008.
- [24] Alf Inge Wang, Terje Øfsdahl, and Ole Kristian Mørch-Storstein. Lecture quiz - a mobile game concept for lectures. In *11th IASTED International Conference on Software Engineering and Application (SEA 2007)*, 2008.

- [25] Alf Inge Wang and Bian Wu. Using game development to teach software architecture. *Int. J. Comput. Games Technol.*, 2011:4:4–4:4, January 2011.
- [26] Bian Wu, Alf Inge Wang, and Yuanyuan Zhang. Experiences from implementing an educational mmorpg. In *Games Innovations Conference (ICE-GIC), 2010 International IEEE Consumer Electronics Society's*, pages 1 –8, dec. 2010.