

Abstract

Educational games have recently caught the attention of educational organizations witnessing newfound potential that is not achievable through traditional lectures. By reviewing findings from authoritative theory, we present the conception and implementation of a prototype educational game for lecture enhancement.

The concept is based on the idea of playing a game during lectures, with students answering multiple choice questions using their own mobile phones and receiving instant feedback by watching a large screen displaying animated graphics. It is shown how such a concept is made readily available for students and schools by using regular mobile phones and computers they already possess.

We describe an example implementation, along with pedagogical guidelines for usage, and the analysis of how the prototype was received in an authentic setting. Students generally found the prototype easy to use and thought it contributed to increased learning outcome. The prototype was perceived as entertaining, and half the students claimed they would attend more lectures if such a system was being used. In spite of this, 10% of the students felt reluctant to pay for the GPRS/3G data transmission fees resulting from playing the game.

Preface

This report is a Master's Thesis at the Department of Computer and Information Science (IDI) at the Norwegian University of Science and Technology (NTNU). It is part of the Software Engineering group's focus on Computer Games, and is a continued work from a student depth project called Lecture Games [1] written in 2006.

We would like to express our gratitude towards our supervisor Alf Inge Wang for his enthusiasm and helpful guidance. We would also like to thank Martin Jarrett and Eivind Sorteberg for their willingness to exchange ideas and experiences from their projects in game development on portable devices.

The students of ITS-015 Tulipan study room at NTNU, as well as the Master of Science in Computer Science classes of 2007 and 2008 deserve credit for eagerly participating in testing the prototype.

Terje Øfsdahl

Ole Kristian Mørch-Storstein

Table of contents

Abstract	i
Preface	iii
Part I ..1	
Introduction	1
1.1 Project Definition.....	3
1.2 Project Context.....	3
1.3 Composition.....	4
2 Research Method.....	5
3 Software Development Method.....	8
4 Research Problems	10
Part II.13	
Findings in the Depth Project	13
5 Theoretical Foundation	15
5.1 Making Games Intriguing	15
5.1.1 Goals	15
5.1.2 Instructional Control.....	16
5.1.3 Intrinsic and Extrinsic Fantasies	16
5.1.4 Triggining Curiosity	18
5.2 Technology and Trends.....	18
5.3 Collaborative Gaming.....	20
6 Depth project findings.....	20
6.1 Characteristics of Good Educational Games.....	21
6.1.1 Characteristics.....	21
6.2 Collaborative Learning.....	23
6.2.1 Fighting the Intrinsic Fantasy Gap.....	23
6.2.2 Extended Malone Model.....	23
6.3 Taxonomy of Educational Games.....	25
6.3.1 Our taxonomy in practice	28
6.4 Genre Relevance Comparison	28
6.4.1 Theoretical Learning	28
Part III	31
State of the Art.....	31
7 Existing Classroom Applications	33
7.1 TVREMOTE Framework.....	33
7.2 Classroom Presenter	34
7.3 WIL/MA	35
7.4 ClassInHand	36
7.5 Ez ClickPro	37
7.6 Buzz! The Schools Quiz.....	39
8 Evaluation of Previous Solutions.....	40
8.1 Prototype Concept Comparison	41
Part IV43	
The Prototype	43
9 Concept	45
9.1 Concept Foundation.....	45

9.2	Concept Presentation	46
9.3	Desired Characteristics of the Prototype.....	48
9.4	User Guidelines	49
10	Technology Rationale	50
10.1	Server technology platform.....	51
10.1.1	Java.....	51
10.1.2	.NET	52
10.1.3	Choice Rationale	52
10.2	Mobile Technology Platform.....	53
10.2.1	Java 2 Platform Micro Edition (J2ME)	53
10.2.2	Microsoft .NET Compact Framework	54
10.2.3	Mobile Platform Selection Rationale.....	54
10.3	Protocols.....	54
10.3.1	Transmission Control Protocol (TCP)	54
10.3.2	User Datagram Protocol (UDP)	55
10.3.3	Protocol Selection Rationale.....	55
10.4	Communication bearers	55
10.4.1	Ethernet over twisted pair	56
10.4.2	GPRS, EDGE and 3G.....	56
10.4.3	Wi-Fi (IEEE 802.11 a,b,g,n)	57
10.4.4	Bluetooth.....	57
10.4.5	Data bearer selection rationale	58
10.5	Graphics Library.....	59
10.5.1	OpenGL.....	59
10.5.2	Microsoft DirectX.....	59
10.5.3	Graphical Library Selection Rationale.....	60
10.6	Database solution	60
10.6.1	MySQL.....	60
10.6.2	Oracle.....	61
10.6.3	Other Database Solutions.....	61
10.6.4	Database Selection Rationale	61
11	Prototype Design and Architecture	62
11.1	Architecture	62
11.1.1	Inter Process Communication.....	63
11.2	Design and Implementation.....	65
11.2.1	Server.....	65
11.2.2	Mobile Client.....	66
11.2.3	Web Client	67
11.2.4	Teacher Client	70
Part V.77		
The Experiment.....		77
12	Introduction	79
13	Experiment Delimitation	79
14	Method.....	79
14.1	System Usability Scale (SUS)	79
14.2	CIF.....	80
14.3	Subjective Assessment	80

15	Context.....	80
15.1	Participants & Environment.....	81
15.2	User Tasks	81
15.3	Success Criteria	83
16	Experiment Execution.....	83
17	Statistical Results	84
18	Evaluation.....	91
18.1	Feedback Evaluation.....	93
19	Experiment Conclusion.....	94
Part VI		97
Evaluation.....		97
20	Evaluation.....	99
20.1	Research method.....	99
20.2	Development method	100
21	Lessons Learned	101
21.1	GPRS/3G Synchronization optimization	101
21.2	Testing a Distributed System	105
21.3	Last minute changes.....	107
Part VII.....		109
Conclusion		109
22	Conclusion	111
23	Further Work.....	113
23.1	New game modes	113
23.2	Acceptance amongst teachers and institutions.....	114
23.3	Question editor.....	114
23.4	Simultaneous lectures support	114
Appendix.....		117
A.1 Questionnaire.....		119
References		123

Part I

Introduction

The concept of a lecture at educational institutions, as we know it today, dates back to the late 11th century. Teachers at a medieval university would read the original source out loud, and the students would take notes for further distribution or self-study. Ten centuries later, the situation is much the same, though new activities such as chalk-board writing, exercises and student teacher interaction is incorporated as a natural part of the lecture. Digital slides, combined with large screens and microphones have facilitated the process of holding and participating in a lecture, however, the format of communication has remained substantially unchanged for centuries. The traditional lecture has been accused of being both dull and ineffective [2].

This master thesis will experiment with the traditional lecture by introducing computer games and wireless technology into the classroom. The outcome will be evaluated and we hope this can contribute to future research in the art of lecturing.

1.1 Project Definition

This master thesis will explore and develop a concept for games to be used during lectures in higher education. The purpose is to investigate the potential of games as a means to increase the level of learning as well as provide a stimulating environment for the students. Our thesis will describe the implementation of a prototype game for enhancing lectures. A rationale for the choices being made will be founded in theory and a realistic experiment will be conducted in order to give an evaluation of the prototype based on empirical investigation.

1.2 Project Context

This project has been conceived as part of the increased focus of the Department of Computer and Information Science at the Norwegian University of Science and Technology on computer games as a research area. The department has already successfully implemented an educational game

as part of a master-level course in computer hardware [3]. The department has also issued a number of other master theses and three new PhD-level research topics as part of its new research program in Computer Games. In addition to more academic research in educational games, we are also seeing an abundance of commercially available products intended for use in schools.

Under the supervision of associate professor Alf Inge Wang, we (Terje Øfsdahl and Ole Kristian Mørch-Storstein) will survey the theoretical foundation for an educational game to be used in a classroom setting. Based on this research, we will create a prototype to be used in an experiment in a real classroom situation.

1.3 Composition

This document is divided into seven parts. The first gives an introduction to the project and illuminates the motivation of the project. Part II, Findings in the Depth Project, investigates the philosophical and psychological research done in the field of computer aided instruction, and sums up the results from the depth project this master thesis is built upon [1]. Part III, sums up the available solutions for lecture enhancement in a state of the art study. In Part IV the prototype is described in detail along with technological, architectural and design decisions. Part V describes the execution of the main experiment, where the prototype is tested in an authentic setting, along with a discussion of the experiment results. An evaluation of development and research method is presented in Part VI, elaborating on different problems, issues and positive experiences. In part VII, conclusions are drawn across the previous parts of the document, and further work is outlined.

2 Research Method

When performing research with the aim of conducting experiments later on, it is imperative to have a sound research method. Basili outlines the three most frequently used research methods in software engineering [4]:

The empirical method: A statistical method is used to verify a hypothesis through collection of empirical data. This method can e.g. be used to determine if a new technology is an improvement over older technologies [5].

The mathematical method: this method is based on mathematical and formal methods. A formal theory has to be constructed and results are in turn derived from this theory. Empirical data can be used for comparison with the derived results.

The engineering method: The engineering method is *"the use of heuristics to cause the best change in a poorly understood situation within the available resources"* ([6] p. 28) and fits our study in its ability to deal with systems of major complexity by consequently finding improvements from self conducted validation methods.

Our research is of both theoretical and creative nature, thus two validation methods will accordingly be applied as a part of the engineering method. After defining our research problems, the problems of theoretical nature will be subject to validation. As a validation tool we include *historical methods* [7] and thus give grounds for our assertions by studying previous work. When previous literature is reproduced or summarized, or when our findings build directly on previous work; internal validation is preserved. Likewise, some of our findings are merely a presentation of theory that is commonly agreed upon among relevant researchers, and rely on the respective validation upon which these assertions are based. Some of our findings are results of our

subjective assessments, indistinctly based on our literature study, and are not subject to validation. When such findings are presented, the lack of formal integrity is explicitly expressed.

Based upon the theoretical validation, we will design and develop a prototype application and a questionnaire to collect empirical data. The prototype will subject to a **controlled test**. The controlled method is based on experiments providing input for the validation [5].

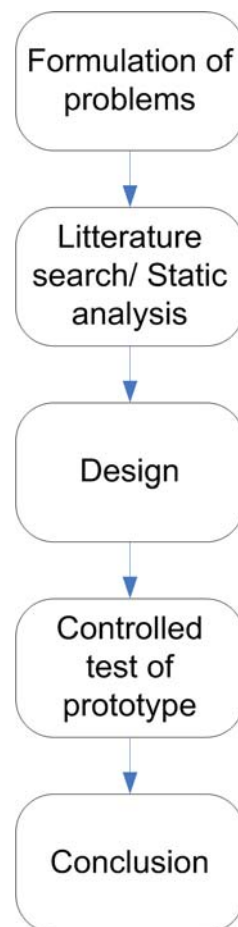


Figure 1: Method of Research

As illustrated in Figure 1, the initial task is a clear formulation of the research problems. By studying authoritative literature in psychology, computer game theory as well as available products, heuristics toward an improved solution is discovered and established in a literature search. Based on these heuristics,

the design of a prototype is formed and implemented. As a preparation for the empirical data collection, the measurable points of interest are decided upon resulting in a questionnaire. Empirical data is provided from a controlled simulation experiment. Finally a conclusion is drawn across the literature search and the experiment phases.

3 Software Development Method

We are basing the software development method in this project on the iterative, as shown in Figure 2, and agile software development processes [8]. The classic iterative process was meant for use in long term projects, where each iteration could take from weeks to months.

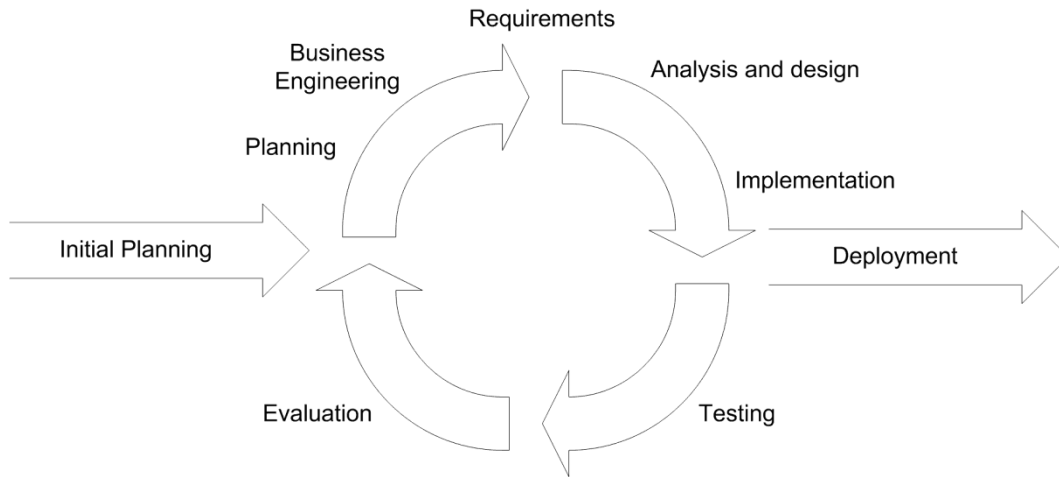


Figure 2: Traditional Iterative Development [8]

In this project, we will have limited time available for development, thus each iteration will take at most two weeks. As a result, we have identified the need for a customized software development process, as shown in Figure 3, based on the agile version of iterative development[8] and adapted to the needs in this project. The planning, requirements, analysis and design phases are unchanged in regard to Figure 2, but the implementation step has been changed. We have chosen to modify the iterative process with a special prototyping step where we will prioritize implementation and testing of the features in the backlog. We will then evaluate the stability and functionality of the prototype. Such an evaluation often influence which features to implement next, and it can also introduce the need for additional features. The remaining backlog of features will be updated and reprioritized if necessary, and another iteration will begin. This ensures an adaptive process

with focus on quickly producing a functional prototype with basic functionality, and also ensures that the project stays on schedule.

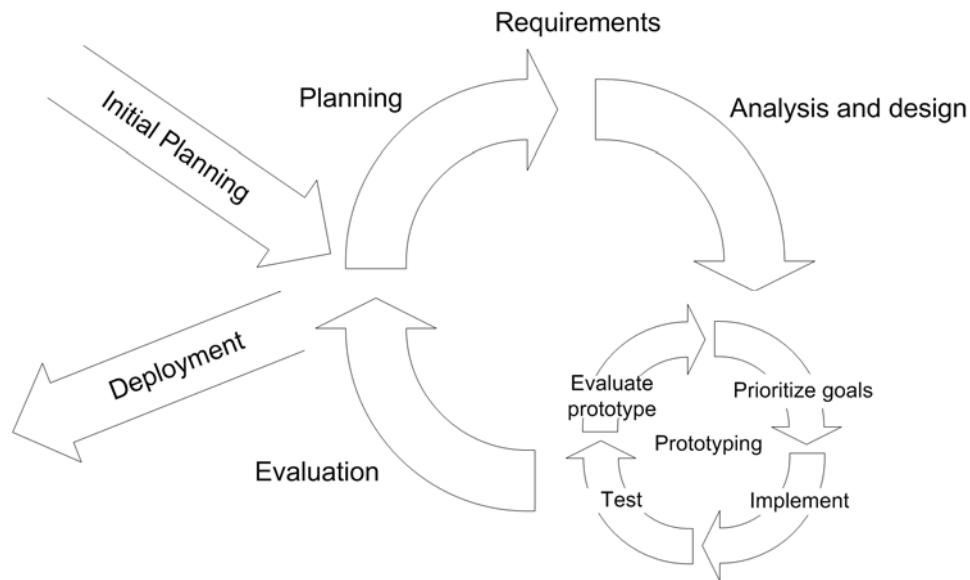


Figure 3: Our iterative development method

The prototyping step also allows us to perform continuous testing of the software by testing each method and each module after a change is done, and makes it possible to correct any unwanted behavior dynamically. After the prototyping step is completed, i.e. the requirements has been met, an evaluation will be executed. This step empowers us to make a decision to whether or not the requirements have to be reevaluated, or if the prototype is ready for deployment.

4 Research Problems

Based on the project definition in Chapter 1.1, we evaluated the critical areas which we sought to center our research around. These areas have been crafted as *research problems* after careful deliberation and evaluation of what we needed to research, and also of what we could contribute to this field. This report will answer the following research problems:

RP1: What technical and pedagogical considerations have to be made when making a lecture game?

With a foundation in established theory, we will present the elements which have to be present to make an effective and intriguing educational game.

RP2: Which technologies are suitable for game enhanced lectures?

Which existing and available technologies can be used to support a lecture enhancement game?

RP3: Implement a fully working prototype of a lecture game.

We will use our findings from RP1 and RP2 to implement a prototype which will be used to evaluate the fitness of lecture games. Key architectural decisions and solutions will be described in some detail.

RP4: What benefits can be observed from the usage of game enhanced lectures?

RP5: What disadvantages may result from the usage of game enhanced lectures?

Our prototype will be used to collect empirical data from a realistic student experiment answering RP4 and RP5.

Answers to the preceding research questions will be presented in the following chapters.

Part II

Findings in the Depth Project

This Part is divided in two chapters. Both chapters present the essential parts of our depth project upon which this Master Thesis is built. For a more thorough presentation of the theory, the reader is referred to our depth project [1]. Chapter 5 presents the underlying theory of educational games: Why are games so captivating, and what makes games particularly suited for educational purposes? In Chapter 6 we present the parts of the result from our depth project which are relevant to the research problems in Chapter 4. This includes a list of desirable characteristics and important features of educational games, as well as a model for how social interaction can make games intriguing. We also briefly present our suggested taxonomy of educational games and discuss shortly how these genres are relevant to theoretical or practical learning.

5 Theoretical Foundation

In this chapter we look at the aspects of computer games that make them captivating and interesting to use.

5.1 Making Games Intriguing

Enjoyable situations, or rather the characteristics of enjoyable activities, can be divided into three categories: challenge, fantasy and curiosity [9]. These are properties of games that are enjoyable and captivating.

5.1.1 Goals

T.W. Malone, a leading authority within the research field of computer games, defines a classification of *intrinsic motivation* [9], i.e. the driving force that makes people play games over and over. Games that lack proper goals are less likely to be challenging over time. Once the novelty factor is over, the game might be forgotten. Where there is no objective, there is also a lack of the desire to finish the unfinished. Goals can be practical, based on a fantasy or the goal of using a specific skill. Malone recommends the use of practical

and fantasy based goals, rather than the goal of just using a skill. An example of this can be the goal of accumulating a certain score versus the goal of the player using her arithmetic skills [1].

5.1.2 Instructional Control

As the player gets more training, and possibly becomes better at the game, the difficulty increases. Thus the player will ideally have a constant feeling of mastering, and will have the same drive towards improving herself. Malone strongly emphasizes that self-esteem plays a key role in relation to a player's experience of a computer game. With success, self-esteem can increase, and likewise failure can cause a decrease in self-esteem. In addition a severe discrepancy between the players' skills and the required skill of the game can cause the player to lose interest in the game. I.e. if it is too easy there will be no motivation for the player to continue playing, and if it is too difficult, then the player will lose self-esteem and be discouraged. This implies that there exists a "sweet spot" unique to every player, where the game is challenging enough so that the player stays interested, but not to the extent that that she thinks she will be unable to complete it.

5.1.3 Intrinsic and Extrinsic Fantasies

An aspect of computer games is the use of fantasies and abstractions to enhance and make them more interesting to the participants. Malone introduces two different ways of using fantasies in games: *extrinsic*¹ and *intrinsic*². Extrinsic fantasies are fantasies that only depend on the use of a set of skills, but the skills are not influenced by the fantasy. The idea is illustrated in Figure 4. An example of extrinsic fantasy is hangman where you save a crudely drawn man from being hung by guessing a word. The player does not

¹ Extrinsic fantasies are fantasies that are independent of the application of skills in the game.

² Intrinsic fantasies are fantasies that are inherent and essential to the game concept.

have to take into consideration any factors of the fantasy when using her skill.

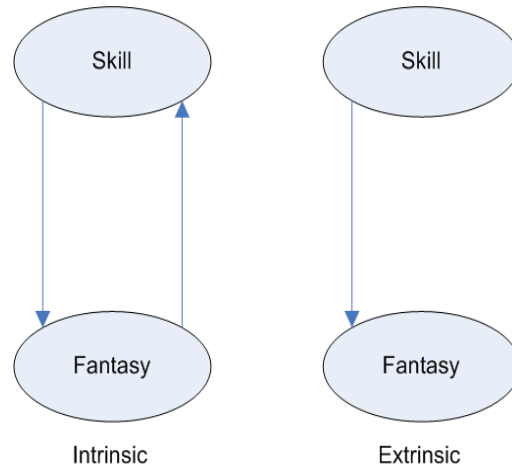


Figure 4: Intrinsic and Extrinsic fantasy in games

Intrinsic fantasies on the other hand have skill and fantasy influencing one another. A flight simulator is an example of an intrinsic fantasy where the fantasy is a direct, albeit simplified, version of reality. Our depth project gives a thorough explanation of these concepts [1].

Malone writes that he generally thinks intrinsic fantasies are more interesting and educational than extrinsic fantasies. He argues that intrinsic fantasies can suggest how a given skill can be applied in real life. Malone suggests that a close relation between the game itself and the material being learned will give players the opportunity to draw from previous experience in the real world. In addition the player can use information from the fantasy to improve her skill directly, without any intermediate steps, like e.g. consulting a text book.

5.1.4 Triggering Curiosity

Motivation to learn and to investigate is called curiosity. Malone distinguishes between two types: *sensory curiosity* and *cognitive curiosity*. A change in patterns, sounds and other stimuli that attracts attention is called sensory curiosity. Audiovisual effects are commonly used in television shows and movies. The usage spans from pure decorative purposes, to enhance a fantasy, to reward the player and finally as a way of conveying information. Cognitive curiosity is, according to Malone, people's desire to bring completeness, consistency and parsimony to their knowledge. He gives us the example that if you read all the chapters but one of a murder mystery, then you will have the desire to read the last chapter, thus giving your knowledge completeness. The importance of feedback cannot be underestimated; indeed it is imperative to the success of a computer game. Feedback should be surprising and constructive. This implies that the feedback should not just underline that a player lacks knowledge or skill, but e.g. rather ways of her to improve her skills. On top of this, surprising but logical feedback stimulates the learner to investigate unusual or perhaps unexpected applications of her knowledge [9].

5.2 Technology and Trends

Traditionally users of computer games have been isolated, either playing alone or with a very limited number of players connected to the same computer or console. With the emergence of the Internet, computer games have gradually shifted focus from the isolated gamer to multiplayer games where 64 or more players can play against each other at a time [10]. Indeed, with the advent of the internet there has been spawned a whole new culture in gaming with a continuing expected future growth [11]. Collaboration in learning is considered the most important field of research in educational games by educational game authority Angela McFarlane [12].

