|  |
| --- |
|  |
| Fantasy & Ash (Final Report) |
| Calum Mathison 40406464  Arran Smedley 40406581 |

|  |
| --- |
|  |

# Table of Contents

Contents

[Table of Contents 1](#_Toc38470098)

[Introduction 2](#_Toc38470099)

[Changes / omissions 3](#_Toc38470100)

[Software Design 3](#_Toc38470101)

[Short Description of Game 3](#_Toc38470102)

[Evaluation 3](#_Toc38470103)

[Summary 3](#_Toc38470104)

# Introduction

## Scope & Content

The scope of this project was to create a fantasy 16-bit RPG style game similar to what final fantasy (1970) is like. This game was to include a menu, overworld and combat scene, the menu screen consisting of the basic elements of Play, Settings, Controls and Quit. The overworld scene consisting of a map in which the player can navigate 4 directions around in order to explore what is around & hittable objects such as water and rocks. The combat scene consisting of a turn-based mechanism with the player having stats such as health and mana power.

The main aim (Game Loop) of the game was to defeat the monsters around the map in order to receive coin and experience, using this coin to be able to be let back into the kingdom which would mean you have won the game. By earning this experience, you would then be able to progress onto harder enemies which would then give you more coin.

The content of the game provides the player with a menu screen in which they can navigate through using the WASD keys. Once beginning the game, the player is then met with an introduction scene, explaining the situation within the game. After the player has read everything, they enter the overworld scene in which they can fight monsters earn gold and rank up. In order to complete the game, the player must defeat a majority of the monsters within the overworld. Upon collision with the monsters the player is put into a combat scene in which they can choose to attack, magic or flee using the up, down, right and left keys.

## Inspiration

- Early JRPGs (Final Fantasy/Dragon Quest)

* Video Game
* the early JRPG game have influenced this title through its story telling, gameplay mechanics and graphical style. Specific influences may be prevalent in the games combat mechanics and overall style.

- Fantasy Novels (LOTR/Redwall/Grimgar)

* Literature
* Fantasy novels will influence this title through their world building and storytelling styles. Fantasy novels can build large believable worlds, and while that will not be in the scope of this project, it is something we will be taking influence from.
* Tabletop RPGs (D&D) o Tabletop games will influence this title through the different gameplay styles the player can use.

# Changes / omissions

Below is a list of planned features that were either omitted from the game or changed through the development process.

**Items**

Items were not made into the final product due to the timescale in which we had to complete, the original idea was for the player to be able to hold potions so that they could heal or restore mana power.

**Final Boss**

Due to the time taken to finish the basic game features, it was decided to omit the final boss character and to make the game more arcade like.

**Controller Support**

Due to the way input was implemented from the start, controller support was cancelled half way through the project as it became apparent that the entire project’s infrastructure would need to be revamped to include this feature.

**Dialogue**

Dialogue was omitted due to time constraints.

**Story**

The story telling aspect of the project was changed to be given simply by exposition at the beginning.

**Remappable Controls**

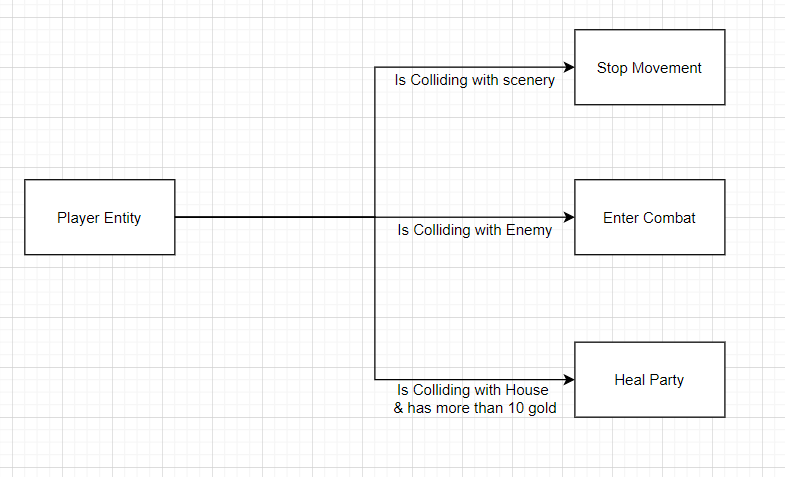
This feature was implemented into the game but was never used due to changes in the input management system.

