



World of Wisdom (WoW) 3D

Kim Daniel Engebretsen

Master i datateknikk

Hovedveileder: Alf Inge Wang, IDI

Norges teknisk-naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap



Department of Computer
and Information Science

TDT4501 GAME TECHNOLOGY
MASTER PROJECT

World of Wisdom (WoW) 3D

Author:
Kim Daniel ENGBRETSSEN

Supervisor:
Dr. Alf Inge WANG

May 16, 2012

Abstract

While there is a growing interest in games for learning, and learning in games. The genre is still quite young. And heavy investments is still to come. But learning games have been a topic of research long enough for the time to take learning games to the next level.

This project started out by studying a learning game, World of Wisdom. The game was made in 2D graphics and was on its way of being discarded. 2D MMORPG games have difficulties attracting players, which in this case is only students in higher learning, and when a learning game struggles to attract players, the point of using games as a learning platform is lost. In order to save the World of Wisdom from being just another old unsuccessful project and utilize new technology to win new players the World of Wisdom should be made into a 3D virtual world based game.

This project looks at the available technology, the state of the art serious games and tries to take wisdom from most common mistakes and other projects hard learned lessons in making a successful serious game.

The goal of this project is twofold. One is to create a successful serious game for learning in higher education. The other is to give all successors a solid base to start building the worlds greatest serious game.

Contents

I	Introduction	1
1	Assignment	5
1.1	Motivation	5
1.2	Project Context	6
1.3	Problem Definition	6
1.4	Project Goal	6
1.5	Terminology	7
1.5.1	Gamification	7
1.5.2	Edutainment	7
1.5.3	Serious Games	7
1.5.4	Epic Win	7
1.5.5	Epic Meaning	7
1.5.6	Urgent Optimism	7
1.5.7	Super-Empowered Hopeful Individual	7
1.6	Readers Guide	8
II	Research Approach	9
2	Research Questions	13
2.1	Questions	13
2.2	Project Hypothesis	13
3	Research Method	15
3.1	Introduction	15
3.2	Experiences from previous projects	15
3.3	Research Methods of Software Engineering	16
3.3.1	Choice of Research Method	16
3.4	Models for Validating Technology	16
3.4.1	Choice of Validation Models	17

III	Prestudy	19
4	Previous projects and works	23
4.1	World of Wisdom 2D	23
4.2	Specialization Project - WoW 2D 2008	25
4.3	Master Project - WoW 2D 2009	27
4.4	Specialization Project - WoW 2D 2009	27
4.5	Specialization Project - WoW 2D 2010	28
4.6	Specialization Project - WoW 3D 2011	28
5	MMORPG and Learning	31
5.1	Introduction	31
5.2	Why Games are a Waste of Time	32
5.2.1	Games are Ruining Education	32
5.2.2	Games Encourage Violence	33
5.3	Why Schools are a Waste of Time	34
5.3.1	Games Inspire	34
5.3.2	Games Engage	34
5.3.3	Games Encourage	34
5.3.4	Games Produce Winners	35
5.4	Success and Failure of Learning MMORPGs	35
5.4.1	3D Learning Environment Archetypes	36
5.4.2	The Risk of Making a Successful Learning MMORPG	37
6	State of the art Serious Games	39
6.1	Analog Electronics	43
6.2	The Evoke Game	44
6.3	Rochester Castle MMORPG	46
6.4	AoC	47
6.5	Moonbase Alpha	48
6.6	S.P.A.C.E.	49
6.7	The Requirements Collection and Analysis Game	51
6.8	SimPort	52
6.9	Virtual U	53
6.10	TARGET	54
6.11	AWEDU	56
6.12	Second Life	57
7	Engines and Technologies	59
7.1	Unity 3	60
7.2	Open Wonderland	61
7.3	Delta3D	64
7.4	JMonkey Engine	65
7.5	CryEngine 3	67

7.6	Unreal Engine	68
7.7	DarkBASIC	70
7.8	Second Life	71
7.9	OpenSim	71
7.10	OGRE	73
7.11	Torque 3D	75
8	Summary	77
IV	Own Contribution	79
9	Software Requirements	83
9.1	Overall Description	83
9.1.1	Product Functions	83
9.1.2	Users	84
9.1.3	Constraints, assumptions and dependencies	84
9.2	Functional Requirements	84
9.2.1	Functional Requirements Acquired from World of Wis- dom (WoW) 2D	85
9.2.2	General Functional Requirements	87
9.2.3	Basic Game Play Functionality	87
9.2.4	Extended Functionality	88
9.3	Quality Requirements	89
9.3.1	Modifiability	89
9.3.2	Testability	89
9.3.3	Performance	90
9.3.4	Non-Functional Requirements Acquired from WoW 2D	90
10	Technology Validation	91
10.1	Technology Validation Models	91
10.2	Technology Evaluation	92
10.2.1	Second Life	92
10.2.2	OGRE	92
10.2.3	OpenSim	93
10.2.4	Unity	93
10.2.5	Wonderland	93
10.2.6	JMonkey Engine	94
10.2.7	CryEngine	94
10.2.8	Unreal Engine	95
10.2.9	Torque 3D	95
10.2.10	DarkBASIC	95
10.2.11	Delta3D	96
10.3	Technology Assertion	96

10.4 Choice of Technology	96
11 Implementation	99
11.1 Pre-Implementation	99
11.1.1 COTS setup	100
11.2 Implementation	100
11.2.1 In the Beginning	101
11.2.2 The Frustration	102
11.2.3 The End	104
12 Prototype	107
12.1 Installation	110
13 Tests	113
13.1 Test Scenario Setup	113
13.1.1 Server	113
13.1.2 Client	114
13.2 Prototype Test Scenarios	114
14 Result	119
14.1 Prototype Test Results	119
14.1.1 General Game Play	119
14.1.2 Communication	119
14.1.3 Game Settings	120
14.1.4 Plug-ins	120
14.1.5 Profile Management	120
14.1.6 Quest and Task Management	120
14.2 Implementation Requirements Status	121
V Reflections	125
15 Evaluation	129
15.1 Prestudy Evaluation	129
15.2 Tests Evaluation	129
15.3 Prototype Evaluation	130
15.4 Research Questions Revisited	132
15.4.1 RQ1: How can World of Wisdom be implemented into a 3D virtual world?	132
15.4.2 RQ2: Can a MMORPG be a useful platform for learning? 132	
15.4.3 RQ3: What advantages can WoW get from using a COTS 3D engine?	132
15.5 Project Evaluation	133
16 Conclusion and Future Work	135

List of Figures

4.1	Game play in World of Wisdom	24
4.2	Game play in World of Wisdom 2D. (1) The question, (2) The time left, (3) The options	26
4.3	Content Editor in World of Wisdom 2D	27
4.4	OpenGL Prototype: A virtual world created by a skybox	29
4.5	OpenWonderland Prototype: A meeting with Nicole Yankelovich, the CEO of The Open Wonderland Foundation, inside the virtual world.	30
6.1	A student working on assignment in the game by Olaf Graven	43
6.2	First mission of Evoke.	45
6.3	Rochester Castle MMORPG	46
6.4	Game play in AoC	47
6.5	Game play in Moonbase Alpha (Repairing the Life Support System)	48
6.6	S.P.A.C.E. logo	49
6.7	Map of the space station in the game S.P.A.C.E.	50
6.8	Exterior of the space station in S.P.A.C.E.	50
6.9	Screen during requirements collection in RCAG	51
6.10	SimPort-MV2 game play	52
6.11	Virtual U 2D game play	53
6.12	Virtual U 3D game play	54
6.13	Game play in TARGET	55
6.14	TARGET: Players virtual lounge	55
6.15	AWEDU	56
6.16	Second Life game play from an adult section of the virtual environment.	57
7.1	A 3D scene made in Unity	60
7.2	Implementation of a social gathering with exchanging information on sticky notes in Open Wonderland	62
7.3	Virtual hospital at Birmingham City University in Open Wonderland	63

7.4	Integration of 2D applications in Open Wonderland	63
7.5	Game play in the Delta3D game engine	65
7.6	The terrain editor in JMonkey Engine	66
7.7	The Statoil Troll oil platform imported from CAD and schematics into CryEngine	67
7.8	Comparison of Unreal Engine 1, Unreal Engine 2, and Unreal Engine 3 rendering capabilities, side-by-side, using the Malcolm model from Unreal Tournament, Unreal Tournament 2004, and Unreal Tournament 3.	69
7.9	Games created with DarkBASIC	70
7.10	A screenshot from Second Life's Dark Alley Neko Club. As mentioned the contents of Second Life can be quite graphic. . .	72
7.11	A theatre created in OpenSim	73
7.12	Screenshot from the OGRE Demos pack	75
7.13	A screenshot from DeathBall Desert made with Torque 3D . .	76
12.1	Prototype: The admin interface for the wonderland server. . . .	107
12.2	Prototype: A view over the town inside the virtual world. . . .	108
12.3	Prototype: The managing modules interface, where the wow-module is uploaded to the wonderland server.	109
12.4	Prototype: The player is given a task from a NPC. World of Wisdom HUD in upper right corner.	109

List of Tables

9.1	The general requirements of WoW 3D	87
9.2	The user level basic game play requirements of WoW 3D	88
9.3	The administrator level basic game play requirements of WoW 3D	88
9.4	The extended functionality (optional) requirements of WoW 3D	89
10.1	A technical overview of the engines previously described	97
10.2	An minus means not having and nearly impossible to imple- ment, while x means have, and empty not having.	98
13.1	Prototype Test Scenario T-UG1	115
13.2	Prototype Test Scenario T-UG2	115
13.3	Prototype Test Scenario T-UG3	116
13.4	Prototype Test Scenario T-AG1	116
13.5	Prototype Test Scenario T-AG2	116
13.6	Prototype Test Scenario T-AG3	117
13.7	Prototype Test Scenario T-AG4	117
13.8	Prototype Test Scenario T-AG5	117
14.1	Status of requirements after implementation of the prototype. .	122
14.2	Status of requirements (cont.)	123
14.3	Unsatisfied requirements after implementation of the prototype.	124
15.1	Summary of test results.	130

Acronyms

AoC	Age of Computers
API	application programming interface
AI	artificial intelligence
AWEDU	Active Worlds Educational Universe
BSD	Berkeley Software Distribution
COTS	commercial of the shelf
FPS	first-person shooter
GaLA	Games and Learning Alliance
GPL	GNU General Public Licence
GLSL	OpenGL Shading Language
GQM	goal question metric
GTGE	Golden T Game Engine
GUI	graphical user interface
HUD	heads-up display
IDE	integrated development environment
IM	instant messenger
JDK	Java Development Kit
JRE	Java Runtime Environment
LGPL	GNU Lesser General Public License
LSL	Linden Scripting Language
LWJGL	Lightweight Java Game Library

- MIT** Massachusetts Institute of Technology
- MMO** massive multiplayer online game
- MMOG** massive multiplayer online game
- MMORPG** massive multiplayer online role playing game
- MUD** multiuser dungeons
- NASA** National Aeronautics and Space Administration
- NAROM** Nasjonalt Senter for Romopplæring
- NPC** non-player character
- NTNU** Norwegian University of Science and Technology
- OGRE** Object-Oriented Graphics Rendering Engine
- OS** operating system
- OpenGL** Open Graphics Library
- OpenAL** Open Audio Library
- OpenCL** Open Computer Language
- PDF** portable document format
- RCAG** Requirements Collection and Analysis Game
- RPG** role playing game
- RQ** research question
- SDK** software development kit
- S.P.A.C.E.** Simulation Program for Astronautical Classroom Engineering
- SRS** software requirements specification
- TARGET** Transformative, Adaptive, Responsive and Engaging Environment
- UDK** Unreal Development Kit
- URL** uniform resource locator
- WoW** World of Wisdom
- WYSIWYP** What You See Is What You Play

Part I

Introduction

Introduction

Part 1 presents the Motivation (chapter 1.1) and Context (chapter 1.2) for this project, and the assignment. This part also defines the Project Goal (chapter 1.4) and Problem Definition (chapter 1.3). A short list of important expressions is included in the Terminology (chapter 1.5) section, as well as a short Readers Guide for the remaining parts and chapters of this report.

Chapter 1

Assignment

1.1 Motivation

There are games everywhere in our society. Now a days even the most common situation is displayed in form of a game. Everything from banking and applying for a mortgage (DNB [13]) and engineering environmental solutions (NASA [41]) to the more teaching and learning environment as game-based hand-ins for students in some courses in colleges and universities. The latter seems to be some of the most challenging to get entertaining enough, and still keep the high demands for goal-oriented content and time use in higher learning studies. The MMO genre has been a huge success for entertainment. And studies show that learning through entertainment can prove to be a success as well. Therefor it was natural that learning and MMO should be the next step. This is why World of Wisdom was developed earlier. Todays version of WoW is in low graphics and try to refer to the first generation of MMO games. But since the popularity of 2D MMO games have declined heavily since the second generation of MMO games where released, it is only natural that World of Wisdom gets a facelift as well. This project will look closer into implementing all logic into an existing 3D engine.

My personal motivation for contributing to this particular project is my affection for the demoscene [59]. The demoscene is a computer art subculture that specializes in producing demos, which are non-interactive audio-visual presentations that run in real-time on a computer. The experience gained from creating demos I believe can be used in creating complex graphical computer games. After several years of studying computer science and programming small applications and games, I wanted to try out full scale 3D game development. Although I had little experience in the field I hope to be able to use my knowledge from nearby fields such as the demoscene.

Another personal motivation I bring to this project, is my affection for open source and free software. This implies for this project to create a product that should work on all three large platforms, Linux, Windows and OS X. The

most important being Linux, since I believe no student should be forced to buy or use proprietary platform to use this projects product.

1.2 Project Context

WoW is a lecture game which development process started by students in a course at NTNU in 2008. The prototype has then been improved by later contributors and quality assured by Alf Inge Wang and Bian Wu as the supervising professors. The prototype has been fully developed to a working game and evaluated by students.

A project finished in December 2011 evaluated a line of approaches of converting the 2D version into a 3D version. The two angles, to rewrite the graphical parts, or implement the logic and other back end code into an existing 3D engine, was prototyped and evaluated.

This project will continue where the previous project stopped, which is the implementation phase and fully product testing. But this report should still be independent and will therefore argue some of the choices and progress in short summaries so that it can be read without any prior knowledge of the earlier stages in research and development for World of Wisdom.

For more elaborated insight into the previous works the previous project reports can be found in the bibliography.

1.3 Problem Definition

World of Wisdom is an MMORPG with focus on learning. The main idea is that instead of fighting monsters by casting spells or fighting with swords, the players will fight a knowledge battle. Elements from standard MMOs should be used like different characters, monsters, leveling, etc.

The virtual world should be divided into various kingdoms that reflect various courses. Each kingdom (course) should get its own characters, quests, monsters, etc.

The focus of this project is to implement a 3D version of the game into an existing 3D engine and during which also improve other aspects of the existing system.

1.4 Project Goal

The main goal of this project is to research possible alternatives for creating a 3D version of World of Wisdom using a 3D engine. This project should also produce a prototype of World of Wisdom 3D. This project aims at creating a open source learning game that can aid classes and be an alternative to classroom teaching in higher learning institution.

The final product of this project should be able to be freely distributed and developed as an open source project. This project should be a contribution to the community of learning games and serious games.

1.5 Terminology

Some terms used throughout this report is explained in short here.

1.5.1 Gamification

The use of game design techniques, game thinking and game mechanics to enhance non-game context.[57]

1.5.2 Edutainment

A form of entertainment designed to educate as well as to amuse[56].

1.5.3 Serious Games

A game designed for a primary purpose other than pure entertainment[58].

1.5.4 Epic Win

A victory or achievement so great and with a such positive result that it can both surprise and shock the player. The phrase can also be interpret as an extra hard achievement with positive result beyond imagination (normally in the context of a computer game) [39].

1.5.5 Epic Meaning

An individual experiences epic meaning, when the individuals action clearly shows direct results of having an impact on a larger scale involving many other people, in which the individual is put responsible for the existence of many others, if not all others individuals. In short, epic meaning can be interpret as the feeling of being important on a global scale [39].

1.5.6 Urgent Optimism

The power meet all challenges with optimism and engagement and see possibility instead of constraints where there is only slight hope of success [39].

1.5.7 Super-Empowered Hopeful Individual

Persons motivated by "Epic Meaning", who are used to used to tackle complex and hard challenges in short time, with "Urgent Optimism" [39].

1.6 Readers Guide

Chapter 2 presents the Research Questions of which is to be answered with the retrieved research. This chapter also briefly explains the constraints sat down by previous projects, and thereby also limiting the context of this project by presenting the Project Hypothesis.

Chapter 3, Research Method, describes detailed how the research was done, and what was the techniques for retrieving and processing, and what methods was used to validate the research.

Chapter 4 contains the reports of Previous Projects about the game World of Wisdom.

Chapter 5 consists of published articles, books and doctor and master thesis about games and learning, with special focus on MMORPGs.

Chapter 6 reviews the State of the art Serious Games.

Chapter 7 reviews the possible technologies to use when developing WoW 3D.

Chapter 9 contains the Software Requirements for WoW 3D.

Chapter 10 validates the technologies with the requirements.

Chapter 11 describes the implementation process.

Chapter 12 describes the prototype.

Chapter 13 describes the tests who shall decide if the prototype meets the desired requirements and goals.

Chapter 14 describes the results of the complete process from setting requirements to implementation and testing.

Chapter 15 evaluates the results from chapter 14, all parts of the project and the project it self.

Chapter 16 tries to conclude and give direction for further work in this project and the field of serious games and learning games.

Part II

Research Approach

Research Approach

Every master project needs to do research. While this is an obvious fact, a more detailed plan of the approach to take when retrieving and processing this research is paramount to maximise the quality and thereby reducing the quantity. The goal is to first retrieve as much relevant research as possible and then cut down the amount until what is left is only the research that can contribute to the project.

Part II presents the Research Questions (chapter 2) of which is to be answered with the retrieved research. This part also briefly explains the constraints sat down by previous projects, and thereby also limiting the context of this project by presenting the Project Hypothesis.

An important chapter of part II is the Research Method (chapter 3) which detailed describes how the research was done, and what was the techniques for retrieving and processing, and what methods was used to validate the research.

Chapter 2

Research Questions

2.1 Questions

The research questions that creates the base of this report is:

RQ1 How can World of Wisdom be implemented into a 3D virtual world?

RQ2 Can a MMORPG be a useful platform for learning?

RQ3 What advantages can WoW get from using a COTS 3D engine?

The research questions is derived from the previous version of World of Wisdom and from the goals of this project. RQ1 is important to find the alternative ways of proceeding with choice of technology and techniques to use for creating a prototype. RQ2 is an important subject to be reviewed as to if all elements of massive multiplayer online role playing games (MMORPGs) should be used in combination with learning. Some aspect of RQ2 is also to take a stand regarding use of violent scenes in game play, or encouraging violent behavior in the game. RQ3 is following up RQ1, with specific interest in 3D engines that could suit this project. RQ3 also raises the question whether a 3D engine is the right tool for the job.

2.2 Project Hypothesis

The previous and the current projects is based on these hypothesis. The hypothesis is interpret from studying the older World of Wisdom projects and the project problem definition. They will not be further elaborated. This is mainly because this is outside the scope of this project. This project is supposed to find the best solution to a problem not to investigate why and when this problem has occurred and, what could have been done to avert it. By this I mean that there are possible flaws in the earlier design of the game, as the battles being the best way of getting high learning potential of using

the game environment. Some of these doubts is being elaborated in suitable sections of this report but is only meant as commentaries and not a field of research. This is because the main goal of this project always was and still is to create a 3D game out of the 2D version of the World of Wisdom, while improving other peripheral aspects of the game. So even if I may not agree with all decisions made earlier in this game, I will work from a perspective as if I would.

- The battles of World of Wisdom 2D as questionnaires can be transferred to the 3D version.
- Battles as a questionnaire gives high learning potential.
- Battles is the most entertaining way of presenting the curriculum through a MMORPG.
- Administration of curriculum is difficult when it's not formed as questionnaires in the game.
- There is no other user interface of curriculum administration that is simpler than the current.
- Implementation of other evaluation of learning than questionnaires, will be complex, hard, and heavily time consuming.
- Keeping the battles as is, gives the best trade-off between learning potential and entertainment value.

Chapter 3

Research Method

This chapter presents my research method, where it have been derived from, and how I plan to utilize it to achieve a comprehensive yet solid base for this master project.

3.1 Introduction

This master thesis is mainly a software engineering project. Although the project only involves one person, it is still important to have clear plans of both the outlines of the final product, and the main stages of the process.

”Software development cannot be performed like manufacturing, because it contains human-intensive and creative activities. The goal of research within software engineering is to make tools, methods and models enabling software to be produced more effectively, with better quality, on time, and spending less resources.” [51]

Although this quote may indicate a strong need for research within software engineering, Wang also write that such research is very time-consuming and expensive to perform, and that it is impossible in smaller projects. But the research method is never the less required to examine if new technology, methods, tools, processes and so on really improves goals like the software quality, and less development time.

The research method will mainly be derived from Basili [3], and Zelkowitz and Wallace [64] quoted in Wang’s thesis [51].

3.2 Experiences from previous projects

The specialization project I wrote in the fall of 2011 was the beginning of this research and this project. It is therefore natural to begin with a short summary, and derive the conclusions and experiences from it.

3.3 Research Methods of Software Engineering

Basili claims that there are three main research methods that are commonly used in software engineering projects. The engineering method which emphasize on test-evaluate-improve cycle, and is only restricted by the final product. The empirical method which starts by a hypothesis that is adapted throughout the project, and uses data to falsify or verify its own correctness. And the mathematical method where software development is seen as a mathematical transformation process, and is mostly based on mathematical and formal methods of doing experiments [3].

3.3.1 Choice of Research Method

From the methods above I see that the best choice for this project will be the engineering method. This choice emphasize my own goals of prioritizing a completion of a prototype to be tested in the end of this project, and also gives the research less restrictions in the academic execution.

The Engineering Research Method

The engineering research method is the way of the engineer, with the trial and error approach. This method comprises a hypothesis of which an engineer build and test a system. The method opens up for a more creative and less restrictive engineering process. The only restriction of this method is the resulting product and is hence also a less controlled method. This is important to keep in mind when the work amount is fairly limited. The results of this method is presented in Chapters 11 through 14.

3.4 Models for Validating Technology

Since the research method is chosen in respect to a software engineering project it is only natural to incorporate this in the validation methods as well. Of the existing models there have been a particular study of which models suits such a project. The methods described here is derived from Zelkowitz and Wallace [64] taxonomy for software engineering. They have performed a literature study of which they examined what validation methods was used in software engineering research papers. Of this study they came up with a new list of twelve different models they see as favorable for software engineering projects.

The twelve different models can be grouped into three broad categories: Observational, historical and controlled methods.

In the observational methods category we find Project Monitoring, Case Study, Assertion and Field Study. These methods give respectively: better global control of status in the project, better local control of one element in

the project, better validation if one proposed technology is better than the alternatives, and better validation of multiple simultaneous projects.

In the historical methods category we find Literature Search, Legacy Data, Lessons Learned and Static Analysis. These methods give respectively: a falsified or verified hypothesis, quantitative data used in other projects, a transfer of project experience and knowledge, and quantitative data of completed projects and their products.

In the controlled methods category we find Replicated Experiment, Synthetic Environment Experiment, Dynamic Analysis, and Simulation. These methods give respectively: a review of multiple approaches of solving the same problem, a smaller and more focused task force for specific issues, statistics from after launch of product, and data from use of test subject in end product environment.

3.4.1 Choice of Validation Models

A model for validating technology must be able to examine if new technology, methods, tools and processes improves product goals, such as software quality and low development time.

Zelkowitz and Wallaces taxonomy describes 12 different approaches divided into three groups. I have chosen two methods from different groups that suited this project best. The assertion method and literature search combined will in my opinion give both depth and the necessary width for such examinations, with better validation if one proposed technology is better than the alternatives, and a falsified or verified hypothesis.

The Assertion Method

The observational assertion method is a method of comparing results of objective measurable tests on two or more competing technologies. In this method I will compare the chosen technology with it's real alternatives. This comparison will be in form of a short discussion. The result of this method applied is presented in Chapter 9 with stating the software requirements and Chapter 11, with evaluations regarding choice of a commercial of the shelf (COTS) game engine.