At the same time, in the sphere of educational games, the traditional approach has been to develop *drill and practices* games. These are games where the player is presented with a number of exercises, often in the form of multiple choice questions. These exercises are, in their simplest form, very similar to the exercises found in text books, and thus not very innovative. But *Drill and practice* exercises build on well proven methods, which have been refined throughout the history of educational institutions. Examples of successful implementations include *Internal Force Master* [13] and *Breast Cancer Detective* [14].

Recently there has been an increase in the creation of more imaginative educational software. This is in part the result of a closer collaboration between the computer game industry and academic circles [12]. These are games that are the result of the application and research into educational theory.

Kirriemuir and Elder both suggest the use of 'lite' or stripped down mainstream games as a compromise [12, 15]. Squire, However, suggest that theoretical understandings of edutainment is not as productive as innovative attempts at producing fresh concepts to support learning in games [10].

Another trend in computer games is the emergence of popular titles on handheld devices such as Sony's *Playstation Portable* (PSP) and *Nintendo DS*. These are dedicated gaming devices but PDAs and mobile phones can also be used as gaming platforms. New technology opens new doors to game producers and the market is expected continuous growth. This trend reveals possibilities for brand new modes of learning through context sensitivity and improved collaboration [11].

In addition to the separate advances and trends mentioned, there is a convergence between different gaming hardware. Home based consoles, PCs and handheld devices implement common interfaces for communication new

to the gaming scene. Examples of these are Bluetooth wireless connectivity and open internet standards allowing collaboration and communication across heterogeneous hardware [11].

Evidence of benefits from computer games when correctly integrated in a teaching environment is becoming clear through anecdotal, empirical and pedagogic evidence. Within a longer perspective, however, it is hard to predict the future trends.

5.3 Collaborative Gaming

Collaborative does not necessarily mean competition between teams, or otherwise an adversarial approach [16]. A goal that requires a collaborative process, like solving a puzzle does create a conflict in the form of the interaction within the game[17], but it is not a contest amongst adversaries. The team has to cooperate to reach a common goal. That being said, there is always the potential for a conflict amongst the members of the group as a result of varying or indeed conflicting visions, motivations and strategies.

Up until recently, the lack of proper means of communication and interaction has made it difficult to support collaboration in computer games, and there exist few actual true collaboration games on the market. Notable exceptions are two player arcade cooperation games. But these are on the periphery of true collaboration games, where the players cooperate to solve puzzles and difficult tasks, not merely helping each other to defeat a violent and powerful enemy.

6 Depth project findings

Based on authoritative literature in the field of computer games, as well as empirical studies of educational games we present our findings from [1] relevant to this Master Thesis. Our findings include:

- a checklist for quality characteristics in educational games
- a model for making games intriguing through social interaction
- a taxonomy for educational games
- suitability chart of the game genres for theoretical learning

6.1 Characteristics of Good Educational Games

In the process of studying available literature in the field of educational games, we have found that some characteristics are considered desirable by most researchers while other characteristics are subject to heated debate or considered contra productive. Based on such characteristics, evaluations of existing solutions can be conducted.

6.1.1 Characteristics

Combining theoretical knowledge with empirical experiences found in the literature, we have identified the characteristics that may characterize quality when used in a sensible way:

- **Variable instructional control** – the level of difficulty is adjusted to the skills of the player, or the player herself can adjust complexity [9, 18].
- **Presence of instructional support** – whenever the player is incapable of solving a task, some sort of hints or supplementary background information is available to the player [18, 19].
- **Necessary external support** – successful use of computer games in an educational setting demands careful considerations of external factors [18]. For instance availability of hardware, personal follow up and available guidance might be necessary for everyone to have a positive experience.
- **Inviting screen design** – users should feel they are playing a game in contrast to operating a program. The screen design might motivate

players by being playful and inviting without affecting the credibility of the game negatively [18].

- **Practice strategy** – players may practice without affecting their score or status negatively [18, 19].
- **Sound instructional principles** – this characteristic might be obvious but nonetheless an important quality attribute. Examples of such principles are motivating abstractions of theoretical syllabus [20], collaborative learning [18, 21] or use of recognized cognitive psychological principles such as repetition and incremental learning [22].
- **Concept credibility** – the theory or skills need to be abstracted in a way that maintains the integrity of the instruction. Empirical studies show that when abstractions become too conceptual or the game become too focused on abstractions instead of instruction, users find the game silly and loose interest [15].
- **Inspiring game concept** – Thomas Malone is the main authority on theory identifying what makes a game fun. Successful implementation of his principles might make a game inspiring, but the only way of determining such qualities is through empirical studies of the specific concept. Ideally the players lose track of time, experience curiosity, have an enjoyable experience and want to continue playing whenever they need to stop [9, 12].

Based on our research, these are the main areas of concern when conceiving an educational game. This list should be used as a reference when designing game concepts. In our opinion, the absence of one or more of these qualities can make a game less suited for its purpose. We are convinced that these aspects have to be covered for an educational game to maximize its potential. This does not mean that a game where one of these aspects is excluded will necessarily be unsuccessful or unpopular amongst the users, but the integration of the missing characteristic can further enhance the game.

In our opinion the most important characteristics are as follows (unordered): *Sound instructional principles*, *Concept credibility*, and *Inspiring game concept*. These are the characteristics that have the most profound impact on the educational value. They should thus be considered carefully when designing a game. From our perspective it is very difficult to justify the exclusion of these qualities from any educational game.

6.2 Collaborative Learning

As suggested by Angela MacFarlane [23], a further research on the strategic use of collaboration in educational software is a necessity. We will suggest how one can overcome one of the most important barriers when making good educational games by exploiting the potential of collaborative learning. Collaboration is extensively used with great success in modern commercial games. The immense success of titles such as *Buzz* [24] and *World of Warcraft* [25] exemplifies this trend.

6.2.1 Fighting the Intrinsic Fantasy Gap

The perfect educational game has high credibility and captivating gameplay. Realizations of such games are often games involving intrinsic fantasy with interesting and amusing abstractions of the real world domain. However, making games of intrinsic fantasy presupposes that the game is custom made for the subject being taught. This is expensive and diminishes re-use possibilities in other subjects. We argue that through creative use of collaboration one can restructure a game concept neglecting intrinsic fantasy into obtaining what might otherwise seem unattainable in terms of captivating qualities.

6.2.2 Extended Malone Model

In Chapter 5 we have described how a tight feedback loop between skill and fantasy is desirable in computer games, and how the making of such games

prove hard and expensive. By extending Malone's model with the entity *Social Interaction* in Figure 5, we see that the feedback from fantasy to skill can be reproduced through social interaction.

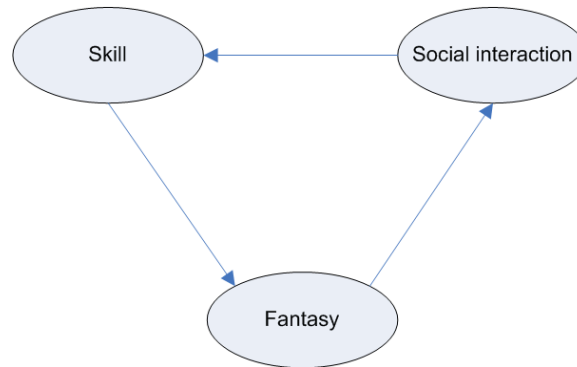


Figure 5: Extrinsic Fantasy, Social Interaction

This means that even though the fantasy of the game is irrelevant to the skill, the players are provided with direct feedback from the contribution of other players. A barrier in making good educational games is the fact that habitually a player is trained in a skill she only needs on her final exam or in real world future challenges. By providing constant feedback through collaboration the theory remains the same, but the real challenge takes place there and then instead of in a potential future. A simple yet illustrating example might be a straightforward quiz game compared to the group quiz game "*Buzz!*". A quiz game is independent of its contents and extrinsic by nature. By adding multiplayer possibilities the challenge is no longer just answering the questions, but to cooperate with or beat your opponents. The social element of the game provides a context, where the player receives a great deal of information from his opponents. To give an example, imagine a group of friends playing a group quiz. One of the participants tries to answer a question, but he gives the wrong answer and one of the others will be allowed to try. This person then gives the correct answer. The first person can then interact with the person with the correct answer and ask her follow up questions to obtain possible additional information relevant to the

question. Thus the important feedback from the game is indirect, obtained via the other players. This feature turns a simple idea into an amusing experience that proves successful even competing with mainstream intrinsic fantasy console games.

6.3 Taxonomy of Educational Games

Here we present a taxonomy of educational games, inspired by the early work done by Malone [9] and other existing taxonomies [26]. The taxonomy will be referenced as a rationale for our choice of game concept. For an explanation of how the taxonomy was deduced and example classifications, consult our depth project [1].

The three criteria for our taxonomy are summarized below.

- **Player interaction** - the existence or absence of possibility for several players to interact in some way through the game.
- **Fantasy and skills interaction** – whether the fantasy of the game is intrinsic or extrinsic according to Malone’s definition of these terms as described in Chapter 5.1.3.
- **Game Concept Type** - classification of the genre of computer game.

A graphical presentation of the taxonomy is depicted in Figure 6.

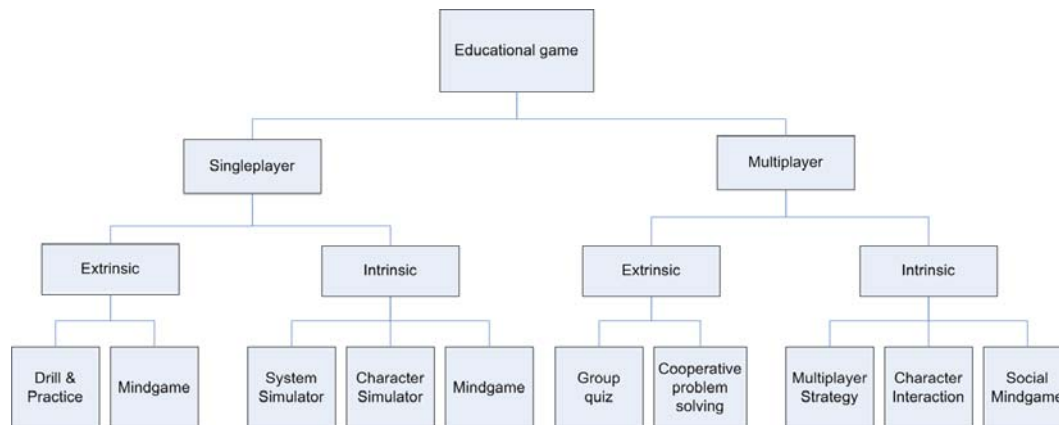


Figure 6: Suggested Taxonomy for Educational Games

Drill & Practice: The user answers questions from alternatives, visual recognition or textual input. The user is tested in factual recall and recognition. *Example: Internal Force Master, quiz games.*

Extrinsic Mindgame: Presents challenges including reasoning of some sort. The fantasy of the game does not affect the reasoning challenge. *Example: Minesweeper, Mastermind*

System Simulator: A complex system consisting of many parts and different sets of rules are simulated, and the user input parameters affect the simulation. Typical for a system simulator is that the player controls more than one character or aspect of the game at any one time. *Example: Civilization, Sim City, The Incredible Machine*

Character Simulator: Simulation from the point of view of a personified character/avatar. The player controls only one character or a very limited number of characters. Control of the environment where these characters exist is beyond the immediate control of the player. *Example: Americas Army Training Missions (single player)*

Intrinsic Mindgame: Presents challenges including reasoning of some sort relevant to the fantasy in a way where the fantasy gives feedback to the challenges.

Example: Illustrated math games, the niche of elementary learning games.

Group quiz: Drill & Practice game where interaction among players in the game itself or outside the game is strongly encouraged and is an essential part of the game.

Example: Buzz

Cooperative Problem Solving: Mindgame where the user parameters affect the game of other users without the fantasy of the game affecting the challenge at hand.

Example: Battleship

Multiplayer Strategy: Multiplayer games where the input from the different players affects the game of one another and the fantasy of the game is relevant.

Example: Command & Conquer, Civilisation (multiplayer)

Character Interaction: multiplayer character simulator where different players control separate characters that interact within the game.

Example: Americas Army (multiplayer)

Social Mindgame: Game involving reasoning where the players interact actively in the challenges. The fantasy of the game gives feedback to the challenge at hand.

Example: Multiplayer illustrated math games

6.3.1 Our taxonomy in practice

Any educational game can be categorized in our taxonomy given that certain characteristics of the game can be identified; namely the player interaction type, the fantasy skill interaction type and the game concept type [1].

6.4 Genre Relevance Comparison

In this chapter we give an evaluation of the different genres of games for educational purposes. As a basis for our assessment we use our own taxonomy presented in Chapter 6.3.

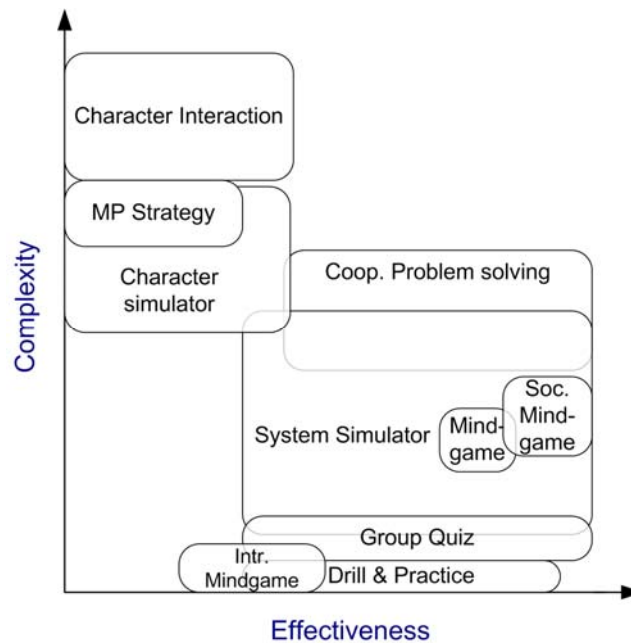


Figure 7: Genre relevance for theoretical knowledge [1]

6.4.1 Theoretical Learning

By theoretical learning we mean the internalization of written or lectured subjects. Figure 7 shows our evaluation of the different categories for theoretical education [1]. We see that the least complex genres are also the most effective genres for teaching theoretical subjects. We believe conveying

theoretical information through the perspective of a character might lead developers into more problems than benefits. The system simulator genre is considered a promising category for theoretical learning, though we have not seen many attempts at this category. Several categories that are considered well suited for theoretical learning do not feature intrinsic fantasy, and are thus eligible for reuse in different domains [1].

Part III

State of the Art

This part describes existing applications for lecture enhancements. The solutions are evaluated and the different features are compared. Finally the key features of our concept are presented and a comparison to the previous solutions is carried out.

7 Existing Classroom Applications

Several solutions have been designed for student participation during lectures. These solutions are presented here along with findings and experiences from using these systems.

7.1 TVREMOTE Framework

The TVREMOTE framework was designed to allow for student participation in lectures counting hundreds of participants at Darmstadt University of Technology, Germany [27]. The system features polling of student opinions and electronic question submission. The teacher can also broadcast links and notes, as well as multiple choice questions. The teacher collects the feedback and reads it from a private display, from which she can select a question for display on a second public screen. The multiple choice quiz provides the teacher with a statistical distribution of correct and incorrect answers. Studies of using the TVREMOTE show that students generally appreciate a short explanation as to why a given answer is correct.

The TVREMOTE framework uses GPRS for data transmission and Bluetooth support is planned as a future feature. Surveys show that students are reluctant to pay for the data transmission fees from sending data over GPRS [27].

7.2 Classroom Presenter

Classroom presenter is a Microsoft PowerPoint plug-in developed at the University of Washington. Version 3 was released in April 2007 and allows for students to write comments directly onto the digital slides with a stylus or textual keyboard input using a tablet PC handed out in the beginning of the lecture. Exercises are broadcasted to the tablet PCs, and the student writes their answers onto a blank space of the slide. The teacher can then browse through the replies. Use of Classroom Presenter is illustrated in Figure 8 and Figure 9.



Figure 8: Students using Classroom Presenter

The application is open source and has been used in over 100 courses, notably in courses teaching software engineering and algorithms [28, 29].

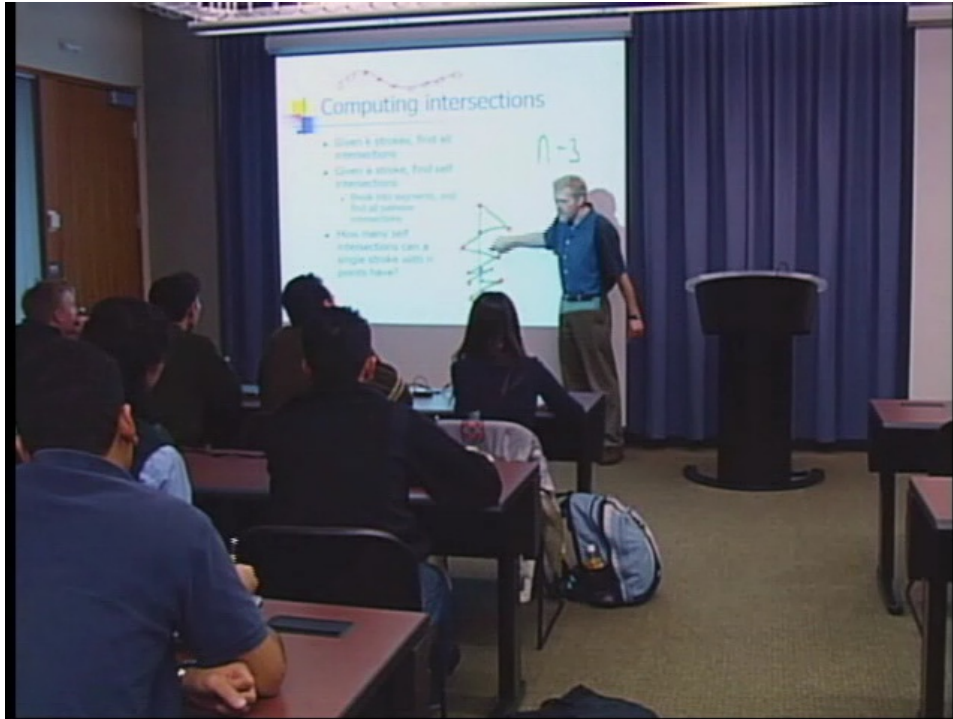


Figure 9: Classroom Presenter lecture example

7.3 WIL/MA

WIL/MA is a tool developed at the University of Mannheim, Germany and is a Java implementation of digital hand raising, spontaneous student comments and multiple choice questionnaires. PDA's supporting J2ME (previously *Personal Java*) are handed out to the student participants at the beginning of the lecture. Data communication is obtained over WLAN coverage. The teacher obtains the student data on his PC and reads it from a private screen. A survey of similar classes where one class used WIL/MA and the other attended traditional lectures showed the learning outcome of using WIL/MA superior [30]. Example screenshots from student feedback and multiple choices are shown in Figure 10.

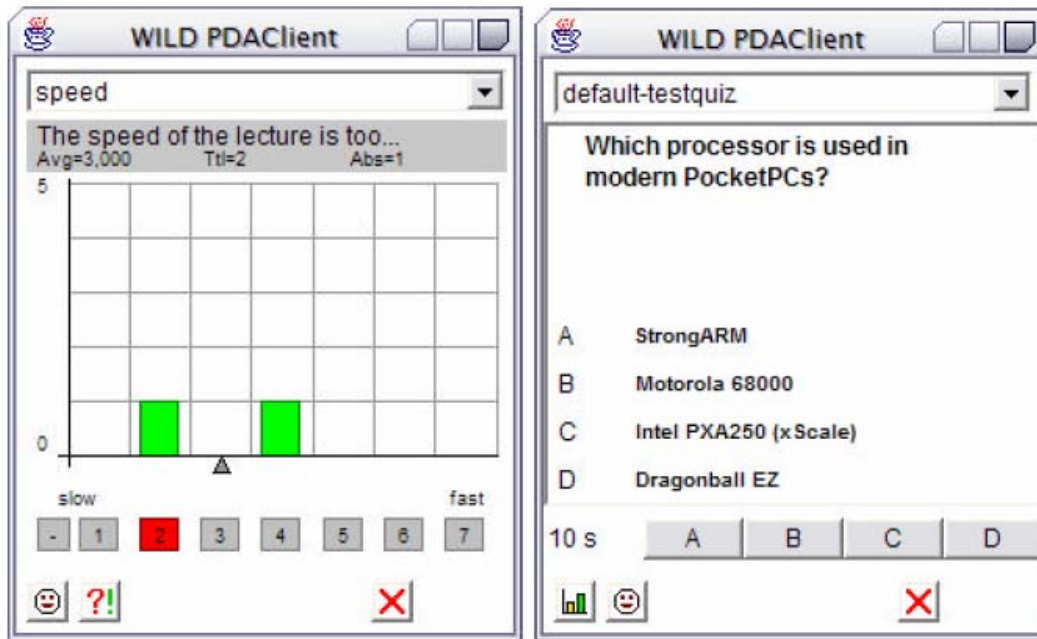


Figure 10: WIL/MA PDA Screenshots

7.4 ClassInHand

ClassInHand is developed at Wake Forest University in the USA and features a presentation controller, real time quiz and student/teacher interaction. The system requires PDAs supporting Windows Mobile 5 or Windows Mobile 2003 for PocketPC. The teacher only uses his private PDA screen for information retrieval and then presents this info to the students independently of the ClassInHand system [31, 32] as shown in Figure 11.

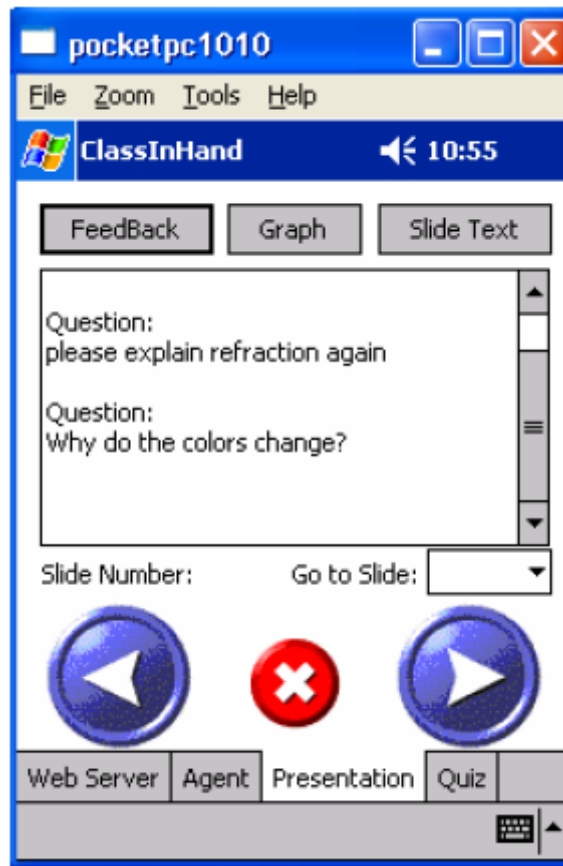


Figure 11: ClassInHand PDA screenshot

7.5 Ez ClickPro

Ez ClickPro is a commercial classroom polling application developed by Avrio Ideas for teaching in elementary school. It is commercially available for £3.450 GBP for the maximum set including 100 custom remote controls.

