C++ Role Playing Game

Richard Came

Registration Number 13030333

Supervisor – Diego Perez

Second Assessor – Michael Gardner

Computer Science BSc

# Acknowledgements

Firstly I would like to thank my supervisor Diego Perez for his efforts in keeping me on track and providing insight into technical problems that I was facing throughout the project.

Perlin noise

Library for finding sdl2 on linux

lazyfoo

# Abstract/Summary

The project is to create a C++ Role Playing Game. The game is a 2-Dimentional fantasy Role Playing Game that is created using the C++ language and the SDL2 (Simple DirectMedia Layer 2) library. The game is designed to have a retro feel with pixelated graphics and simple real time combat system which is based around a levelling system.

The games terrain is completely procedurally generated, this will allow for greater re-playability as each time a new game is started it will be different to the last.

The objective of the game is to complete the quests that you are given. Some quests will be given to you at the start of the game, and others will be found while you play the game. Quests will range from locating treasure to defeating monsters. Once all quests have been completed the game is completed and the player wins.

Players level up their character by defeating monsters and completing quests. The monsters and quests will scale in difficulty with the players level, so that the game is always a challenge.

Contents

[Acknowledgements 2](#_Toc448677419)

[Abstract/Summary 3](#_Toc448677420)

[List of Symbols 5](#_Toc448677421)

[Technical Achievement 6](#_Toc448677422)

[Introduction 6](#_Toc448677423)

[Literature Survey 6](#_Toc448677424)

[Procedural Content Generation 6](#_Toc448677425)

[Pathfinding 6](#_Toc448677426)

[Project Planning 7](#_Toc448677427)

[Conclusions 8](#_Toc448677428)

[References 9](#_Toc448677429)

# List of Symbols

The following are acronyms that are used throughout this document:

SDL2 – Simple DirectMedia Layer version 2

NPC – Non-player character

Replayability – The amount of enjoyment that is received after playing a game multiple times, the higher the replayability the more times you can play a game while still enjoying it.

# Technical Achievement

## Introduction

This section will provide the main body of this report and will describe the processes that went into creating the game as well as the key decisions that were made during the games development. It will also cover the areas of the development cycle that were particularly challenging.

## Literature Survey

This project has mainly tried to solve the same problem that almost all games try to solve, and that is to solve the problem of boredom and the general tedium that is everyday life. The project tries to extend on the solutions to this problem that have been presented by other games in the past by trying make use of procedural content generation so that the game can be re-played many times while still providing a fresh experience to the player. This, it is hoped will help solve the problem for a larger period of time compared to a game that only has static content and is the same every time it is played.

This problem of boredom has been one that has existed for as long as humans have had free time, and probably even before that. It is fundamental in our drive to make use of our time in new and exciting ways, however sometimes in our modern lives there is nothing to do and our boredom can run wild resulting in unhappiness. This is where games can come in, they can fit into someone’s daily life and fill the spots where there is nothing else to do and provide some much needed excitement or pleasure. This can be hard to find in other mediums where you’re not actually partaking in the action, like films or books.

### Procedural Content Generation

The vast majority of the project was solving the problem of procedurally generating the games terrain in a way that looked realistic while providing enough features to make it interesting to both the eye and from gameplay perspective. To this end perlin noise [1] was essential in creating the terrain of the game, the continuous noise that is produced by the noise function allowed for a realistic looking terrain to be created. The noise produced can then be manipulated to create a terrain that suits the specific requirements.

### Pathfinding

The game makes use of pathfinding to create roads, as well as making use of it to move the NPCs (non-player character) around the map. As the game uses a simple 2D grid based system the pathfinding used was A\*. [2] A singular general pathfinding class was used in the project, this was then used by both the NPC characters and placing the roads on the terrain.

## Legal

## Project Aims

The main aims of the project are to create a fun and immersive gaming experience that has a large amount of replayability. This will be achieved by using procedural content generation to create a unique experience every time the player starts a new game, by generating a new game world each time. Alongside this the game should make use of procedural content generation to create quests and objectives for the player that make use of the games map so that the game has different quests each time making the player feel like they are in a completely different world compared to the last.

The game should also contain items, weapons and armour that provide bonuses for the player character and allow an extra element of strategy. These items should be created procedurally so that the player has different items to use rather than using the same items in each game they play. These items should be purchasable in shops in the game, as well as found as treasure and achieved as rewards for completing quests.

The game should have NPCs that the player can interact with; these characters will provide the player friendly interaction as well as a combat enemy. The two main types of NPCs will be the friendly characters that the player can interact with and talk to and the enemy characters that the player can fight with.

The game should have a fluid combat system, where the player can fight the enemy NPCs. The combat system should be designed around the fighting parties’ statistics, where the players level, and items denote how much attack and defence they have and the enemy characters attack and defence is denoted by their level and status.

The game should be displayed using a 2D overhead world view where the characters, buildings and certain terrain features such as trees are the only objects that appear to be in sudo-3D.

The game should run well without any bugs that cause it to crash.

## Project Objectives

The project objectives are as laid out in the Interim Report [3] the final state of these objectives will be given below, including any changes that were made to these objectives since the Interim Report.

PRIMARY GOALS

1 - PROCEDURALLY GENERATED TERRAIN

2 - PROCEDURALLY GENERATED STRUCTURES AND TOWNS

3 - PROCEDURALLY GENERATE

4 - PLAYER CHARACTER IMPLEMENTATION

5 – NON-PLAYER CHARACTERS (NPC’S) AI

6 - COMBAT MECHANICS AND HIT

SECONDARY GOALS

1 - LEVELING SYSTEM

2 - PROCEDURALLY GENERATED ITEMS

3 – MULTIPLAYER

4 - PROCEDURALLY GENERATED QUESTS

# Project Planning

# Conclusions

# References

|  |  |
| --- | --- |
| [1] | K. Perlin, “An Image Synthesizer,” *SIGGRAPH,* vol. 19, no. 3, pp. 287 - 296, 1985. |
| [2] | D. Perez, *High-Level Games Development lecture Slides,* Colchester: University of Essex, 2015/16. |
| [3] | “SDL - Simple Directmedia Layer,” [Online]. Available: https://libsdl.org. [Accessed 17th April 2016]. |