Literature Search Method

The historical literature search method is meant to gather related written material and use experiences from other similar projects, such as the specialization project from 2011, and other written material that can illuminate any new developments and confirm that the hypothesis is up to date with the existing research in the field. The result of this method applied is presented in Part III with emphasis on Chapter 4 and 6.

Part III
Prestudy

Prestudy

Part III is the results of utilizing the Research Methods (chapter 3) of part II. The research is here split in four: The reports of Previous Projects (chapter 4) about the game World of Wisdom, published articles, books and doctor and master thesis about MMORPGs (chapter 5), games and learning, State of the art Serious Games (chapter 6), and review of commonly used Technologies (chapter 7) for creating games in general and with special focus on learning.

Chapter 4

Previous projects and works

This chapter will show the history of World of Wisdom (WoW).

The idea of World of Wisdom was first started by Bian Wu as a Phd Student at NTNU. The concept of World of Wisdom was to create a hand-in system for a course at NTNU, which would be more exiting and social interaction based, than traditional paper excercise hand-ins.

The development of the game started in 2008 as a specialization project of Thor Grunde Krogsæter and Henrik Halvorsen and Esben Andre Føllesdal, three students at NTNU. It was afterwards picked up by several other students in later projects giving WoW a step by step completion while all plans being edited along the way. The goal have been the same through it all, to create a game that emphasize learning and gives both great learning outcomes and entertainment.

4.1 World of Wisdom 2D

The World of Wisdom is an interactive learning game in 2D, mainly made for use in courses at NTNU and other college and universities. The game is an massive multiplayer online game (MMOG) and enables multiple players to interact with each other in a virtual world. The goal of the game is to have students challenge the non-player characters (NPCs) in the game, and win battles by answering right on the questions the NPC poses. When enough right answers has been given the student gets a level up, and will then be able to progress in the game, until the final level where the game ends after winning the last battle.

This game should provide a kind of framework for courses at the university to participate with their own game content. The game is divided into zones which represents towns. Each course has one or more towns they can fill with suitable content to their training or problem solving needs. The game content is mainly focused around quests, minigames, and other tasks. The games have rules that encourage group play and collaboration inside the game world. The



Figure 4.1: Game play in World of Wisdom

game is design in such a matter that previous knowledge and experience in the game will be rewarded [31]. This experience and knowledge should to some degree represent the knowledge and experience of the course curriculum and practical use.

The creation of good exercises and relevant use of knowledge and experience in the game is left to the content creator. This is why this role is to be distributed to teachers in each course. This on the other hand demands an easy to use interface for teacher to interact with the game when creating the content. It is also demands to the variety of content, to prevent a visually boring game environment, that easily loses the players interest.

The World of Wisdom was first implemented as a 2D game, based on several requirements made by a specialization project group in 2008. The game consisted of a database server, a lobby server and a world server, which worked as one server for a client that logged on. The clients graphical user interface (GUI) was in 2D, giving a appearance that reminds of the Zelda game for Super Nintendo. Important parts of the game was also the world editor, map editor and world object editor, which allowed an administrator to add change or delete worlds (courses), terrain and paths, and objects to be placed in the world. The game was also configured through the questions editor, quest editor and an item editor, which managed questions, and control the story the player must follow, and configure the items, like weapon and

armor, the player can utilize.

The goal of the World of Wisdom project 2D was:

- Create a educational 2D MMORPG
- Create playable exercises that can be a substitute for traditional paper exercises in higher education
- Provide toolsets for creating playable content without programming skills
- To be an example of how to create an educational MMORPG

The overall goal is to trigger a better learning situation for students through computer games to diversify the ordinary lecture teaching. This goal and the others above will have an important role in the development process when evolving from 2D to 3D.

4.2 Specialization Project - WoW 2D 2008

The development of WoW started in 2008 as a specialization project of Thor Grunde Krogsæter and Henrik Halvorsen and Esben Andre Føllesdal.

The goal of the first project was to research ways in which elements of MMORPGs could be used in educational games in higher education. The project especially focused on game concepts from the MMORPG genre, and what architecture would be best suited for an educational MMO game.

The project also prototyped a solution for implementing an educational MMORPG. The prototype was based on several servers or server functions that opened up for good scalability in possibility of separating many functions over multiple servers.

The game got a game play with characteristics similar to other MMORPG games, with fighting an enemy with a weapon and progressing in the game by "killing" most enemies and monsters. The enemies and monsters was all NPCs.

The 2008 project creates the base of World of Wisdom when it puts down 40 functional requirements and 5 non-functional requirements. Some of these will be directly transferred to this projects requirements, while others will inspire for new requirements or may be left out. [31]

The ones that was left right away as deprecated or obsolete, were all the functional requirements regarding character development, as these will require a evolving character system which the project of 2008 did not have time to complete, and have a complexity that challenge the time frame of this project. An important choice occurs when reviewing the original requirements of 2008. The 2008 version aims at game play that emphasizes violence and killing as a natural part of the game and paramount to players progression. This was

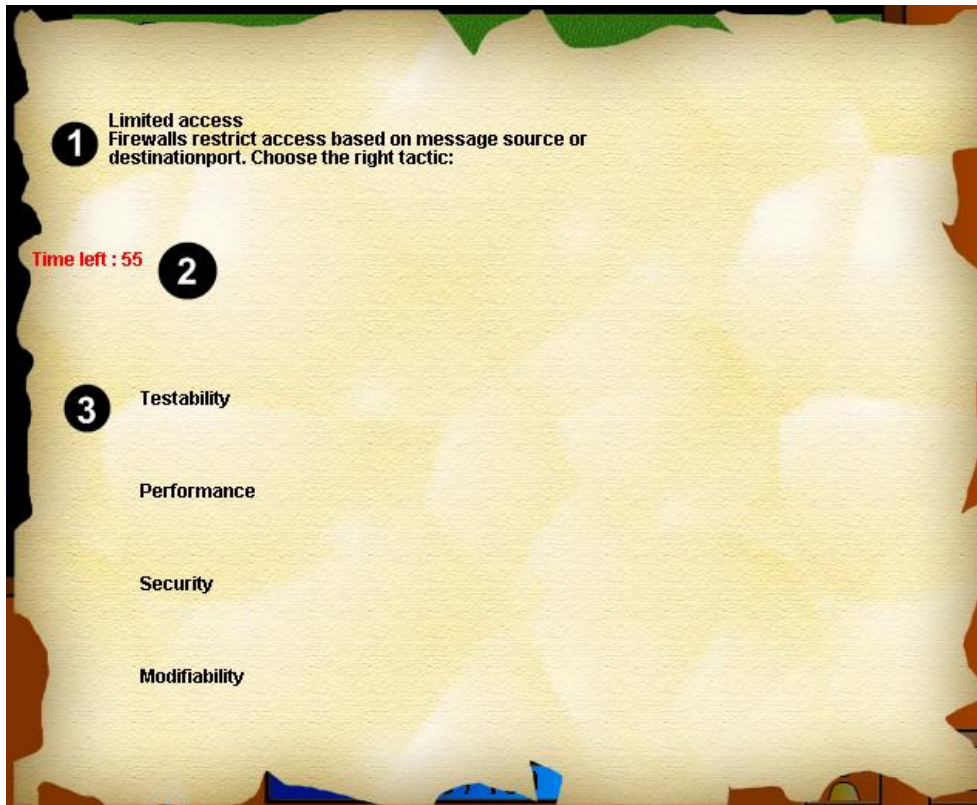


Figure 4.2: Game play in World of Wisdom 2D. (1) The question, (2) The time left, (3) The options

maybe not a topic before 2011 in Norway, but is at best controversial now a days. I see no clear answer to why this has to be such a big part of this game in the 2008 report and this project will therefor try to angle the game play more to an easier accepted concept for leveling and progress in the game that emphasizes social networking events. This means that progress will be measured in other activities than killing other players or general violence.

The requirements that was left out as the progress and leveling changed was FR: 5-11, 15,16, 19, 22, 26, 40. Only one non-functional requirement had to be left out, NFR3 which stated that the game would have 2D graphics, which now should be 3D. [31]

The game from 2008 was based on the Golden T Game Engine. This will no longer be necessary, as it does not support 3D and this project will try to replace it functions with another 3D game engine.

4.3 Master Project - WoW 2D 2009

The 2009 master project was creating a world editor for the World of Wisdom version prototyped in the 2008 project. The master project only involved a prior member of the 2008 project and therefore had a limited scope. The project and the report was written by Thor Grunde Krogsæter. [30]

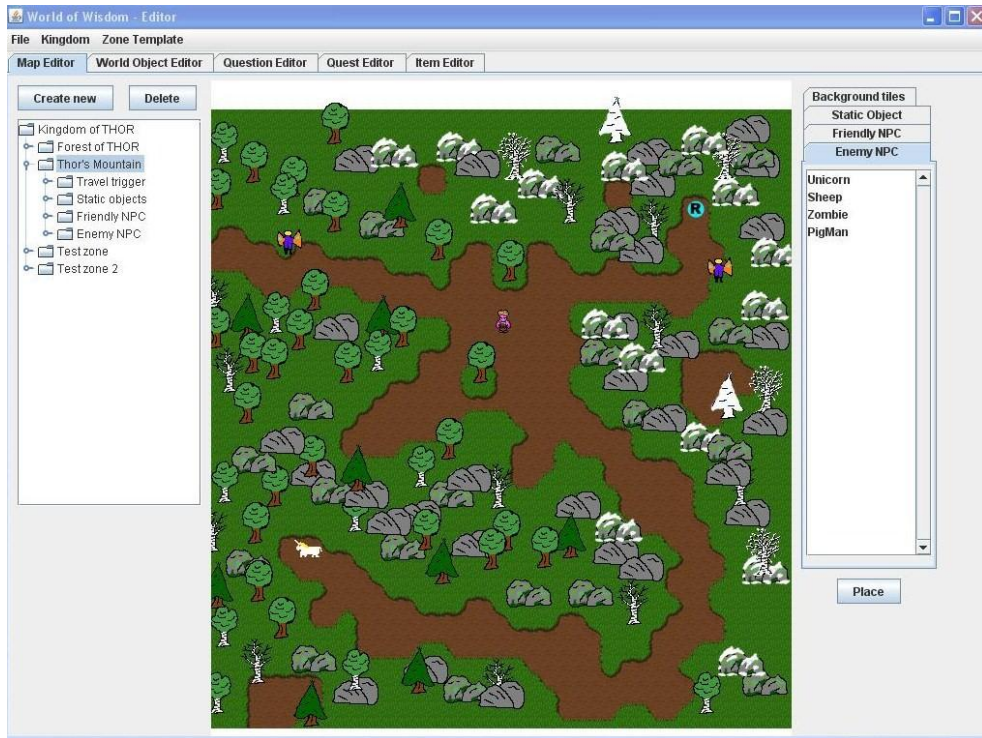


Figure 4.3: Content Editor in World of Wisdom 2D

The master project had many requirements and research on game editors but none of them seems to comply with the extra dimension added in the current project. Although there are bits and pieces that could be used from this project it seems to cost more of locating and integrating these, than trying to find other examples already in 3D, or free prefabricated.

4.4 Specialization Project - WoW 2D 2009

The specialization project in fall 2009, seems to be continuing on along the lines drawn up by the two previous projects. The group of three, Andreas Johnsen, Lawrence Valtola and Sondre W. Bjerkhaug, has continued to implement requirements previously put down, and have had little focus on further evolving the game or otherwise bring new visions or ideas into the game. [28]

It seems like this project has emphasized on describing the 2009 version of WoW, and identifying the next steps in implementation, and implementing them. This projects main contribution is the finalization of the game as a complete prototype.

Some aspects of this project have been difficult to interpret as the final report was incomplete and lacked important parts of information about the project.

4.5 Specialization Project - WoW 2D 2010

The main goal of the 2010 project of World of Wisdom (WoW), done by Dennis Titze, was to implement a evaluation framework for in-game player evaluations.

The project first tested the game prototype and fixed minor bugs. After tests returned successful the implementation of the evaluation framework began. This project listed many questions to use in a evaluation of the game done by a player and could be useful as guidance if this projects needs to evaluate the new version. [47]

Games these days often have better ways of game evaluation just by recording how players play the game, and in that learn from what the majority does right or wrong in the game by viewing statistics and using the recorded data. This technique is called Metric Assisted Game Design [5], and can possibly lower or make the need for an evaluation framework obsolete.

4.6 Specialization Project - WoW 3D 2011

The report of 2011 evaluates and test possible development paths of World of Wisdom 3D. The main goal of this project was to evaluate the technologies that can be used from the point of view of a student with limited 3D programming experience and resources in the context of creating a 3D version of the game World of Wisdom. This specialization project of 2011 was done by me, Kim Daniel Engebretsen.

This project was a preparation phase for this master thesis. The report looked at research and existing games and developed prototypes for proof of concepts and testing of technologies.

The two paths studied in this project was to create a lightweight 3D framework that could replace the 2D game engine that WoW was using, or use a COTS 3D engine or game engine that could replace the Golden T Game Engine.

The report concludes that the best available game engine restricted by economy and time requirements was the Open Wonderland project which is more of a virtual world simulator than a traditional game engine. The reason for this was that Open Wonderland was used as a learning environment in

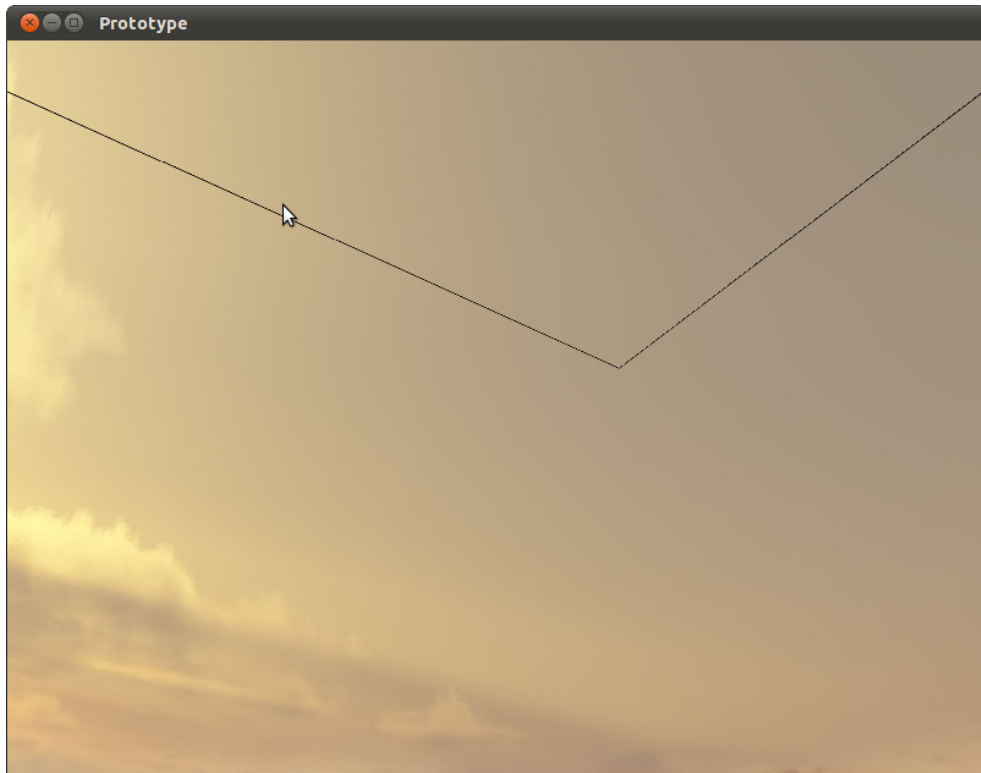


Figure 4.4: OpenGL Prototype: A virtual world created by a skybox

other institutes of higher learning, it already had a large community of participating universities and colleges, and that it was open source and free to use. A prototype with Open Wonderland was tested. The other path with developing a lightweight framework ended in a small code base in C++ utilizing OpenGL for rendering. The evaluation in the end concluded with the lightweight framework not even being near to fulfill the requirements and had to be discarded. The Open Wonderland prototype was not fulfilling all requirements either, but it was still recommended for choice of technology to go with due to many requirements being completed by using plug-ins, and available user guides for implementing the rest of all the mandatory requirements needed to create a prototype ready for user tests. [14]

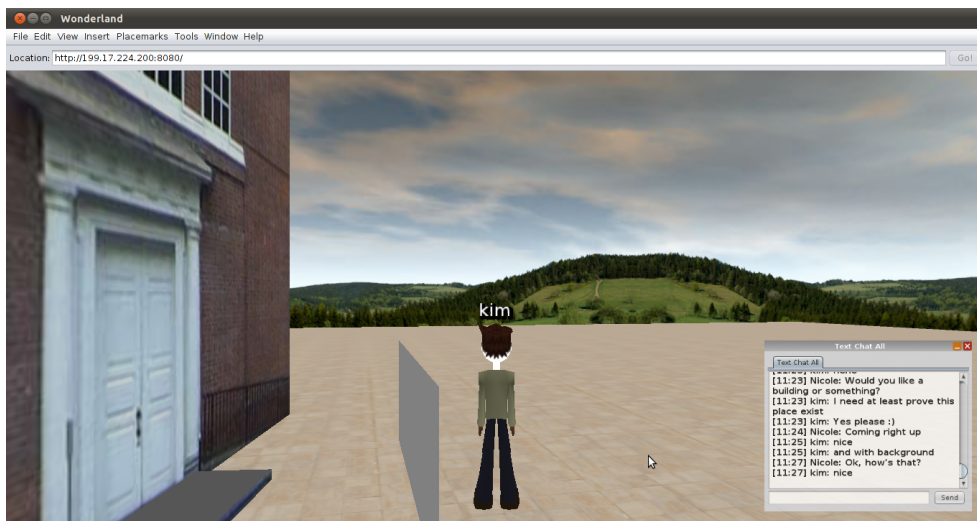


Figure 4.5: OpenWonderland Prototype: A meeting with Nicole Yankelovich, the CEO of The Open Wonderland Foundation, inside the virtual world.

Chapter 5

MMORPG and Learning

This chapter will look at the possible union of MMORPG and human learning. I will present both the arguments research results advocating this union, and the ones against. Since the goal of this master project is to create a game (or at least a simple prototype of a game) that can give both entertainment and an increase in knowledge in a preselected curriculum, it is only natural to investigate the historical contradictions and earlier experiments in this area.

5.1 Introduction

There have been many claims of that playing games is strictly entertainment, and could even have more negative influence than positive for learners. These claims might have been true if the game was seen as a distraction from a traditional classroom learning session. Many trials have been done with trying to integrate computer games into regular curriculum. And of these most of them failed either to entertain, or to give any significant learning advantage. [29][6]

Some other trials have tried to integrate parts of a curriculum into a game, thus keeping as many elements of the game as possible to keep it entertaining [22][44]. What is clear in both of these types of trials is that there is a balance between the learning value and the entertainment value of a game. But newer research have in the last years been exploring more of the essentials behind a successful game [6]. And they have been able to identify some key aspects. They have also seen that playing an advanced game that demands full attention and concentration of a player actually puts a player in a learning situation [44][25]. This learning situation is very different from the typical classroom learning sessions in schools. It is actually more like the learning methods used in many occupations today, since many companies now use simulator-based training. Among these are Statoil, Norwegian Army and St. Olav Hospital. In USA, simulator-based training is required by law for safe operations of nuclear plants. And many other government facilities world wide

is required to train and test their employees regularly to meet international standard and agreements.[4]

While a class in school often uses textbooks and blackboards to present the theory of the subject, a workplace often have more trial and error situations. This is also the way a player in a game experience and learn from the errors and mistakes he makes. The feedback in games is even faster and more reliable than in most situations on a workplace [39]. I will look further into the discussion of how these learning situations differs and why games provides the ultimate environment for human learning. I will first present some of the experts in this field.

One have also seen the increase in the use of the expressions "gamification" and "serious games". These two expressions represents the common game experiences adopted by otherwise non-game situations.

5.2 Why Games are a Waste of Time

Computer games have many critics. Many has tried to draw lines between problems in society and the influence of computer games. Some researcher have tried to find a correlation between these.

5.2.1 Games are Ruining Education

Anyone who sees a hurricane coming should warn others. I see a hurricane coming.

Over the next generation or two, ever larger numbers of people, hundreds of millions, will become immersed in virtual world and online games. While we are playing, things we used to do on the outside, in "reality", won't be happening anymore, or won't be happening in the same way. You can't pull millions of person-hours out of a society without creating an atmospheric-level event.

If it happens in a generation, I think the twenty-first century will see a social cataclysm lager than that caused by cars, radios, and TV, combined.... The exodus of these people from the real world, from our normal daily life, will create a change in social climate that makes global warming look like a tempest in a teacup. [7]

Edward Castronova

As some critics points out. The tremendous popularity of immersive games, can endanger the very backbone of society as we know it. And as McGonigal refers to in her TED Talks speech, "a gamer use up to 20 hours a week playing games" [39]. This means that gamers take on a second half time "job". The great problem with this, is that needs the gamer is used to get covered from the real world society, are now being covered to some extent

in the virtual society. Why this is problematic can be elaborated further into the economic requirements of a consumer society. I will limit my argumentation for this to only point out that all medias and government in Norway agreed that in order to get the lowest possible influence of the financial crisis in Norway in the fall of 2008, the citizens were encouraged to keep spending their money as usual, and keep being consumers.

Another interesting point of McGonigal is, at the age of 21, an average gamer (if we don't count the casual gamers and such), has played over 10 000 hours of games. The number of school hours in the American school system between fifth grade and high school graduation, with perfect attendance, is 10 080. This means that for many youngsters their education is nearly 50% from games.

5.2.2 Games Encourage Violence

Norway experienced 22nd of July it's worst terrorist attack ever since world war two. The investigation and interrogation of the suspect revealed that the suspect had video games and computer games as training and inspiration to his actions. The link between violent games and violence in society has been draw by many over the last decade, but this incident has brought the debate up as one of the most important in Norwegian media these days. Many has tried to condemn violent games from ever reaching the Norwegian market, but the politicians have yet to make changes in the law for such.

Much criticism is based on subjective opinions and have little to show for, but some research has been made. A large thorough research was done in cooperation between universities in Tokyo, Japan, Iowa, USA and New York, USA, on the effects violent video games have on aggression empathy and prosocial behavior in eastern and western countries. The research clearly showed that playing computer games had much larger psychological effect than previous research had come to. They first went through all previous research in the field and found out why other had failed to come to the same conclusion, and found that much of the previous research had flaws like to small bias or sample set and to narrow search in only published material while many smaller studies were left out. This research article therefor included much more of unpublished but yet important previous research. They concluded, with all statistical anomalies and other fail sources like multiple stimuli (from TV and Movies) removed, that there was no doubt, violent video games stimuli a deep psychological violent behavior that can burst into physical violence.[2]

Since the debate and trials of July 22nd is still a very public discussion I will not take a stand or try to elaborate further in this topic here. But it is important to keep in mind that use of violence in games as entertainment could cause unwanted effects and should be avoided in learning games.

5.3 Why Schools are a Waste of Time

5.3.1 Games Inspire

In Jane McGonigals book, *Reality is Broken*[37], she presents a completely different point of view on the discussion of whether games are productive or not. She has researched the effect of computer games when they are made to activate thousands of users into collaborative (unconscious) work. In her trials the gamer becomes a user in the games network of other players, and starts to interact with other gamers to achieve a common game goal. McGonigal has made several such games, and completed many trials to support her research. In the later years she has shifted focus from just the academic to the more idealistic. The latest game of hers, *Evolve*, engaged the gamer into a ten week program of which he/she had to interact with many other players world wide in order to gain points and progress in the game. Each of the tasks or quests the players had to do was a combination between interaction on the computer and interaction in the real world. The real world interaction was typically "help out your local library", "plant a tree", or "pick up garbage from your local park". The computer interaction could be "design a solar-powered boat" or "reduce the energy consumption of a house". Many of the computer tasks and quests also work as an inspiration to try it out in real life.[36]

5.3.2 Games Engage

A study by Lyn Henderson[25] has proven games to give beneficial informal educative experiences. The study also supports the cognitive worth of playing games, and in so contradicting the common view on computer games not being a significant cognitive artefact of youth culture.

Jane McGonigal speaks of gamers, people who enjoy games, in a way that both explains their behavior and their individual systemized capabilities [39], on a TED Talks session. She uses the keywords "Super-Empowered Hopeful Individuals", "Urgent Optimism" and "Epic Meaning" (explained in chapter 1.5). She claims that gamers in reality is an untapped source of "Super-Empowered Hopeful Individuals". These are persons who are used to used to tackle complex and hard challenges in short time, with "Urgent Optimism", which gives them the power to see possibility instead of constraints where there is only slight hope of success. These individuals also gets extra motivated by the "Epic Meaning", which clearly shows direct results of their achievement having an important impact on a larger scale involving many other people.