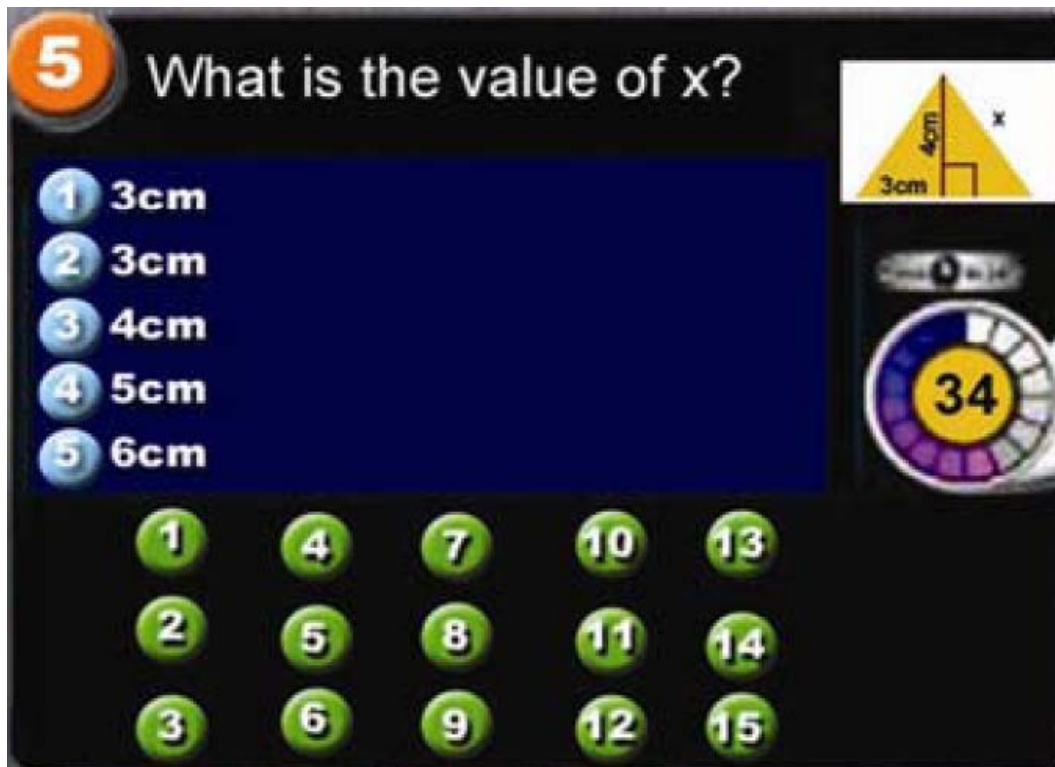


Figure 12: Screenshot Ez ClickPro

Ez ClickPro uses infrared technology and custom produced remote controls, shown in Figure 13, with the teacher running an application on a PC as shown in Figure 12. Multiple choice questions are displayed on a TV or projected onto a canvas, before the students submit their answer using their remote control. Each student is assigned a number that is displayed in a green circle if the answer is correct or otherwise if the answer was wrong. The questions can be presented with elaborating pictures and videos [33, 34]. A new version of the software due 2007, called PowerClip, is integrated into Microsoft PowerPoint.



Figure 13: Ez ClickPro remote controls and sensor

7.6 Buzz! The Schools Quiz

Buzz was originally conceived as a commercial trivia game for Playstation 2 receiving great success in terms of both sales and critics. The game is marketed as “the game show in your living room” and comes with special wired handsets called buzzers displayed in Figure 14. There are currently four versions of the game for sale in a multitude of languages, each testing knowledge in different domains such as sports, music and general trivia [35].

With government funding, a new version of the game, designed especially as a tutoring tool is due released late 2007, named *Buzz! The Schools quiz*. The game will be marketed towards educational institutions and comes with content covering Stage 2 National Curriculum for primary schools in the UK. A new feature, “Create a quiz” is included to allow tutors to hold revision exercises on a given subject. However, the wired Buzzers are still in use, thus

limiting the number of simultaneous players. The content is also limited to the included questions. As a consequence, newer versions are required as the curriculum changes and the game will also be less interesting in countries other than the UK.



Figure 14: Buzz! Cover and handsets

8 Evaluation of Previous Solutions

From Table 1 we see a tabular representation of the features of previous classroom software solutions. All the solutions except *EZ ClickPro* and Classroom Presenter are tools limited to student participation and student-teacher communication in larger lectures. These solutions have limited or no support for displaying feedback on a projector screen, as a consequence of their function as a communication tool rather than a game. All of the solutions except Classroom Presenter feature a quiz mode. It seems the quiz is typically intended for the teacher to monitor whether or not the students are paying attention. There are no elements of competition, goals or amusement besides the actual selection of alternatives and the observation of

the correct answer. All the solutions except TVRemote depend on hardware being handed out to each student before each lecture.

Two solutions stand out from the typical classroom polling. Classroom Presenter is a powerful communication tool, sharing edited Microsoft PowerPoint slides. But the tool does not support any communication beyond the actual sharing of these slides. Feedback is thus given through the teacher's presentation and never directly to the students. Ez ClickPro is a pure quiz game where animated feedback is shown directly on a big screen for the students to enjoy [34].

Table 1: Previous solutions features

Features	TVRemote	Cl. Pres.	WIL/MA	ClassInHand	EZ ClickPro	Buzz!
Digital student comments	X		X	X		
Teacher info broadcast	X	X	X	X		
Quiz mode	X		X	X	X	X
Public feedback display	(X)	X			X	X
Animated graphics					X	X
No custom HW needs	X					

8.1 Prototype Concept Comparison

In Part V, the concept prototype subject to our research is presented. The concept idea is similar to several of the products presented in Table 1. The prototype aims to feature the stimulating gaming experience from "Buzz!", with animated graphics on a public display. Using mainstream technology, the prototype allows all students and educational institutions to use the system without any hardware or software purchases. This is achieved by only requiring hardware and software that are commonly installed in lecture halls to facilitate usage of the prototype. Likewise for participants, only hardware and software that most students already possess are required. Add-on functions such as information broadcast and digital student comments are omitted as the prototype is intended to be perceived as a computer game rather than a software tool.

Part IV

The Prototype

The concept being realized in this thesis is one of two concepts conceived in our previous depth project [1].

9 Concept

In this chapter the concept idea is described along with the theoretic foundation and thoughts this concept is based upon.

9.1 Concept Foundation

When we were searching for a concept for use in combination with lectures aimed at students in higher education, we came across a large number of solutions of varying success. It is encouraged to have intrinsic fantasy in educational games, but most such solutions we have seen are simulators teaching a specified task. Yet we knew that simulators are expensive to make and very dependent on the skills being taught. Specifically the entire game needs to be reinvented for new subsets of knowledge. We have found that most lectures convey theoretical knowledge rather than practical, and we have seen that earlier solutions are both complex and expensive to produce. On the other hand we have seen extrinsic fantasy games with notable success [13, 14].

Malone [9] stated that intrinsic fantasy is useful because the fantasy provides constructive feedback relating to our skills and vice versa. In extrinsic fantasy games we lack this input, but we believe one can achieve the same effect by taking as source of input a community of other players. We claim this will give the same motivation because social interaction proves stimulating and perfect to provide authentic feedback on the user skills. This concept is illustrated in Figure 5.

Using our taxonomy described in Chapter 6.3, we have classified our concept as a “group quiz”. From Figure 7 we see that this genre of educational games is considered quite efficient for learning theoretical knowledge while

complexity is kept low. The “group quiz” genre also encourages re-usability as opposed to genres of intrinsic fantasy nature.

9.2 Concept Presentation

This concept is intended to refresh the memory of the audience in a fun way while attending a lecture. From a cognitive perspective the strengths of the application is the possibility of making students reflect on the theory while it is still fresh in their memory in addition to keeping students alert throughout the lecture. The goal of the game is specific to each game mode. Two game modes have been implemented, one in which the goal is to be the last man standing in a questionnaire shoot out. The other mode gives feedback on how each student scores compared to her other classmates. The goal is more subtle and individually motivated. For instance a student may try and make it to the high score list, or contribute to a positive impression on the teacher. We expect that the social aspect of the game is an important motivational factor for students to participate and enjoy the game. It is important that the graphics presented on the lecture screen is enjoyable and animated in order for the participants to feel they are participating in a game rather than a statistical evaluation. As the client application is run on mobile phones, sparse exchange of data between server and client is imperative. It is also important that the game does not steal time from the lecture, but integrates smoothly without adding dead time to the teaching process.

The students register to a simple client application using mobile phones or online laptop computers. The registration process prompts the students for a nickname and a session code identifying the lecture they are currently attending. The lecturer has prepared questions concerning the lecture theory in advance. During the lecture, when the speaker finds it suitable, a question appears on the main screen with multiple choice alternatives. Simultaneously the alternatives appear on the screen of the connected clients as the audience selects their alternatives.

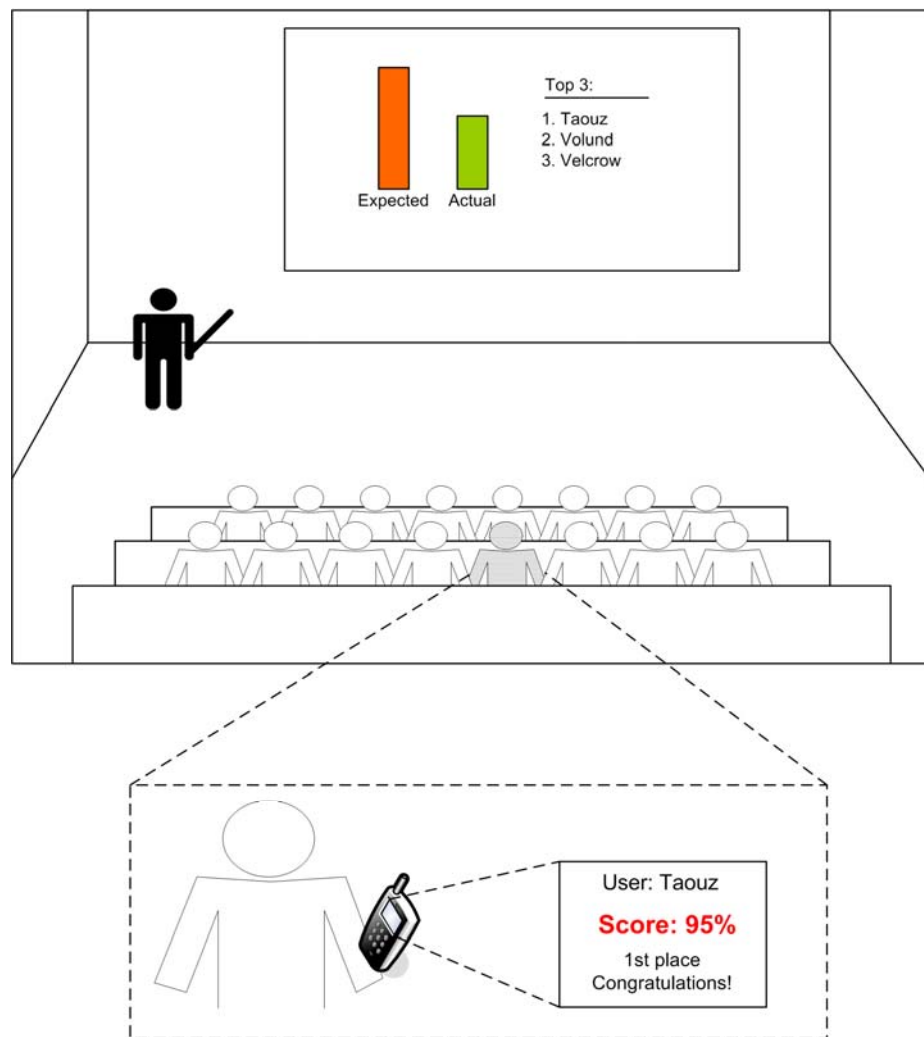


Figure 15: Early concept illustration

When the question times out, e.g. after 30 seconds, the correct answer is stated and mode specific graphics displays the feedback on a large screen. For instance; some indicator of the collectively achieved score appears animated on the presentation screen such as: “cool”, “great”, “not to good” or “awful!” accompanied by a suiting sound effect. The lecturer bundles several questions assigned to one of several game modes. Collective feedback and a temporary high score list sums up each round. The final stage of the game appears when the lecture is over. The top three nicknames appear on the screen. Each player is updated on her performance throughout the lecture. Figure 15 shows an early illustration of how the concept was envisioned.

9.3 Desired Characteristics of the Prototype

In Chapter 6.1 we have summarized the desired characteristics of educational games based on an extensive literature search. We give a discussion of how these characteristics are realized in our concept and how a full scale implementation should be administered to maximize its potential of success.

Variable instructional control is left to the teacher to implement in her choice of questions. It is advised that the students experience a soft start in the sense that everyone experiences personal mastering. The nature of the game makes the target group narrow and homogenous in terms of domain knowledge as they necessarily participate in the same lecture.

Instructional support presents itself when informing the player of the correct answer whenever a user submits a wrong answer. However, the concept lacks a second source of theory when the users do not master the subject. It is left to the teacher to supply literature for students who do not experience a successful participation. Alternatively the teacher can initiate a follow-up discussion of the subject matter at hand.

External support denotes careful considerations of external factors. Availability of hardware is high as we assume that everyone possesses the hardware to play the game. From a pedagogical point of view personal follow up and encouragement to participate is important, but not an integrated part of the game. Students who chose not to participate may feel left out, but still the teacher can improve external support by rewarding participation.

Inviting screen design and carefully planned usability are prioritized subjects and forms the foundation of some of the choices of external tools.

Fun, animated graphics and custom graphics on the mobile phone is prioritized in the prototype implementation.

There is no **practice strategy** as an addition to the actual competition in the game. However, the game is in many ways nothing but a practice for future exams. If the teacher wishes to prepare the students, sample questionnaires may be published in advance of a game session.

The most important characteristic of an educational game is a high quality concept. We have already provided arguments for our **sound instructional principles** based on our literature study as described in the previous Section 9.3. Whether the prototype proves to be of **high credibility** remains a question for our empirical data gathered from our experiment in Part V. However, as there is no abstraction of the real world, the students are less likely to find the prototype silly or technically deficient in relation to their theoretical knowledge. As with the level of credibility, there is no way to recognize an **inspiring game concept** other than empirical studies. This will therefore be subject to investigation in our experiment.

9.4 User Guidelines

Based on the preliminary findings in Chapter 6, a checklist describing optimal external factors of using the prototype is deduced. The checklist is intended to be a guideline for teachers who wish to use the prototype in a pedagogically effective manner. The guidelines are general in the sense that they are valid for all implementations of lecture games, such as TVRemote and EzClickPro mentioned in Chapter 7, as well as future implementations of lecture enhancement systems.

- Early questions should be easy in order for the students to become comfortable with the system, and experience a sense of mastering.

- The teacher should read the question and alternatives out loud to integrate the game as a natural part of the lecture.
- The questions should alternate between simple multiple choice questions and exercises to prevent a monotone experience of the game.
- Whenever the students do not succeed in answering a question, a following discussion of the theory and an explanation of the correct answer are in place.
- The teacher should encourage participation by offering a prize for the winner, or reading the nicknames of the best students out loud.
- The teacher should show an interest in the student scores, by commenting on the results and sharing her enthusiasm with the participants. However, the teacher should not prioritize this to the point where she becomes more of a game show host than an educator.
- Questions should alternate between recapturing recently presented theory and summarizing general theory presented at an earlier stage. Further variation can be provided by grouping several questions in bulks while distributing question over the entire course of the lecture.

10 Technology Rationale

When creating a multiuser classroom application, there is considerable concerns regarding which technologies are suitable. Costs in terms of time to

develop the application and in terms of actual price of the technology are some of these factors discussed in the following subchapters. Figure 16 shows how we imagined the system based on the concept discussed in Chapter 9.2. The students communicate with the system using their mobile phones with an Internet connection. Likewise the teacher uses her PC to communicate with the system over the Internet. The feedback graphics are displayed on the PC and needs to be shown on a large screen so the students can observe the results. The server interprets the requests from the different clients and keeps track of program flow and synchronization between clients.

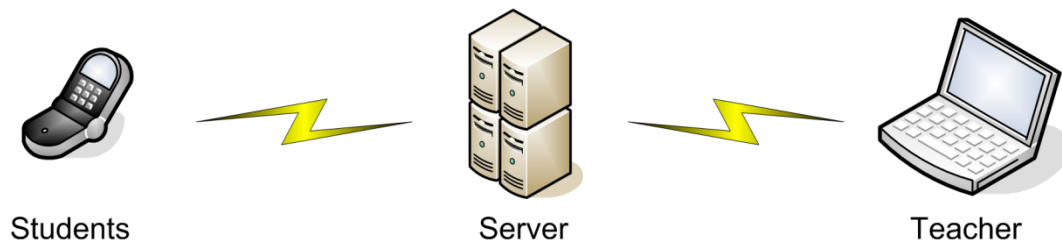


Figure 16: Early system overview

10.1 Server technology platform

We will here summarize the rationale behind our choice of technology platform for our server application.

10.1.1 Java

The Java programming language is an object oriented, high level language developed by Sun Microsystems. Code written in Java is compiled to bytecode, which in turn is interpreted by a Java Virtual Machine (JVM). Early versions of the JVM were considered quite computational inefficient due to the fact that the virtual machine interpreted non-optimized bytecode. But later implementations of the JVM have improved vastly in this regard, and today Java is considered as fast as C and C++ for some platforms. In addition Java is safer to execute [36]. This is due to the JVM and its automatic memory

management, thus sparing the programmers of the burden of manual memory management.

One of the key philosophies behind the Java language is platform independence [37]. The bytecode produced by the Java compiler can be run on any implementation of the JVM. Java runtime is available at no cost for a number of platforms, including but not limited to: Windows XP/Vista, Linux, Solaris and Mac OSX [37].

10.1.2 .NET

The Microsoft .NET Framework is a platform primarily available for the Windows family of operating systems [38]. It is included with Windows Vista, and freely available for download for older versions of the operating system. Other implementations also exist, like Mono for Linux. At the heart of the .NET architecture is the Common Language Runtime (CLR) which is part of the Common Language Infrastructure specification (CLI). The main purpose of the CLR is to make the platform language independent and to provide automatic memory management as well as security features. Code targeted for the .net framework is compiled to Common Intermediate Language (CIL) (previously known as Microsoft Intermediate Language). The CIL byte-code is then compiled at runtime to machine code [38].

10.1.3 Choice Rationale

Portability is one of the most important factors when choosing the technology platform for the server. The ideal situation is to have a server application which can run on any platform. This has several advantages: no need for additional investments in hardware and software to run the game. Also there is the possibility to run the server application on any modern computer and a low threshold for installing and running it. In this light we

chose Java to build our server. In addition to fulfilling our needs for a portable solution, Java has an excellent reputation as a server and middleware platform [39]. There is also a good selection of freely available tools for Java which we consider a bonus.

10.2 Mobile Technology Platform

Several platforms for programming on handheld devices exist. A short presentation of our alternatives is given along with rationale for our choice of platform.

10.2.1 Java 2 Platform Micro Edition (J2ME)

J2ME is Sun's contribution to the wireless market. It is a small scale version of Java designed for smaller and typically handheld devices such as mobile phones, PDAs and other consumer electronics such as car navigation systems. Much like Java a virtual machine interfaces to the specific operating systems, making the actual code portable across any hardware supporting J2ME within a specific configuration. There are configurations for classes of devices such as a configuration for mobile phone, CLDC, and one for more powerful devices such as advanced PDAs called the CDC.

To make the J2ME Runtime Environment complete, a profile constitutes a programming API for the programmer. The only profile available for the CLDC configuration is the MIDP profile. MIDP 1.0 is the first released API. An upgraded version, MIDP 2.0, supports custom game graphics and multimedia possibilities [40]. Almost every mobile phone being sold at the moment of writing has built in Java MIDP 2.0 support [41].

10.2.2 Microsoft .NET Compact Framework

.NET Compact Framework is a small scale implementation of the .NET framework optimized and limited to run on small devices. It contains a platform adaptation layer that allows different operating systems on different hardware to run the .NET Compact Framework applications. However, only a few operating systems such as Microsoft CE, Microsoft Pocket PC and Smartphone offer support [42].

10.2.3 Mobile Platform Selection Rationale

.NET Compact Framework and J2ME offer the same potential and freedom of implementation in our concept. However, few students own a device supporting the .NET Compact Framework. Availability is an important feature of our concept, thus J2ME is the chosen platform of implementation. To offer a game-like feel to the mobile application the concept is implemented for MIDP 2.0. It is likely that some students possess a phone that lacks MIDP 2.0 support, but we consider this a small loss compared to the increased usability that MIDP 2.0 allows.

10.3 Protocols

The choice of protocol is an important design decision, and imposed its own set of constraints on the prototype with regards to robustness and reliability. In the following subchapters we briefly introduce the protocols evaluated for use in our prototype: TCP and UDP. These are the two protocols that form the core of the Internet Protocol (IP) suite.

10.3.1 Transmission Control Protocol (TCP)

With TCP two peers can establish a connection to one another, having one stream socket at each end. A connection is maintained as the sockets at each end are open. The protocol guarantees in-order delivery of data between the two parties. TCP controls that no packets has been lost in transmission via

sequence numbers on the packets. When a packet has been correctly received, TCP sends an acknowledgment of which packets has been received from the sender. In the case of lost or presumably lost packets, the packets will be retransmitted. There is also a checksum in each packet to ensure that the data is not corrupted [43].

10.3.2 User Datagram Protocol (UDP)

UDP does not have any of the mechanisms for reliability which TCP has. There is no sequence numbers for guaranteed in-order reception of data, nor is there any checking of whether or not the data has arrived properly. These features ensure that UDP is fast and efficient, especially for transmission of large quantities of small data. An example of this can be broadcasting of data. Applications using UDP must tolerate lost or duplicate data [43].

10.3.3 Protocol Selection Rationale

We decided to use the TCP protocol in this project. The reason for this is the inherent robustness and reliability of this protocol. Our concept requires for instance that the student clients installed on the mobile phones and laptop computers have a persistent connection which the server can use to push out messages. These needs take precedence over the overhead created by TCP in comparison with UDP. We chose to use the same protocol for all the components of the prototype, ensuring interoperability and reducing design complexity.

