# *Muspellheimr - a 2D adventure game*

a dissertation presented

by

Robert Muschong

to

The Department of Computer Science

in partial fulfillment of the requirements

for the degree of

Thesis

in the subject of

Computer Science

West University

Timisoara, Romania

July 2015

Thesis advisor: Cosmin Bonchis Robert Muschong

# *Muspellheimr - a 2D adventure game*

Abstract

This bachelor thesis describes a case study, focusing on developing a 2D adventure game, using a process based upon Scrum, Agile and the evolutionary approach. The thesis will cover the implementation of the graphics and physics engine, logic systems as well as sound effects and background music. In the end, the case study will show that this development process was an appropriate choice for the project.

# Contents


1 Introduction **vi**

1.1 Methodology and Concept . . . . . vi

2 Game Design **viii**

2.1 Development process . . . . . viii

2.2 Game type and specifics . . . . . ix

3 Thoughts **x**

3.1 Particularities . . . . . x

3.2 Acknowledgments . . . . . xi

References **1**

# Author List

The following authors contributed to:

Chapter 1: Robert Muschong and Prof. Cosmin Bonchis.

Chapter 2: Robert Muschong and Prof. Cosmin Bonchis.

Chapter 3: Robert Muschong and Prof. Cosmin Bonchis.

*"The world is a country which nobody ever yet knew by description; one must travel through it one's self to be acquainted with it."*

Philip Stanhope, 4th Earl of Chesterfield

# 1 Introduction

## 1.1 Methodology and Concept

### 1.1.1 Methodology

Developing software is a costly and time consuming process. During the last years, several development methodologies have sprung to address this issue. Examples of such methodologies would be Scrum, Agile and the evolutionary approach. In this particular case an amalgam of practices from the afore-mentioned were chosen to ensure a succesful and less time-consuming development.

### 1.1.2 Concept

The project ended up going through several concepts as more features were added. The initial concept had a game play similar to The NetHack DevTeam's game Nethack[2], where the player character had to go through an elaborate labyrinth, fight various creatures and gain experience points to increase statistics[Ch.2] and items to reach the end of the map. Another idea was to develop a platformer, in the style of the classic Nintendo game, Mario but with various puzzles that the player had to analyze and figure out the best method to complete. As the second idea involved much greater work than the latter one, as each puzzle had to be manually made, it was given a low priority. A merge between the two was decided upon, where players would be able to fight through the labyrinth of monsters, collect coins and use them to make upgrades. Examples of upgrades were better weapons and armor. Such upgrades would be similar to the concept of levelling, something that many players appreciate. Most earlier presented ideas were kept and in addition to other improvements the enemy creature concept was expanded upon. Having only one enemy type would become very boring, very quickly. Thusly, expanding on various and diverse enemy types ended up being a major part of the final game.

*"Form follows function."*

Louis Henri Sullivan

# 2 Game Design

## 2.1 Development process

OpenGL was chosen as a suitable engine. The decision was made with regard to it having many great, in-built tools and provided a good framework, thus allowing the development to start as seamless as possible. A bonus was the fact that OpenGL[6] worked well with the C-Language family[4]. Ever since the beginning of development the vision for the project was for it to have good, quality pixel sprites. Thus, there wasn't a need for a sophisticated modelling program. With these choices made the development environment was the last decision on the list, a simple one for that matter, having prior used Vim with the semantic, clang-based code completion engine, YouCompleteMe. For the audio library OpenAL was the go-to choice.

## 2.2 Game type and specifics

With the decision of choosing a type of game being free from the very beginning, the project was required to have the following concepts: a statistics-based system, multi-threaded engine[1, 3], synchronous parallel models and last but not least a turn-based combat engine. Statistics in role-playing games usually are pieces of data that represent particular aspects of characters. These pieces of data are usually unitless integers or a set of dice.

The most often used types of statistic include:

1. Attributes – describe to what extent a character possesses natural, in-born characteristics common to all characters. Advantages and disadvantages are useful or problematic characteristics that are not common to all characters.

2. Powers – represent unique or special qualities of the character. In game terms, these often grant the character the potential to gain or develop certain advantages or to learn and use certain skills.

3. Skills – represent a character's learned abilities in predefined areas.

4. Traits – are broad areas of expertise, similar to skills, but with a broader and usually more loosely defined scope, in areas freely chosen by the player.

*"Science is what we have learned about how to keep from fooling ourselves."*

Richard P. Feynman

# 3 Thoughts

## 3.1 Particularities

Certain things had to be considered for the project. Specifically, a limit on the game world was required so that the project would not get too bloated and end up requiring an impossible amount of time to finish. Thus, a procedural generation system was decided to be put in place. Having a finite number of enemies in the game world was also a requirement to avoid an inhumane amount of time to complete. During development an issue sprung, the issue common to most aspiring game developers: the naive infinite loop resulted in variable framerate and variable time difference that made the game nondeterministic.[5] Finally, the engine itself needed to be functional aswell as practical.

## 3.2 Acknowledgments

I would like to first thank the members of the dissertation committee - not only for their time and patience, but for their intellectual contributions to my development as a scientist. I am indebted to and I would like to thank my thesis advisor, Prof. Cosmin Bonchis. I would like to thank my teachers for their contribution in helping me become better aquainted with Computer Science. Finally, but not least, I would like to thank my colleagues for the fun and support. Thank you for everything that helped me get to this day.

# References


[1] Bitsquid. Task-management, a practical example. [Online resource].

[2] The NetHack DevTeam. Nethack. [Online resource].

[3] Gamasutra. Multi-threaded game engine. [Online resource].

[4] Dennis Ritchie and Brian Kernighan. The c programming language. 1988.

[5] Fabien Sanglard. Game timers: Issues and solutions. [Online resource].

[6] Dave Shreiner, Graham Sellers, John Kessenich, and Bill Licea-Kane. Opengl programming guide. 2013.

# Milestones

Here follows a list of milestones and what tasks were assigned to which milestone:

1. Milestone 1 Implement fundamental classes, objects. Placeholder models in use
2. Milestone 2 Create a camera handler class, Input should be taken via mouse and keyboard
3. Milestone 3 Game engine implementation
4. Milestone 4 Implement logic system, add models
5. Milestone 5 Add effects, work on enemy pathfinding
6. Milestone 6 Tentative animated backgrounds, fix collision

# Colophon

This thesis was typeset using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 11 point Arno Pro, designed by Robert Slimbach in the style of book types from the Aldine Press in Venice, and issued by Adobe in 2007.