5.3.3 Games Encourage

The "Urgent Optimism" ability a gamer gains through games is what might come most useful later in life. McGonigal tries to inspire to bring the gaming worlds into our own world in order to take advantage of the human resources.

If a student would have a large and urgent optimism attitude against school work he or she would be better suited for the hardest most challenging topics which demand both focus and deep concentration. But in order to gain from this one would have to put the gamer in the right mode.

5.3.4 Games Produce Winners

In games there have always been the possibilities of multiple winners since everyone is playing in their own environment. But in the MMORPGs the main concept is to share an environment and interact with other player. The main criteria of a massive multiplayer online game (MMO) game is that players share a dynamic game state. The problem with this is that not everyone can be a winner at the same time. This called for the game to have hierarchies and dynamic environment to constantly shuffle advantages around between players and get a more "alive" game play[6]. One of the typical implementation is of a currency which player earn and use to buy in game items. Often the games tries to simulate functions in a real society to trigger the players immersion, and in that the player can learn from this simulation-based training.

One example is micro and macro economy. Constance Steinkuehler writes that the economies of some of the virtual countries that exists in World of Warcraft, Civilization, EverQuest, Syberia and Sims, rivals many of the important real world economies. As an example, Norrath, a virtual country in the MMORPG EverQuest, was in 2001 the 77th largest economy in the real world with a GNP per capita somewhere between Russia and Bulgaria. One platinum piece, the virtual currency of Norrath, was trading on real world trading markets higher than both the Yen and the Lira. [45] Steinkuehler got its numbers from a research done by Edward Castronova.

5.4 Success and Failure of Learning MMORPGs

So why are some games a big success while other cease to exist only months after release? Are there any clear recipe how to make a successful game? And even more important, can these success factors be implemented into a learning game?

A book by two scientists from Bloomsburg University and Duke University, address the challenge and claim to have found 12 basic learning archetypes that can be utilized in a 3D learning environment. Many of these are well known, but they summarize them in a short and informal way.

In my research I have tried to find both success and failure examples to learn from. But the small amounts of published material is not helping too much. When participating in conferences and workshops about games in Norway, I reckon there is much pride involved. Very few game developers wants to share their failures. And even college professors are reluctant to

share their experiences in developing games unless they succeeded. And most research are more concerned with AAA-titles and commercial MMORPGs that either makes a billion dollars or a bankrupt game company. These data will help me some, but still won't touch the subject of learning advantages and values in learning games. I have however found some research that tries to give an answer to the questions I pose.

A few of the games I found is listed in chapter 6, State of the art Serious Games.

5.4.1 3D Learning Environment Archetypes

A research [29] by Karl M. Kapp and Tony O'Driscoll claims that learning situations in virtual worlds can be divided into 12 learning archetypes:

- Avatar persona
- Role play
- Scavenger hunt
- Guided tour
- Data visualization
- Co-Creation
- Small group work
- Group forum
- Social networking
- Operational application
- Conceptual orienteering
- Critical incident

These are archetypes that are important factors of high learning advantages in games. The avatar keeps the game personal and immersive. The role play frees the player from his/her normal environment and gives a player a role instead of just a task. This is using the same principle as giving a person a uniform for a particular job. The scavenger hunt keeps the player engaged while still entertained in the game trying to awake the primal instinct of hunting. The guided tour is giving the player an experience opposed to just reading or viewing photos. The game can much better visualize the data of for example weather than viewing it in statistics and tables. The small group work approach gives the users a tighter more communicative bound than that

of a normal class size of 20-30 students. The group forum gives elaborated discussions and good opportunity to understand and reflect. The social network keeps a player much more immersed and available for the game. The social network can act as a substitute of a real life socialization while playing. Operational application means that the player is given control of a piece of equipment or machinery, and has to operate this in conjunction of other players. Conceptual orienteering is good for trying to teach a concept. Like for example learning how a tsunami works through virtual experience. Critical incident is putting the players in a critical situation like a fire, an accident or earth quake. Depending of the immersion of the player, this can have great learning potential.[29]

5.4.2 The Risk of Making a Successful Learning MMORPG

Developing a successful computer game today is much more demanding than just a few years ago. The explosion in performance of high end systems have given both the programmer and the gamer much more abilities. But this also lays pressure on the producers to deliver a better game than yesterday, which also take advantage of todays, or even in some cases, tomorrows technology.

The majority of game productions never reaches the market. Some game production companies have as many as fifty ideas which all end in different stages of the production line. This means that for each game the need to have income to cover more than fifty failed game productions. One reason for this is that each game has to compete with every other game in the large global market. And then it has to compete with the ones in the other smaller markets which is available for the targeted buyer segment.

The risks of a game never getting to the shelves is high for a game based on the general success factors of a computer game. So when a game production also needs to incorporate the learning factors of a specific subject, theme or curriculum it raises the risk factor dramatically. The focus of such a game can easily be shifted to one of the sides, entertainment or learning, depending on the motivation of the project manager and developers.

Since the risks is higher, so is the cost of production. And if the budget is limited, then it goes without saying that something has to be given less priority. [6]

Chapter 6

State of the art Serious Games

Education Games or Serious Games, are games that has a main focus on education and learning and less on entertaining. Serious games are designed for the purpose of solving a problem. Although serious games can be entertaining, their main purpose is to train, simulate, investigate, or advertise. Sometimes a game will deliberately sacrifice fun and entertainment in order to make a serious point. Whereas video game genres are classified by game play, serious games are not a game genre but a category of games with different purposes. This category includes educational games and advergames, political games, or evangelical games. The category of serious games for training is also known as "game-learning". [60]

Within the serious games genre there are some prototypes, some projects and some fully developed prototypes available. Some of these even use 3D virtual worlds in their games, but as with most games made today, many of these are restricted to a few Microsoft operating systems. Other games have a less graphical oriented perspective on user experience, and went for 2D, in bird-perspective-map or sideways-scrolling-world user graphics experience. A thing to notice is also the line between real world and virtual world, which is normally found in computer games, but almost non existing in some serious games. Many serious games tends to immerse the user into the game by including real life elements. Examples of these are The Evoke Game, which mix both physical real world tasks with virtual world tasks in game play, and Living world in Afghanistan which has more physical real life elements than virtual world elements.

The list of state of the art serious games are long. Just to show you how long, I've listed them all below. The list was populated after email exchanging with the organizations Sintef, GaLA, and SGDA which is organizations known for research and contributions to serious games, and own research. From these games I will pick a few of which have same goals or architecture or have a

resemblance to WoW 3D, and look further into them. These are outlined in bold in the list. The list shows the most notable games within the serious games genre:

Analog electronics

Lecture Quiz

A game similar to Buzz for PlayStation only with questions from curriculum instead. From 2011 of Alf Inge Wang.

The Evoke Game

Rochester Castle MMORPG

AoC

Moonbase Alpha

S.P.A.C.E.

3rd World Farmer

3rd World Farmer is an experiment in the genre of Serious Games, it simulates some of the real-world mechanisms that cause and sustain poverty in 3rd World countries.

Cell craft

Cell craft aims to Explore the cell through gaming.

Darfur is dying

Darfur is Dying is a viral video game for change that provides a window into the experience of the 2.5 million refugees in the Darfur region of Sudan. Players must keep their refugee camp functioning in the face of possible attack by Janjaweed militias. Players can also learn more about the genocide in Darfur that has taken the lives of 400,000 people, and find ways to get involved to help stop this human rights and humanitarian crisis.

Pulse

It is a research project designed to show whether sophisticated medical clinical learning can occur in virtual space powered by cutting-edge video-game technologies.

ABN AMRO Retail game

A short flash oriented serious game which helps employees to train themselves in customer service and the core competence of trustworthiness. Attached link is in Dutch but hopefully the images will give you an idea and perhaps Google Translator can help out on the text.

Vogels (english: birds)

Dutch serious game which is designed to support the rehabilitation of patients with paralysis.

Snow world

A game used to alleviate the pain which burn victims experience during the treatments of their wounds by letting them play a serious game set in an ice world thereby not only distracting the patient but also tricking the body's sensory system and thereby reducing the pain experienced.

Market place

An extensive business simulation, with 3D graphics and with detailed links with the educational path. Used in business schools.

Living world in Afghanistan

First Person Cultural Trainer. Prototype used by US army.

Wildchords

WildChords is an iPad game that makes learning to play the guitar fun, addictive and motivating. Its played with a real guitar. The user hypnotizes animals by playing their favorite chords.

Braintraining (Nintendo)

A game for the Nintendo DS console. Memory training, responsiveness training and other cognitive training programs in a suit.

Logistic koffer

Board game that can be used for explain logistics.

Beer game

For explaining the bull whip effect within the supply chain. This game can be played in an online and a board game version.

Zombie

For teaching Maths specifically division.

The Requirements Collection and Analysis Game

Virtual leader

This game develops leadership skills.

Synergy

A game addressing the Challenge of Managing Change and fostering Collaborative Innovation in Organizations used in many Management

Schools and large organizations (like IKEA, Fiat Group, Alcatel Lucent, Scottish Government) as well as in the public sector (e.g. Scottish Government)

Whataday

An intensive team experience simulating everyday challenges of managers asked to respond continuously to requests for help, to a large inflow of emails and calls, to the resolution of people-related conflicts, to the need of motivating collaborators and peers, to ask good questions, seize opportunities for new projects, and to react to incoming customers requests and expectations.

EIS

A computer-based multimedia management simulation involving the implementation of organisational change. During the simulation, participants can develop and implement change strategies, select among many different tactics to meet their goal and incrementally change the attitude of the managers of large European commercial businesses, influencing their willingness to adopt the proposed innovation.

Climate game

A Dutch game made by Tygron. Simulates climate changes.

SimPort-MV2

Get in line

Turning attending long lines into a fun experience.

Virtual U

Eagle racing

A game that address the challenge of working in diverse and distributed teams. Developed in collaboration with Ferrari, and used by many managers in business schools and organizations worldwide.

Sharkworld

A game about project management. It is used to provide future entrepreneurs with some specific competences on project management.

Re-mission

Remission is one of the most well known and researched serious games. Aiming at promoting adherence to treatment, this game has managed to combine both fun and instructional design together to address issues related to cancer treatment.

Americas Army

A FPS game that advertise the excitement and adventure of being a soldier in the army of USA.

Microsoft Flight Simulator

One of the first simulators on the market, giving an advanced and complex simulator of flying many different types of planes.

Innov8

A game by IBM, to learn the fundamentals of business process management (BPM). Interact with other virtual employees, participating in their daily activities in the fictitious company, After, Inc. In the process, learn about BPM, discovering, collaborating on, and optimizing the company's business processes.

TARGET

AWEDU

Second Life

6.1 Analog Electronics



Figure 6.1: A student working on assignment in the game by Olaf Graven

As the interest in serious games have been a growing in the academic area, many games have, in comparison with WoW, been developed and researched.

This game doesn't even have a title. The game about analog electronics is made by Olaf Graven at Buskerud University College in 2007. The game has a clear educative goal in that it incorporates assembly of and conduction of trials of analog electronics. The game is supposed to simulate real life events like building an differential amplifier and a voltage reference. The challenge with having students doing these trials in real life is connected with high expenses in material and space in electronics labs on campus. The game solves these issues with giving unlimited of the two challenging resources. The use of the game also aids in the approval process of students tasks. This gives the professor more time for helping students in the game.

The game is based on the Torque 3D (7.11) game engine. The development also utilized the Torque MMO Kit, which gives the engine extra MMOG functionality.

Their prototype consists of a virtual world. This world consist of several areas with different tasks and quests. The main story is that the player is stranded on a deserted island. In order to escape the island the player has to build some analog electronics devices, one of them being a bicycle cellphone charger. The students as players should also be able to see and communicate with each other in the game. And collaboration is highly encouraged as this will gain the learning potential in students and help slow students advance faster.

The game as never been publicly released although several articles of the research have been published. Among them "Prototyping Games-Based Environments for learning C++ programming" [22], "Prototyping a Games-Based Environment for Learning" [20], "Learning C++ using an MMORPG with embedded learning content" [21] and "Computer-based Role Playing Game Environment for Analogue Electronics" [23]

6.2 The Evoke Game

The Evoke game engage users for only 10 weeks, and challenge them to do most missions, discussions and research, in order to achieve a collective goal. The users are given an "EVOKE" once a week, from the producers or game starters. This is an message with information around a problematic global topic and a concrete problem of which the player must try to find solutions to. Each player gets the EVOKE at the same time Wednesdays at midnight. When the weeks EVOKE is out, the player which finds the best solution in the shortest time scores most points. If a team of players deliver an answer all player get equal amount of points.

The Evoke Game is a blend of both physical interaction from players and virtual interaction as in normal computer games. The game revolves around missions which is dealt every Wednesday each week of the 10 week long game play period. The main goal is to achieve contact between players and a col-

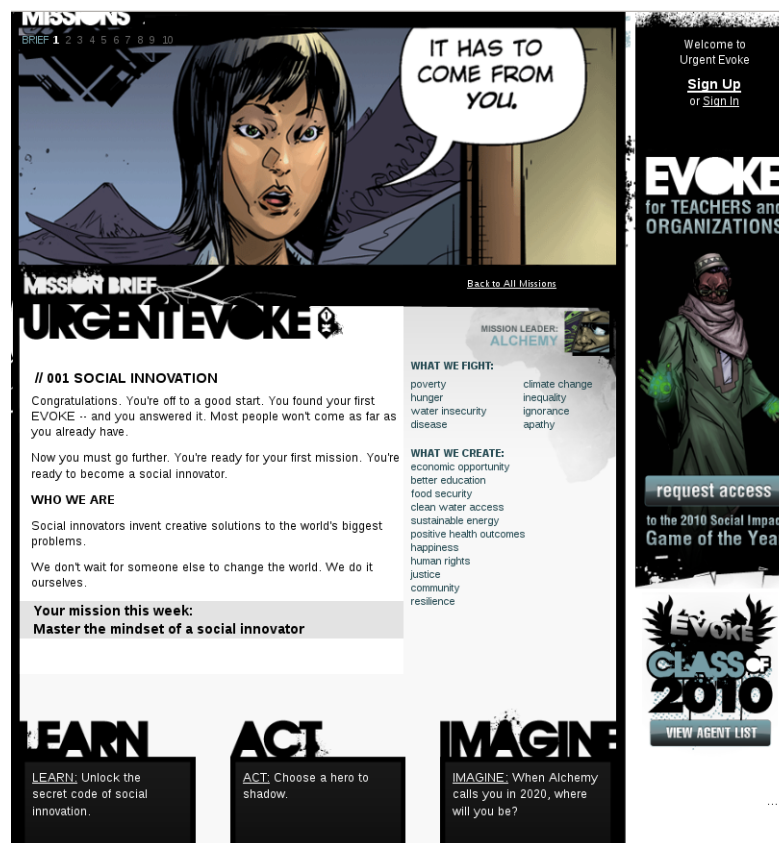


Figure 6.2: First mission of Evoke.

laborative environment which can outlive the ten week period of the game and be transferred out of the game into the real world. Since all players are trained to read about, investigate and choose to take action, in problems no player is capable to solve alone these are conditions create highly valued skills in the global real world problem solving arena.

To site an old saying used in the Evoke game, "If you have a problem and you can't solve it alone, evoke it." [38] The word evoke it self can be translated as "calling up others to invest emotions in the situation" [34]

The Evoke game is an inspiration to all game developers in the innovative way of integrating physical tasks with computer based game successfully. The game ran for many seasons of 10 weeks each, even though the intent was only one run. The goal of reaching both third world countries, Americans and Europeans was achieved. Even though the game has more in common with board games combined with out door activities, the choice of deadlines, blogs, discussions, and videos, sound and pictures clearly benefits from the computer game format. The latter also verifies the placement in the serious games category in my opinion.

This game is developed by Jane McGonigal and The World Bank Institute in 2010.

6.3 Rochester Castle MMORPG

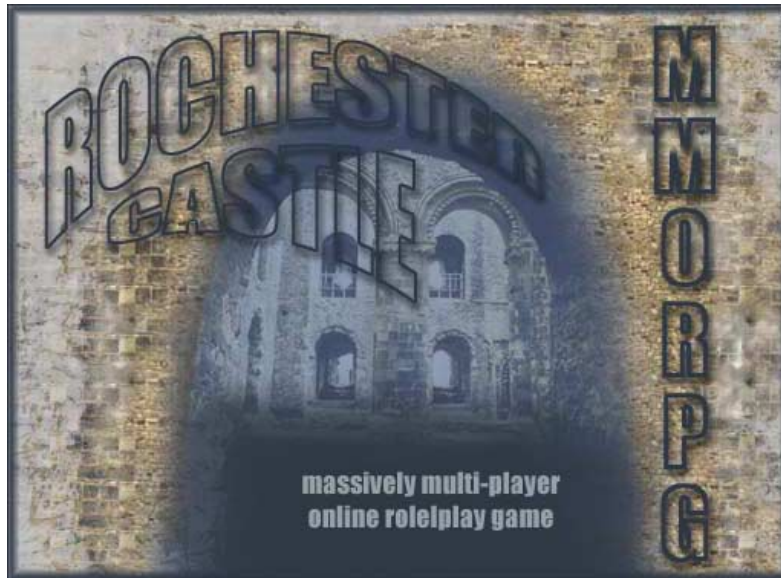


Figure 6.3: Rochester Castle MMORPG

Rochester Castle MMORPG is a MUD-style 2D game from 2005. The game was made during research of instructional gaming and collaborative learning at an Australian school. The intent was to test how a MMORPG could be used in teaching of English and Science at a senior high school. The pilot study involved teachers and students.

The game development had inspiration from the current MMORPGs, and was a game which input educational components or content into a well known game concepts. The social aspect was also an important concept to bring from the MMORPG genre which earlier was less incorporated as a core of the games.

The development phases had focus on being historical correct by using sketches history books and general history knowledge of life in 1215 AD. The teachers of the project therefore had a central part in development. The game it self was based on an enCore Xpress enhanced MMO (MUD object oriented). This is what makes the game more of multiuser dungeons (MUD) based game than the 3D MMORPGs we find today. The MUD games tend to be more simple in design and game play with low graphics and a more complex story or artificial intelligence (AI).

The AI is written in a Prolog like programming language and is what makes most of the games backend. The game also have separate roles for players, one as programmers which student can train their skills by programming bots inside the game.

What is most important of Rochester Castle MMORPG is not the 2D graphics and mostly text based interaction, but the pioneer act of trying to build a game for a educational purpose. This game is of the few I have seen from 2005 and earlier that have such advanced game logic and still incorporates their own story well. [33]

6.4 AoC

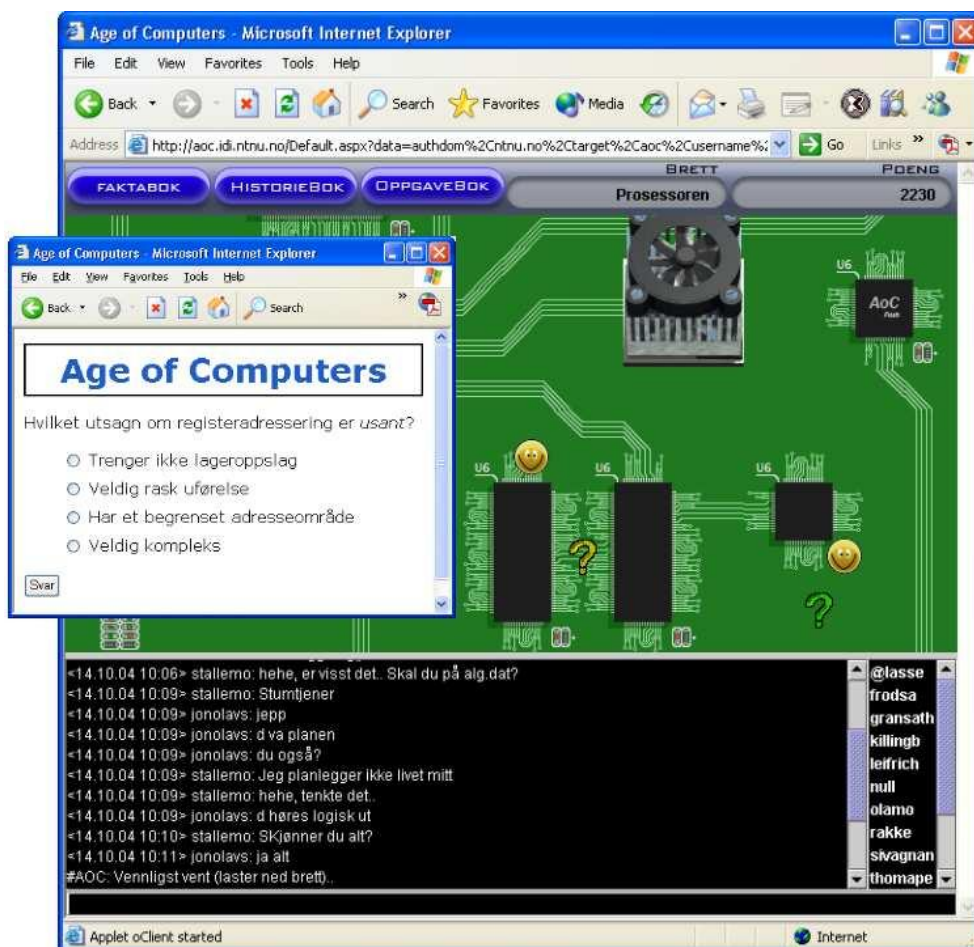


Figure 6.4: Game play in AoC

”Age of Computers (AoC) is an online multiplayer game used in teaching

of computer fundamentals in the M.Sc. study in computer science at NTNU in Trondheim, Norway.” [42]

AoC have many similarities with the current version of WoW 2D. At least if we were not to look ”under the hood”. They both provide the user with a bird-perspective-map where the user moves in a plane, and interacts with entities in the map. The entities differs in AoC with the lack of NPCs and the use of zones, for the user to enter, instead of facing a villain as in WoW 2D. The data models and the client-server communication is very different in the games, which is where the deviation between the underlying architecture is most clear. AoC has still only 2D graphical user experience, and have still to solve the porting from a 2D environment to a 3D environment. But whether or not it needs to be ported to a 3D version is another question I shall leave unanswered.

6.5 Moonbase Alpha



Figure 6.5: Game play in Moonbase Alpha (Repairing the Life Support System)

Moonbase Alpha is a learning game developed by National Aeronautics and Space Administration (NASA). NASA has in the last years started an new educational program called eEducation. This is to contribute to projects with technology and resources to both give students and educators an virtual, immersive educational experience, and an opportunity for NASA to develop a proof of concept for their new content, in example the lunar architecture, in combination with cutting edge game engine.

The Moonbase Alpha is built with Epic Games Unreal Engine 3 as its

back-end. The game statistics, leader boards, browsing and distribution is handled by Valve's Steam network. [41]

6.6 S.P.A.C.E.



Figure 6.6: S.P.A.C.E. logo

During an event held in Oslo in April 2012, JoinGame Workshop XIII, a new Norwegian serious game was presented. The game is a project between Nasjonalt Senter for Romoppl ring (NAROM) and Science [&] Technology AS. While NAROM provides the pedagogical, testing and promoting, S[&]T handles development of the game engine, graphics and is the project leader.

The game, Simulation Program for Astronautical Classroom Engineering (S.P.A.C.E.), is intended for Norwegian students from 13 to 19 years old. The main goal is to advertise the value of knowledge in the major sciences such as physics, math, nature science, biology and so on. The students will be challenged with events for a astronaut on a space station, and by that see the need for all subjects mentioned above in challenges as blowing up an asteroid on collision course, fixing a broken biosphere and taking the right precautions when maintaining a nuclear reactor on the space station.

The game should also inspire students to choose college and universities which emphasize the subjects mentioned. Of course some should also be inspired to study space technology, and astronomy.