10.4 Communication bearers

The concept to be implemented requires ease of use for large numbers of participants. We here present a short introduction to the considered communication bearers.

10.4.1 Ethernet over twisted pair

Ethernet over twisted pair cable is standardized as 10BASE-T, 100BASE-TX and 1000BASE-T. The transfer rates are 10Mbit/s, 100Mbit/s and 1000Mbit/s allowing for high transfer rates over short distances, typically 100 meters or less [44]. Data is converted into electrical impulses, which are transmitted over the wires. Ethernet over twisted pair features high transmission rates, low latency and little noise. The limitation of this bearer is the need for the receiver to be physically connected to the sender and the limit on cable length [44].

10.4.2 GPRS, EDGE and 3G

General Packet Switched Data (GPRS) is a packet-switched data service available in GSM mobile networks. Many users share the same communication channel, and transmit data only as needed. This means that the user can be connected to a server for a very long time, yet only pay a small fee if low volumes of data is transmitted [45]. The users are charged for data transfer rather than the time they have stayed connected as is the case when using Circuit Switched Data (CSD). GPRS uses one or more timeslots to transfer data, where the maximum speed for a regular GPRS slot is 20Kbit/s. Regular configurations are 3 or 4 bundled slots for upstream data transfer and 1 for downstream. Giving upper theoretical speeds of 80 or 60 Kbit/s upstream and 20Kbit/s downstream. The more slots used, the more increases the probability of an interrupt caused by CSD needs, such as voice calls [45], which will cause delays.

An enhanced version of GPRS is currently being deployed, Enhanced Data rates for GSM Evolution (EDGE) or Enhanced GPRS (EGPRS). EDGE increases data speeds by increased utilization of the GSM radio signal. EDGE requires compatible handsets and software upgrades at the base stations to support the new signal encoding. Maximum speed per slot is 59.2 Kbit/s, giving 236.8

Kbit as a theoretical maximum speed for an EDGE connection using 4 bundled slots, and 473.6 Kbit/s for one using 8 slots [46].

UMTS (3G) is the newest mobile network technology to be deployed in Europe and Norway. UMTS supports up to 384Kbits/s downstream rates, or 3.6Mbit/s in High-Speed Downlink Packet Access (HSDPA) enabled networks. The UMTS network has been designed with both voice and data communication in mind, whereas the GSM (2G) networks was originally designed to accommodate voice communication and CSD [47].

10.4.3 Wi-Fi (IEEE 802.11 a,b,g,n)

Wireless LAN, also known as Wi-Fi, is based on the IEEE 802.11 specifications. Wi-Fi uses radio waves in the 2,4 GHz spectrum for transmission of data [48]. Wi-Fi enabled network interface cards is typically bundled with laptops, high-end mobile phones and PDAs. Wi-Fi has typically transfer rates of 54Mbit/second (802.11g) or 11Mbit/second (802.11b). This is substantially lower than Ethernet over twisted pair [48].

10.4.4 Bluetooth

Bluetooth is a specification for wireless Personal Area Networks (PANs). Bluetooth uses radio waves in the 2,4 GHz spectrum for transmission, i.e. the same spectrum as Wi-Fi [48, 49]. Transmitters are divided into three classes, where class 1 has a range of 100 meters, class 2 has a range of 10 meters and class 3 has a range of 1 meter. Bluetooth devices are connected in groups, so called piconets, with one master and up to seven active slave devices. Up to 255 other Bluetooth slave devices can be inactive and the master can at any time activate them, forcing a currently active device to become inactive. Typical Bluetooth applications are file transfer between mobile phones or

mobile phones and computers or wireless connectivity between input devices such as keyboards and computers [49].

10.4.5 Data bearer selection rationale

Since the development framework used in this project will be Java and TCP/IP, which implies the use of socket programming, the data bearer is essentially transparent to the applications per se. This, however, does not mean there is not a preferred or expected data bearer for the different components of the prototype. The server will typically be a stationary workstation or server computer which is always on and always connected to the internet. This warrants the use of Ethernet over twisted pair, as it is a stable and widely available communication bearer on university campuses. The teacher client will typically connect to the server from a class room. Both Wi-Fi and Ethernet over twisted pair are suitable for this, as these communication bearers are usually available in a class room. In some cases we imagine 3G, EDGE or even GPRS can also be used if neither Wi-Fi nor Ethernet over twisted pair is available. The amounts of data to be transferred between the server and the teacher clients are miniscule, so the bandwidth will not be an issue.

The student clients will be run by the students, situated inside the lecture hall, on either a mobile phone supporting J2ME or a laptop computer. This requires wireless connectivity; preferably GPRS, 3G or Wi-Fi, as most new mobile phone support one or several of these communication bearers. As UMTS (3G) will increase its market penetration, we must expect a reasonable amount of student clients will be connected over UMTS in the future. Some newer handsets also support Wi-Fi, and in lecture halls with Wi-Fi coverage, we will not be surprised if the students choose to connect over this communication bearer, as it will be free of charge for the users.

10.5 Graphics Library

There are two powerful widely accepted low level libraries available for PC users offering full screen 3d graphics: OpenGL and DirectX. A presentation of the two solutions follows along with arguments for the selected product. The two solutions are considered equally competent in terms of performance.

10.5.1 OpenGL

OpenGL is a 3d graphical library supported on common operating systems including OSX, Windows and Linux. Compared to DirectX the OpenGL hardware support is inferior. The library keeps the current graphical structures in memory and offers over 100 procedures available to the user for manipulation on subsets of the graphics. OpenGL was originally designed for C and C++ programming but is now available for other languages through so called OpenGL bindings including Java [50]. However, such bindings need to be separately installed on the computer making the installation process more complex. We also found that OpenGL has high level objects for smooth text manipulation through its TextRenderer class.

10.5.2 Microsoft DirectX

DirectX offers 3d full screen graphics through its Direct3d graphical library. It is the most used graphical library in the computer game industry, and it has wide support in the available graphics hardware accelerators. However, Direct3d is only supported in releases of the Microsoft Windows operating system. Direct3d arguably lets the programmer operate at a lower level of programming and directly on buffered graphical structures and corresponding function calls consequently routed directly to the Direct3d engine for processing. The control of algorithms are decided to a greater extent by the programmer and less by the current drivers, providing extra possibilities yet more spacious code compared to OpenGL. DirectX code is programmed in languages C++ or C# [50].

10.5.3 Graphical Library Selection Rationale

The need for a graphical library that's widely known and supported is important for future implementations or additional game modes to our implementation. We do not recognize a need for cutting edge technology, yet the ability to impress through enchanting graphics is important. Portability across operating systems is an important factor as teachers may use a Mac or run Linux. Animated text is a natural part of our program and benefits from suitable high level text manipulation tools in OpenGL. We also see the potential for re-use of code across the two types of clients. A Java implementation is only possible if we select OpenGL. Therefore OpenGL is our preferred library for the concept implementation.

10.6 Database solution

There are several competing Database Management Systems (DBMS) on the market today, each powerful and feature rich in their own respects. Here we will briefly present the rationale and background for our choice of DBMS.

10.6.1 MySQL

MySQL is a DBMS owned and sponsored by MySQL AB, a Swedish based company. MySQL AB also owns most of the copyrights to the codebase [51]. MySQL is a key component of the LAMP (Linux, Apache, MySQL and PHP) solution stack, which is commonly used to run dynamic web sites [52]. The MySQL DBMS comes in two different variants: Community Server and Enterprise Server. Both share the same codebase and are released under the GPL. The Enterprise Server, however, is aimed at a commercial market and has product support and only publicly available source (not binaries). The Community server is released on an unspecified schedule; with binaries being released with every major update free of charge (smaller incremental updates may not have binaries included). As the Community Server is easily available, free of charge and requires no compilation when using the

provided binaries for the intended platform, this version has proven itself immensely popular with developers of free software and web sites [52].

10.6.2 Oracle

The Oracle RDBMS (Relational Database Management System) is a commercial, closed source database system which is available on a number of platforms including Windows, Linux, Solaris and Mac OSX.

10.6.3 Other Database Solutions

PostgreSQL is free software, Object-Relational Database Management System (ORDBMS). Its codebase is not controlled by a single company, but rather the community which develops and maintains the code [53].

Firebird or (FirebirdSQL) is a RDBMS released under the InterBase Public License [54], an open source software license. The application has been in development for over 20 years, starting in 1984 and became open source in 1999.

10.6.4 Database Selection Rationale

Portability, price and easy installment, as well as previous experience on our part have contributed heavily towards our decision of choosing the MySQL DBMS for our solution over the other alternatives. All the different database applications mentioned here has readily available Connectors for use with Java which is our development environment. MySQL has a broad user base, and there exist large amounts of tutorials created by the community and other developers, which are freely available on the internet. The version of MySQL used in this project is Community Server 5.0.27 for Win32, which are the newest release containing binaries at this time.

11 Prototype Design and Architecture

With basis in our choices of technologies; a concept architecture and design has been deduced. The architecture is described through the logical, process and physical views as described by Kruchten [55]. The corresponding scenarios can be found in Chapter 15.

11.1 Architecture

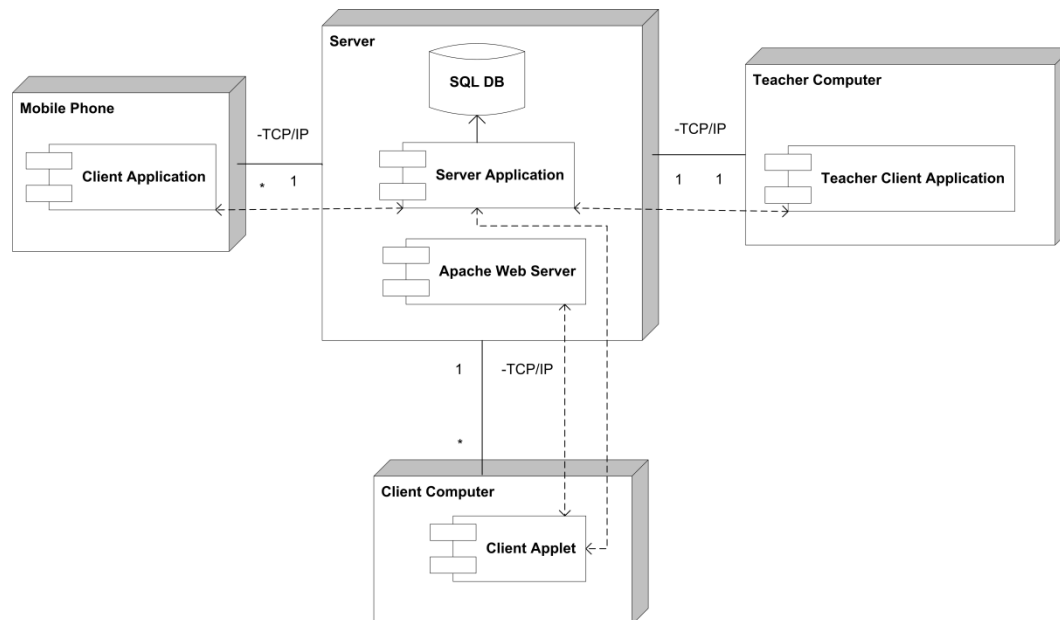


Figure 17: Prototype UML deployment diagram

A UML deployment diagram illustrates the physical entities involved in using the system in Figure 17. Several mobile phones runs the J2ME client application which communicates with the Server using TCP/IP protocol. The signal bearer is transparent to the system and selectable by the user. The system is tested for bearers GPRS, UMTS 3G and Wi-Fi. Users of client computers can also select their type of Internet connection as they please.

Mobile Phones have to be MIDP 2.0 compliant, while any computer with a browser supporting Java Applets can run the Client Applet.

The communication bearer is transparent to the system and optional to the user of the Teacher Client Application as well. The Teacher application uses TCP/IP and is tested for Wi-Fi and Ethernet over twisted pair connections. An installation of Java Bindings for OpenGL version 1.3 or higher is required and 3d accelerator hardware is recommended for smooth and fast graphics. The teacher client is successfully tested on operating systems Mac OSX and Windows XP. The computer running the teacher client application needs to duplicate the display onto a public display such as a projector, and connect the audio output to a speaker system.

The computer running the server application has to have Java SE 5 or higher installed. The system is developed in Java, and thus platform independent. The communication bearer is, as mentioned, transparent to the system, but a physical connection to the internet is preferred to ensure a stable and reliable connection. In the case of a firewall being installed between the server and the internet, the TCP ports 22 (used by the student clients) and 119 (used by the teacher client) has to be open.

11.1.1 Inter Process Communication

Communication across the three types of entities is illustrated and commented in a UML sequence chart in Figure 18. The entity Mobile Client represents any number of both Web Clients and Mobile Phone Clients. As shown in Figure 17, there is only one instance of each of the entities Teacher Client and Server.

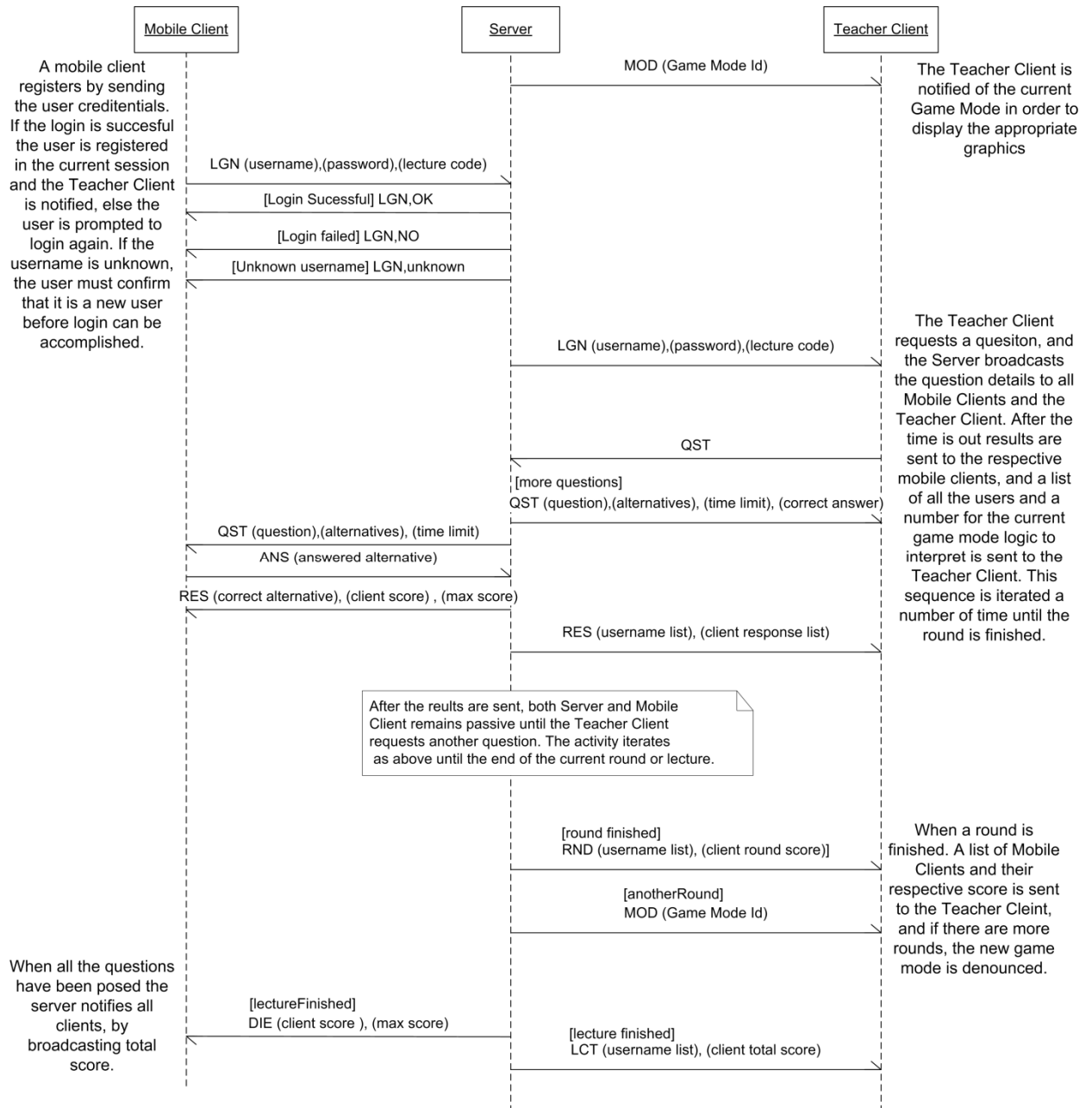


Figure 18: Sequence Chart of Inter Process Communication

11.2 Design and Implementation

In this subchapter we describe the fundamental design of the different applications that constitutes our prototype. The class diagrams presented is somewhat simplified for improved readability.

11.2.1 Server

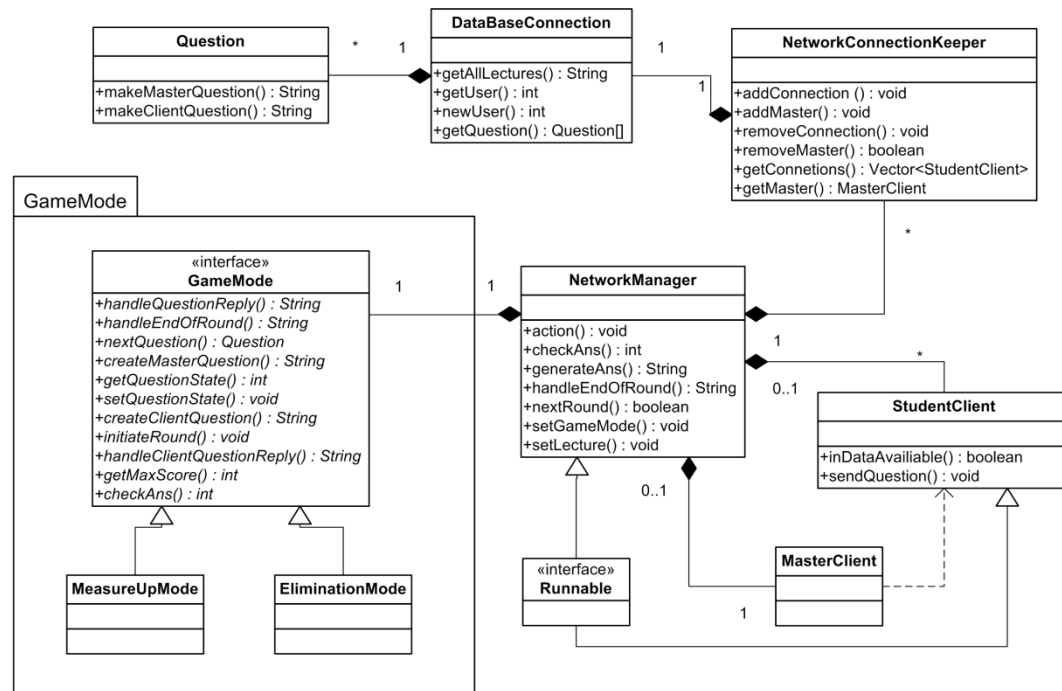


Figure 19: UML Class Diagram Server Application

The server's main responsibility is to facilitate the task of tracking connected clients, routing messages to and from the Teacher (Master) client and reading to and from the database containing the actual content to be used in the lectures. In addition, the server handles the score system and performs game-mode specific operations on the data received from the clients. The server is implemented in Java SE 6. Figure 19 shows the core classes as well as the *GameMode* package, which allows the application to be extended with new functionality. External developers can quite easily make a new class in the *GameMode* package which implements the *GameMode* interface. The new game-mode has to be given a *GameMode* identifier, i.e. a serial number from

00-99 to identify it. Reserved game-modes are 10 (measure up) and 11 (elimination). The server's nerve-center is the *NetworkManager* class. An instance of this class is started as a separate thread by the main method in the initialization class. It also initializes an instance of the *NetworkConnectionKeeper* class, a central repository for storing connections from both the teacher client and the student clients. In turn the *NetworkConnectionKeeper* creates a *DataBaseConnection* which has the responsibility of connecting to the database. The class uses the freely available *MySQL Connector/J* from MySQL to connect to the database. The database used with our prototype is MySQL 5.0.27 for Win32 (community server). The dedicated listener thread (not depicted) listens for incoming connections and stores them in the *NetworkConnectionKeeper* instance. Other classes exist for the maintenance of connections with the clients and detection of unresponsive clients.

11.2.2 Mobile Client

The mobile client is a J2ME application using TCP/IP sockets for Internet communication. Example screenshots from the mobile client are shown in Figure 21. The application requires a phone that supports Java MIDP 2.0 and an Internet connection. The application is tested with GPRS, UMTS and Wi-Fi as communication bearer.

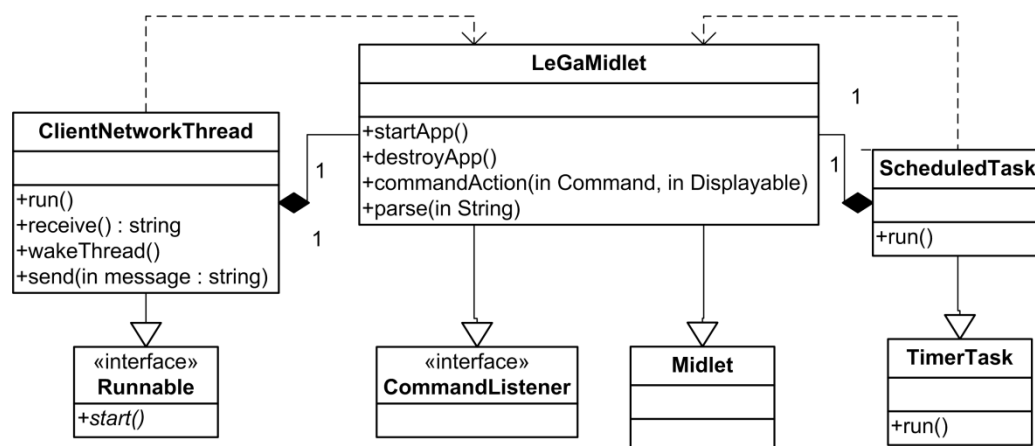


Figure 20: UML Class Diagram Mobile Client

The UML class diagram is depicted in Figure 20. The program consists of two core classes. The `NetworkThread` class and the `LEGAMidlet` class work together in a producer-consumer relationship: When there is need for network communication the `ClientNetworkThread` is woken by the `LEGAMidlet` class. When there is need for graphical rendering, the `LEGAMidlet` class is woken by the `ClientNetworkThread`. When the program depends on user input the `LeGaMidlet` thread is woken through the `CommandListener` interface. This makes it possible to process network flow without freezing the graphical rendering. For special cases of program flow such as countdown timers, and cases where both threads are sleeping simultaneously, the class `ScheduledTask` is started as a separate thread using the Java `TimerTask` class, invoking the appropriate thread for execution.

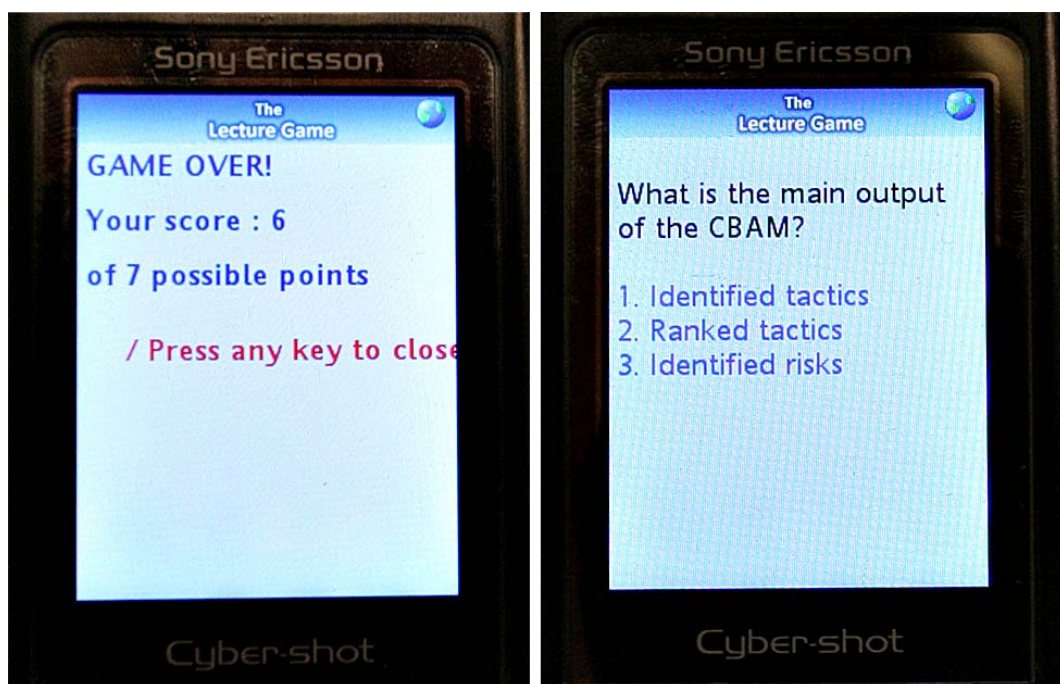


Figure 21: Mobile client screenshots

11.2.3 Web Client

The open source emulator *MicroEmulator* runs the J2ME application described above inside a Java Applet. Input is given by clicking on a graphical representation of a mobile phone. To achieve this, graphical rendering other

than the standard J2ME components are avoided due to compatibility issues. The program can run in any web browser supporting Java Applets. The web interface is displayed in Figure 22 and Figure 23.

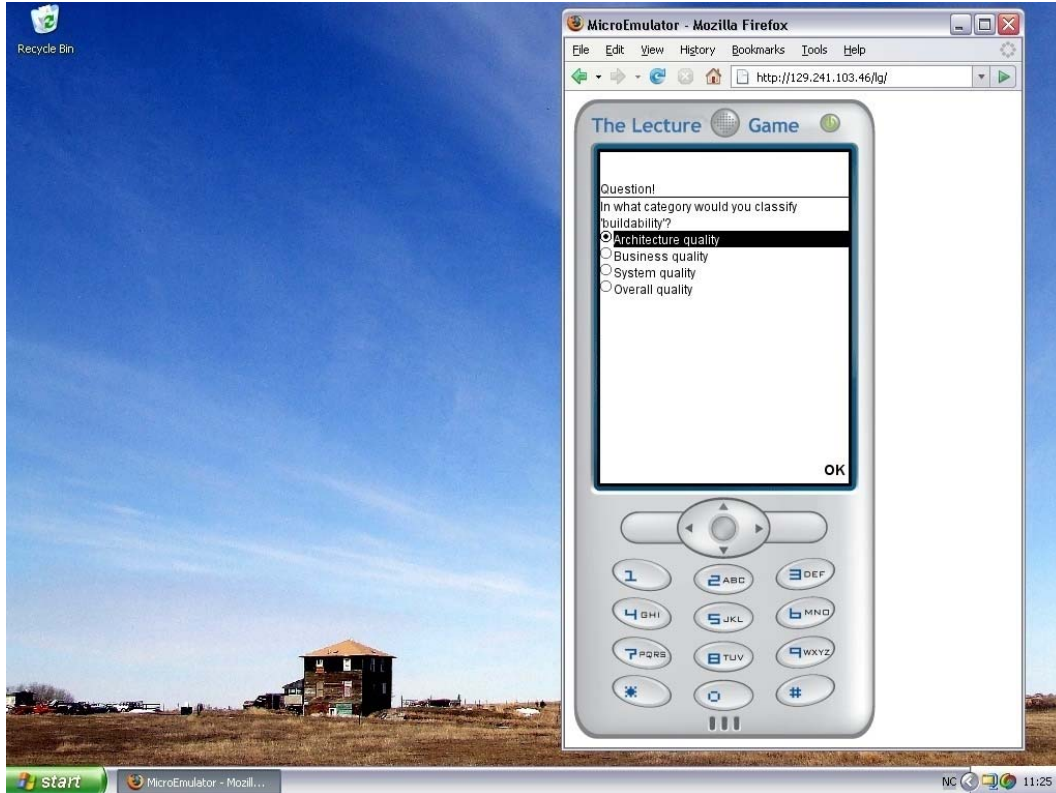


Figure 22: Web client screenshot

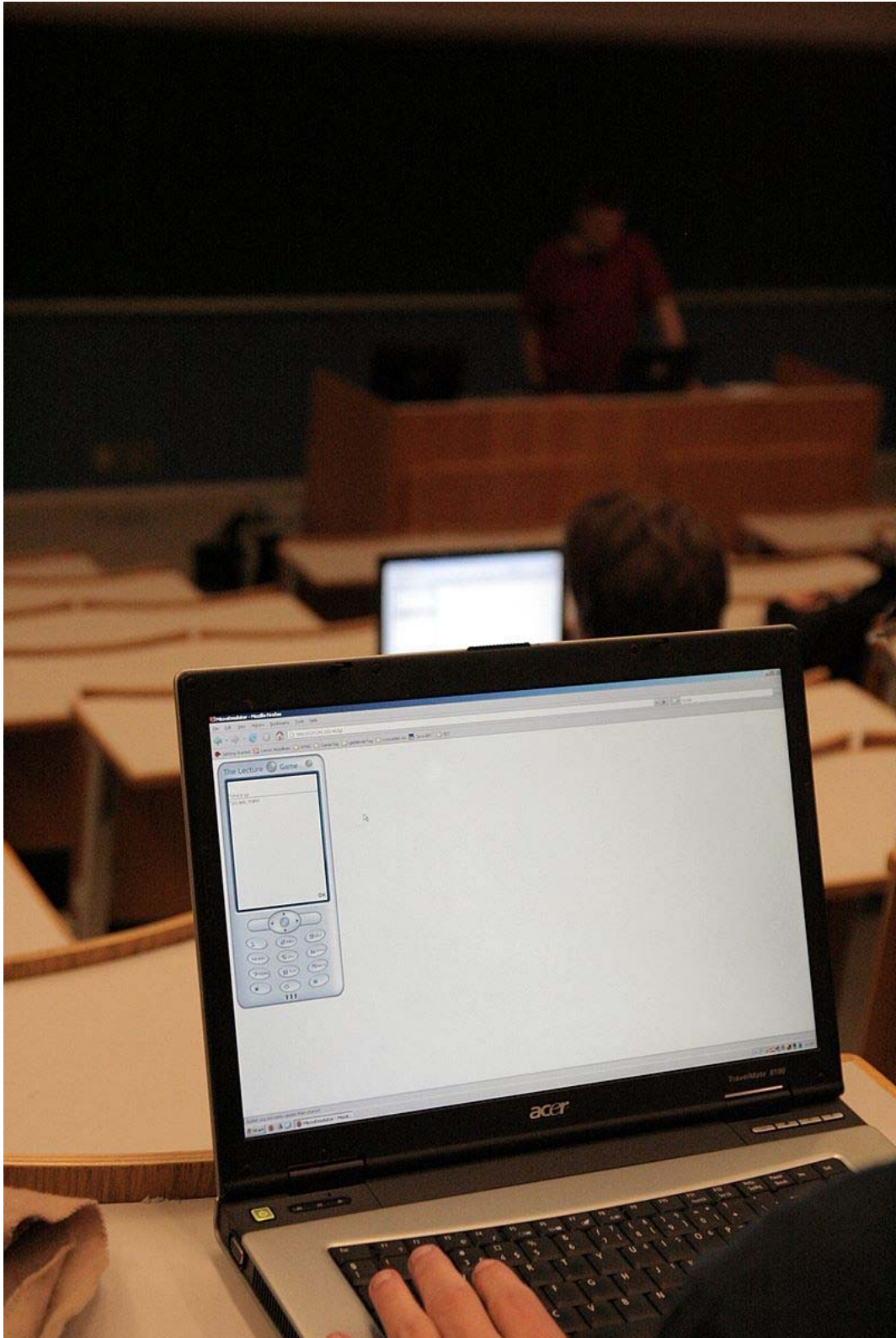


Figure 23: Web client in an authentic setting during the experiment

11.2.4 Teacher Client

The Teacher Client requires an installation of *Java Bindings for OpenGL* (JOGL), which is available from <http://jogl.dev.java.net> . The latest stable release is 1.3. A typical gameplay screenshot is shown in Figure 24.

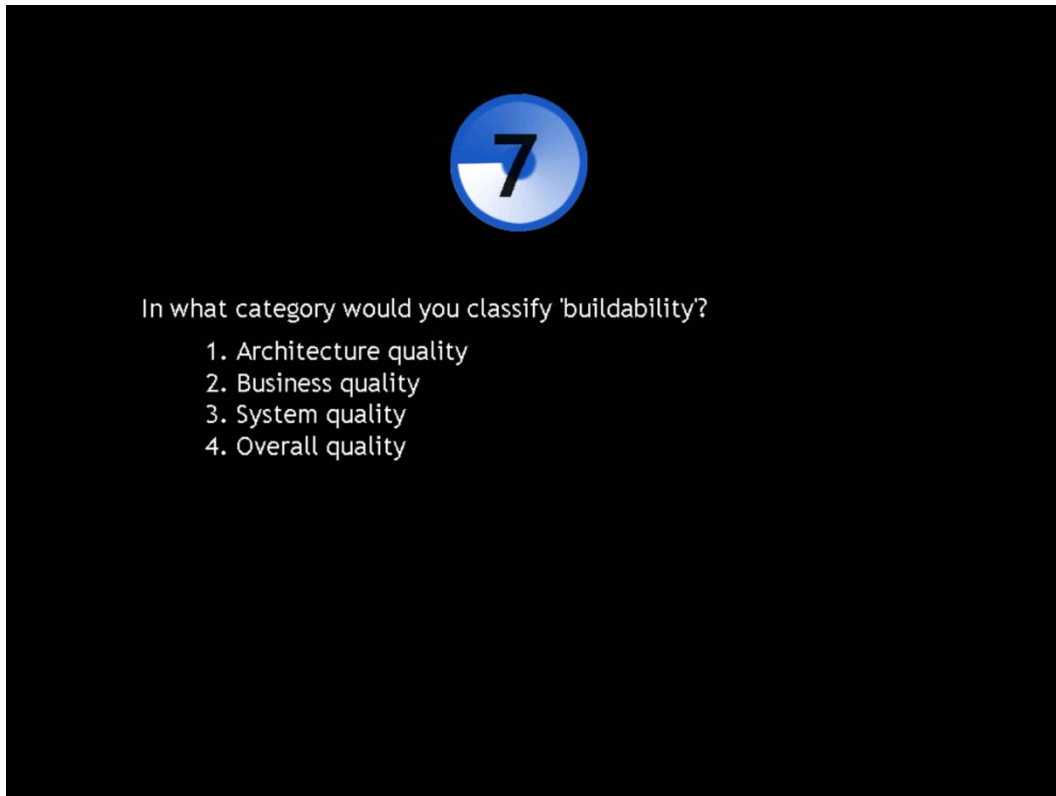


Figure 24: Teacher Client screenshot; question is posed and countdown has started.

Figure 25 shows a simplified UML class diagram of the Teacher Client. The code is divided in two packages: The package Core is dealing with network activity, parsing and basic operations. The LectureGame class is the entry point of the application and launches the game, dealing with login and initialization. The ClientNetworkThread runs at all times listening for network input and calling appropriate methods according to the parsing of the incoming messages. A separate thread, InputSimulatorThread, is not spawned during normal operation, but can be used to simulate any communication from the server. For instance it is hard to log on 100 clients

or more manually, but this thread makes it easy to test such scenarios without having the server running. All processing concerned with the actual graphical or game mode specific processing is routed to the Graphics package.

All communication between the Core and the Graphics packages is routed through the GraphicsFacade class, which is also responsible for keeping record of the incoming questions and serving them to the Renderer class. The renderer class has two responsibilities: keep a consistent state of the flow of the program and to assign appropriate methods for graphical rendering based on its state information. The Renderer class is based on the Neon Helium tutorial sets [56], as are the multitude of OpenGL-related classes. To keep the diagram uncluttered only the GLDisplay class is included in Figure 25, to represent the set of independent OpenGL-specific classes.

The program is made for easy exchange of game-modes and simple creation of new ones. This is achieved through the interface GameMode which every game mode needs to implement. The renderer calls these methods for the

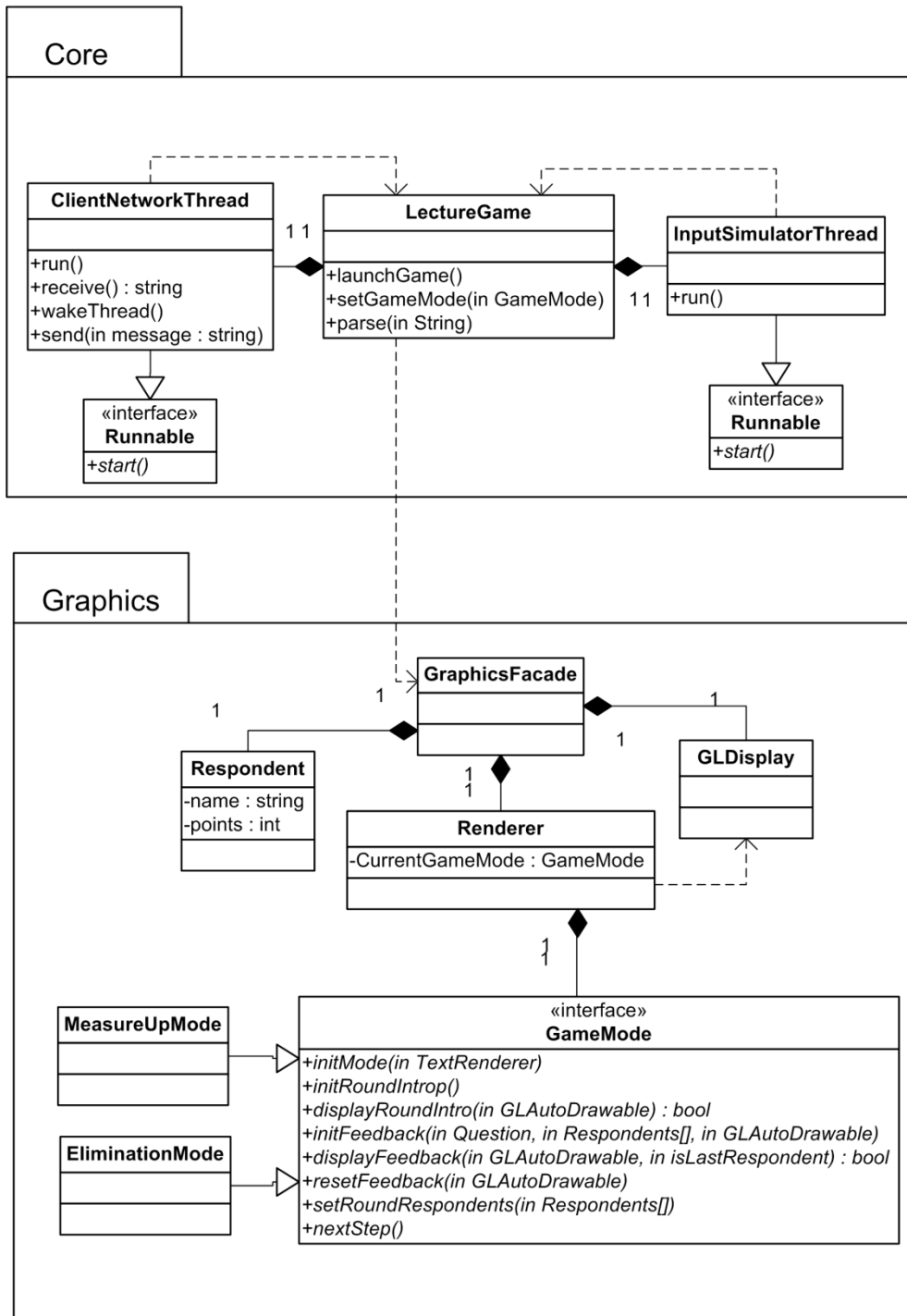


Figure 25: Teacher Client UML Package diagram

currently selected game mode with the OpenGL graphics object as input along with appropriate input such as current answers to a recently posed question. For every drawn frame, the **Renderer** keeps calling the same

method of the game mode until it returns the value false, in which case the concurrent step in the program flow is executed. The core methods are `displayRoundIntro` and `displayFeedback`. The rest of the methods are optional initialization methods. The input parameters to the `GameMode`-interface is intended to be complete and generic so that any game mode can be implemented without interfering with other code than the actual `GameMode` implementation. Two game modes are currently implemented: The Measure Up mode, depicted in Figure 26 and Figure 27, and Elimination mode shown in Figure 28 and Figure 29.

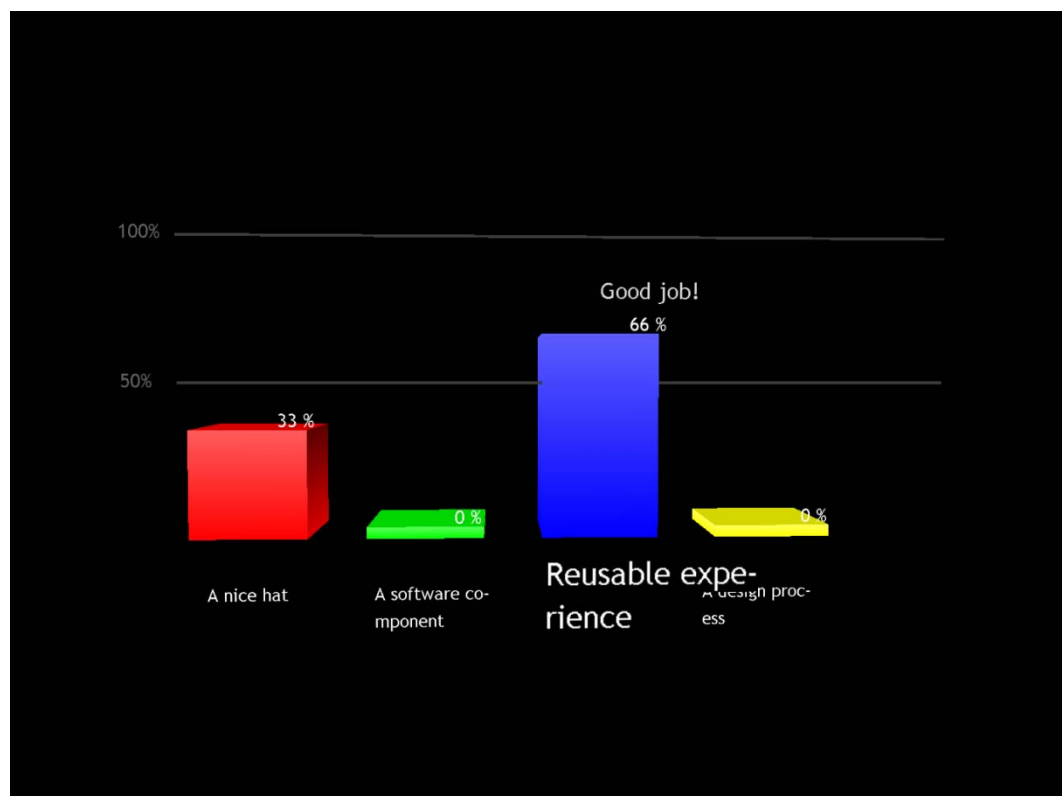


Figure 26: Teacher Client screenshot; positive feedback in Measure Up mode.

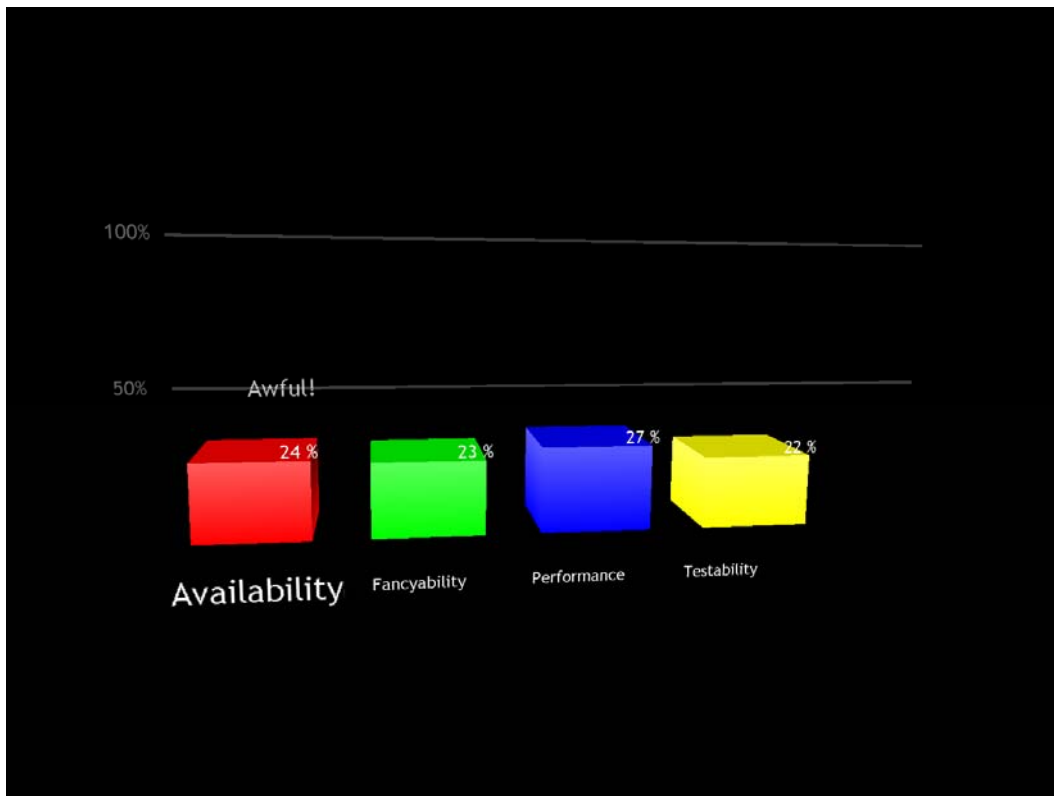


Figure 27: Teacher Client screenshot; negative feedback in Measure Up mode.