As a goal of the project is also research in the use of this type of games as an tool in educating unpopular or difficult topics. The project also tries

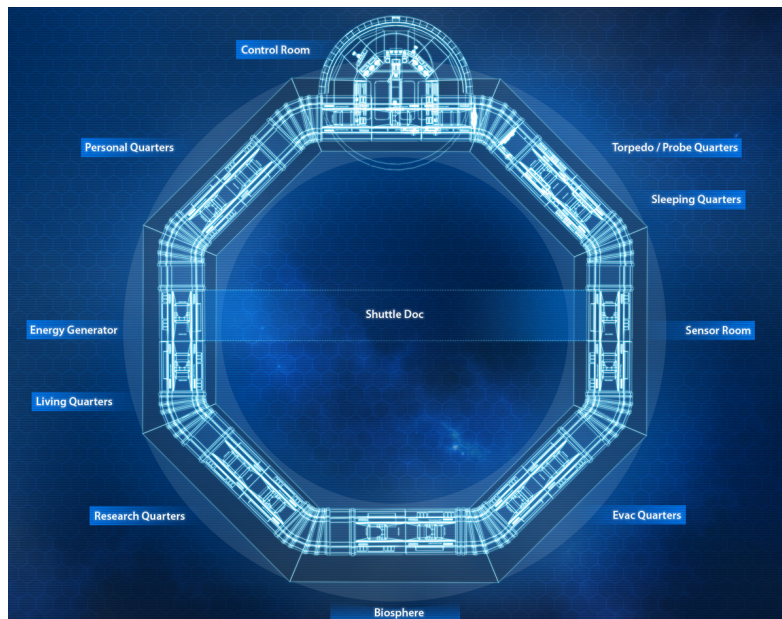


Figure 6.7: Map of the space station in the game S.P.A.C.E.

to connect already established initiatives in the Norwegian space research community, and point out directions for the students.

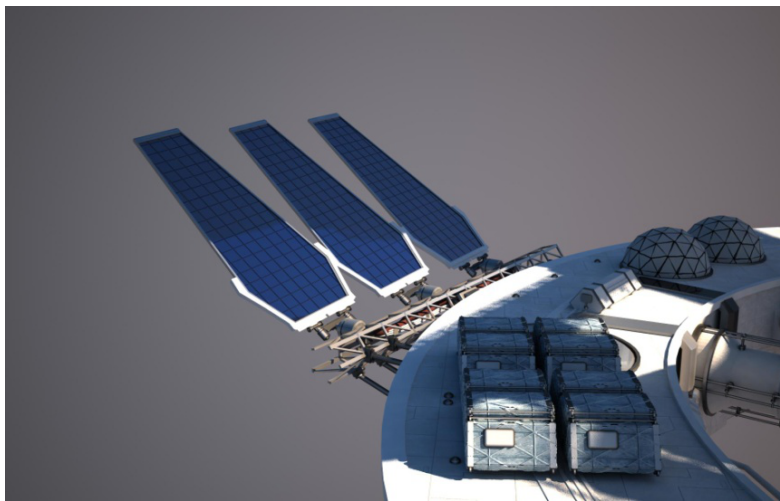


Figure 6.8: Exterior of the space station in S.P.A.C.E.

The game itself is graphical and made with 3D models and compares well to any story-driven games available to the same age group. The game is less massively multiplayer than other games mentioned in this chapter, but they have both logic and graphics better than most. The game is in its start

phase with the first rounds of testing and development, but is expected to be released publicly by the end of 2012. [40]

6.7 The Requirements Collection and Analysis Game

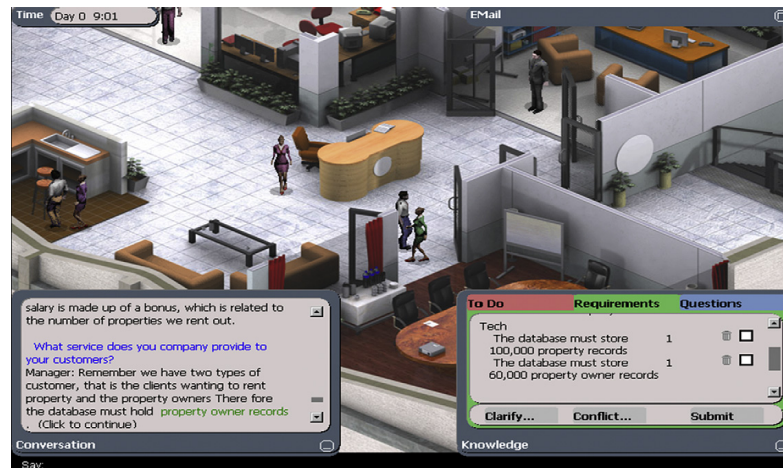


Figure 6.9: Screen during requirements collection in RCAG

The Requirements Collection and Analysis Game (RCAG) is like many other serious games an research project started at an university. The project has seen the disadvantages of traditional approaches in teaching software requirements collection and analysis, and tries to identify and overcome the problems with this approach. The traditional approaches such as lectures, role-play and paper-based case study did not motivate the students to see the potential use when entering employment. The students often end up in or around construction of software and the techniques from requirements collection and analysis could be important if not paramount. The solution is a serious game.

The game is played by moving around in a 3D virtual world and communicating with other players and NPCs. The players all have roles of which some are developer, and project leaders and more. The roles is the same is in any software development project, and the game play look very similar to ordinary work scenarios for a software development company. By this the virtual world looks like an office floor with cubicles and meeting room and social areas.

The game logic is incorporated into the NPCs, which is the customers of the software that is to be developed in the game, then hand out requirements and other relevant and non-relevant information. It's the developers task to gather the right information and process it. The project leader is then sat to evaluate requirements and come up with a timetable and a plan of progress.

If the prioritizing of requirements is wrong the project will fail. If the some of the important requirements is left out the project will fail.

The game looks like the COTS game The Sims, which simulates real life and lets a user be a puppeteer for their avatar. Difference here is the limitation to an office and the goal of creating, managing and delivering a software development project in the end of the game. [24]

6.8 SimPort



Figure 6.10: SimPort-MV2 game play

SimPort-MV2 is situated at the docks. The player starts up as a crane man which mans the docks in a arbitrary city. The player has to have good logistics skills to advance in the game, and money and time is resources which has to be balanced to progress.

The game was originally made for the Port of Rotterdam, where it was supposed to be a training game for personnel. But the game has also become available outside the training program.

”SimPort is a computer-supported simulation game that mimics the real processes involved in planning, equipping and exploiting the Second Maasvlakte (MV2) in the Port of Rotterdam. Briefly, the Second Maasvlakte is a port expansion project to be situated on newly reclaimed land adjacent to the existing port area.” [48]

The game gives a player both insight and training in strategy and logistics involved with managing a port. The players must works in teams of 3-6

persons can coordinate their actions. The long term effects of all choices is also shown to the players during game play. This gives more instant feedback than for example storing a container of nuclear waste in a particular spot in the port for a long period, and then seeing the unfortunate results of the surrounding containers and personnel in the next years.

The game is limited in that it only emphasizes the port, but still utilizes 3D graphics and tries to incorporate a sense of realism which engage the users.

SimPort is developed in Java with JMonkeyEngine 7.4, as it's rendering engine. This is the same engine used by Open Wonderland. All game data is stored in XML and is therefore little user friendly to edit. But the project as plans for making an editor which can easily set up a new port (Since current version only supports Maasvlakte 2), and control economy, customers, buildings, performance factors and other more in detail. [49]

6.9 Virtual U



Figure 6.11: Virtual U 2D game play

Virtual U is virtual universities. The game is all about running an university on daily basis. The game was created by Dr. William F. Massy, who served as CFO and vice-president of business and finance at Stanford University for 14 years and is currently president of the Jackson Hole Higher Education Group.

He had collected a vast amount of raw data about running an university, and he wanted to visualize his data. Instead of just ending up with a

large number of spreadsheets he decided to simulate them in an game engine. Virtual U uses the Enlight Software game engine.



Figure 6.12: Virtual U 3D game play

The game engine used was a product of the company Enlight Softwares Seven Kingdoms game. The company had success with implementing economy engines and political engines in the Seven Kingdom and wanted to create something special with Virtual U.

The game play is simple, but still complex and advanced. "Each Virtual U campus can have up to 15 separate departments, with as many as 500 individually modeled professors teaching up to 10,000 unique students. You don't have control over each student, but you can try to establish programs or create funds that attract minority students and increase the university's diversity. It also is possible to meddle with professors' salaries, research funding, workloads, and other variables to make each department as productive and happy as possible." [19]

The game also have random events as professors winning Nobel Prices and embezzlement scandals, which affects the game play. The game ultimately gives leadership training even though the game was never intended for learning or training. The game was only meant to simulate data for a few persons involved, but as the game became such a success the game has been made public, although not free of charge. [19]

6.10 TARGET

"The TARGET project is a collaborative project partially funded by the European Community under the Seventh Framework Programme (Grant Agreement N 231717)." [1]



Figure 6.13: Game play in TARGET



Figure 6.14: TARGET: Players virtual lounge

Transformative, Adaptive, Responsive and Engaging Environment (TARGET) is a serious game project. Its goal is to learn how to use games to leverage knowledge. The project is sponsored by EU, and has a budget of 9 million Euro. The project management is mainly SINTEF staff. And SINTEF has a large contribution in the project. A total of 16 countries contributes to, and are involved the project.

TARGET is a more of a learning platform, than a stand alone game. It includes a manager software for creating content and story for the serious game and the lounge. The serious game module, the manager module, and the lounge is all available through a TARGET server. The lounge is like a 3D virtual world, shared by all users on the same TARGET server. The serious game is a 2D interactive situation for a user to play in. These games is also in a module on the server.

One of the project developers says in a video on their home page [1], that some of their inspiration comes from the game Civilization. While Civilization is a game for entertainment, TARGET aims at the industrial users, with Nokia and Statoil, as two of the companies that use TARGET for rapid learning for their employees, among the companies currently involved.



Figure 6.15: AWEDU

6.11 AWEDU

Active Worlds Educational Universe (AWEDU) is the subset of Active Worlds, dedicated for exploring educational applications of Active Worlds. AWEDU also provides lower cost of license and new functionality, but the main reasons for AWEDU is for Active Worlds to take share in the research done in such environment, by collaborating with educational institutions.

Active Worlds is a platform for delivering and sharing interactive 3D content over the web. This makes Active World a Second Life clone. Of course not identical but still same concept.

Active Worlds tries to take web browsing to the next level, and reaches out to businesses and their customers, to provide a more immersive browsing experience. Businesses can integrate their products in the virtual world, and customers may enter and view and interact with the products, giving a whole new dimension to online shopping. While setting up a business or own private area in any of the worlds in the Active Worlds universe, being a tourist or customer is free of charge. Although many services requires a registered user, such as instant messenger (IM) protected citizen name and access to expanded avatar gallery.

As Second Life Active Worlds have small mini games for users to play alone or with others. Some of these are free while other may cost a small fee. [26]

The campus of Gløshaugen at NTNU have been created as a world in Active Worlds. This was done in project where students learned about 3D

modelling and 3D virtual worlds in one of the courses at NTNU. This virtual world emphasized the main building at Gløshaugen while others were present but less detailed. [43]

6.12 Second Life



Figure 6.16: Second Life game play from an adult section of the virtual environment.

Second Life is perhaps the most like Open Wonderland which was a huge part of the specialization project about WoW 3D in 2011. The use of a virtual world where users avatars can meet, the use of text-chat and sharing of media is essential for this platform.

Second Life emphasized in it's early beginning the collaborative tools of enterprise users, who due to distance or other constraints needed a common platform for communication and collaboration. The aim of the professional user soon diverted into the more social and less formal users. These users wanted more of a socialization experience with abilities to customize and design both avatars and inventory in the virtual world. The two different types of users have both increased since the start and Second Life has become one of the most used application for social networking on the Internet.

Second Life tries to offer a second life to it's users. A sort of escape from the boundaries of the real life. With this many users tend to act up and live out both fantasies and desires. This huge amount of freedom for all users have actually diverted into virtual crime. User inside Second Life are free to do many actions from the developers. Many of the available actions are maybe a unfortunate combination of harmless actions while others seems intended from the developers. Some actions even allow the user to change their avatar to look either female or a child, and then dress up the avatar as a prostitute,

and then try to sell time with their avatar with other users avatars.

A side from the suspicious, and dark. The Second Life has also implemented campuses inside the virtual world which is open for students and professors to join in and expand their social network, and collaborate using the collaborative tools implemented in Second Life. [32]

Chapter 7

Engines and Technologies

In 2008 the game World of Wisdom (WoW) was made with the Golden T Game Engine. This game engine only supported 2D graphics. The obvious constraints of this game engine consequently forced a search for a better suited technology to replace it.

When the World of Wisdom had to be made into a 3D game, the choice of whether to create a simple lightweight framework for 3D rendering or to use a COTS was reviewed. The amount of work and maintenance of having a large code base was seen as an unavoidable downside to creating the framework, and a 3D engine was chosen instead.

The 3D engine was previously picked among only a couple others and so to clarify the choice, an explanation and a description of the previous nominees is revised here. This selection of 3D engines are based on the availability, and previous experiences of these from people in my communities. There could be others more relevant but since some advance of using this 3D engine should be a large community and a well tested and known engine the point of doing a wider search for others would be pointless.

Modern interactive virtual environments are usually implemented using game engines[15][16][32], which provide the core technology for the creation and control of the virtual world. A game engine is an open, extendible software system on which a computer game or a similar application can be built. It provides the generic infrastructure for game creation, in example input and output, game play control and resource and asset management facilities. The possible components of game engines include, but are not limited to: rendering engine, audio engine, physics engine and animation engine.

The choice of 3D engine is somehow wide interpreted as both 3D engines, and the 3D virtual worlds open for "mods" (short for modifications), which provides a finished virtual world to put new objects and game story in. Some advantages and drawbacks come to each. An technical overview of the engines below is presented in table 10.1.

7.1 Unity 3



Figure 7.1: A 3D scene made in Unity

There are two kinds of engines reviewed in this report. The ones that offers an easy way of programming graphics and setting a scene, but stops just where it's about to get interesting. And the ones that involve multiple editors, rendering tools, huge amounts of pre-fabricated content, and solves everything from networking, audio and controls for you. Unity is of the latter.

Unity, like other heavyweight engines, offers a development platform. It starts you of by letting you create a 3D virtual world which later becomes the stage of your game play. The development environment has a strong focus on testing, and lets the developer rapidly test out the "world" between each changes. This emphasizes a test driven development and rapid prototyping. The different editors easily gives the ability to add filters, effects and scripts self written or pre-made. The only real strong hold back is that Unity never let go of the controls. Many of the pre-fabricated functionality is hidden in the closed source code and thereby nearly impossible to change unless through the user interfaces made available through the bundled integrated development environment (IDE).

Unity has gotten many good reviews in the media, among them the Wall Street Journal, which awarded it with the Technology Innovation Award in 2010. The game company was also names as one of the "Top 5 Game Companies" by Gamasutra in 2009, and a runner-up for best use of graphics on Mac OS X in 2006.[55]

What's different and innovative about Unity is the great support for previously overlooked platforms for games, like OS X on Mac, iOS on iPhone/iPad and Android. Deployment to these mobile platforms off course puts some restrictions on the game. Unity is cross platform. In this Unity support both mobile devices and PC, but unfortunately not Linux.

Some game engines try to be cross platform, but can restrict you to choose only one type of hardware architecture per game. So even if the cross platform support is there, it is up to you to avoid the caveats when choosing functionality features and effects. This is handled well in Unity when it automatically

translate between shader languages to cross compile and deploy. [46]

Unity ships with many batteries included, among them are:

- Triggers, easy creation of "trigger zones" in the game.
- Play Testing, start playing the game to test during development.
- Easy object placement with snap to grid and vectors.
- Prefab objects free and ready to use.
- IDE
- Audio through FMOD.
- TrueType font support.
- Extensive 3D model formats support.
- Particle system.
- Instant previews.

Unity is in its third stable version. It has been used for many commercial games like the MMO game Battlestar Galactica Online, and Tiger Woods PGA TOUR Online. The Unity engine is licensed under a proprietary license.[55]

7.2 Open Wonderland

Open Wonderland started out as the, Sun Microsystems funded, Project Wonderland. In its early days the code base depended on the Java3D scene graph, but as Sun ceased funding in early 2010, Project Wonderland changed into Open Wonderland which now depends on JMonkey engine instead.

Open Wonderland is a pre-fabricated 3D platform. Ready to use for collaboration or social meeting in virtual reality. It offers a wide variety of plug-ins and extensions that can be downloaded through the Wonderland Module Warehouse. It is also possible to program own plug-ins and extensions. The main programming language is Java. The core of Open Wonderland is Open Source and freely available under GNU Lesser General Public License (LGPL). [15]

To quote the Executive Director Nicole Yankelovich of the OpenWonderland Foundation, which is the Company behind Open Wonderland, "Open Wonderland is an open source toolkit for creating 3D virtual worlds. I emphasize toolkit because unlike Second Life or World of Warcraft, Wonderland is not a destination in and of itself. Its a set of tools that people can use to create virtual world destinations. The toolkit is quite rich, so out of the box,



Figure 7.2: Implementation of a social gathering with exchanging information on sticky notes in Open Wonderland

you can build some pretty nice worlds. And with a bit of software development effort, you can create highly customized, special-purpose virtual worlds.

Think in terms of the world wide web. There's no single, giant document collection. On the web, millions of individual organizations own and operate highly specialized web sites both inside and outside their firewalls: travel sites, video sharing sites, endangered turtle tracking sites, charitable giving sites, financial planning sites, and so forth. To create these web sites, developers use tools like CSS, PHP, content management systems, Flash, blogging software, and many others which you can collectively think of as the web toolkit.

The Wonderland toolkit is analogous to the web model. Individual organizations can use the Wonderland tools to create their own specialized virtual worlds. For example, worlds for learning Roman history, worlds for conducting physics experiments, worlds for designing gardens, worlds for simulating molecules, or worlds for remotely controlling laboratory equipment. And these worlds can be linked together, allowing a visitor to the Roman history world to teleport to the garden design world, just like visitors to the travel web site can link to the endangered turtle tracking web site. Like federating web sites, federating virtual worlds enables the concept to scale without requiring any individual world to be extremely large.” [62].

Open Wonderland is actually just the name of the open source code of Wonderland. Wonderland can be used without adjustments, as it is a finished product on its own as a virtual world simulator. Wonderland has an extensive repository of plug-ins, in what is called The Warehouse, which is like the

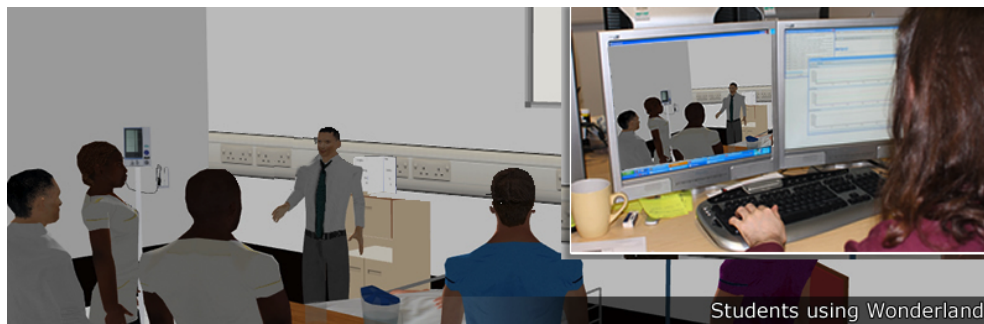


Figure 7.3: Virtual hospital at Birmingham City University in Open Wonderland

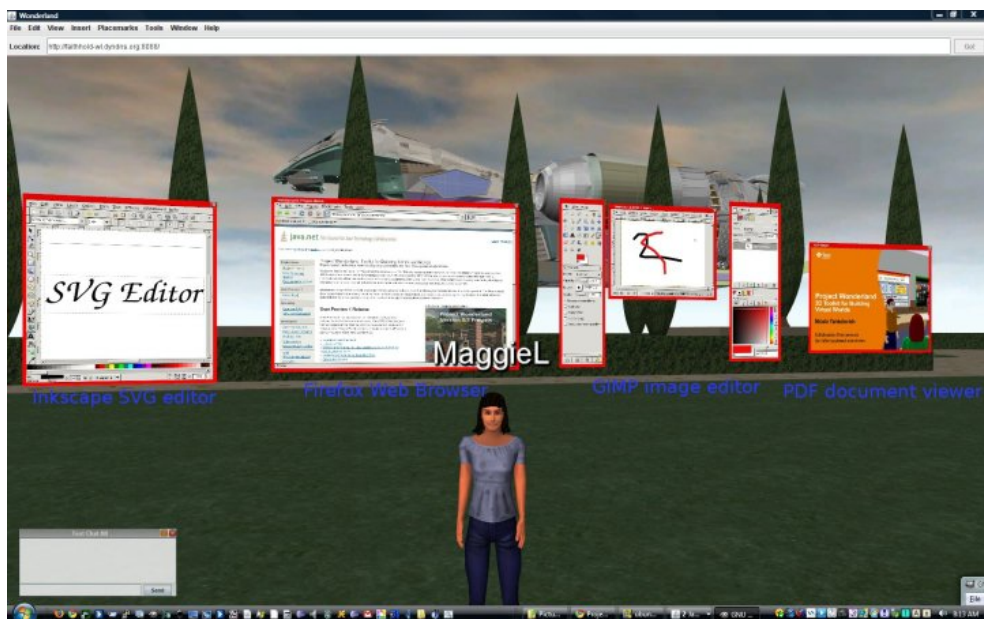


Figure 7.4: Integration of 2D applications in Open Wonderland

AppStore is in the iPhone/iPad world. Wonderland Foundation encourage its users to create plug-ins and upload them to the Warehouse. Some of the plug-ins can therefore be licenced and have a small cost of download, but most are free of charge.

Open Wonderland latest stable release is version 0.5. This means that Open Wonderland is yet to pass the important version 1.0 which in computer science is a users assurance, of quality and a software less prone to error, from the developers. (A version that is not a .0-version should be viewed as test release, Beta or Alpha release, and not as a finished product. This means that huge changes can be introduced to the software.)

The key features of Open Wonderland can be summarized as:

- Written in Java and uses JMonkeyEngine for rendering
- Dependent only on Java core libraries, the rest is bundled
- Good API based on Javadoc
- Runs within JVM and can therefore be ran in most environments (linux, windows, mac osx)
- Large and active community with extensive documentation, guides and tutorials.
- Large Warehouse of Features that can be downloaded.
- Core source code and most of features is GPL or LGPL
- Direct support for 2D X applications (Any program that runs on Linux)
- Uses jVoiceBridge for audio sharing (in-game audio chat and recording)
- Security with support for enterprise solutions and LDAP

7.3 Delta3D

Delta3D is an Open Source simulation/game engine. It is licensed with LGPL. The main objective of Delta3D is to wrap all commonly used tools in a simulation based development environment. In short, an IDE for developing a game. It includes mostly other open source software, but also includes commercial software bridges to such as 3D Studio Max and Maya. Delta3D offers only the tools you need, and therefore leaves the developer to develop from scratch. [10] "Delta3D is a widely used and well-supported open source game and simulation engine. Delta3D is a fully-featured game engine appropriate for a wide variety of uses including training, education, visualization, and entertainment. Delta3D is unique because it offers features specifically suited to the Modeling and Simulation and DoD communities such as High Level Architecture (HLA), After Action Review (AAR), large scale terrain support, and SCORM Learning Management System (LMS) integration." [11]

Delta3D is dependent on networking, audio and graphics through the OpenGL, OpenAL and OpenCL libraries. These are platform independent. There is few or none limitations due to this. The programming language is C++. Delta3D simply wraps a lot of libraries and common operating system calls into a simple API. This makes it versatile, but still a bit confusing, as it gives abilities to do anything, but leaves doubt as to how this is done properly.

Delta3D does not come with a finished and easy solution for quick prototyping and testing, and demands a great effort to start up for a developer



Figure 7.5: Game play in the Delta3D game engine

unfamiliar with the engine. Since stating the requirements and evaluation of these against each engine isn't failsafe, this means that possible unsolvable issues will first be discovered far out in the test period. And thereby posing a greater risk, than smaller and simpler engine or pre-fabricated environment.

Delta3D have had a uncertain future the past year, and have just recently began releasing updates again. The main users of this game engine is American government institutions, such as the military and fire and police departments. Delta3D is not known for any big productions, triple A games, or even indie games (games that targets a small audience, and is a kind of a niche product). This means that existing reviews and evaluations will be hard to come by, and puts a greater press on this reports evaluation.