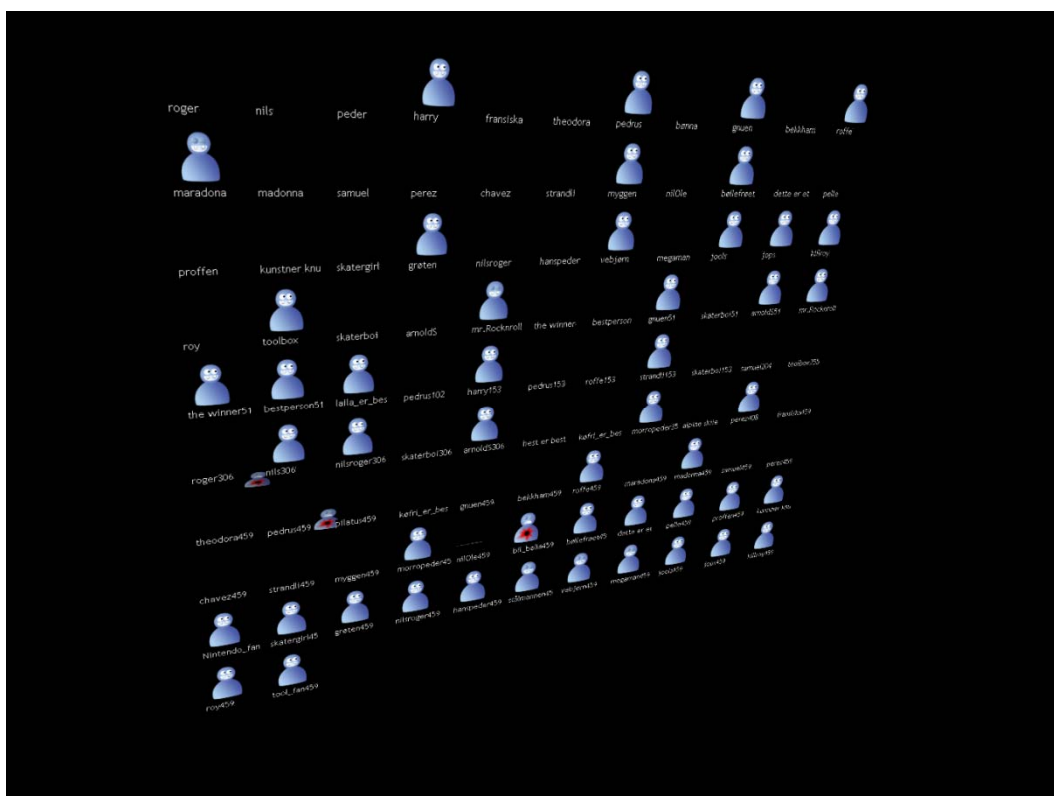


Figure 28: Teacher Client screenshot; Elimination mode with many students connected



Figure 29: Teacher Client screenshot; only one player left in Elimination mode.

Part V

The Experiment

12 Introduction

This part presents an empirical experiment where our prototype application is tested in a realistic setting. Relevant theory, a description of the experiment and the results will be presented.

13 Experiment Delimitation

The purpose of this experiment was to investigate how the Lecture Game concept is received in a real life scenario. A statistical analysis and proven results are outside the scope of our research. We measured the subjective satisfaction and idea of effectiveness and efficiency of the experiment objects. We point out trends and tendencies, but we have left a thorough psychological analysis as well as an objective measure of learning outcome to future studies.

14 Method

We used recognized methods combined with our own assessments to obtain measures of usability defined by three goals of usability, as defined in ISO 9241-11 [57]:

- Satisfaction – Freedom from discomfort, and positive attitudes towards the use of the product
- Effectiveness – The accuracy and completeness with which users achieve specified tasks
- Efficiency – The resources expended in relation to effectiveness

14.1 System Usability Scale (SUS)

System Usability Scale (SUS) is a generic questionnaire with 10 questions for a simple indication of the system usability as a number on a scale from 0 to 100 points. Each question has a scale position from 1 to 5. For items 1,3,5,7 and 9 the score contribution is given by subtracting 1 from the scale position. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. This

implies that each question has a SUS contribution of 0-4 points. Finally the sum of the scores are multiplied by 2,5 and divided by the number of replies to obtain the SUS score. SUS is proven to give similar results as more advanced measures of usability. The questionnaire is publicly available and is commonly used in a variety of research projects [58]. A complete measure of SUS is included in our questionnaire. The results will be discussed in Chapter 18.

14.2 CIF

The *Common Industry Format for Usability Test Reports* (CIF) defines the format of a usability test document. The purpose of these documents is to form a basis for business decisions and as a basis for evaluation of the system usability [57]. We include all the requirements of the CIF specification in Chapters 14 through 17. Due to layout concerns we do not follow the formatting required by a CIF compliant document.

14.3 Subjective Assessment

In addition to the 10 generic questions from SUS, we have added 11 questions specific to features of the system that are especially interesting in our experiment. These are properties of the efficiency and effectiveness goals. Responses to these questions will be subject to a non-formal evaluation to discover how the system was commonly perceived.

15 Context

This experiment tested the usability of *The Lecture Game* prototype version 1 written by Terje Øfsdahl and Ole Kristian Mørch-Storstein. The test was prepared and lead by Terje Øfsdahl, Ole Kristian Mørch-Storstein and facilitated by associate professor Alf Inge Wang. The test was conducted on May 8th 2007 at NTNU, Trondheim, Norway. The purpose of the test was to

provide empirical data regarding the suitability of game enhanced lectures and specifically the concept implemented through the prototype.

15.1 Participants & Environment

The participants in the experiment were the students of the course TDT4240 Software Architectures. Precisely 20 students participated in the experiment of a total number of 88 registered students. The student population was 85 % male, and predominantly around 23 years of age. One might suspect a high interest in technology due to the technological nature of their study. No one of the test subjects had tried the product before and there is no clear segmentation of the population. The teacher was the facilitator of the experiment and controlled the progress and alternation between game and lecture. The experiment was conducted in a lecture hall with the Teacher Client graphics displayed by a projector. Students had the Mobile Client running on their respective mobile phones and the central server was running at a different location. All applications were communicating over TCP/IP. The content being taught during the experiment was a review of the relevant literature from the entire semester in an examination preparation lecture. The participants and environment were authentic in the sense that students attending higher education lectures were the intended users of the system. A complete reference of the system setup was found in Chapter 12. Several informal prototype tests have also been conducted. Feedback from these tests is included in technical discussions, but other results only consider the actual experiment that is described in this part.

15.2 User Tasks

The user tasks are presented in the following textual use case scenarios [59]. The preparation of the lecture is beyond the scope of the experiment and is not included as a user task.

After the projector, audio system and an Internet connection was properly connected to the teacher's computer, the teacher initiated the game enhanced lecture as described in Table 2.

Table 2: Use Case Teacher

Use Case 1: Teacher facilitates
1. Present theory
2. Start teacher application
3. Trigger question
4. Receive feedback
5. Stop feedback animation
6. Repeat step 3-6
Extension
6a. no more questions : Trigger high score list

The students need access to the Internet through a MIDP 2 Java compatible device or a web browser. If the students are using a mobile phone they need to download the mobile application. This is independent of the application itself and is not described as a user task. Use Case for student participation is presented in Table 3.

Table 3: Use Case Student

Use Case 2: Student participates
1. Pay attention to lecture
2. Start application on mobile phone or web browser
3. Submit username, password, lecture code
4. Receive question
5. Submit answer
6. Receive feedback
7. Repeat step 4-7
Extension
2a. Application not installed : download application

from specified URL
7a. no more questions : Receive total score

15.3 Success Criteria

The success criteria are articulated as hypotheses listed below. Confirmation of the hypotheses constitutes success. However, we emphasize that a statistically valid confirmation of the hypotheses is beyond the scope of our experiment.

H1: The students retain more information when using the system than they do in a traditional lecture.

H2: The system is not conceived as intrusive on the lecture.

H3: The students found the system inspiring and fun.

H4: Students possess the necessary equipment to participate.

H5: The system has high usability.

16 Experiment Execution

35 students participated in the lecture where the experiment took place. 20 of these chose to participate in the experiment. The lecture was the final lecture in the subject *TDT4240 Software Architecture*. Before the experiment was conducted, theory from the entire semester was summarized. A prize was announced for the winner of the game.

The students were shown instructions on how to download the mobile client and install it on their mobile phone. Two of the students had problems and

received the application via a Bluetooth connection instead. The installation process went smoothly and only took a couple of minutes. As the Teacher Client was started, the network connection proved defect. Because of this, everyone had to walk to the neighboring lecture hall. 15 out of the 35 students left during the change of location.

There was a good and relaxed ambience in the classroom. The students chuckled as the feedback appeared on the screen, and were apparently having a good time. Due to the technical error described in Chapter 21.3, we were not able to enter the second part, eliminator mode, of the program. However, we did show a demonstration of how eliminator mode worked before the survey was conducted. Video footage of the experiment is available on the CD-ROM, delivered as part of this master thesis, along with a technical demonstration and an informal round of eliminator mode with 20 participants.

All 20 students participated in the survey. The questionnaire is found in Appendix A.1. The ten first questions constitute the SUS data gathering. The rest of the questions are subject to an open assessment of discovering central trends.

17 Statistical Results

Here we present the results of the answers to the questionnaire. The ten first questions of the questionnaire are from the SUS, thus incorporated in the SUS score. We therefore only handle the answers to the questions which are specific to our project. For the questions 11-17 the respondent could choose to answer from 1-5, where 1 is "I strongly disagree" and 5 is "I strongly agree". The questions 18-21 only had two options: true or false. In all we received 20 questionnaire forms in return from the test group. We

encouraged everyone to fill out this form, even those who didn't participate due to technical issues.

Q1 – Q10: System Usability Scale Standard Questions (SUS)

Our prototype got SUS score 74,25 of 100 possible points. Table 4 shows the average scale points and the corresponding SUS score per question. See Chapter 14.1 for an explanation of how SUS is calculated.

Table 4: SUS score

#	Question	Average	SUS/question
1	I think that I would like to use this system frequently	3,6	2,6
2	I found the system unnecessarily complex.	1,85	3,15
3	I thought the system was easy to use.	4,02	3,05
4	I think that I would need the support of a technical person to be able to use this system.	1,35	3,65
5	I found the various functions in this system were well integrated	3,2	2,2
6	I thought there was too much inconsistency in this system	1,95	3,05
7	I would imagine that most people would learn to use this system very quickly	4,35	3,35
8	I found the system very cumbersome to use	1,95	3,05
9	I felt very confident using the system	3,55	2,55
10	I needed to learn a lot of things before I could get going with this system	1,95	3,05
		SUS score	74,25

Q11: I think I paid closer attention during the lecture because of the system (Figure 30).

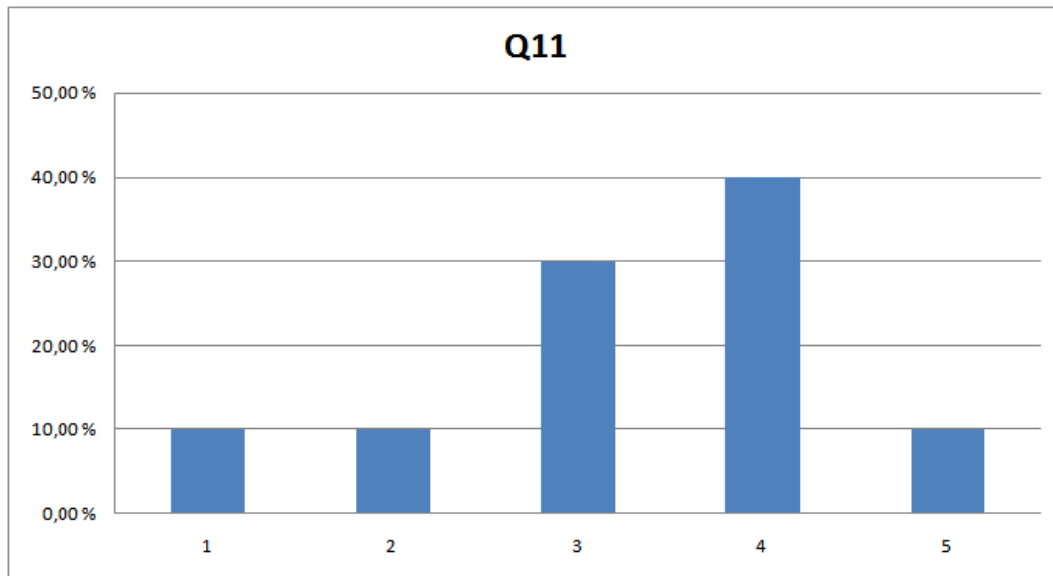


Figure 30: Q11

Q12: I found the system had a distracting effect on the lecture (Figure 31).

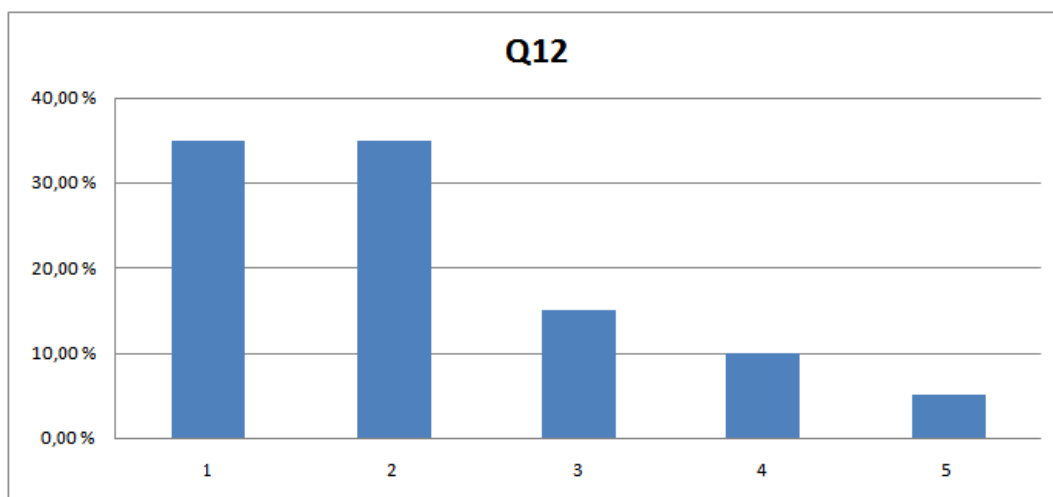


Figure 31: Q12

Q13: I think I learn more during a traditional lecture (Figure 32).

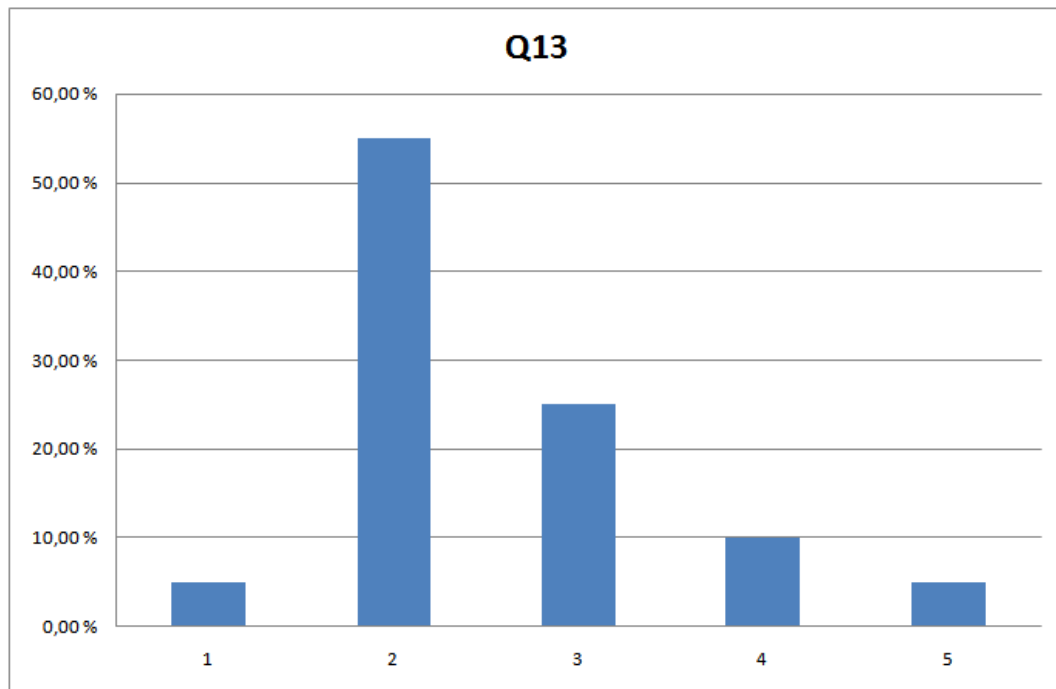


Figure 32: Q13

Q14: I found the system made me learn more (Figure 33).

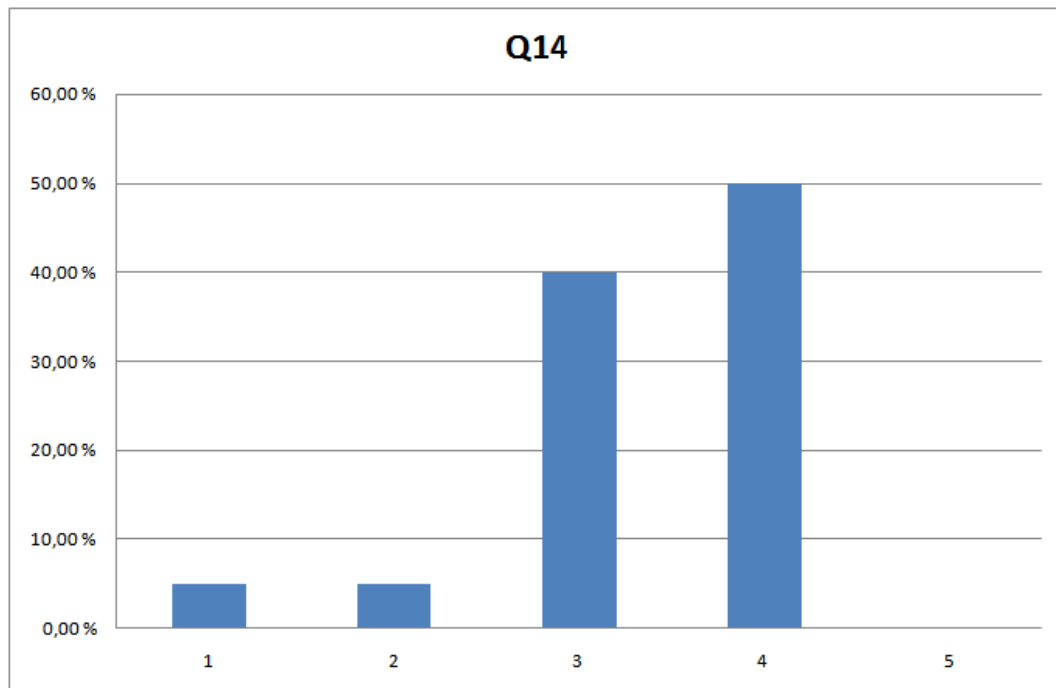


Figure 33: Q14

Q15I: found the system made the lecture more fun (Figure 34).

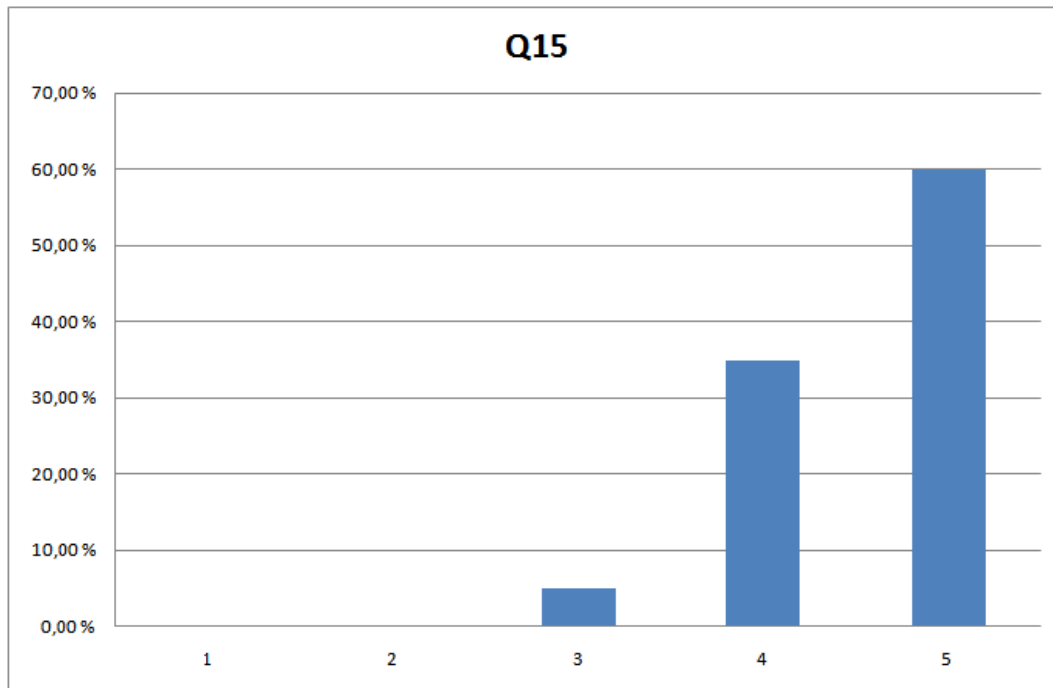


Figure 34: Q15

Q16: I think regular use of the system will make me attend more lectures (Figure 35)

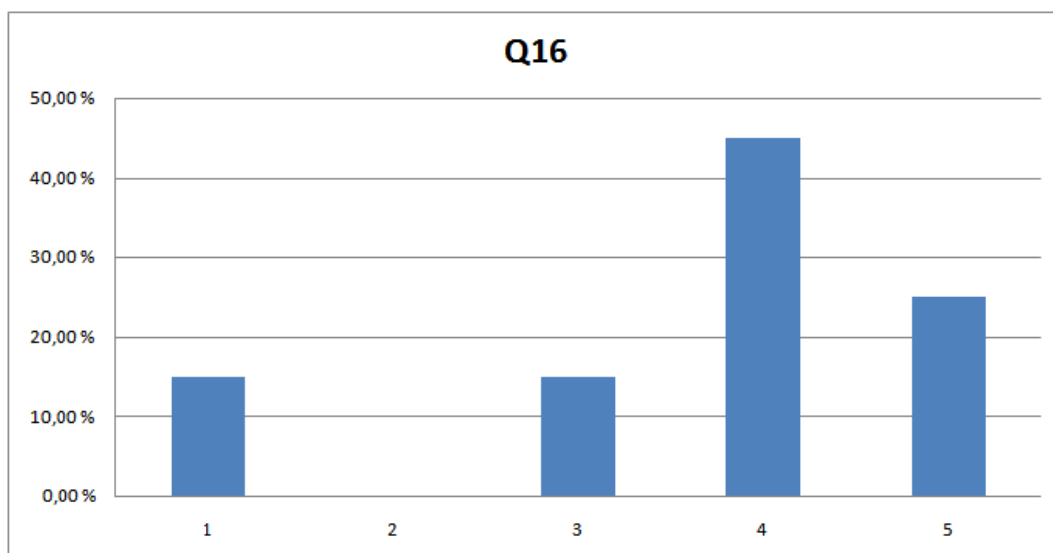


Figure 35: Q16

Q17: I feel reluctant to pay 0,50 NOK in data transmission fee per lecture to participate in using the system (Figure 36).