7.4 JMonkey Engine

JMonkey Engine is a high level game engine written in java, which basically gives access to OpenGL in Java with a extensive documentation, an customized IDE and GLSL (shaders) support. It depends on jME3, JMonkeys code base is completely Java and runs rendering through LWJGL. JMonkey also offers an software development kit (SDK) included in its third version. The software is licensed under the new BSD license. JMonkey depends heavily on shaders and therefore also shader support on the graphics cards, depending on the use of shaders in the final product.

JMonkey also have a great community of users as it is a popular choice

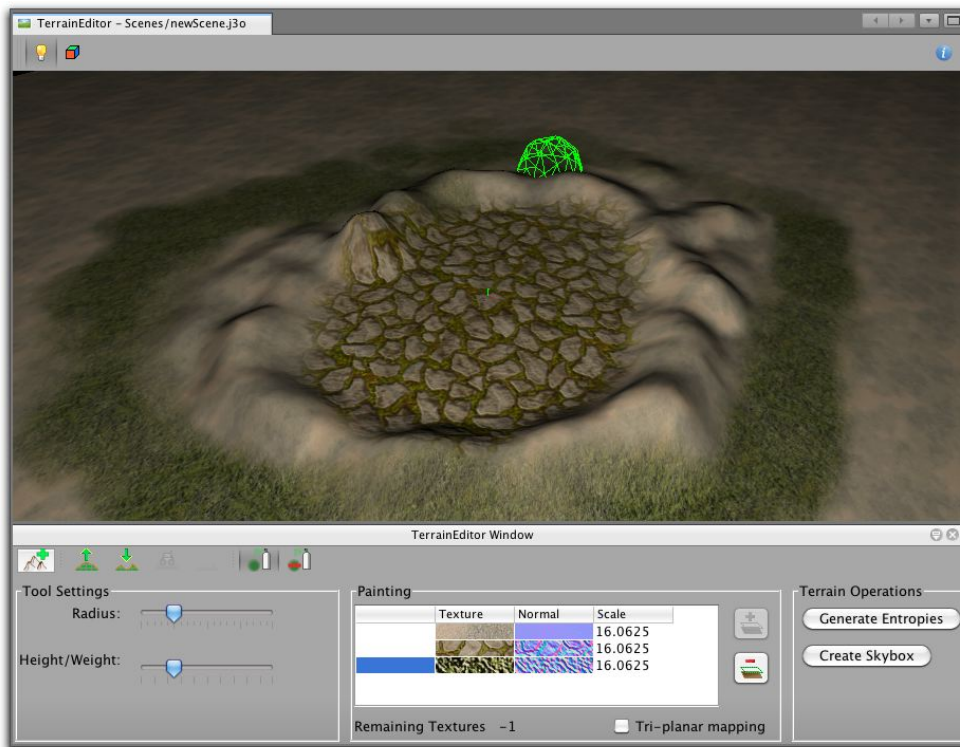


Figure 7.6: The terrain editor in JMonkey Engine

among educational institutions and commercial game studios. JMonkey is also the game engine Open Wonderland is based on. [61]

JMonkey is made upon LWJGL and is a full fledged game engine written in Java, and gives many features for free. One of them being cross-platform, gaming controls, camera features, and more. It has often updates and a respectable community. Since Open Wonderland is based on this game engine, it also gives a good acknowledgement.

The documentation and tutorials available on the projects website is good. The community is large and very helpful to new users. The game engine has been under constant updates since its first release in 2003, and is now in its third stable generation.

As with Delta3D, JMonkey does not come with a finished and easy solution for quick prototyping and testing, and demands a great effort to start up for a developer unfamiliar with the engine. This means that possible unsolvable issues will first be discovered far out in the test period, and thereby posing a bigger risk, than smaller and simpler engine or pre-fabricated environment.

The use of JMonkey Engine will demand a greater self maintained code base, and more self developed content. But it will also give more freedom to decide what to include and what to exclude in the finished game.[50]

7.5 CryEngine 3

CryEngine was initially created by CryTek for a technology demo for Nvidia graphics cards. During development CryTek saw the potential and made it into a game engine and developed *Far Cry*, a first-person shooter (FPS) game for the Windows platform. The game engine was later adopted by other projects by CryTek, such as *Crysis*, and became a product on its own when CryTek licensed the engine to NCSoft and their MMORPG *Aion: Tower of Eternity*. Later the earliest version of CryEngine was sold to Ubisoft.

CryEngine 3 is a game engine developed by Crytek, and is the successor to CryEngine 2 and CryEngine. CryEngine 2 is used in Crytek's game *Crysis*, among others. CryEngine 2 has been licensed to Ringling College of Art and Design for educational purposes. CryEngine 3 is used in tens of games released (or to be released) after 2010. CryEngine 3 as a proprietary engine is the first to support PC (Windows only), Xbox 360, PlayStation 3, and Wii. The game engine is described as the ultimate benchmark test, as it requires heavily processing power from the computer running it. And was often used as a measurement of high-end computer graphic cards. The Norwegian company Statoil has been using CryEngine 3 for some time to visualize their complex CAD models of oil rigs and other offshore installations.



Figure 7.7: The Statoil Troll oil platform imported from CAD and schematics into CryEngine

CryEngine will only work on Windows platforms for PC, and all development and distribution of the game made for this engine is restricted to license.

CryEngine is one of the more extensive engines. It offers both the IDE

and the features and extended functionality one can expect from game engines in this class. But just to list up, these are the attributes that separates the CryTek's CryEngine 3, from the rest:

- Large terrains instead of "fake backdrops" to give illusion of large terrain.
- What You See Is What You Play (WYSIWYP) editor.
- Road and River tools.
- Vehicle creator.
- Visual Budget System, gives the developer to balance memory use in a scene.
- Offline rendering.
- AI editing system.
- Stereoscopic 3D support.
- Deformables and soft body physics.
- Rope physics.

[9]

7.6 Unreal Engine

The Unreal Engine was initially created by Epic Games during development of Unreal, a FPS game for the Windows platform. The game engine was later adopted by other projects by Epic Games, such as Unreal Tournament, and has now become a product on its own. The engine which was primarily developed as a FPS game engine has proven useful in many other genres as MMORPG role playing game (RPG) and stealth. It is shipped today as the Unreal Development Kit (UDK) under a license. And when one talk about the Unreal Engine one often only speaks of it as the kit. The engine is written in C++ which makes it low level programmed. Therefor it has much less bottlenecks in it's graphical pipeline and performs much better than rivals written in other languages such as Java and Python. The engine is cross platform in that it relies on Direct X for Windows platforms, OpenGL for Mac OS X and PlayStation3, and Stage 3D for Flash Players.

The name Unreal Engine is of the name of the first game made with it, Unreal. The game Unreal is again named for it's at the time (1998) "unreal" graphics. The graphics of the game was cutting edge and amazed many in it's ground breaking graphical experience.



Figure 7.8: Comparison of Unreal Engine 1, Unreal Engine 2, and Unreal Engine 3 rendering capabilities, side-by-side, using the Malcolm model from Unreal Tournament, Unreal Tournament 2004, and Unreal Tournament 3.

The Unreal Development Kit (UDK) is the bundled IDE and Unreal Engine. The key features can be summarized as:

- Complete IDE with UnrealEd editor
- Multi-threaded rendering system
- State-of-the-art Animation Studio
- High-level scripts support
- Physics engine
- Advanced Light and shadowing system
- Cinematics support for cut scenes
- Full LAN and direct IP support
- Real-time shaders
- 3D location-based sound

- Unreal Cascade particle system
- AI
- Distributed computing
- Destructible environment
- Blink video codec
- SpeedTree terrain and grass engine
- FaceFX Facial Animation

The Unreal Engine is now in version 3. But version 4 has expected release in May 2012. [17]

7.7 DarkBASIC

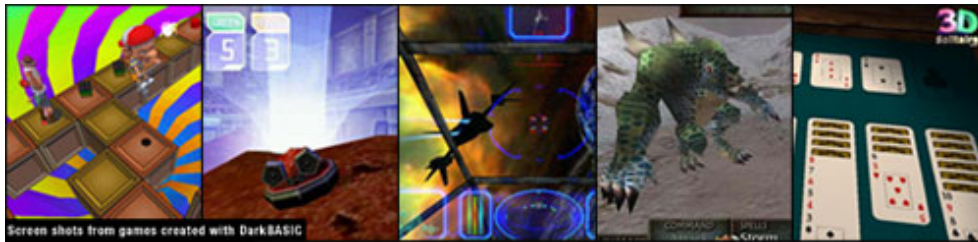


Figure 7.9: Games created with DarkBASIC

DarkBASIC is a programming language for Windows based PC. The language is similar to other languages from Microsoft with Visual BASIC and QBASIC. The syntax and semantics are mostly the same. But DarkBASIC also gives game specific commands to control 3D objects, landscapes, sprites, music, sound and more. DarkBASIC advertises as game engines do, and offers many peripheral products to aid a developer in creating a game. Both 3D modelling, particle studio, vegetation and landscape generation are products one should buy to complete the toolbox.

DarkBASIC claims being the easiest game engine / programming language to use. A lot of finished material ready for use in game is included and many features such as game controls and editors are all bundled. This engine/language have been used to create many games both commercial and hobbyist games. The engine/language have a cost of use in development and is restricted to a proprietary licence.

DarkBASIC is supposed to work as a high level programming interface to help developers avoid programming directly to Direct X which communicates

with most sound and video hardware. DarkBASIC gives mostly the same as OpenGL but only for Windows platforms.

Although the engine seems easy to use and quick in making small games, the engine relies on Direct X and is so forth only for Windows platforms. The Game Company which released DarkBASIC and DarkBASIC professional has created a community with many tutorial and resources for the support of novice users, and uses this as one of the arguments for choosing the DarkBASIC language. [8]

7.8 Second Life

As with Open Wonderland, Second Life might be the answer if one are to go for the fastest solution, and a pre-fabricated virtual world ready to be populated with students classroom and own scripts for gameification of course curriculum. Unfortunately Second Life is a proprietary piece of software, developed and own by Linden Research Inc. It's main attributes is the virtual world, and they do not just hand the source code out so anybody can run the virtual world on other servers than theirs. This means that one would be forced to program scripts that only run on their servers, without the ability to control that server and it's virtual worlds. Other learning games have tried to base their project on Second Life and other such proprietary virtual worlds. But many competitors to Second Life in the past went belly up, and it's no guarantee that Second Life is gonna last forever. There is great risks being dependent on a third party company for a service. And as many of the projects was instantly shut down due to imploding virtual world, this does not advocate the use of Second Life for this project. So without going further into the pros and cons of Second Life I will characterize not being able to run on at own servers as a deal breaker. [63]

Second Life's technology is otherwise used in creation of OpenSim, and gives no clear reason why one should go for a closed source, proprietary product one has to use on others server, instead of one which you can download, run on own server and control, modify and dive into the source code if needed. Second Life does give the advantage of not having to run a server and administer users and give support and such. But within this projects scope, it seems wrong to depend on a "black box" out on the Internet which could cease to exist without any warnings, just because it's easy.

7.9 OpenSim

OpenSimualtor, or OpenSim, was created as a counterweight to Second Life, which as increasingly tighten the grip of its users. The control Linden Research Inc., the company that developed Second Life, has over all data inside Second



Figure 7.10: A screenshot from Second Life's Dark Alley Neko Club. As mentioned the contents of Second Life can be quite graphic.

Life is extensive. And to many colleges who wanted what we want from World of Wisdom, third party involvement is just not suitable.

"OpenSimulator is an open source multi-platform, multi-user 3D application server. It can be used to create a virtual environment (or world) which can be accessed through a variety of clients, on multiple protocols. OpenSimulator allows virtual world developers to customize their worlds using the technologies they feel work best" [16].

OpenSimulator is cross platform. It relies on the .NET framework for Windows platforms, and the Mono framework for Unix-like platforms (Mac / Linux). The code is also open source, so any needed changes to the OpenSimulator can be made and recompiled into source. Although WoW 3D mainly will try to take advantage of the pre-fabricated material in each engine, the ability to make such changes in the future is important.

OpenSimulator is made with regards to the same protocols of Second Life and in that is a out-of-the-box Second Life Viewer. This could easy and transition from Second Life dependent software into this. OpenSimulator aims not to be a clone of Second Life, but rather a "bare bones" version without the problematic capabilities available in Second Life that professors and teachers have been complaining about. One of these is the offensive sexual content made by users inside Second Life, that has created a whole new culture. A culture one would like separated from educational environments. [63]

OpenSimulator is currently in version 0.7.3. This is as mentioned earlier problematic in that software with version number less than 1.0 is to be con-



Figure 7.11: A theatre created in OpenSim

sidered alpha or beta releases. But a stable release is not far, and could come during 2012.[16]

The main features of OpenSimulator is:

- Supports Online multi-user 3D environment.
- Cross platform and multiple clients and protocol support.
- Clients can create 3D content in real time.
- In-game scripting through LSL/OSSL, C# and VB.NET.
- Supports plug-ins for extended functionality

[16]

7.10 OGRE

”Object-Oriented Graphics Rendering Engine (OGRE) is a scene-oriented, flexible 3D rendering engine (as opposed to a game engine) written in C++ designed to make it easier and intuitive for developers to produce applications utilizing hardware-accelerated 3D graphics.” [52]

OGRE is an open source project, started in 1999. The first version, 1.0 "Azathoth", was released in February 2005, and has been actively evolved since then. OGRE have had some of the most active communities in the open source world and was named project of the month in mars 2005 by SourceForge. The founder Steve Streeting, was unhappy with the available 3D engines at the time, and started OGRE to get a more flexible 3D engine, not tied to a single scene type, a single rendering API, one type of game, one content pipeline, etc. The need for a more adaptable 3D engine was created the idea of OGRE. And thus giving the engine the ability to render scenes that were structured differently just by adding a plug-in. [27]

The main idea was to make the engine modular, in that being able, for a given project, to pick the best renderer, the best physics engine, the best sound library, etc., for that project, in contrast with other engines that gave you the whole suit of other functions and became "all or nothing" alternatives. [27]

It's important to emphasize the distinction 3D engine from game engine, since OGRE on purpose has not implemented typical game engine functions as physics engine and audio support. [52]

The softwares intended audience is developers needing a top-class 3D rendering engine for whatever project they have in mind. [27]

OGRE has had slots in Google summer of code in 2006, and have just announced new slots in 2012. [35]

The key features of OGRE is:

- Written in C++.
- Direct3D and OpenGL support.
- Cross platform. Windows, OS X and Linux.
- Integrates with Visual C++ and Code::Blocks IDEs (Windows and Linux).
- Supports Cg, DirectX9 HLSL, and GLSL shaders.
- Supports plug-ins for extended functionality.
- Supports multiple texture formats.
- Sophisticated skeletal animation support
- Particle Systems
- Support for skyboxes, skyplanes and skydomes
- Ribbon trails
- Transparent objects automatically managed

[35]



Figure 7.12: Screenshot from the OGRE Demos pack

7.11 Torque 3D

The Torque Game Engine, or TGE, is a 3D computer game engine originally developed by Dynamix in 2001 for, the FPS game, Tribes 2. The Torque engine is licensed from GarageGames. A company that has new owners but still many former members of the Tribes 2 team at Dynamix. Torque3D are still continuing to be developed and supported.

Torque comes in several variations depending on which kind of license is used. But the most common includes both IDE, rendering engines, physics engines and terrain editors. Most notable may be that Torque does not ship with a particle engine, but this is unlikely to be of any use to this project anyways.

Torque is not open source, but proprietary, and developers needs to buy a license to use it. The licenses do come in a educational type, and a under 250 000\$ income per year type.

Most notable for this engine is the extensive suite with all tools included in the combination with full cross platform support. Linux, Windows and OS X is supported. Unfortunately only for deployment. For development one is still bound to a Windows platform.

Many notable games have been created with Torque, of them even a

learning game made by Olaf Graven in the University of Buskerud, Analog Electronics. Other games are Marble Blast Ultra, Wildlife Tycoon: Venture Africa, Minions of Mirth and the indie game, Space Pirates and Zombies (S.P.A.Z.). [54]



Figure 7.13: A screenshot from DeathBall Desert made with Torque 3D

The key features of Torque is:

- IDE
- Physics engine
- Networking. Server and client for MMO games.
- Cross platform. Linux, OS X and Windows.
- 3D model support (DTS and DIF).
- Terrain engine that automatically chooses best level of detail.
- Starter kit for first-person shooter (FPS) games.
- Starter kit for off-road racing games.
- Geological Information Survey data integration.
- Large number of add-ons available for purchase, like AI, artwork and extra models.

Chapter 8

Summary

From the previous project I see that much work has gone in to the World of Wisdom (WoW) 2D, and that the product is well tested since there is in total five projects before this one. This and the last is the only ones with intend of making a new 3D version.

The MMORPG and learning chapter gave knowledge of the success of using MMORPG as an learning platform on its own, and exiting new ideas of what resource gamers really are, and how they can be utilized. This part also gave insight into the schools versus computer games "fight". It seems like computer games has the upper hand.

The state of the art games is many in number, but few can compare to WoW 3D in type. Some like AWEDU, Analog Electronics, RCAG and Second Life seems to be heading in the same direction. A note of these is that non of them are free software. The source code is not available, and finding research done by independent parties is hard. Some of these games is not available for testing at all or only by buying license.

Many engines for creating 3D worlds and games is available. Most of them charge for use or downloading the source code/library or is just too stripped down to be useful for this game, when implementation time is short. Good candidates are Open Wonderland and OpenSim.

Part IV

Own Contribution

Own Contribution

Part IV is my contributions. Here I present the Software Requirements (chapter 9), an updated description of how the WoW 3D should work, and what this demands from both used technology and resulting developed source code. The requirements is followed up by a Validation (chapter 10) of the Technologies (chapter 7) from part III, and tries to distinguish between compatible or not compatible for use in the Implementation (chapter 11) of the new game version.

This part also presents a summarized log of the Implementation (chapter 11) process, and tries to share the main knowledge and lessons learned in the process. The resulting Prototype (chapter 12) is then presented in a description with short guides of how to use the game and see the implemented parts. The Tests (chapter 13) of which the Prototype (chapter 12) is being tested with is also presented in part IV. The end of part IV is the Results (chapter 14) of the complete process from setting requirements to implementation and testing.

Chapter 9

Software Requirements

Since the resulting product of this project is a software computer game, it is only natural that the software requirements is a part of the prestudy.

This chapter describes the software requirements of World of Wisdom (WoW) 3D.

9.1 Overall Description

Some of the requirements was already set in the specialization project in 2008 (from chapter 4), and will be used again in this project, while others are additive to those of World of Wisdom in earlier projects.

The most important change from the 2D version to the 3D version is the change of concept. The old concept was to acquire weapons and kill NPCs in the game to gain point and new tasks quests and resources as weapons. This concept contradicts the ethical values of which some might link to learning games. And since games used in education can become mandatory, the game concept should not encourage violence and killing. This concept change will make many functional requirements from WoW 2D depreciated, while other must be updated. The requirements acquired from the older version is therefor shown in their updated form in the next section.

9.1.1 Product Functions

The product, World of Wisdom (WoW), is a 3D learning game. It has the versatility to be used in many courses in higher learning as an additive to regular classroom teachings. The game which is situated in an virtual world, give a player the ability to socialize with other students and teachers while solving tasks and puzzles, either alone or by collaborating with other players. The game keeps track of scores and can be evaluated by a teacher to follow progression of the students. The game also gives the players artistic freedom

to shape elements in their virtual world, like their avatar or objects made available by a teacher.

The players all use clients to connect to a server with the state of the virtual world in which players interact. The server is a running process on server computer on a remote location. A player only needs one URL to acquire and run the clients which automatically connects with the server.

9.1.2 Users

There are two sets of users of this product. Students as players, and Teachers as administrators and stagers (a person that stage a situation for others). The students will be enrolled in a higher learning program, in a university or college. And the teacher will be professors or professors course assistants.

9.1.3 Constraints, assumptions and dependencies

The inclusion of a COTS software in the product must not constrain the user, or the administrator, to only one type of operating system, or hardware architecture. It is therefore very important that the game and development platform runs a version of Windows, Linux and Mac OS X. The most important is still Linux since this is the only open source operating system which is free and free of charge. I believe these are important factors for students to accept the game, as well as this is helping a school economically utilize the game in their courses.

Two important aspects of this "running on Linux" requirement is the goal of influencing more than just a few selected students that will play this game in the future, but also to encourage more game developers to view cross platform and especially Linux as an important platform for games. This is extra important when developing a learning game.

The other important aspect is how this report can be a contribution to the community of learning game developers. This report and all source code should be made freely available after this project finishes. This implies that licensing on technology used must be such that further distribution is not hindered or requires extra license or permissions.

9.2 Functional Requirements

The functional requirements is divided into four categories in addition to the ones that is imported from the first SRS of WoW:

- General functionality (GF)
- User (player) level game play functionality (UG)
- Administrator level game play functionality (AG)

- Extended functionality (EF)

The requirements is open for interpretation since it is different options for different game engines, and that there is multiple ways to solve the functional requirements, as long as these does not contradict the quality requirements.

The requirements can be summarized and further deduced from these in prioritized order:

1. Create a new quest
2. Create a new task
3. Complete a task
4. Complete a quest
5. Communicate with another player
6. Communicate with a teacher/student assistant as a student
7. Communicate with a student as a teacher/student assistant
8. Change reward for a specific task
9. Edit a user profile
10. Import a feature from Warehouse

9.2.1 Functional Requirements Acquired from WoW 2D

The requirements numbering does not conflict with the new requirements numbering and is therefor kept for historical reasons, although some have small modifications.

- FR1 The player can give his/her character a name.
- FR2 The visual character can be customized(eg. hair color,gender).
- FR3 The characters progress in the game will be represented by their level.
- FR4 The character stats will improve during the game.(Level up)
- FR12 The character may bind to a respawn point in friendly zones.
- FR17 Each character has a knowledge-book, which will contain all the questions the character have answered. If the player have answered correctly, the correct answer will be stored.

- FR20 When the player clicks on a location, the character avatar moves through a legal path to the location, or teleports when available.
- FR21 The players can chat in several chat channels, team channel for current party, area channel to everyone in the current area, world channel that goes to everyone on the server, and guild channel that goes to everyone in the players guild.
- FR23 When completing certain tasks, the player will be rewarded with a badge. This badge will have no effect on the character, but will be used to encourage the players to complete this tasks.
- FR24 A friendly NPCs will provide information, give quests and/or trade with players.
- FR27 Only the course staff(professor and assistants) may add or remove game specific elements in their world.
- FR28 It should be easy and quick to add new quests and monsters to the world.
- FR29 When adding a monster, the course staff selects a theme topic
- FR30 It should be easy and quick to alter old quests and monsters.
- FR31 There will be an external editor where the course staff may add new questions, monster types and new zones.
- FR32 The professor may appoint a number of assistants which can help the professor maintain the world and add new content.
- FR33 There should be optional templates for worlds, so that the professor only need to add the task content. Which should greatly reduce the time needed to make a world ready for players.
- FR34 Question answering or problem solving is the only way to win a battle.
- FR36 When a character wins a battle, he/she should gain experience.
- FR37 It should be possible to create a party. Each party can maximum contain X players, where one is the party leader. Players can join or leave a party as they like.
- FR38 Graphics should be seperated from the rest of the client code, so that new GUIs can be easily made.

The requirements not listed her did not comply with the new 3D graphics or game concept. Mainly fighting/violence and weaponry. Guild functionality is switched with group functionality.

Table 9.1: The general requirements of WoW 3D

Id	Name	Description
G1	Avatar	The user must be visualized in the virtual game world through an avatar
G2	Sound effects	The game should have implemented basic sounds for fighting, moving and other common actions.
G3	Start game	The player should be able to start the game, and move freely in the virtual world.
G4	In-game chat	The game should support an in-game chat.
G5	NPC-players	The game should have programmable NPC (Non Playable Characters).
G6	Profile management	The game should store a “profile” between sessions which keeps track of both progress and Key-bindings per user.

9.2.2 General Functional Requirements

The requirements listed in table 9.1 is mandatory for the game to be ready for testing. These requirements will make the game slightly usable. The requirements must be seen as a whole for fully test of the game. Although more requirements could be listed they can be deduced from these or are obviously mandatory for other requirements to be met.

9.2.3 Basic Game Play Functionality

User level

The player (aka user) has to be able to play the game as intended for the game to be usable. The users experience is an important aspect of evaluation, as the user will be able to give good insight into usability and performance from a subjective point of view. Had this been a greater project a continuous testing-evaluation-implementation cycle would have been ran until goals of usability and user experiences was satisfied. But because of limited resources and more focus on research and the dividend of learning game as concept.

The players should be able to complete tasks and quests to progress in the game. In order to progress the player should also be able to aid or be aided by other players. This demands a common chat for written communication,

Table 9.2: The user level basic game play requirements of WoW 3D

Id	Name	Description
UG1	General Game Play	Complete a task
UG2	General Game Play	Complete a quest
UG3	Communication	Communicate with another player

Table 9.3: The administrator level basic game play requirements of WoW 3D

Id	Name	Description
AG1	Change settings	Change reward for a specific task
AG2	Plug-in	Import a new plug-in
AG3	Profile Management	Edit a user profile
AG4	Quest Editor	Create a new quest
AG5	Communication	Communicate with a student

visible avatars for visual communication, and voice audio from other players when proximity to them for audible communication. Table 9.2 lists the user level requirements.

Admin level

An administrator is able to change settings while server is up and players logged in. This requirement enables part of the god function that a teacher as a user should have over students as players in the game.

The administrator should be able to install/import plug-ins to achieve extended functionality. This process is likely to happen during a live phase or at least a phase after building a prototype of the game since it is important to test the game concepts and game play for evaluation.

While the game is active and used in a course, some support functions has to be made by an administrator to edit profiles and give support to user. Some of this support should also be available by voice chat with in-game players.

An administrator is responsible for content in the game, and a natural assignment will be to create update and delete tasks and quests. Table 9.3 lists the administrator level requirements.

9.2.4 Extended Functionality

Functionality that would come useful to the player or administrator id listed in table 9.4. These functionalities is not necessary for the game to work, but is seen as a secondary goal in development, or a feature. These requirements can also be seen as optional since they should not steal focus from the game play requirements in implementation phase.

Table 9.4: The extended functionality (optional) requirements of WoW 3D

Id	Name	Description
EF1	Keyboard configuration	It should be possible to define different key-bindings.
EF2	Story mode	The game should have a gameplay that includes a series of event that form a story which the player participates in. This can include events that alter the storyline as well.
EF3	Music	The game should have a background music that the player can turn on and off.
EF4	Support for gamepads	It should be possible to play the game with gamepads

9.3 Quality Requirements

The quality requirements are the non-functional requirements of the World of Wisdom. The quality requirements for the game in prioritized order is:

1. Modifiability
2. Testability
3. Performance

9.3.1 Modifiability

Modifiability encompasses both maintainability and flexibility. The requirement AG3, as well as tasks and story editing (EF2), is just some examples of how the game needs to be versatile and easy to modify. This also includes lowering or raising the scores or difficulties in the game to get the desired result (AG1). The use of open source is desirable due to better control of source code and, rights to publish and distribute. The separation of Engine source code and World of Wisdom game functionality is paramount of good modifiability. The COTS constraint of cross platform can also be seen as a modifiability matter.

9.3.2 Testability

It is important that all game play can be pretested to ensure players do not encounter unstable game states or otherwise experience unintended game play. The game should also be developed in testable phases to improve time in trail and error mode. With only implementing a few functions and features at a time it will reduce complexity of each implementing-testing-evaluation cycle.

9.3.3 Performance

The game must be able to have hundreds of simultaneous users, which include several administrators. The game must also provide large enough virtual space for all players to achieve their in-game-goals (tasks and quests). Since the success of this game heavily depends on a test period with a small numbers of users, both modifiability and testability supersedes the performance requirement.

The most important performance requirement is still that it does not give to high network latency. And in that is able to provide and render content on average every 40ms. This is half of a typical frame rate for LCD computer screens on 60Hz.

9.3.4 Non-Functional Requirements Acquired from WoW 2D

- NFR1 The game should be cross- platform.
- NFR4 The client and server will be developed in Java.

The requirements not listed her did not comply with the new 3D graphics or game concept.

Chapter 10

Technology Validation

In Chapter 7, the most relevant and known 3D engines and 3D virtual worlds for implementing the World of Wisdom was presented. The assertion tests was deduced from the requirements in chapter 9. And these tests shall compare the given game engines in a binary matter, while the material in the rest of Part III Prestudy will be used to discuss and subjectively evaluate each game engine. Some of the assertion tests are interpretations of what must be included in the game to meet other requirements.

The result of the technology assertion tests is listed in the end of this chapter.

10.1 Technology Validation Models

The choice of technology was evaluated in the specialization project in 2011 and was therefor easier to do second time around. A heavyweight framework for a deep and solid evaluation was not used, but a more user-friendly lightweight evaluation should suffice to see which technology proves best. If there is doubt of which the best alternative is another deeper evaluation should be performed.

In order to evaluate the game engines described in chapter 7, the validation models from chapter 3.4.1 will be used.

The Assertion Method will be used to compare the technologies based on easily measurable assertion tests. These will be based on the functional requirements.

The assertion tests:

1. Does the game engine work on a Windows platform, a Mac OS X platform and a Linux platform?
2. Does the engine natively support audio effects and music?
3. Does the game engine have a in-game chat function?

4. Can the engine have player profiles, and can an administrator manage them?
5. Can the game engine have both users and administrator users?
6. Is the fee for setting World of Wisdom in production with this game engine less than 100\$ each year (also including first year)?
7. Does the game engine provide a simple way of creating non-player characters (NPCs)?
8. Is the game engine bundled with user-friendly editors, where user-friendly means intuitive for an average college professor at NTNU?
9. Are there ready to use virtual worlds bundled with the game engine?
10. Does the game engine offer a simple way of using artificial intelligence in the game?

The Literature Search Method will be used to subjectively get a perspective through other users of the same technology, the technology company and other available research. This will be based on the quality requirements and non-functional requirements.

10.2 Technology Evaluation

Each engine is here listed and given a short summarize of biggest pros and cons. A technical overview of each engine is also presented in table 10.1.

10.2.1 Second Life

Second Life does not fit this project due to the restrictive user agreements. The Second Life servers is ran by a third party which takes no responsibility and doesn't guarantee any performance, modifiability or availability of its services. Thereby if the service are to be unstable or down or even cease to exist due to closed project at Linden Lab, the company that developed Second Life, this project can easily suffer the same fate. The risk of being that dependent on another company's good will seems unfortunate. [63] [53]

10.2.2 OGRE

The OGRE project aims at not being a game engine. The founders of the project clearly stated that OGRE was an alternative to other game engine that bundle too much functionality and force the bundled engines, editors, modules and other to be used. OGRE takes care of the graphics while other stuff like 3D models sound and physics is left to technologies of users choice.

This means that OGRE by itself don't leave us with much more than an API to work with. But on the plus side, OGRE is free and runs on all platforms in the specification. [52]

10.2.3 OpenSim

OpenSim tries to distance itself from Second Life, but is clearly a clone or at least very similar in idea. OpenSim is in many aspects the best choice among the engines listed in this report. It does fulfill most of the requirements, but still is a bit short when it comes to prefabricated material.

OpenSim is cross platform as required, open source and free. The Open Simulator Project is a living community with many contributors and have come far in developing against a stable version.

The main differentiation between OpenSim and Second Life is the ability to run own servers. OpenSim was developed with especially learning institutions in mind, and have therefore emphasized both tutorials and simple guides. Another community that also have sprung out from OpenSim offers to run OpenSim on their hardware and let you administer the instance, with or without support agreements. The price is also low, or if ran on own servers completely free.

OpenSim does not offer the same simple setup for games, as many game engines would, but it does give MMO capabilities. [16]

10.2.4 Unity

Unity is one engine that comes bundled with extra tools and features. Unity is a clean game engine and offers many possibilities for creating a game in 3D. The use of Unity would require a computer with Windows for development. The "suit" that comes with Unity prohibits the developer to change default engines and technologies chosen and bundled with Unity. But all in all, Unity is well composed and would probably suit this project if deployment was possible for all three platforms in the requirements specification, and if this project had a much wider and longer development phase.

Unity is used by the Norwegian game company RockPocket Games as primary game engine for all their games [12].

Unity fulfills few of the requirements but is a highly recommended game engine. But unfortunately it is not the right tool for the job this time. [46]

10.2.5 Wonderland

Wonderland is open source and free. The best thing about Wonderland is that it's written in the Java programming language with dependencies on the JMonkey Engine which is also written in Java. This means that everything that runs Java will be able to run World of Wisdom, if we chose Wonderland.

The fact that Wonderland bundles with many samples and tutorials, and even a prefabricated virtual world ready for test or modification is also a very positive property.

Wonderland lacks some of the advantages from game engines that is prepared for games. Some of these like scoreboards, extended player profile and the ability to lead a player through a story, on quests and so on is missing. But since the requirements of game play in World of Wisdom is limited compared to other games, these might be easier to develop or try to integrate from WoW 2D.

Wonderland seems like a step in the same direction as OpenSim and these two might have been even more competitive had it not been for the strictly simulation environment emphasized by OpenSim. Wonderland aims more at being available for collaboration and integration of small shared applications. Wonderland also encourage educational institutions to contribute and use the software in education. [15]

10.2.6 JMonkey Engine

The option of using JMonkey seems only fairly compared with OGRE or Delta3D. These three have all an engine with very few extras included. The extras that do come bundled, like the sound module, is not made difficult to exchange with other and better alternatives.

JMonkey does also come with an IDE that ease the development, but again you have to do the development your self. No drag and drop or prefabricated samples to work on. This might be a good alternative due to the fact that Wonderland is based upon JMonkey, but it cannot compete with other more advanced (and complex) alternatives in this review, although JMonkey Engine is free and open source. [50]

10.2.7 CryEngine

This Engine is one of the most used engines based on the published games created with it. It is like Unity big and comprehensive, but includes everything one should need when creating game. Editors and many other advanced features shall make the user become a great artist, a good developer and musician over night. At least if that were to happen this game engine might have been the thing, but even as sophisticated as it is, the user, the developer has to have the skill set to utilize all of it's functionalities. And in the time restrictions this project has, the learning of such advanced tools would take up too much time.

The fact that CryEngine has been used by companies as Statoil to visualize their oil rigs, does not outweigh the fact that it does not comply with the cross platform requirements, or fit within any small low budget university projects. [9]

10.2.8 Unreal Engine

The Unreal Engine is a well known name in the game development industry. Unfortunately so is the cost of using it. There have been several examples of licenses for universities and other non-profit organizations that were nearly free, but as this game has an requirement of being able to freely distribute, this does not comply. Another negative is as with many other the lack of support for Linux, and therefor failing the cross platform requirement.

The Unreal Engine, or UDK, the whole bundle would have been a good choice since this is one of the more extensive engines, and could cover many aspects of the further development of WoW. Another positive is that several computer labs at NTNU has the UDK installed on their computers, and would be available for other students in the future.

This engine is one of the few to offer support for NPC AI and a ready to use virtual world. So technically this engine could be the fastest way from implementation to prototype, but this still doesn't outweigh the missing requirements. [17]

10.2.9 Torque 3D

Torque was used in many of Olav Gravens educational games. His review of the game engine lead me to believe that this engine would be a strong candidate in these evaluations, but sadly no. The engine lacks many things to compete with the other bigger engines mentioned above. It would be a much better candidate if it were free and open software, but since it's not it will be compared with Unity, CryEngine and Unreal Engine. And all of these have more features and functionalities. But then again, non of these large bundled kits provide for the simple requirements for WoW and therefor loses in these evaluations.

On the positive note Torque 3D is cross platform, and do comply with the requirements in this area. Torque 3D also have AI if you purchase an extension pack. Many of the standard functionalities in the other large engine kits, comes as purchasable extension packs in Torque [18]

10.2.10 DarkBASIC

DarkBASIC is more of a language than a game. That is at least what the company behind it writes. DarkBASIC is supposed to make a game developer out of anyone, and offers tools and very extensive documentation with guides and tutorials to follow. If the product here was to be only a onetime hobby project, DarkBASIC might have been better considered, but to make an cross platform MMORPG learning game, DarkBASIC is unsuited. DarkBASIC fails to fulfill many requirements of WoW and should be discarded as an alternative. [8]

10.2.11 Delta3D

Delta3D offers the bundling of most popular and common open source libraries that together make up the base of a simulation game. This makes Delta3D very modular and possible to only include the needed parts in development. But this engine doesn't give to much prefabricated to the user. And even if this engine is in the same league as JMonkey Engine (which it outranks due to better modularity and performance), it cannot compete with the other engines that provide much extra and therefor speed up development.

If it had not been for the demand of quick development of a prototype, this engine should be closer reviewed as an alternative. [10]

10.3 Technology Assertion

The assertion tests (table 10.2) shows that there is a tie between Wonderland and Second Life, but the OpenSim engine can also be an alternative. While many engines fail to be cross platform in the way this assertion test is done, many of them claim cross platform abilities for supporting 3 or more platforms, example Unity which supports PlayStation, Wii, Xbox360, and Windows. Another issue many engines fail on is low costs to use in production. Since there is no guarantee that this project will be funded, it is paramount that costs is held at a minimum.

10.4 Choice of Technology

Since the choice in the specialization project in 2011 was Wonderland, there had to be a clearly better alternative to it for the choice to be different in this evaluation.

The chosen engine was The Wonderland engine/platform. It has the versatility it needs to adjust and use most of World of Wisdom (2D) previous content. It can also as a test or start phase run the World of Wisdom (2D) inside of it as an X application. The choice of using Open Wonderland was discarded as it currently is no need for changes in Wonderland (making a "mod"). But the availability of Open Wonderland is off course no downside when choosing Wonderland.

Wonderland does not have the best performance or usability, but as a complete evaluation only wonderland outweighs positives by negatives. Although it seems the requirement of "running on Linux" might have had a huge impact on these results, I believe that this concious choice can lead to better life expectancy of World of Wisdom, and have great positive effects now and in the future on game development and learning games.

Table 10.1: A technical overview of the engines previously described

Name	Primary programming language	Cross-platform	Scripts	Notable games	License	Notes
Unity 3	C++	Yes	Mono, Unity-Script, C#, Boo	Battlestar Galactica Online, Tiger Woods PGA Tour Online, WolfQuest	Proprietary	
Open Wonderland	Java	Yes	Java-Script	Wonderland	GPLv2	Open source Supports Linux
Delta 3D	C++	Yes	Python	VESSEL Damage Control Trainer, Northrop Grumman's Virtual Training System (VTS)	acsLGPL	Open source Supports Linux Modular libraries
JMonkey Engine 3	Java	Yes		Monkey Island, Open Wonderland	BSD	
CryEngine 3	C++	Yes	Lua	Far Cry Crysis Nexuiz	Proprietary	
Unreal Engine 3	C++	Yes	Unreal-Script, Unreal Kismet	Unreal Tournament, Mass Effect 3, Epic Citadel	Proprietary	Unreal Engine 4 is expected in May 2012
DarkBASIC	BASIC	No			Proprietary	
Second Life	Unknown	Yes	LSL		Proprietary	
OpenSim	C#	Yes	LSL		BSD	
OGRE	C++	Yes	Lua	Ankh Torchlight Garshasp	MIT	Blender integration Bullet physics Open Source Supports Linux
Torque 3D	C++	Yes	TorqueScript	Minions of Mirth, S.P.A.Z.	Proprietary	

Table 10.2: An minus means not having and nearly impossible to implement, while x means have, and empty not having.

Tech	Delta3D	Unity	Wonderland	JMonkey Engine	CryEngine 3	UDK 3	OGRE	Second Life	Torque 3D	DarkBASIC	OpenSim
X-platform (win/mac/linux)	x	-	x	x	-	-	x	x	x	-	x
Audio music and effects	x	x	x	x	x	x		x	x	x	x
In-game chat			x					x			x
Profiles			x					x	x		x
Users and admin users			x					x			x
Low cost of license	x		x	x	-	-	x	x			x
NPC-support					x	x		x	x		
User-friendly editors		x	x	x	x	x			x	x	
Ready to use virtual worlds			x			x		x			x
Artificial Intelligence					x	x			x		

Chapter 11

Implementation

This chapter will briefly describe choices made during implementation phase, and obstacles and challenges met along the development.

11.1 Pre-Implementation

Before starting implementation. A pretest had to be run to confirm that chosen technology complies with the evaluations of bundled features and standard functionalities. This means testing that Wonderland does in fact do the things it claimed to.

Just like the documentation stated, it was simple and easy to install. It was off course many steps to get it to work but all these were described in details. The errors that did occur were only made due to small misunderstandings when reading the user installation guide. The test was set up, and guest users was invited in to try out the new virtual World of Wisdom, which off course was empty of any art and game logic from the actual game yet. The functions that was expected to work is there and working, except for audio support on Linux platform. This is still to be further tested, but discussions with CEO of Open Wonderland Foundation confirmed that this has been an issue for some period of time now. "Sometimes it works, sometimes it doesn't. It's probably dependent on your audio configuration and whether you are using ALSA or pulseaudio." Nicole Yankelovich.

Otherwise the plug-ins work overall, better than expected. The expected was that many plug-ins would not work due to version differences or bad programming from volunteer developers. But non of the above have been any issue yet.

The server was installed on a private domain and test users were friends that read about the unannounced test on facebook. A total of 17 users tried out the virtual world during a test run of 36 hours. They were given no clues in how to install use or utilize the functionalities in the virtual world. But none of them needed any guidance. Even this was not a scheduled or planned

test, the results was surprisingly positive. The test was initiated by coincident with the facebook announcement. From this test I got confirmed that users can hear and speak to each other (no verified Linux test from users on that), chat, create new objects inside, move freely and explore, share applications like whiteboard, share files like PDFs and pictures and videos.

Short summarized, only audio on Linux systems seems to be a issue at the moment. But this is a known issue by Wonderland developers, and is likely to be solved by the community in the near future.

11.1.1 COTS setup

In order to get the functionality we want from Wonderland, it has to be configured first. This has to be done ,first of all, for security. After starting the server (by running `java -jar Wonderland.jar`):

Step 1: Remove the `security-session-noauth` module and replace it with `security-session-auth` in the admin web interface automatically created by Wonderland. Restart the server.

Step 2: Change default passwords for default users from 'admin' to the same password used in the `wonderland.password` file.

Step 3: The config files must be placed in a chosen location which is referenced in the `my.run.properties` file. This file must also be argument when running server. If the custom bash script is not used.

Step 4: Load the module `BasicSecurity`, to prevent everyone from editing the world. Create groups, and place users who shall have edit access in separate groups and other user in the `unprivileged` or "view-only" groups. Restart server.

Step 5: Ready for implementation! Or if prototype is ready, load the `WoW` module in the admin web interface.

11.2 Implementation

The implementation it self can be split in three stages. The beginning is where all hope is left but no code is out. Inspiration and eager is high and expectations of the prototype is at its highest. The middle is where all frustration breaks loose and the implementation process comes to a stand still and even resets a couple of times. Hope has here reached rock bottom and the possibility of actually failing this master does not seem that far. The end is where things finally comes together and starts both to compile and work. This is where hope has just began growing again and inspiration is picking up pace. Unfortunately time is of the essence and limits the very good productivity of this stage, and the implementation period ends prematurely.

11.2.1 In the Beginning

The Implementation can finally start when the configuration and tweaking of the standard Wonderland v0.5 is done.

First thing to do is fetch the svn-trunk of Open Wonderland. This is necessary in order to develop a module, which of course depends on classes in the wonderland source code. I also had to install Subversion, Ant and NetBeans in order to have tools for this job. Another useful element to fetch before startup is an empty-module from the online wiki of Open Wonderland.

The empty module have some files that needs be updated, mainly configuration files. After this the framework is in place, and Java is the way to go next. An important lesson learned here is that one has to use the GroupLayout class for Java Swing GUIs.

The first choice met was that the game should be implemented as a wonderland module for good modularity and modifiability, instead of a in-game object script which could be just as complex and advance but with fewer possibilities of splitting everything up in packages that can be separately installed and modified. This choice makes it possible to put the finished game in a jar file and import into any running wonderland instance. Making it independent of updates and small changes of Wonderland. It also benefits of not being effected or in hinder of security updates.

The old framework of WoW 2D, was based on running a Postgresql database. Unfortunately this database was made on an old version of Postgresql which have known issues with converting beyond version 8.3. This meant that the first 4 days of which should have gone into developing the wonderland module, was used for troubleshooting the database and cleaning it of errors. Many of the errors discovered was due to the use of non standards like the use of "" and CamelCase relation names instead of making them case insensitive without the quotation marks. The hard coded names also became quite an obstacle to overcome.

During the database trouble shooting, a quick code review also revealed many of the same development choices as in the database. The wide use of the special (Norwegian) characters like æ, ø and å and preference of Windows-1252 over UTF-8, also gave my compiler something to trouble about. This experience alone has at least thought me never to use anything but standard characters and character encoding, ever. And by standard I mean ISO standards, not Windows standards.

The code base was previously put together in the IDE Eclipse, but since there were no Eclipse project file included in the source code or any other material handed over from the previous projects, I had to sort this out on my own. The code base has a web of dependencies which constraints the separate parts as World Server and Database Server to be split into separate projects. Even if the architecture obviously has planned for having a shared library with shared functions and types, many dependencies is hard linked

across different separate elements of this client-server-architecture game. The source code was also without the necessary dependency libraries as the Golden T Game Engine and JDBC Driver. Luckily the game utilize .ini-files to parse configuration, but again the location of these files are hard coded through out the source code.

Perhaps the most noteworthy here is that the code base was chosen in such language and with technology dependency that it should be platform independent. But as seen in the many days of troubleshooting, this does not work if one do not strictly follow standards. And that Windows often neglects standard in favor of a proprietary solution, tends to impede this. (The last developments in the project was done in editors in Windows.)

After finally getting all server modules to run as they are supposed to, I notice that the client module won't take input from keyboard, only mouse. But the good news is that the editor module works as supposed to.

I will therefor leave the client module for now, since it's not going to be used anyway.

The goal now is to make an object to place in Wonderland that can utilize the database or server protocols of WoW 2D.

11.2.2 The Frustration

I would lie if I told you that, implementation was straight forward or easy. But the previous research of the old reports had me believe that I knew how and why WoW 2D was as it was. In the past when I was researching this game in the end of 2011 I did not try to decouple the framework from the Golden T Engine, and I did not try to decouple the dependencies in the source code so that the individual parts could both be compiled and exist without each other. After seeing how the client-server architecture has been implemented in Open Wonderland I see how wrong things have gotten in WoW 2D.

It may seem like a cheap shot criticizing what other people have done in this project before me, but an honest opinion is still unavoidable. I see how the game was never meant to be anything else than it already is. An educational massive multiplayer online role playing game (MMORPG) which in fact, do implement all the common factors of most MMORPGs. But with the limitation of only comparing to other such games from before 2008 and in 2D. So, if the game has achieved it's goal in previous projects why do I criticize? I think the one to be criticized is me. I made the decision in 2011 to go forward with this project based on the knowledge I had then. Had I tried to tear apart the source code and force implementation of new functionality or at least implemented tests (which the first project has completely missed), I think the problems that now occurs would not have been overlooked.

The problematic is that:

1. I though that each part of the source code could be separated and only

depend on it self, or the classes in the "shared" directory.

2. There exists no tests for WoW 2D, which implies that each change in source comes with great risk.
3. The client-server architecture speaks neither in a standard "language" over standard protocols or speaks with Strings that can be parsed and translated.
4. Database management is done with an extra standalone module which provides no extended features do defend its demand of a standalone process in the system at runtime.
5. Each module is made as a standalone application with a GUI, which complicates running in a non graphical environment (as most Linux servers).
6. The game is cross platform and runs on Linux but will not compile correctly on Linux, hence development of the game depends on Windows.
7. The relations (tables and sequences) in the database consists of many small tables only used by one source code function. The complexity is wast even for a modern MMORPG. Many relations is made to fulfill a Java function, giving it many on-to-one relations between code and database tables.

After finally coming to the conclusion that the old code from WoW 2D was practically useless, I discarded the code and began building the game from scratch. The problem now was that this project had not put down a time frame for a such major development of the World of Wisdom (WoW) 3D. The plan had all along the way been to improve of the code in WoW 2D and just rewrite the code that controlled graphics. Actually the code to do graphics was supposed to be switched with the code from Open Wonderland, and therefor only to copy the game logic from the old code. So with all this debugging and troubleshooting a critical situation emerged when this choice was made.