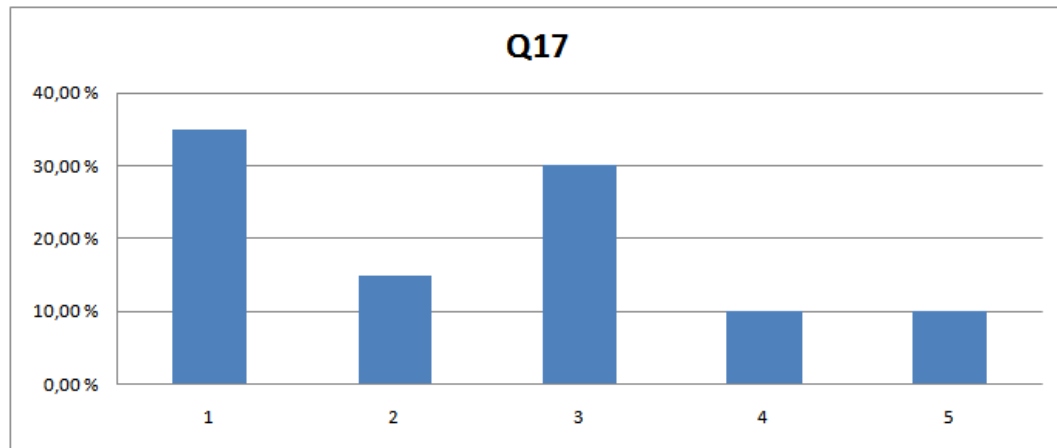


Figure 36: Q17

Q18: Did the client software work properly on your phone (Figure 37)?

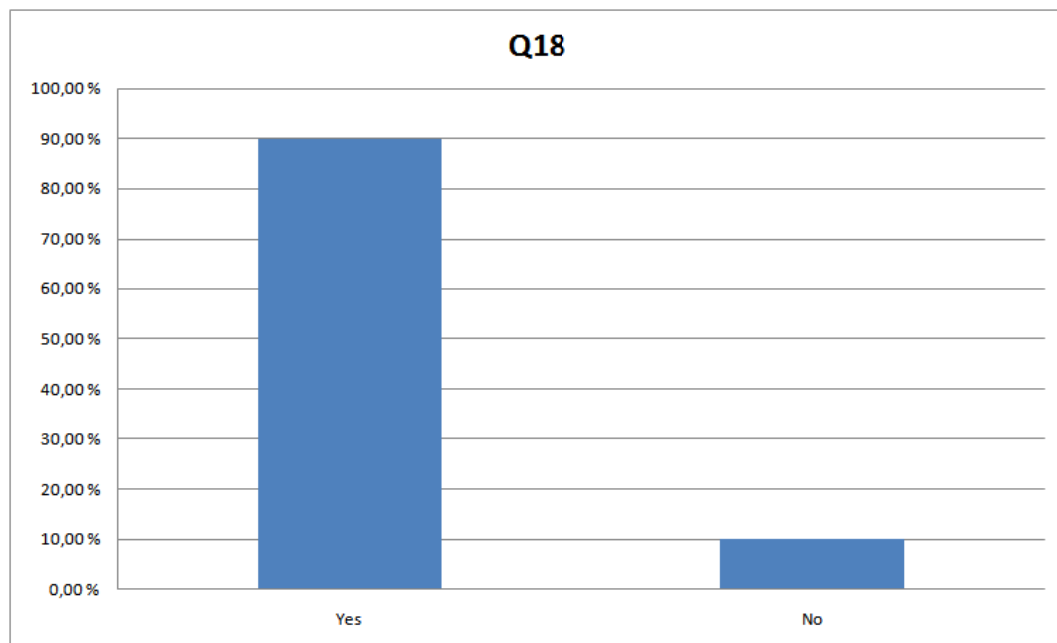


Figure 37: Q18

Q19: I own a laptop with Wireless LAN (Figure 38).

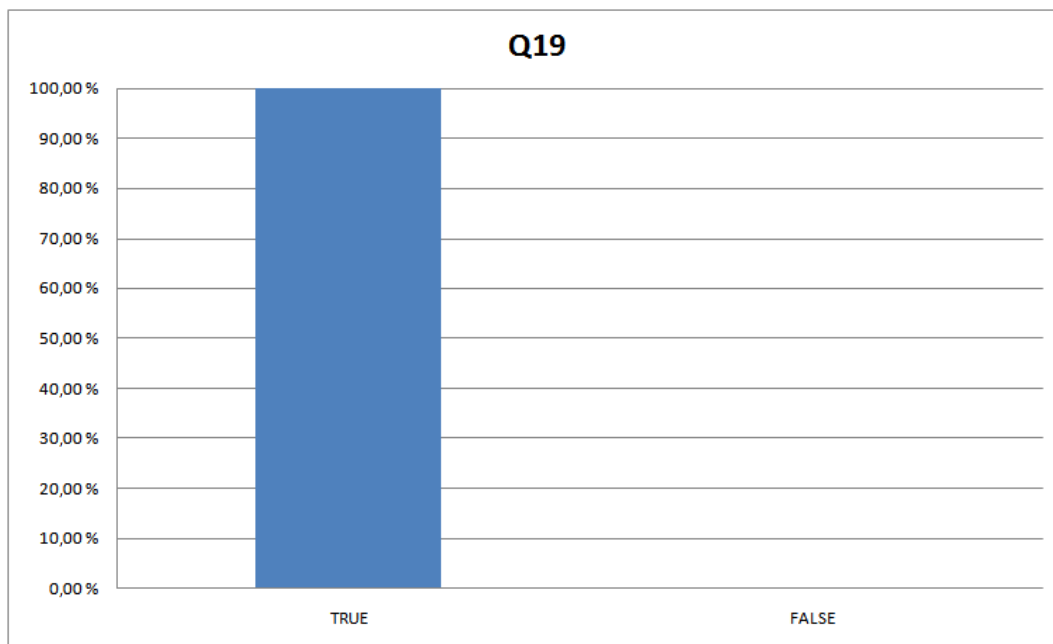


Figure 38: Q19

Q20: My mobile phone has Bluetooth support (Figure 39).

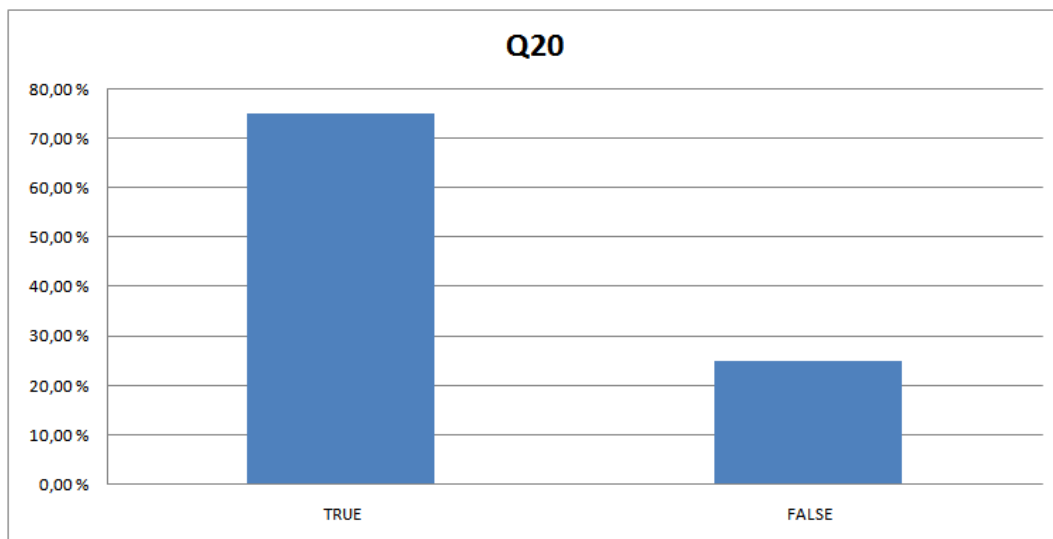


Figure 39: Q20

Q21: I used a laptop to participate (Figure 40).

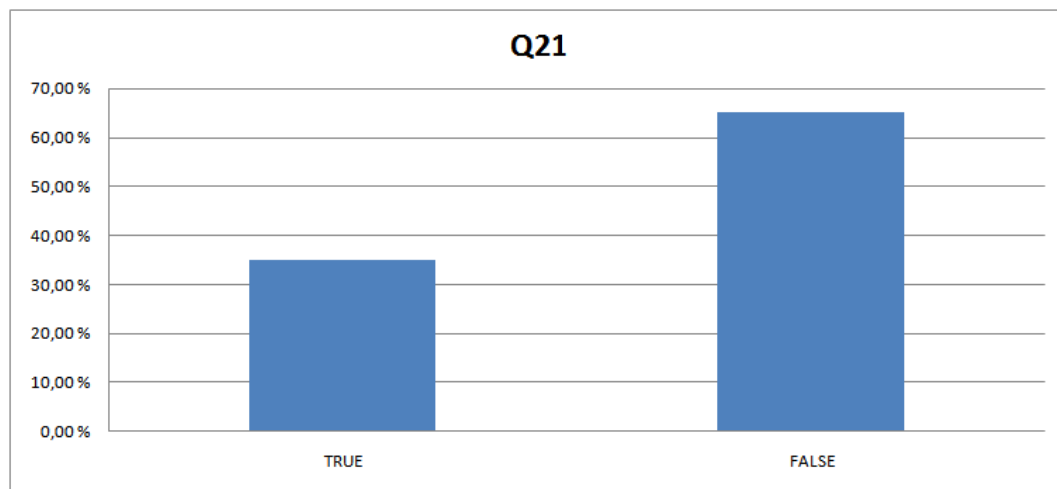


Figure 40: Q21

18 Evaluation

The discovery of trends and opinions in our survey are presented here and evaluated along with written and oral feedback from test. The results are presented for each success criteria.

H1: The students retain more information when using the system than they do in a traditional lecture.

50% of the participating students agreed that they were paying closer attention to the lecture when using the system, while 20% showed skepticism on this point (Q11). In our experiment the interaction with the prototype was not interleaved with the actual lecture, but postponed until the end. However, the students knew they were going to be tested, and in the opinion of half the students it had a positive effect on their concentration. The novelty factor has to be taken into consideration as the students did not know what to expect from the prototype. If using the prototype makes half the students pay closer attention, this property alone makes the system very useful. However, further analysis is needed to make such a conclusion.

With only 15% stating they thought they learn more during traditional lectures (Q13), and 50% thinking they learn more using the prototype (Q14), the students seem to believe in the pedagogical concepts implemented in the prototype. However, 40% did not state a positive nor negative opinion concerning increased learning outcome. This might be connected to the malfunctioning of the program during the experiment or the need for further usage before they make themselves an opinion. However, the students own assessments of learning outcome may not represent the actual learning outcome. We did not have the opportunity to perform an objective and statistically valid experiment measuring actual learning outcome. Our results merely give an indication of the subjective learning effect.

H2: The system is not conceived as intrusive on the lecture.

70% of the students did not find themselves distracted from the lecture when using the system, while 15% agreed of being distracted (Q12). The fact that the students had to change lecture hall may in addition to the technical problems have had a negative impact on our results. Further research is necessary to obtain a valid result, but the vast majority did not find the system intrusive.

A second testimonial that the students did not seem to dislike the usage of the systems were stated as 75% thought they would attend more lectures if the system was frequently used (Q16). 45% agreed to some extent to this statement, while 25% strongly agreed. As a preliminary result, we consider this result to exhibit a very positive initial reception.

H3: The students found the system inspiring and fun.

When asked whether the prototype made the lecture more fun, 95% of the students agreed to some extent with the statement. 60% of these students

stated they *strongly agree* (Q15). The last 5% were neutral. These results show that the prototype had the intended entertaining effect on the students, although the novelty factor, as mentioned earlier, may have positively biased the results.

H4: Students possess the necessary equipment to participate.

Only 10% of the students experienced problems with the client application. From the comments of these students it seems one of the students blames the server error and the other had problems running the web client in his browser. 100% possess a laptop with wireless internet connectivity, but as some of the students may have left because their phone was not compliant, we may not draw conclusions as to how many possess a MIDP 2.0 compliant phone.

H5: The system has high usability.

A *System Usability Scale* score of 74,25 indicates overall good usability. The lack of a comparative study with a similar product prevents us from drawing specific conclusions. We encourage further research on similar products to use this score for comparison.

18.1 Feedback Evaluation

We received both written and oral feedback from the experience and other prototype testing. Some students with phones lacking a high resolution display requested the possibility to scroll when the question and alternatives were too long to be displayed on their screen. However, they were not inhibited from answering as all information was displayed on the lecture hall screen. Several students commented that the correct answer was not stated in elimination mode, preventing the students from learning. There were also some confusion of which bar represented the right answer, even though it

was flashing and the accompanying text was animated. One student suggested that the other bars could be grayed out to further clarify the correct answer.

The experiment also showed that 10% of the students felt reluctant to pay 0.50 NOK in transmission fees to participate in a lecture (Q17). However, 0,50 NOK is the cost of downloading the client application and 45 minutes of participation for the most expensive subscription available. Further the experiment shows that 75% of the students own a phone that features Bluetooth support (Q20). A Bluetooth implementation of the prototype would make the system free of charge, but previous Bluetooth solutions require installation of several Bluetooth access points in the lecture hall affecting availability and ease of use from the perspective of the teacher negatively [27].

During the experiment we did not encounter technical problems on the clients except for one student having problems with slow execution on the web client. During informal prototype testing the phone I-mate SP5 resulted in vibration that did not stopped until another vibration was triggered. We believe this to be an issue with the MIDP 2.0 implementation on this particular phone. Informal testing also showed that some people lost their connection to the server, and GPRS/3G delays of up to 10 seconds. Notable GPRS and 3G delays were discovered only for students subscribing to supplier Telenor.

19 Experiment Conclusion

During the execution of the experiment, technical problems made the experience incomplete and more tedious than intended. The prototype was well received by the students, but we suspect an even more positive result could be achieved if the experiment was problem free. The results of the experiment indicate the following trends:

- The students think they retain more information when using the system than they do in a traditional lecture.
- Most students do not find usage of the prototype intrusive on the lecture.
- The students found the system inspiring and fun.
- Students possess the necessary equipment to participate.
- The system has high usability.

All of the success criteria were confirmed by the test subjects. Success criteria H2 were only agreed upon by 70% of the students. Therefore the resulting trend above differs slightly from the initial success criteria.

Part VI

Evaluation

20 Evaluation

This chapter evaluates our iterative method of development and our general research method, as well as how we executed our research and development according to the given guidelines. An evaluation of the experiment execution and the prototype is included in Part V.

20.1 Research method

We chose to use the engineering method in our research. After declaring our research problems in Chapter 4, we summarized important findings from a literature search in relevant authoritative literature. The assertions build directly on published works and rely on the respective validations performed by the different authors. The taxonomy in Figure 6, the extended Malone model in Figure 5 and the genre suitability evaluation in Figure 7 are based on our subjective assessments and are not to be interpreted as validated facts. However, they build upon the findings from our literature search. A validation of the quality of the taxonomy is included in our depth project [1]. The other two theories are not meant to reflect theoretical facts, but to give an insight into our assessments and are not candidates for theoretical validations. On the contrary we encourage further improvement and critics of these models in future research.

The prototype was designed based on our subjective assessments and historically validated theory. Even though the prototype design is partly based on historically validated theory, it is mostly based on our subjective ideas of a success solution. Creation of computer games is a creative task that can only be validated by empirical studies. The engineering method has given us the possibility to experiment with our own ideas combined with findings in theory, which seems reasonable when performing research in the borderline of creative and theoretical research.

Ideally a thorough and statistically valid empirical experiment should have uncovered qualities and weaknesses with our solution. Due to limited resources and time we have chosen only to discover the indication of trends towards validating our findings. Even though the final results are statistically frail, we have witnessed a relatively uniform pattern in the answers, which makes the findings candidate for further improvements and may trigger experiments to confirm our findings with more statistical integrity.

20.2 Development method

We chose to use an agile variant of the iterative development method in this project, with a focus on prototyping and short, incremental development steps. We regard this process as highly successful. During the whole project we held short fifteen minute standup meetings each morning, prioritizing the goals for the day and summarizing progress from the day before. This enhanced the process vastly, and increased communication amongst us as a team as well as overall motivation. The short iterations and quick development of a working prototype was also a motivating factor by enabling us to quickly see the results from our work.

As with any process we see the room for improvement. The testing process was at times tedious and complicated, with the server, student clients and teacher client having to be operated at the same time. At later stages in the project some of the aspects of this testing process were automated with e.g. the stress testing module, a tool developed specifically for testing server durability. In retrospect, we should have written some of the tests beforehand, and started developing testing tools earlier.

21 Lessons Learned

This chapter describes problems we encountered, and issues we resolved.

21.1 GPRS/3G Synchronization optimization

Already during the planning of the system, we were aware of initial latencies with an upper bound of 6-10 seconds for TCP/IP packets sent over GPRS [60]. These frequent and long delays were undesired in the prototype implementation as it would make the game unfair and cumbersome to use. In an effort to minimize the perceived delay, we adjusted our initial prototype and experimented with ways to minimize the latency.

Subscription: Chess 3G

Date: February 2, 2007

Roundtrip time	Sleep before send
1953 ms	0 ms
1219 ms	0 ms
1234 ms	0 ms
1219 ms	0 ms
1218 ms	250 ms
969 ms	500 ms
719 ms	750 ms
469 ms	1000 ms
453 ms	1250 ms
438 ms	1500 ms
453 ms	1750 ms
438 ms	2000 ms
453 ms	2250 ms
437 ms	2500 ms
422 ms	2750 ms
422 ms	3000 ms
422 ms	3250 ms
406 ms	3500 ms
406 ms	3750 ms
406 ms	4000 ms
391 ms	4250 ms
390 ms	4500 ms
391 ms	4750 ms
375 ms	5000 ms
359 ms	5250 ms
360 ms	5500 ms
343 ms	5750 ms
469 ms	6000 ms
328 ms	6250 ms
453 ms	6500 ms
453 ms	7000 ms

Subscription: Tele2 3G

Date: February 2, 2007

Roundtrip time	Sleep before send
1828 ms	0 ms
1848 ms	0 ms
1828 ms	0 ms
1704 ms	0 ms
1718 ms	250 ms
1578 ms	500 ms
1203 ms	750 ms
969 ms	1000 ms
703 ms	1250 ms
813 ms	1500 ms
703 ms	1750 ms
703 ms	2000 ms
813 ms	2250 ms
703 ms	2500 ms
687 ms	2750 ms
672 ms	3000 ms
672 ms	3250 ms
1156 ms	3500 ms
782 ms	3750 ms
1932 ms	4000 ms
765 ms	4250 ms
2125 ms	4500 ms
1969 ms	4750 ms
1500 ms	5000 ms
1468 ms	5250 ms
2329 ms	5500 ms
2204 ms	5750 ms
1594 ms	6000 ms
2797ms	6250 ms
2047 ms	6500 ms
1672ms	6750 ms

Table 5a (left) & b(right): Test of optimal packet interval. Roundtrip time is measured after waiting for the indicated number of milliseconds. New data is not send before receiving the ping answer

We realized that the initial latency differed considerably between the different mobile network operators see Table 5 and Table 6. For some operators the initial delay was at times seconds longer than for others. However, this was only a problem for the first packets sent after a longer break. By sending small amounts of data directly after the mobile client returned an answer, the average latency was dramatically reduced. For instance we observed that the initial roundtrip latency using a Chess GPRS connection is measured to 1828 milliseconds. After sending a second data packet, the roundtrip latency was reduced to 344 milliseconds.

To synchronize the slowest clients by sending one packet of dummy data, a delay of up to 10 seconds was necessary. This intentional delay is shown in Figure 41 as **Δ_{wait}** . But because 10 seconds is substantially longer than the initial delay of the average client, an additional unnecessary delay was observed in these clients in addition to the 10 seconds of the intentional delay. After evaluating the trade-offs from introducing a conscious delay (Δ_{wait}) versus assuming lower average delays for all clients, we decided to send one byte of dummy data and the actual data with an interval of 2 seconds. The Δ_{wait} value of 2 seconds was found after studying measurements such as those presented in Table 5a and Table 5b. This concept is illustrated and further explained in Figure 41. The technique reduces the difference in latency with approximately 1,0-1,5 seconds for the clients experiences long initial delay. An example of how this technique reduced initial latency for clients using the Norwegian operator Telenor with a 3G connection is shown in Table 6. The delay was reduced from a maximum of 7-8 seconds to approximately 300 milliseconds.