The focus on troubleshooting fast changed to another troubleshooting session. The module framework of Wonderland is supposed to be pretty straight forward. The community around this open source project has seemed wast and active. And so when I found back to the tutorials and guides for developing I had found in 2011, I had still a solid hope of success. Unfortunately I must aim some criticism against my self again. Open Wonderland, which is the name of the open source project, had changed its name from Project Wonderland a few years back in 2009. My assumptions was that this transition would not affect the module framework too much, that any huge changes would be throughly discussed in both the tutorials and the forum. This was

wrong. The source code has made a few quantum leaps forward from 2009. But most guides and tutorials was from after or at least in the middle of the transition. These guides tends to be more of a developers own best practices with tips and tricks involved. So for me as a fresh developer I had little critical eye to spot the changes and see the obvious replacement class dependency, file structure, or functions. But the good news is that with experience and a careful case study of other open source modules that do work, and do compile, I have been able to separate the obsolete documentation from the new, and thus being able to follow even complex guides that points to old parts of project wonderland.

The requirements in developing the open wonderland source code and modules was also a bit of a surprise. It seems that It's made nearly impossible to use other IDEs than NetBeans for development. This was not a huge problem since NetBeans is available for all platforms. My problem was then odd complications with file system on Linux and the file handling in NetBeans. I've tried to find the reasons for it but as fast as it came crashing into my project it vanished. Suddenly NetBeans got problems saving files. Especially files that was not made by NetBeans. It turned out that it had parsing error, but could still both read and compile the files, which was only 5-6 out of 30. The problem did have a couple of solutions on web forums, but non of them seemed to work. I troubleshooted more than 14 hours trying to find out why. Even reinstalled NetBeans deleted whole projects and tried changing files outside of NetBeans and restarting NetBeans. The oddest ting is that the problem suddenly stopped occurring, and when it came back the problem was solved by saving everything and just restarting NetBeans. I'm still puzzled of why this works now and not then.

11.2.3 The End

This is the section where I actually tell you about the implementation. The previous sections were only the hard and troubled way to freeing the hands of the developer.

The implementation is done by taking a module named "empty-module" from the repository of Open Wonderland and setting it up in NetBeans. In the module I utilize the HUD and the Object function of Wonderland. The latter is a standard way of implementing an object (a new "Cell" extending Java class) in Wonderland. My new Cell is attached to a mouse click listener which I have implemented to change shape and print messages in the chat HUD. Which brings us back to the first. A HUD is an abbreviation for Head-Up Display, which in this case means, that the WoW 3D module displays on a small portion of the screen with semi-transparent background. In the HUD the module displays questionnaires and can possibly be more of a toolkit than just an answer and question window.

Now the concept is that quest information and talk with NPCs is done

through the text-chatting while questionnaires only in the HUD. The main idea is still to create more advanced tasks. An in game example would be to travel to a world filled with computer parts. In this example a garbage dump for old computers. This can easily be created by drag and drop of images from google or other places straight into the virtual world. And tagging some images with a unique label. The task is then to assemble a computer of old parts. The HUD will then be a storage place for the items that is gathered. The player must then assemble the computer by laying down one item at a time in the right sequence in front of the NPC that sent the player on a quest.

The HUD utilizes the HUD and HUDComponent from the wonderland sources and uses Java Swing with JPanels for GUI. The visible object is implementing using Cell and the text-chat from NPC is done through fetching the (player) clients text-chat HUD and sending messages to it. This ensures that interaction happens client side for information while questionnaire which saves state and score, is done server side. Since the text-chat thinks it is connected to a user named NPC it doesn't know that the user NPC doesn't exist, and lets the player talk into an empty chat room. This ensures that interaction between NPC and player can go on mainly on client side, which improves both performance and give NPCs with unique states per client. The click on the object placed inside the world only invokes the client side functions to be activated, while it can later send the client side functionality extra data as example position or other if needed.

Chapter 12

Prototype

This chapter describe how the prototype works, and what functions it provides.

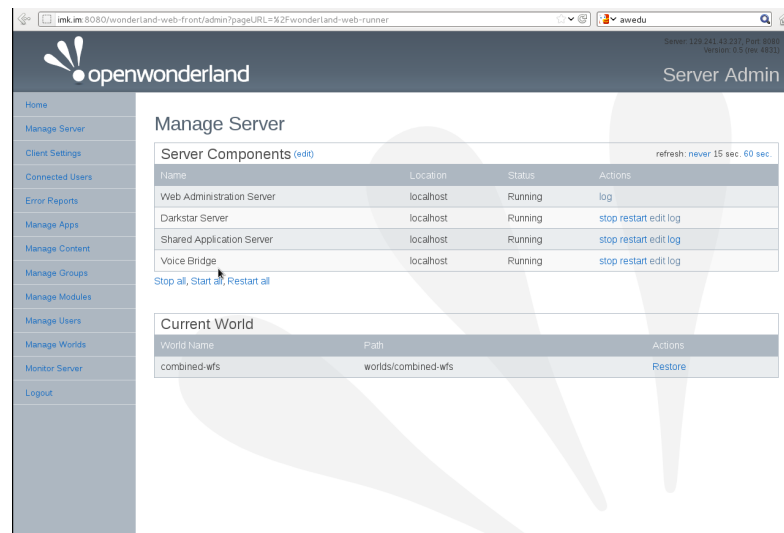


Figure 12.1: Prototype: The admin interface for the wonderland server.

The prototype is now made up by an mix of heads-up display (HUD) and NPC functionality. The HUD element is available through the Window menu for the client. When activated the HUD positions in the top right of the screen. The HUD can be seen as a kind of toolkit. It has both the ability to store items picked up for use in tasks and quests and has the ability to give questionnaires for the player to answer and by that score points and progress in the game.

The NPC functionality works as an object that can be placed around in the world. It can also be programmed to move around or make gestures, but this is likely too advanced for any teacher without programming skills

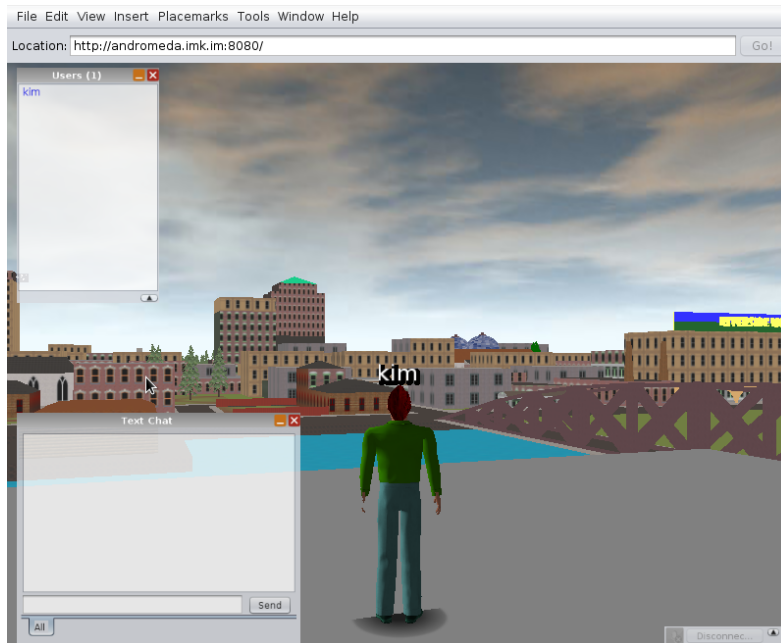


Figure 12.2: Prototype: A view over the town inside the virtual world.

to want to do. The NPC writes phrases into the text-chat when clicked on. These phrases can be edited same way as the object is moved, simply by right clicking and choosing edit. The player can then choose to seek a NPC to gather information, by clicking on them. The NPC can also give out information about quests and thereby leading players with tips and tricks.

The game play is rather simple. This was part of the plan for this project. It was supposed to be simple in order to get at least a solid platform to start from when further developing the concept. This is more of a proof of concept of that 3D world can be a framework of implementing advanced functionality for fun and educational games. The game might be simple now, but has lots of potential.

The prototype also include much more functionality than the basic game functionalities. This was also part of the requirements. A "first time player" should use the right click function on it's own avatar and choose configure. The player can then personalize their avatar with hair styles lengths colors, body contours, cloths and utilities as glasses, canes, or hats. Many more features than required is available in this section. An Evolver model importer is also in place, so that users may design their avatar externally (in a process that even uses a facial picture to better render the avatar), and upload through the client. Unfortunately Evolver Technologies have been acquired by Autodesk which has closed down the project which earlier was open and free to download and use.

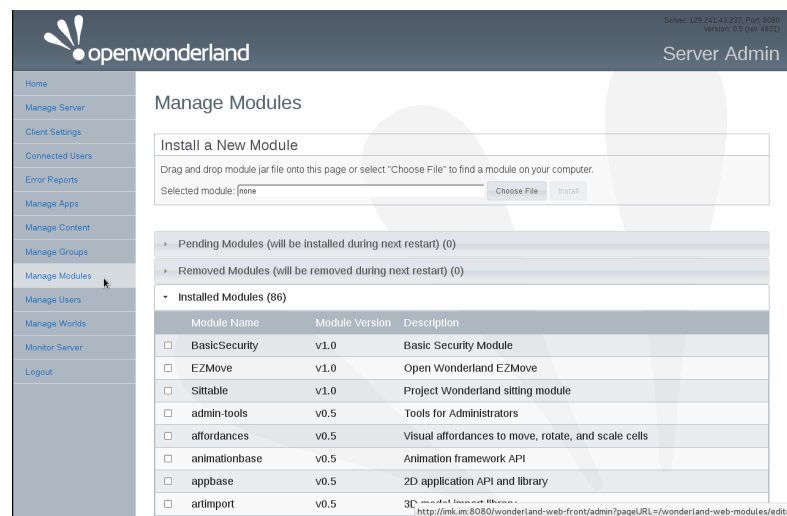


Figure 12.3: Prototype: The managing modules interface, where the wow-module is uploaded to the wonderland server.

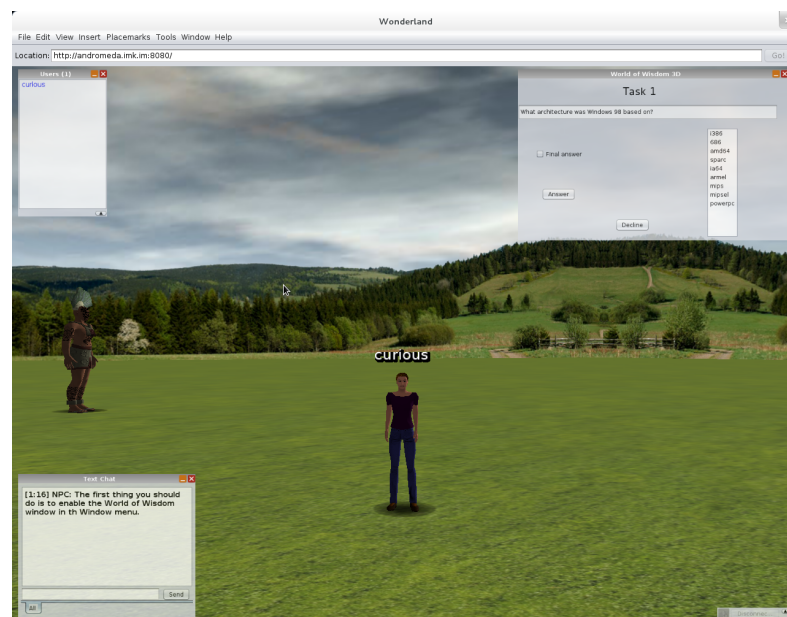


Figure 12.4: Prototype: The player is given a task from a NPC. World of Wisdom HUD in upper right corner.

All the basic functionality as student-student and teacher-student text and voice chat, setting favorite points in map, and sound effects and music are also included in the prototype.

12.1 Installation

The prototype currently consist of a basic Wonderland instance and plug-in modules. World of Wisdom (WoW) 3D is one of these plug-in modules. In my case the server is an average desktop computer with Debian Lenny (stable) amd64 server. And I used 7.03 version of the Sun JDK from Suns website.

1. Set up Sun Java Java Development Kit (JDK) on the server.
2. Download the Wonderland.jar file from <http://openwonderland.org/download/binary>
3. Place the jar file and the content of the settings folder from the zip-file submitted with this report, into the same location on the server.
4. Make the wonderland-file, which is a bash script, executable and change the path in the "my.run.properties" file.
5. Run the wonderland-file.
6. Navigate using a browser to `http://<server-fully-qualified-domain-nameorIPAddress>:8080/wonderland-web-front/admin`
7. Use the Manage Modules interface and remove the "security-session-noauth" module.
8. Use the Manage Modules interface and install all modules in the extra-modules folder from the zip-file.
9. Install the wow-module in the same interface.
10. Restart the server either in the admin interface or by ending the running instance on the server with Ctrl+c and restart it.
11. Use the Manage Users interface and update each of the users with the password found in the wonderland.password file.
12. Use Manage Server interface and edit the component Darkstar Server, and add the property "sgs.password.file" with the correct location of wonderland.password with full path.
13. Use Manage Server interface and edit the component Shared Application Server, and add the property "sas.password.file" with the correct location of wonderland.password with full path.
14. Use Manage Server interface and edit the component Voice Bridge, and add the property "voicebridge.password.file" with the correct location of wonderland.password with full path.

15. Restart the server either in the admin interface or by ending the running instance on the server with Ctrl+c and restart it.

Now the Wow object should be available along with the World of Wisdom menu-item inside the game. The admin interface is now only available with username admin and the password from wonderland.password.

The game is also available for testing purposes on the site <http://imk.im> with username "curious" and password "GwowAwowMwowE". Since this site may experience some period of downtime, you may contact me or Alf Inge Wang if the game is unavailable.

Chapter 13

Tests

This chapter describe some test scenarios for prototype evaluation. The result of these tests is in chapter 14, while the results will be discussed and evaluated in chapter 15.

13.1 Test Scenario Setup

The test environment consists of one server and one client. The client launches on a client computer while the server is already running on server computer. Both machines have been subject to normal use before testing started. Both machines installed/upgraded Java and installed the server/client before starting testing.

13.1.1 Server

- CPU: Intel Core 2 Duo 2,4 GHz
- HD: 1 TB
- RAM: 4 GB
- GPU: None
- Net: 100 MB/s
- URL: <http://imk.im>
- OS: Linux 2.6.32-5-amd64
- Java: Sun Java JDK 1.7.0_03-b04

Wonderland server was ran as normal a user on server computer with Java (Java JDK), while client was launched as normal user at client computer. Client downloads and launches software by opening a web browser at server URL and utilizes Java Web Start (Java JRE).

13.1.2 Client

- CPU: Intel i5-2520M 2,5 GHz
- HD: 64 GB SSD
- RAM: 4 GB
- GPU: Integrated Graphics Controller
- Net: 11 MB/s
- OS: Linux 3.2.0-2-amd64
- Java: Sun Java JRE 1.7.0_03-b04

The client was a laptop connected to the Internet through a wireless connection limited to 11 MB/s, stationed in Oslo. While the server stationed in Trondheim was connected to the Internet through the core network of NTNU limited by a 100 MB/s wired connection. Client downloads and launches software by opening a web browser at server URL and utilizes Java Web Start (Java JRE). User of client either authenticates as registered administrator or logs in as guest.

13.2 Prototype Test Scenarios

The tests that will be ran are:

- T-UG1 Complete a task (table 13.1)
- T-UG2 Complete a quest (table 13.2)
- T-UG3 The player should be able to communicate with another student(table 13.3)
- T-AG1 Change reward for a specific task (table 13.4)
- T-AG2 Import a feature from Warehouse (table 13.5)
- T-AG3 Edit a user profile (table 13.6)
- T-AG4 Create a new quest (table 13.7)
- T-AG5 Communicate with a student (table 13.8)

Table 13.1: Prototype Test Scenario T-UG1

T-UG1	General Game Play
Goal:	Complete a task
Dependencies:	G1 G3
Pre-condition:	Started game as player, player is inside task area
Artifact:	Player interacting with game, player answer question correctly
Response measurement:	Player score/level increments, new tasks available
Failure criteria:	Player completes a task without answering question right, player not given a task in task area

Table 13.2: Prototype Test Scenario T-UG2

T-UG2	General Game Play
Goal:	Complete a quest
Dependencies:	G1 G3 UG1
Pre-condition:	Started game as player, player is inside quest area, quest is available
Artifact:	Player interacting with game, player answer question correctly
Response measurement:	Player score/level increments, new tasks available, new quests available
Failure criteria:	Player completes a quest without answering question right, player not given a quest in quest area, player given quest without being qualified

Table 13.3: Prototype Test Scenario T-UG3

T-UG3	Communication
Goal:	The player should be able to communicate with another student
Dependencies:	G1 G3 G2/G4
Pre-condition:	Player is in game (and in proximity to another player for oral communication)
Artifact:	Player opens a in-game-terminal and write/read messages to/from other players, Player speaks into microphone and listens to game audio
Response measurement:	Visible in-game-terminal, Voice audio from other players
Failure criteria:	No voice sounds from game, missing or static in-game-terminal

Table 13.4: Prototype Test Scenario T-AG1

T-AG1	Change game settings
Goal:	Change reward for a specific task
Dependencies:	UG1
Pre-condition:	Started game as administrator and existing tasks
Artifact:	Administrator writing new values for reward to task
Response measurement:	Task gives different reward in game play
Failure criteria:	No existing task, Task not updated

Table 13.5: Prototype Test Scenario T-AG2

T-AG2	Plugin
Goal:	Import a feature from Warehouse
Dependencies:	G3
Pre-condition:	Started game as administrator
Artifact:	Import feature through menu and update game
Response measurement:	The new feature can be started by a user in game
Failure criteria:	Feature not available in any users list of features, or cannot be started by any user

Table 13.6: Prototype Test Scenario T-AG3

T-AG3	Edit user profile
Goal:	Edit a user profile
Dependencies:	G3 G6 (EF1)
Pre-condition:	Started game as administrator, opened edit user profile dialog
Artifact:	Edit values in edit profile dialog box
Response measurement:	user profile has changed
Failure criteria:	user profile changes is not saved

Table 13.7: Prototype Test Scenario T-AG4

T-AG4	Create quest
Goal:	Create a new quest
Dependencies:	G3 G5 UG2
Pre-condition:	Started game as administrator, new quest dialog
Artifact:	Input values in new quest dialog, refer depending tasks
Response measurement:	New quest available in game
Failure criteria:	No new quest in game for player

Table 13.8: Prototype Test Scenario T-AG5

T-AG5	Admin communicate
Goal:	Communicate with a student
Dependencies:	G3 G4 (G2)
Pre-condition:	Started game as administrator, open in-game chat terminal or proximity to other players
Artifact:	Speak in microphone or text in message
Response measurement:	answer in form of voice sound, visible message in terminal
Failure criteria:	no sound or no message visible in-game chat

Chapter 14

Result

The Result chapter shows the results of the tests in chapter 13, and gives an overview of the status of the implementation (chapter 11) of World of Wisdom (WoW). The chapter ends with a summary of each requirement from chapter 9, Software Requirements, and a summary of remaining implementation work based on unmet requirements in the current prototype in table 14.3.

14.1 Prototype Test Results

The results of the implementation is shown through running the tests from chapter 13 on the prototype. The results is divided into six categories which elaborate the test results. Not all requirements is discussed in full but a summary is presented in table 14.1, which shows the current status of all requirements of WoW 3D.

14.1.1 General Game Play

The tests for the UG1 and UG2 requirements (table 9.2) of general game play, T-UG1 (table 13.1) and T-UG2 (table 13.2), should test whether or not the prototype can be used to complete a task and a quest for a player. This of course depends on that an administrator has already created the tasks and quests, which again depends on the AG4 9.3 requirement.

The result of the test T-UG1 (table 13.1) is failed. The result of the test T-UG2 (table 13.2) is failed. Currently then game does not save the states of the tasks and quests. The game provides an interface for the user to retrieve and complete tasks and quests but does not save this progress for an administrator to easy evaluate.

14.1.2 Communication

The requirements UG3 and AG5 from table 9.2 and 9.3 is tested in T-UG3 (table 13.3) and T-AG5 (table 13.8). The requirements is for an administrator

(teacher) to be able to communicate with a user (student), and a user to communicate with another user.

Both the tests, T-UG3 and T-AG5, results is passed. The communication is open for global or one-one through text chatting or voice audio sharing. The privacy of voice audio can be sat up in-game by the administrator inviting a user into a virtual private room, or by choosing to have a private conversation in the users HUD in the user interface.

14.1.3 Game Settings

AG1 from table 9.3 requires the game to be able to change game settings and in specific a reward for a specific task. The score system in the game is currently not in function and will therefor fail this test.

The test T-AG1s (table 13.4) result is failed.

14.1.4 Plug-ins

AG2 from table 9.3 requires the game to have an easy access to installing plug-ins. This is available from the manage server website which is part of the Wonderland platform. Some plug-ins require the running server process to be restarted.

The results from test T-AG2 (table 13.5) was passed.

14.1.5 Profile Management

The AG3 software requirement from table 9.3, stated that the game needs an interface of managing the profiles of the users and especially edit a users profile. For example reset the players score.

The functionality available through the server management website can change name, username and password, but the extended profile of which is part of the score system is not available. But still test T-AG3 (table 13.6) only states that one should be able to edit a users profile as an administrator, and this is available.

The result of test T-AG3 is passed.

14.1.6 Quest and Task Management

The AG4 requirements from table 9.3 requires a quest editor in the game order to be fulfilled. The quests will often also contain or be based on a series of tasks and therefor depends on a task editor, and the availability of tasks and quests in the game. The quest and task editor, which should be a combined editor for both, is not implemented but it is possible to create quest information and have the NPCs in the game distribute to players. But since this must happen by editing properties of the NPCs one by one manually, this can not be counted as a passed test.

The result of test T-AG4 (table 13.7) is failed.

14.2 Implementation Requirements Status

There are some requirements that have not been solved by the implemented prototype. These are to be implemented in future projects. Some of the requirements may also be developed by the Open Wonderland community as there are ongoing development av modules for the Wonderland platform that can easily be installed in this version of Wonderland in the prototype.

The requirements are all listed in the tables 14.1 and 14.2 below with status and level of completion. An overview of only the remaining tasks is listed in table 14.3.

Table 14.1: Status of requirements after implementation of the prototype.






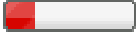












Req.	Description	Status
G1	The user must be visualized in the virtual game world through an avatar	
G2	The game should have implemented basic sounds for fighting, moving and other common actions.	
G3	The player should be able to start the game, and move freely in the virtual world.	
G4	The game should support an in-game chat.	
G5	The game should have programmable NPCs.	
G6	The game should store a “profile” between sessions which keeps track of both progress and Key-bindings per user.	
UG1	Complete a task	
UG2	Complete a quest	
UG3	Communicate with another player	
AG1	Change reward for a specific task	
AG2	Import a new plug-in	
AG3	Edit a user profile	
AG4	Create a new quest	
AG5	Communicate with a student	
EF1	It should be possible to define different key-bindings.	
EF2	The game should have a game play that includes a series of event that form a story which the player participates in. This can include events that alter the storyline as well.	
EF3	The game should have a background music that the player can turn on and off.	
EF4	It should be possible to play the game with gamepads	

Table 14.2: Status of requirements (cont.)






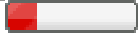
















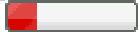

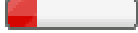
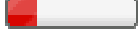















Req.	Description	Status
FR1	The player can give his/her character a name.	
FR2	The visual character can be customized(eg. hair color,gender).	
FR3	The characters progress in the game will be represented by their level.	
FR4	The character stats will improve during the game.(Level up)	
FR12	The character may bind to a respawn point in friendly zones.	
FR17	Each character has a knowledge-book, which will contain all the questions the character have answered. If the player have answered correctly, the correct answer will be stored.	
FR20	When the player clicks on a location, the character avatar moves through a legal path to the location, or teleports when available.	
FR21	The players can chat in several chat channels, team channel for current party, area channel to everyone in the current area, world channel that goes to everyone on the server, and guild channel that goes to everyone in the players guild.	
FR23	When completing certain tasks, the player will be rewarded with a badge. This badge will have no effect on the character, but will be used to encourage the players to complete this tasks.	
FR24	A friendly NPCs will provide information, give quests and/or trade with players.	
FR27	Only the course staff(professor and assistants) may add or remove game specific elements in their world.	
FR28	It should be easy and quick to add new quests and monsters to the world.	
FR29	When adding a monster, the course staff selects a theme topic	
FR30	It should be easy and quick to alter old quests and monsters.	
FR31	There will be an external editor where the course staff may add new questions, monster types and new zones.	
FR32	The professor may appoint a number of assistants which can help the professor maintain the world and add new content.	
FR33	There should be optional templates for worlds, so that the professor only need to add the task content. Which should greatly reduce the time needed to make a world ready for players.	
FR34	Question answering or problem solving is the only way to win a battle.	
FR36	When a character wins a battle, he/she should gain experience.	
FR37	It should be possible to create a party. Each party can maximum contain X players, where one is the party leader. Players can join or leave a party as they like.	
FR38	Graphics should be seperated from the rest of the client code, so that new GUIs can be easily made.	