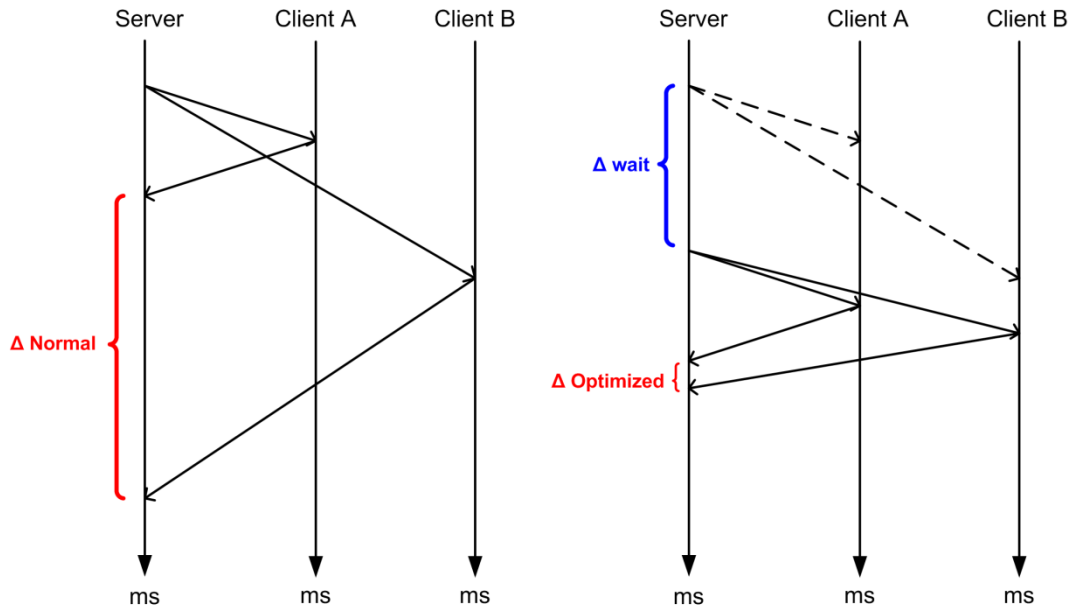


Figure 41: Illustration of GPRS/3G synchronization between the server and a client with a fast connection, Client A, and a client with a slow connection, Client B. (left) The first packet of data is sent and a large difference in round trip time is typically observed between Client A and Client B. (right) Dummy data is sent to both clients. After a predetermined delay, the actual data is sent resulting in a significantly smaller difference in round trip time between the two clients, thus masking the initial delay to the users.

The implemented delay of 2 seconds is camouflaged in the Teacher Client by playing the sound of a bell and displaying the timer graphics as a question is triggered. To further increase the perceived synchronization, the Teacher Client waits an additional 300 milliseconds before displaying the question and starting the timer. This is done to conceal the overhead delay of GPRS/3G compared to the Ethernet over twisted pair connection of the Teacher Client.

Subscription: Telenor 3G

Date: March 12, 2007

Algorithm	Roundtrip time	Sleep before send
Optimized	203 ms	0 ms
Plain	7500 ms	11932 ms
Optimized	328 ms	0 ms
Plain	8375 ms	10146 ms
Optimized	312 ms	0 ms
Plain	7953 ms	9931 ms
Optimized	219 ms	0 ms
Plain	8407 ms	14981 ms
Optimized	218 ms	0 ms
Plain	15531 ms	13037 ms
Optimized	313 ms	0 ms
Plain	5360 ms	12732 ms
Optimized	281 ms	0 ms
Plain	1593 ms	5473 ms

Table 6: Testing of optimization algorithm. Rows marked *Optimized* indicates that dummy data is sent 2 seconds prior to a ping that is routed back to the server before measuring. Rows marked *Plain* indicate a direct ping after waiting for a random interval between 5 and 15 seconds. No data was sent before starting the test.

Between early prototyping and actual testing of the finished prototype, Telenor dramatically reduced the initial delay for their GPRS/3G subscribers. It seems that problems of initial delays are currently being solved by operators. Our method may, however, come to more significant benefit in other locations, and for subscribers of other mobile phone operators.

21.2 Testing a Distributed System

From the very beginning it was clear that our prototype would consist of several components, all communicating with each other via network connections. Early in the project, in the infancy of the prototype, it was fairly

trivial to carry out tests on the system as whole. We used the J2ME emulator bundled with the Wireless Toolkit from Sun to run the mobile clients. These were connected to the server to e.g. see if questions were pushed, or similar tests. The test could be done swiftly, and it was quickly seen if the tested functionality worked. This involved starting all the separate subsystems to test a single component. We wanted to eliminate some of the time dissipated with the overhead created by this process. Thus we implemented simulation of connections to the server on the teacher client. This empowered us to develop components such as the graphical user interfaces without having to spend time on starting the server and logging on using usernames and passwords. A similar system was implemented on the teacher client, which then allowed us to test it without running the server. On the student (mobile) client, however, we chose not implement such a simulation due to this being a simple thin client. Simulation of server responses would involve porting much of the server functionality over to the mobile client, which we deemed too time consuming, or hard coding of server responses, which might actually cascade bugs and create surprises for us later in the project.

Another step towards simplifying the test process was the creation of an automatic stress test module, which logs on a given number of student clients to the server. Each student client can be regarded as a different, albeit simple, agent, generating traffic by logging in, answering questions etc. The Artificial Intelligence (AI) of these agents simply consisted of the ability to register a new user if necessary and replying to questions at a random time within or after the given time limit or not replying at all. This was sufficient for our purpose of testing how the server responded under the strain of multiple connections at the same time, and saved us the time of gathering a group of volunteers to do the same test. At the most we tried to have 1500 clients connected, which worked flawlessly. Server CPU load was only 10% and the application occupied roughly 100MB of RAM. The only issue was with the teacher client painting all the 1500 avatars in Eliminator mode, something our developer machine did not have a powerful enough graphics card to do.

At critical intervals, however, it was necessary to test the system as a whole. This was e.g. at the end of a prototyping iteration where we had to see how the different components of the system actually interacted in a real environment. Many critical factors of the actual live system were difficult to accurately simulate, such as the properties of a GPRS connection.

21.3 Last minute changes

Making certain that everything should run smoothly was an absolute priority when preparing for the experiment. In the two weeks preceding the demonstration, we tried to test the prototype as thoroughly as possible to uncover any remaining bugs or issues. We ran long running tests with student clients logging on and off hundreds of times to test the stability of the server. We ran stress tests with the stress test module. We tried to find special cases with users logging on and off at unusual times, or terminating the student clients at critical times. We gathered friends and coworkers and tried to make the system crash or unstable. Naturally we discovered some issues that had eluded us during our small scale testing, and testing with simulated clients. An example of this was that we discovered that the clients logging on in the middle of a Measure Up round did not receive any points for answering questions correctly. This was quickly corrected.

In the same time period, we added a console window to the server to make the server application more user-friendly when running it outside a developer environment. This console window was implemented in Java SWING and had fairly limited functionality, i.e. printing text to screen.

The first part of the experiment ran flawlessly, with students managing to log on with little or no help at all. All communication between the clients and the server went smoothly. But then suddenly, about three thirds of the way through the first round in the lecture something went terribly wrong. The

teacher client stopped receiving feedback from the server. We quickly went ahead to try and fix the issue. We started an elimination mode demo lecture we had made previously for another occasion. The first question arrived as expected, but then no more communication was received from the server. It had crashed. After wrapping up the experiment we proceeded to check what had happened at the server. We discovered quickly that it was the Java Virtual Machine (VM) which had crashed. After investigating the crash logs generated by the VM, we came to the conclusion that the culprit must have been the console window. The sheer volume of messages passed to the window had caused an unexpected error. We were able to reproduce this error with our stress tester by connecting a very large number of clients which in some rare cases caused the window to become unresponsive. After eliminating this window, we have been unable to reproduce this error again, even with 1500 connected student clients connected over a long period of time.

The lesson learned from this event is that one should not add non-essential features such a short while before a critical delivery. We should rather have created the console, but waited with integrate it with the server. The console, albeit simple to create in Java, relies on more complex operating system specific native code, which indirectly adds to the complexity of the component, and also makes it more error prone.

Part VII

Conclusion

22 Conclusion

In this thesis, we have studied the desired characteristics of educational games and heuristics for making them intriguing. These have been used to form a game concept with extrinsic fantasy, which means the game is independent of the theory it teaches. The outcome of this research answered our research problems in the following way:

RP1: What considerations have to be made when making a lecture game?

A document with usage guidelines for lecture enhancement games, based on authoritative literature in pedagogy and empirical studies of educational games, was created. Increasing level of difficulty, flaunted enthusiasm and constructive theoretical debriefings are key elements.

A discussion of how the potential of lecture enhancement games can be increased, and how our prototype supports these properties, was provided. High availability, ease of use, entertainment and subject matter credibility are top priorities along with the implementation of sound pedagogical methods.

RP2: Which technologies are suitable for game enhanced lectures?

The prototype implementation shows that GPRS/3G is a suitable technology for client server data transmission in a lecture enhancements system. However, slight differences in reception times for broadcasted data, must be taken into account.

For the system implementation free software tools are sufficient. In our implementation these include OpenGL, Java Bindings for OpenGL, MySQL, Eclipse, Sun Java Wireless Toolkit and MicroEmulator.

RP3: Implement a fully working prototype of a lecture game.

The final prototype has been tested with random input for several hours consecutively without any errors or halts indicating a fully working prototype. However, we have not performed the proper test to exclude errors as a consequence of slow GPRS disconnections, or other network related irregularities.

The prototype features:

- Broadcast of multiple choice questions, and reception of answers for game mode specific processing
- Animated feedback on a public screen and sound effects
- 2 different game modes which may be combined in a session as desired
- Easy creation of new game modes
- No need for equipment or software other than that which is normally available in any given lecture hall
- Tested with 20 real players in an actual classroom
- Tested for 1500 simultaneous artificial users

RP4: What benefits can be observed from the usage of game enhanced lectures?

From our experiment we have uncovered the following preliminary trends after one session of 20 participants in an authentic setting:

- The students think they retain more information when using the system than they do in a traditional lecture.
- Most students do not find usage of the prototype intrusive on the lecture.
- Most students think regular usage of the game would encourage them to attend more lectures.
- The students found the system inspiring and fun.

- The system has a high usability score (SUS)

RP5: What disadvantages may result from the usage of game enhanced lectures?

From our experiment we recorded that some students (15%) found usage of the prototype intrusive on the lecture. Some of the students (20%) were reluctant to pay the necessary data transmission fees to their mobile phone operator company.

23 Further Work

Our research has focused on the development on a prototype and its reception among students. We suggest further research focus on additional game modes further enhancing the idea of lecture enhancement games as well as the willingness to incorporate such methods into a classroom setting. Also we encourage future research to focus on how lecture enhancement games can gain accept in educational institutions. Due to the fact that the developed system is merely a prototype some features remain to be implemented such as server support for simultaneous lectures and a graphical interface for the teacher.

23.1 New game modes

Two different game modes are currently implemented in the prototype. However, the architecture is designed for easy creation of new modes. With more time, more astonishing graphics and more entertaining modes of feedback can be implemented.

During the months we have worked on this thesis, the latency issues of GPRS and 3G have been reduced dramatically by the largest mobile phone operator in Norway. This phenomenon opens possibilities for students to be rewarded

for fast answers, or other time dependent processing without the game becoming unfair.

23.2 Acceptance amongst teachers and institutions

The object of our experiment has been to measure and detect the opinions of students. However, the attitudes of educators and willingness to incorporate lecture enhancement games into lectures in organizations are left for further research. Missing features or poor usability from a teacher's perspective might be discovered in experiments or research directed towards teachers and organizations.

23.3 Question editor

A critical part of the game concept is the questions. Both the quality and variety of the questions are important to the success of our prototype in a real life situation, where the existence of good tools is critical as described in [61].

Currently questions have to be added to the database manually. This is a tedious and error prone process, something which is not useful when adding questions frequently. Thus we see the need for a tool supporting the creation of lecture and adding questions. This tool could also be used for rearranging questions, activating lectures and creating new lectures based on old ones where individual questions are replaced. We would highly recommend any further development of our prototype to include the creation of such a tool.

23.4 Simultaneous lectures support

The server currently only supports one lecture running at the time, although the database can be accessed by several different servers running at the same time. There are two approaches for enabling several lectures to run

simultaneously: The first is to run several instances of the server application, each with a different address, and then modify the student clients to ask for the sever address when connection. The second approach is more elegant. The server has one unique address, as is the case currently. The clients connect to the server, and they have to enter a lecture code to choose the correct lecture. This can be accomplished in many ways; one is to have a master server spawning new server as needed when lectures have to run in parallel.

Appendix

A.1 Questionnaire

Questionnaire Lecture Game – Software Architecture May 8th 2007

Male Female

Gender

☐☐

What brand is your mobile phone?

What model is your mobile phone?

**Strongly
disagree**

**Strongly
agree**

1. I think that I would like to use this system frequently.

☐☐☐☐☐

2. I found the system unnecessarily complex.

☐☐☐☐☐

3. I thought the system was easy to use.

☐☐☐☐☐

4. I think that I would need the support of a technical person to be able to use this system.

☐☐☐☐☐

5. I found the various functions in this system were well integrated.

☐☐☐☐☐

6. I thought there was too much inconsistency in this system.

☐☐☐☐☐

7. I would imagine that most people would learn to use this system very quickly.

☐☐☐☐☐

8. I found the system very cumbersome to use.

☐☐☐☐☐

9. I felt very confident using the system.

☐☐☐☐☐

10. I needed to learn a lot of things before I could get going with this system.

--	--	--	--	--

11. I think I paid closer attention during the lecture because of the system.

--	--	--	--	--

12. I found the system had a distracting effect on the lecture.

--	--	--	--	--

13. I think I learn more during a traditional lecture.

--	--	--	--	--

14. I found the system made me learn more.

--	--	--	--	--

15. I found the system made the lecture more fun.

--	--	--	--	--

16. I think regular use of the system will make me attend more lectures.

--	--	--	--	--

17. I feel reluctant to pay 0,50 NOK in data transmission fee per lecture to participate in using the system

--	--	--	--	--

Yes No

18. Did the client software work properly on your phone?

--	--

If no; describe the problem...

--

True False

19. I own a laptop with Wireless LAN

--	--

20. My mobile phone has Bluetooth support

--	--

21. I used a laptop to participate

--	--

Please write any comments/feedback/thoughts below...

References

1. Terje Øfsdahl and O.K. Mørch-Storstein, *Lecture Games*, in *Department of Computer and Information Science*. 2006, Norwegian University of Science and Technology: Trondheim.
2. Bligh, D.A., *What's the Use of Lectures*. 2000, San Francisco: Jossey-Bass.
3. Natvig, L., S. Line, and A. Djupdal, "Age of Computers:" *An Innovative Combination of History and Computer Game Elements for Teaching Computer Fundamentals*. *Frontiers in Education*, 2004. FIE 2004. 34th Annual, 2004.
4. Basili, V.R. *The Experimental Paradigm in Software Engineering*. in *Experimental Software Engineering Issues: Critical Assessment and Future Directions*. 1993. Dagstuhl Castle, Germany: Springer Verlag.
5. Wang, A.I., *Using a Mobile, Agent-based Environment to Support Cooperative Software Processes*, in *IDI*. 2001, NTNU: Trondheim. p. 407.
6. Koen, B.V. *Toward a Definition of the Engineering Method*. in *ASEE-IEEE Frontiers in Education Conference*. 1984. Philadelphia.
7. Wallace, M.V.Z.D.R., *Experimental Models for Validating Technology*. *IEEE Computer*. **31**(5): p. 23-31.
8. Craig Larman, V.R.B., *Iterative and Incremental development. A brief history*. *Computer*, 2003. **36**(6): p. 47-56.
9. Malone, T.W. *What Makes Things Fun to Learn? Heuristics for designing Instructional Computer Games*. in *The 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*. 1980. Palo Alto, California, United States: ACM Press.
10. Squire, K., *Video Games in Education*. *International Journal of Intelligent Simulations and Gaming*, 2003. **2**(1).
11. Kirriemuir, J., *Video Gaming, Education and Digital Learning Technologies: Relevance and Opportunities*. *D-Lib Magazine*, 2002. **8**(2).
12. Kirriemuir, J. and A. McFarlane, *Litterature Review in Games and Learning*, in *Futurelab Reviews*, Futurelab, Editor. 2004.
13. Ebner, M. and A. Holzinger, *Successful implementation of user-centered game based learning in higher eduvation: An example from civil engineering*. *Computers & education*, 2005.
14. Piontek, M.A.R.C.M.C.M.E., *Development and Evaluation of an Interactive Web based Breast Imaging Game for Medical Students*. *Academic Radiology*, 2002. **9**(10).
15. Elder, C.D., *Problems in the Structure and Use of Educational Simulation*. *Sociology of Education*, 1973. **46**(3): p. 335-354.
16. Manninen, T. and T. Korva. *Designing Puzzle for Collaborative Gaming Experience - CASE: eScape*. in *DiGRA 2005 Conference: Changing Views - Words in play 2005*. Vancouver, Canada: DiGRA.
17. Crawford, C., *The Art of Computer Game Design*. 1982: Osborne/McGraw Hill.

18. Lowe, J.L. and I. Elwood F. Holton, *A Theory of Effective Computer-Based Instruction for Adults*. Human Resource Development Review, 2005. **4**(2): p. 159-188.
19. Privateer, P.M., *Academic Technology and The Future of Higher Education: Strategic Paths Taken and Not Taken*. The Journal of Higher Education, 1999. **70**(1): p. 60-79.
20. Boocock, S.S. and J.S. Coleman, *Games with simulated environments in learning*. Sociology of education, 1966. **39**(3): p. 215-236.
21. McFarlane, A. and J. Kirrimuir. *Use of Computer and Video Games in Classroom*. in *Level Up Digital Games Research Conference*. 2003. Universitet Utrecht, Netherlands.
22. Schick, J.B.M., *The Decision to Use Computer Simulation*. The History Teacher, 1993. **27**(1): p. 27-36.
23. McFarlane, A., *The games people play*, in *eLEARNING AT BRISTOL*. 2005. p. 3.
24. Relentless. *BUZZ!* [website] 2006 [cited 2006 1/12/2006]; Available from: <http://www.buzzthegame.com/>.
25. Blizzard. *World of Warcraft Community Site*. 2006 [cited; Available from: <http://www.worldofwarcraft.com/>.
26. H.Maier, F. and A. Größler, *What are we talking about? - A taxonomy of computer simulations to support learning*. System Dynamics Review, 2000. **16**(2): p. 135-148.
27. Bär, H., E. Tews, and G. Rössling, *Improving Feedback and Classroom Interaction Using Mobile Phones*. 2005.
28. UW Classroom presenter. [Web site] [cited 24.04.2007]; Available from: <http://www.cs.washington.edu/education/dl/presenter/>.
29. Linnell, M., et al., *Supporting Classroom Discussion with Technology: A Case Study in Environmental Science*. 2006.
30. *UCE Servers & Clients, LectureLab: WIL/MA, Wireless Interactive Lectures in Mannheim*. [cited 2007 22/4]; Available from: <http://www.lecturelab.de/>.
31. *ClassInHand Software User Guide*. 2003 [cited 2007 22/4].
32. *ClassInHand*. [Web site] [cited 2007 22/4]; Available from: <http://classinhand.wfu.edu/>.
33. Limited, A.I. *EduClick*. 2004 [cited 2007 22/4]; Available from: <http://www.avrio.co.uk/shop/educlick.htm>.
34. *Aclass Technology*. [cited 2007 1.may]; Available from: <http://www.aclasstechnology.com/>.
35. *Government backs "Buzz! for schools"*. [cited 2007 24.april]; Available from: <http://www.mcvuk.com/news/25249/Government-backs-Sonys-Buzz-for-schools>.
36. Pancake, C.M. and C. Lengauer, *High-performance Java: Introduction*. Communications of the ACM, 2001. **44**(10 (2001)): p. 98-101.
37. *Java Technology*. [Web site] [cited 16.05.2007]; Available from: <http://java.com/en/about/>.
38. *Basics of .net*. [Web site] [cited 16.05.2007]; Available from: <http://www.microsoft.com/net/basics.mspix>.

39. Martin Karlsson, K.E.M., Erik Hagersten, David A. Wood. *Memory system behaviour of Java-based middleware*. in *The ninth International Symposium on High-Performance Computer Architecture*. 2003.
40. Helal, S., *Pervasive Java*, in *IEEE Pervasive Computing Magazine*,. 2002.
41. *MIDP Telephones*. 2007 [cited 2007 16.may]; Available from: http://www.club-java.com/TastePhone/J2ME/MIDP_mobile.jsp.
42. Neable, C., *The .NET Compact Framework*, in *IEEE Pervasive Computing Magazine*. 2002.
43. Jeong, S.H., *QoS support for UDP/TCP based networks*. Computer communications, 2001. **24**(1): p. 64.
44. Azadet, K. *Gigabit Ethernet over unshielded twisted pair cables*. 1999.
45. Haggman, S.N.S. *GPRS performance estimation in GSM circuit switched services and GPRS shared resource system*. in *Wireless Communications and Networking Conference*. 1999. New Orleans, LA, USA.
46. Olofsson, A.F.S.M.F.M.H., *EDGE: enhanced data rates for GSM and TDMA/136 evolution*. Personal Communications, IEEE, 1999. **6**(3): p. 56-66.
47. Samukic, A., *UMTS universal mobile telecommunications system: development of standards for the third generation*. IEEE transactions on vehicular technology, 1998. **47**(4): p. 1099.
48. Crow, B.P., *IEEE 802.11 Wireless Local Area Networks*. IEEE communications magazine, 1997. **35**(9): p. 116.
49. Haartsen, J., *Bluetooth-The universal radio interface for ad hoc, wireless connectivity*. Ericsson review, 1998. **3**(1): p. 110.
50. Jensen, B.E., *Voxelbasert 3d visualisering i OpenGL*, in *Universitetet i Oslo, Matematisk and Natural Sciences, Informatics*. 2003.
51. *MySQL FAQ*. [Web site] [cited 15.05.2007]; Available from: <http://www.mysql.com/products/enterprise/faq.html>.
52. *lampware*. [Web site] 2007 [cited 2007 15.05.2007]; official web site of the LAMP solution stack community. Available from: <http://www.lampware.org>.
53. *Postgre SQL FAQ*. [Web site] 2007 [cited 2007 15.05.2007]; Available from: <http://www.postgresql.org/docs/faqs.FAQ.html>.
54. *Interbase Public License v 1.0*. [Web site] 2007 [cited 2007 15.05.2007]; Available from: <http://www.firebirdsql.org/index.php?op=doc&id=ipl>.
55. Kruchten, P., *The 4+1 View Model of Architecture*. IEEE Software, 1995. **12**(6): p. 42-50.
56. *OpenGL tutorials*. 2007 [cited 2007 1.may]; Available from: <http://nehe.gamedev.net/>.
57. NIST. *Common Industry Specifications for Usability - Requirements*. [Specification] 2007 [cited; Available from: <http://www.nist.gov/iusr>.
58. Broke, J., *SUS - A quick and dirty usability scale*, in *Usability Evaluation in Industry*, P. Jordan, B. Thomas, and B. Weerdmeester, Editors, Taylor and Francis: London. p. 189-194.
59. Fowler, M., *"UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition"*. 2003.
60. Calveras, A., et al. *Optimizing TCP/IP parameters over GPRS and WLAN networks*. 2003.

61. Reynolds, T.T.B., *Post mortem: Big Huge Games' Rise of Nations*. Game Developer.