Table 14.3: Unsatisfied requirements after implementation of the prototype.

Req.	Description	Status
G5	The game should have programmable NPCs.	
G6	The game should store a “profile” between sessions which keeps track of both progress and Key-bindings per user.	
UG1	Complete a task	
UG2	Complete a quest	
AG1	Change reward for a specific task	
AG3	Edit a user profile	
AG4	Create a new quest	
EF1	It should be possible to define different key-bindings.	
EF2	The game should have a game play that includes a series of event that form a story which the player participates in. This can include events that alter the storyline as well.	
EF4	It should be possible to play the game with gamepads	
FR3	The characters progress in the game will be represented by their level.	
FR4	The character stats will improve during the game.(Level up)	
FR17	Each character has a knowledge-book, which will contain all the questions the character have answered. If the player have answered correctly, the correct answer will be stored.	
FR21	The players can chat in several chat channels, team channel for current party, area channel to everyone in the current area, world channel that goes to everyone on the server, and guild channel that goes to everyone in the players guild.	
FR23	When completing certain tasks, the player will be rewarded with a badge. This badge will have no effect on the character, but will be used to encourage the players to complete this tasks.	
FR29	When adding a monster, the course staff selects a theme topic	
FR30	It should be easy and quick to alter old quests and monsters.	
FR31	There will be an external editor where the course staff may add new questions, monster types and new zones.	
FR36	When a character wins a battle, he/she should gain experience.	
FR37	It should be possible to create a party. Each party can maximum contain X players, where one is the party leader. Players can join or leave a party as they like.	

Part V
Reflections

Reflections

The part V of this report tries to give an Evaluation (chapter 15) of the Results (chapter 14) found in part IV, and the project as a whole. This part also gives a Conclusion (chapter 16) of whether or not this project succeeded in achieving the Project Goal (chapter 1.4) presented in part I, as well as give an insight in Future Work (chapter 16) and next steps after this project.

Chapter 15

Evaluation

This chapter emphasize reflection around the results from prestudy, with what lessons could be learned from other projects and available research in the field. The evaluation chapter also evaluates the software requirements, implementation, the prototype and the results of prototype tests.

15.1 Prestudy Evaluation

The prestudy showed status and progress in the previous projects. Unfortunately the previous reports seemed incomplete and were missing essential parts of information. The prestudy did not discover the problems with source code from previous projects. Here the prestudy could have involved studying and testing the source code as well.

The research into the field gave many new insights that helped create a new understanding and motivation for creating the game WoW 3D. The particular speech from Jane McGonigal on TED Talks [39] can be highly recommended.

The state of the art became a very wide search and a more specific to the virtual world type. This could have made the chapter more comprehensive on the interesting parts and neglected some of the games found to have little in common with the new 3D version of WoW.

The search for a engine or 3D virtual world that satisfied the needs for WoW 3D, was conducted, and two technologies stood out. The ongoing tests and asserting with requirements followed the plans laid in chapter 3.

The research was wide and narrow, which showed the field but also elaborated the important and interesting subjects of the prestudy. The prestudy have also been of great use to the succeeding parts of this report.

15.2 Tests Evaluation

In the test chapter (chapter 13), eight tests was performed to evaluate the functional requirements of the prototype. The tests was made to evaluate

several requirements and by that be harder to pass. The summary in table 15.1 shows the test results.

Table 15.1: Summary of test results.

ID	Description	Result
T-UG1	Complete a task	Failed
T-UG2	Complete a quest	Failed
T-UG3	Communicate with another player	Passed
T-AG1	Change reward for a specific task	Failed
T-AG2	Import a new plug-in	Passed
T-AG3	Edit a user profile	Passed
T-AG4	Create a new quest	Failed
T-AG5	Communicate with a student	Passed

As the table 15.1 shows, 4 of 8 tests have failed. This can be evaluated as the implementation was not successful. The reasons for this is likely to be a too small time frame for implementation. The original time frame for this project calculated extensive use of the old code base from WoW 2D, and when this plan had to be discarded the new did not with comply with the time frame available to implementation phase.

15.3 Prototype Evaluation

Sadly many of my plans for this game only ends in the planning stage. Partly this is because the project has diverted away from the original plans and will slightly miss both time schedules and product goals. The main idea was to keep both concept and game play of the old game, but this was not possible of practical reasons with old code being incompatible to such extensive changes as an extra dimension.

The prototype is far from complete, but under prestudy, setting of software requirements and implementation, the idea and concept of this game changed. This may not be clearly visible throughout the previous chapters but has been an ongoing process throughout this project. In the same time this report has been written throughout this project and is more clear on reflections and thoughts and experiences in the last chapters than in the first.

The change is from the old concept of using weapons and violence to simulate the student as a player who needs to fight and go to war against monster and unfriendly NPCs. This concept has been a subject of controversy when Norway has had to deal with a terror incident where the terrorist used violent video games as inspiration and learning platform. This and own personal opinion had strong influence on the desire to change the concept. The reason the concept was not changed much earlier was the belief that there was no

way of completing a prototype without utilizing a large part of the old source code to build on. But when early prototyping clearly showed the old code was incapable of being split and used as module, and the prototype was decided to be made from scratch, the choice of concept was without the previous constraints.

The change in concept also freed up constraints from the old concept and gave room for new requirements in the new version. These new requirements was not clearly stated but some of them ended up in the extended functionality table in requirements chapter [9].

While the prototype may seem empty it has many features which are of the type mentioned over, but not clearly stated. These are functionality as soft phone (in-game phone for voice chat between player not in the same game world area), visualization of physics experiments and ability for users to create their own space and personalize it.

Another aspect not directly addressed but mentioned in many of the research papers used in this report is the need for collaboration in the game, and the need for a learning situation that goes beyond and not only simulates the traditional learning situation in a classroom, for it to give high educational value. Mentioned in the YouTube video [29] by Karl M. Kapp and Tony O'Driscoll, they emphasize the use of collaboration in learning games, and especially the many learning situations that give high learning potential, mark that blackboard teaching is not one of them. The same does Nicole Yankelovich in her presentation of Open Wonderland, where she emphasize that the learning platform should not be used only to give classed in a virtual classroom instead of a physical one.

The original concept only contained a questionnaire and an unclear story about monsters and unfriendly characters in the game. This might not be traditional classroom education, but with only the questionnaire as educational integration, the original concept is more of a simulation an older concept than a new concept. The concept now is something quite different. With the possibility to work with objects and assemble and disassemble them the result will be much more interactive. I think this will hide the fact that students in the game are being tested and measured in the questionnaire less obvious. Because as Michael Zyda writes in his article, about the educational platform moves towards the virtual reality and game development area:

Pedagogy and story integration involve determining theories and developing practices for inserting learning opportunities into story, such that participants find the story immersive and entertaining because the embedded instruction remains subordinate to it. The game industry has already witnessed the failure of edutainment, an awkward combination of educational software lightly sprinkled with game-like interfaces and cute dialog. This failure shows that story must come first and that research must focus on combin-

ing instruction with story creation and the game development process.[65]

The World of Wisdom (WoW) 3D has already many educational characteristics as more than a game, with collaboration tools as portable document format (PDF) viewer, image viewer, video player, audio player, shared whiteboard, posters and even scrum board. These were never part of the requirements but is certainly a positive for the wider use of World of Wisdom to become a learning platform for all courses because of the social networking capability and community building functionality.

15.4 Research Questions Revisited

The research questions was reviewed with the summary of Part III, Prestudy. The questions will now be revisited once more to enlighten new interpretations and possible update the answers with new knowledge.

15.4.1 RQ1: How can World of Wisdom be implemented into a 3D virtual world?

The answer is Yes. It can be done in multiple of the technologies in the prestudy. There are several options but with most of game engines and 3D virtual worlds being a subject of high price license, the ones that pointed out as particular useful for the task is Wonderland and OpenSim. Both is suitable to the demands in chapter 9.

15.4.2 RQ2: Can a MMORPG be a useful platform for learning?

From the prestudy and own experience when implementing, yes. Several studies have been made in the field of computer-based learning, and there are few critiques to the fact. There may be disagreement of to how useful MMORPG games in general is for learning, but non of the articles I have come across denies the fact that MMORPG even without educational content provides a learning situation for the player. Especially in areas as collaboration, leadership and tactics.

15.4.3 RQ3: What advantages can WoW get from using a COTS 3D engine?

The advantages was greatest from using a 3D virtual world opposed to using a clean 3D engine. This research question should have earlier been pointed directly to the virtual worlds instead of also studying the 3D engines which could not match in evaluation against the software requirements.

15.5 Project Evaluation

The project as a whole has been very exiting and instructive. The preparation to this master project by the specialization project in the semester before gave answers to many questions that is often hard to see and understand in the beginning of a such huge project.

For me the project has been done alone. This has had mostly disadvantages in that many discussions have been left out and new ideas and insight have been missed. The diversity of this master report may have suffered as well as the prototype which could have been made much better by having a second person helping in troubleshooting and development.

The main obstacle occurred when all the code base from WoW 2D had to be discarded. The outlooks from that point seemed very bad. It was very late in the project, and the mistake of thinking I had lots of code finished ready to use and therefor lots of time was fatal. The turn happened very late as I tried for as long as I could to make the old code work. But the complexity and hard chaining to the old graphics engine were to much to tackle on my own. Since the turn was so late I haven't had the time to implement all the features I wanted and planned to do. This will of course look bad in a perspective of finishing with half a prototype. But I believe that the research done before and under this project will shine through.

The choices I have made of going as open as possible with open source, no license fees, large community, and active developers and users (If you post a commentary of any modules in the warehouse, the module will be checked and a feedback will come in a couple of days), is deliberately and will contribute to the serious games and learning games community. I have done what so many other games in the list of serious games have not. I have chosen to give this contribution to anyone that sees potential and wants to further develop the source code and the prototype. I feel this is a shortage when it comes to other games, like the analog electronics game, using a game engine which prohibits the author of the game to redistribute the game to others. Or like Virtual U and AWEDU, both with proprietary game/virtual world engines. What is the point of creating a serious game that solves the problematic of mixing games and education, if you are not allowed to share this solution, so everyone can learn from it? I am sure this also can be a reason why so many serious games can be perceived as clones of one and other. Since few is allowed/wants to share their source code and knowledge of how to create a successful serious game, many projects reinvent each other instead of being able to continue where other stopped.

To reflect on what's been learned I would say that taking old code for granted can be fatal, writing the mandatory parts in a report, doing research and trying to finish as much as possible of the report first may give a comprehensive report but doesn't necessarily help with programming the game. As

I have written many times in projects through last five years of this master program, "should have planned better", I come to realise that almost no plans actually is followed 100%. Every plan is broken from the start since a such matching plan would have fail in at least one area with either to fast or slow pace according to reality, and should one stress more and slow down, just to be according to plan? The answer is no. The plan has to adjust during the project to the people in it, not the other way around. And since I have been keeping a high pace, I see that working from a perspective with a plan as the authoritative timekeeper and boss, all flexibility needed for creative work will fail.

In the end I hope I have at least inspired or contributed to a discussion within the field of serious games, and maybe made a good start for someone else to continue.

Chapter 16

Conclusion and Future Work

The project as a whole was evaluated in the previous chapter. But as a continued discussion I have come to conclude several things from this project.

1. There is unproven that playing computer games is harmful, but still many claims this.
2. Serious games tend to be dependent on close source code, a strict license agreement, or never be publicly available to other researchers.
3. Creating a game with high learning potential is a matter of mixing both a game developers interests and a teachers interests.
4. WoW 2D was never possible to translate into a 3D version.
5. Open Wonderland is a well equipped tool for making virtual worlds that have huge potential for implementing successful learning environments.

With these conclusions I see that creating the World of Wisdom as a 3D game emphasized much more than just writing some new code and hacking together a game. The research and study before hand is what makes the choice of creating a such game possible. I see that the choice of this task as my master project actually was turn upside down. But luckily I also have the interest and motivation to find the research, study it, find technologies, study them, and ask questions like, why games create learning experiences. This all has lead me to conclude that a game like World of Wisdom 3D not only can be made, but must be made.

It is the next natural step in education. Integrate the parts of the world that young students feel familiar with, and they feels speak to them. Give students a chance to be met on their terms not the professor which tends to be one or two generations older than them.

The future work of this project should be to further develop the module to integrate in the Open Wonderland project. This module can utilize all

the open source code in modules available in the Open Wonderland Project, and will therefore follow a standard and be a part of a greater software. By following through one can easily collaborate with other universities (many of which have contributed to the project by making modules) and by that creating our own path of how the next MMORPG learning game should be.

The next project about learning games should emphasize the possibility of collaborative work between multiple institutes and universities when creating the biggest most successful serious game the world has ever seen.

Bibliography

- [1] Bjorn Andersen. Transformative, adaptive, responsive and engaging environment, 2011 (accessed December 8, 2011). <http://www.reachyourtarget.org/>.
- [2] Craig A. Anderson, Nobuko Ithori, Akiko Shibuya, Edward L. Swing, Brad J. Bushman, Akira Sakamoto, Hannah R. Rothstein, and Muniba Saleem. Violent video game effects on aggression, empathy, and prosocial behavior in eastern and western countries: A meta-analytic review. *Psychological Bulletin*, 136(2):151–173, Mar 2010.
- [3] Victor R. Basili. The experimental paradigm in software engineering. *Experimental Software Engineering Issues: Critical Assessment and Future Directives*, pages 3–12, September 1992.
- [4] B. Braunschweig and R. Gani. *Software Architectures and Tools for Computer Aided Process Engineering*. Computer-Aided Chemical Engineering. Elsevier, 2002.
- [5] Charles Butler. Metric assisted game design. Lecture at the xiii joingame conferece, NITH, April 2012. <http://www.idi.ntnu.no/grupper/su/joingame/media/wsxiix/05.pdf>.
- [6] David Callele, Eric Neufeld, and Kevin Schneider. Requirements engineering and the creative process in the video game industry. *13th IEEE International Conference on Requirements Engineering (RE'05)*, 1(13):240–252, 2005.
- [7] Edward Castronova. *Exodus to the Virtual World: How Online Fun Is Changing Reality*, chapter preface, pages xiv–xv. New York: Palgrave Macmillan, 2007.
- [8] The Game Creators. Darkbasic, 2012 (accessed April 18, 2012). http://www.thegamecreators.com/?m=view_product&id=2030.
- [9] Crytek. Cryengine, 2012 (accessed April 18, 2012). <http://mycryengine.com/index.php?conid=2>.
- [10] Rudy Darken, Perry McDowell, and Erik Johnson. The delta3d open source game engine. *IEEE Computer Graphics and Applications*, 25(3):10–12, May/June 2005.
- [11] Rudy Darken, Perry McDowell, Joe Sullivan, and Erik Johnson. Delta3d: A complete open source game and simulation engine for building military training systems. *JDMS, The Society for Modeling and Simulation International*, 3(3):143154, July 2006.

- [12] Jørgen Djuve. Mr. melk winter games - fra konsept til ferdig spill, 2012 (accessed April 25, 2012). <http://www.idi.ntnu.no/grupper/su/joingame/media/wsxiix/07.pdf>.
- [13] DNB. Dnb nor impossiball, 2011 (accessed December 19, 2011). <http://www.impossiball.no/>.
- [14] Kim Daniel Engebretsen. World of wisdom (wow) 3d. Specialization project, NTNU, December 2011.
- [15] Open Wonderland Foundation. Open wonderland documentation wiki, 2011 (accessed October 03, 2011). <http://code.google.com/p/openwonderland/wiki/OpenWonderland>.
- [16] The Overte Foundation. Opensim feature matrix, 2012 (accessed April 18, 2012). http://opensimulator.org/wiki/Feature_Matrix.
- [17] Epic Games. Epic udk - features, 2012 (accessed April 18, 2012). <http://udk.com/features>.
- [18] Garage Games. Torque 3d, 2012 (accessed April 18, 2012). <http://www.garagegames.com/products/torque-3d/overview>.
- [19] Gamespot.com. Virtual u, 2011 (accessed May 8, 2012). <http://www.gamespot.com/features/virtual-u-2694634/>.
- [20] Olaf Graven and Lachlan MacKinnon. Prototyping a games-based environment for learning. In Curtis J. Bonk, Mimi Miyoung Lee, and Tom Reynolds, editors, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2008*, pages 2661–2668, Las Vegas, Nevada, USA, November 2008. AACE.
- [21] Olaf Graven and Lachlan MacKinnon. Learning c++ using an mmorpg with embedded learning content. In Zoraini Wati Abas, Insung Jung, and Joseph Luca, editors, *Proceedings of Global Learn Asia Pacific 2010*, pages 3564–3574, Penang, Malaysia, May 2010. AACE.
- [22] Olaf Hallan Graven and Lachlan MacKinnon. Prototyping games-based environments for learning c++ programming. In *Proceedings of HCI Educators*, Dundee, United Kingdom, 2009.
- [23] Olaf Hallan Graven, D. Samuelsen, and Lachlan MacKinnon. Computer-based role playing game environment for analogue electronics. *International Journal of Online Engineering (iJOE)*, 5(1), 2009. Paper published in International Journal of Online Engineering (iJOE), 2009, Volume 5 - Special Issue: IRF'09.

- [24] Thomas Hainey, Thomas M. Connolly, Mark Stansfield, and Elizabeth A. Boyle. Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level. *Computers & Education*, 56(1):21 – 35, 2011.
- [25] Lyn Henderson. Video games: A significant cognitive artifact of contemporary youth culture. In *DiGRA*, Vancouver, Canada, 2005.
- [26] Active Worlds Inc. Active worlds, 2011 (accessed May 8, 2012). <http://www.activeworlds.com/>.
- [27] Geeknet Inc. Sourceforge.net: Project of the month, 2005 (accessed April 23, 2012). <http://sourceforge.net/potm/potm-2005-03.php>.
- [28] Andreas Johnsen, Lawrence Valtola, and Sondre W. Bjerkhaug. World of wisdom. Specialization project, NTNU, December 2009.
- [29] Karl M. Kapp and Tony O’Driscoll. 3d virtual world learning archetypes, February 2010 (accessed April 20, 2012). <http://www.youtube.com/watch?v=eNjV8Fvre4U>.
- [30] Thor Grunde Krogsæter. World of wisdom - world editor, user-interface for creating game worlds for world of wisdom. Master’s thesis, NTNU, 2009.
- [31] Thor Grunde Krogsæter, Henrik Halvorsen, and Esben Andre Føllesdal. World of wisdom. Specialization project, NTNU, December 2008.
- [32] Linden Lab. Second life, 2011 (accessed May 8, 2012). <http://secondlife.com/whatis/?lang=en-US>.
- [33] Mark J.W. Lee, Ken Eustace, Geoff Fellows, Allan Bytheway, and Leah Irving. Rochester castle mmorpg: Instructional gaming and collaborative learning at a western australian school. *Australasian Journal of Educational Technology*, 21(4):446–469, 2005.
- [34] Dictionary.com LLC. Dictionary:evoked, 2012 (accessed May 8, 2012). <http://dictionary.reference.com/browse/evoked>.
- [35] Torus Knot Software Ltd. Ogre, 2012 (accessed April 18, 2012). <http://www.ogre3d.org/>.
- [36] Jane McGonigal. Be a gamer, save the world. *The Wall Street Journal*, January 2011. <http://online.wsj.com/article/SB10001424052748704590704576092460302990884.html>.
- [37] Jane McGonigal. *Reality is broken*. Popular Science. Jonathan Cape, London, United Kingdom, 2011.

- [38] Jane McGonigal. Evoke, 2011 (accessed May 8, 2012). <http://www.urgentevoke.com/>.
- [39] Jane McGonigal. Gaming can make a better world, February 2010. http://www.ted.com/talks/jane_mcgonigal_gaming_can_make_a_better_world.html.
- [40] NAROM. S.p.a.c.e. Lecture at the xiii joingame conferece, S[&]T, April 2012. <http://www.idi.ntnu.no/grupper/su/joingame/media/wsxiii/08.pdf>.
- [41] Terry Hodgson (NASA). Nasa - moonbase alpha, May 13, 2011 (accessed December 8, 2011). <http://www.nasa.gov/offices/education/programs/national/ltp/games/moonbasealpha/index.html>.
- [42] Lasse Natvig. Age of computers (aoc), 2007 (accessed December 8, 2011). <http://www.idi.ntnu.no/~lasse/AoCshare.php>.
- [43] Ekaterina Prasolova-Frland, Theodor G Wyeld, and Mikhail Fominykh. *Virtual Campus of NTNU as an Arena for Educational Activities*, pages 244–251. ACTAPRESS, 2010.
- [44] Rodney P. Riegler and Wesley A. Matejka. The learning guild: MMORPGs as educational environments. *22nd Annual Conference on Distance Teaching and Learning*, 2006.
- [45] C.A. Steinkuehler. *Cognition and learning in massively multiplayer online games: a critical approach*. University of Wisconsin–Madison, 2005.
- [46] Unity Technologies. Unity 3d, 2012 (accessed April 18, 2012). <http://unity3d.com/unity/engine/>.
- [47] Dennis Titze. Evaluation of world of wisdom. Specialization project, NTNU, December 2010.
- [48] Tygron. Simport introduction, 2011 (accessed May 8, 2012). http://www.simport.eu/index.php?option=com_content&task=view&id=24&Itemid=45.
- [49] Tygron. Technical overview simport, 2011 (accessed May 8, 2012). http://www.simport.eu/index.php?option=com_content&task=view&id=27&Itemid=50.
- [50] Kirill Vainer. jmonkey engine, 2011 (accessed December 18, 2011). <http://jmonkeyengine.com/engine/>.
- [51] Alf Inge Wang. *Using a Mobile, Agent-Based Environment to support Cooperative Software Processes*. PhD thesis, NTNU, 2001.

- [52] Wikipedia. Ogre, 2012 (accessed April 18, 2012). http://en.wikipedia.org/wiki/OGRE_3D.
- [53] Wikipedia. Recreation in second life, 2012 (accessed April 18, 2012). http://en.wikipedia.org/wiki/Recreation_in_Second_Life.
- [54] Wikipedia. Torque 3d, 2012 (accessed April 18, 2012). [http://en.wikipedia.org/wiki/Torque_\(game_engine\)](http://en.wikipedia.org/wiki/Torque_(game_engine)).
- [55] Wikipedia. Unity, 2012 (accessed April 18, 2012). [http://en.wikipedia.org/wiki/Unity_\(game_engine\)](http://en.wikipedia.org/wiki/Unity_(game_engine)).
- [56] Wikipedia. Edutainment, 2012 (accessed April 24, 2012). <http://en.wikipedia.org/wiki/Edutainment>.
- [57] Wikipedia. Gamification, 2012 (accessed April 24, 2012). <http://en.wikipedia.org/wiki/Gamification>.
- [58] Wikipedia. Serious games, 2012 (accessed April 24, 2012). http://en.wikipedia.org/wiki/serious_game.
- [59] Wikipedia. Demoscene, 2012 (accessed February 7, 2012). <http://en.wikipedia.org/wiki/Demoscene>.
- [60] Wikipedia. Serious game, December 3, 2011. http://en.wikipedia.org/wiki/Serious_game.
- [61] Wikipedia. jmonkey engine, December 6, 2011. <http://en.wikipedia.org/wiki/JMonkeyEngine>.
- [62] Nicole Yankelovich. Open wonderland, Mars 30, 2010 (accessed December 13, 2011). <http://mfeldstein.com/open-wonderland/>.
- [63] Jeffrey R. Young. After frustrations in second life, colleges look to new virtual worlds. *The Chronicle of Higher Education*, February 14, 2010. <http://chronicle.com/article/After-Frustrations-in-Second/64137/>.
- [64] Marvin V. Zelkowitz and Dolores R. Wallace. Experimental models for validating technology. *IEEE Computer*, 5(31):23–31, May 1998.
- [65] Michael Zyda. From visual simulation to virtual reality to games. *IEEE Computer*, 38(9):25–32, 2005.