# Software Design

## State Model

### Collision State Diagram

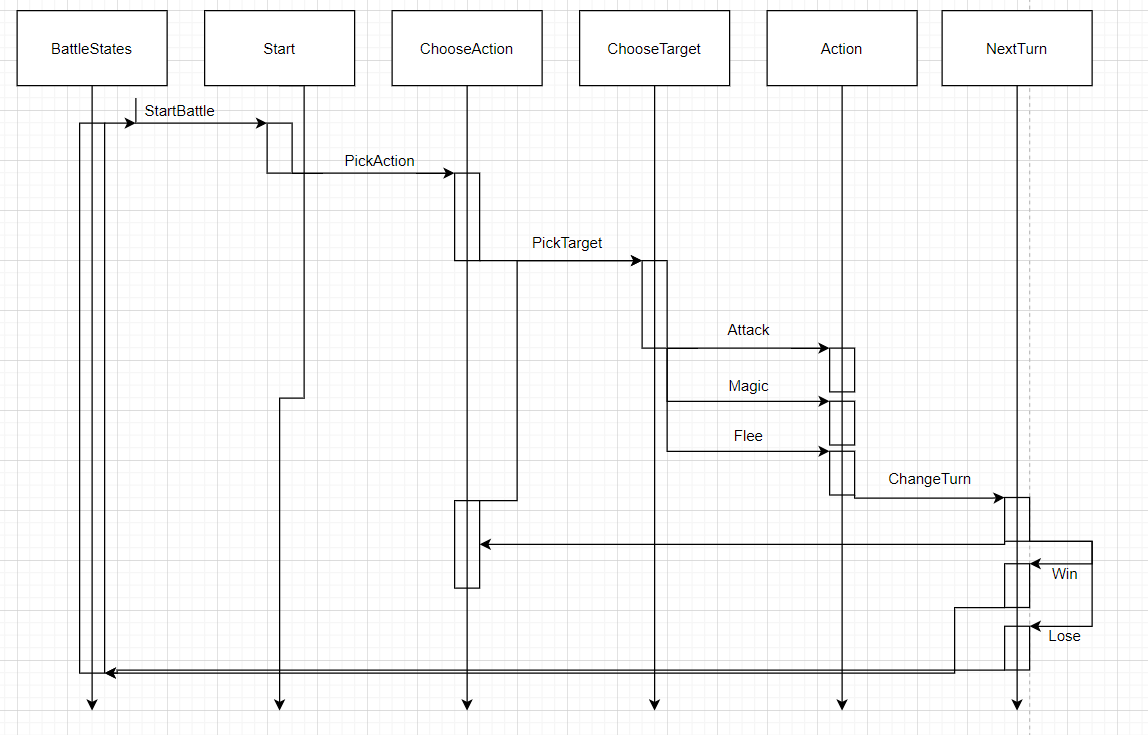
Below is a state diagram highlighting the possible collision states.



## Sequence Model

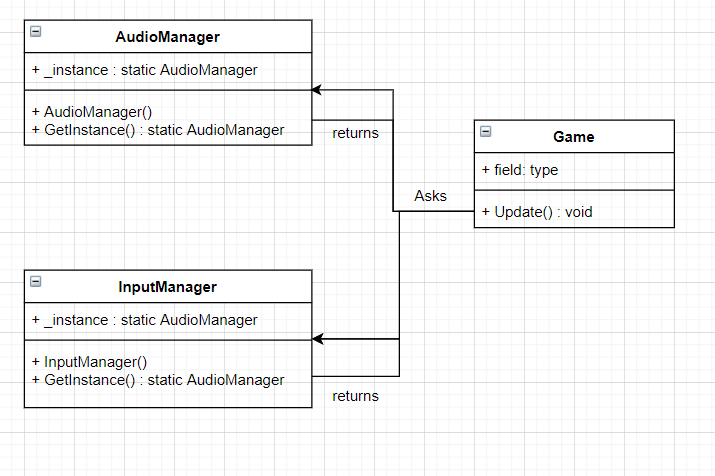
### Combat Sequence Diagram

Below is a sequence diagram representing the interaction between its different states.



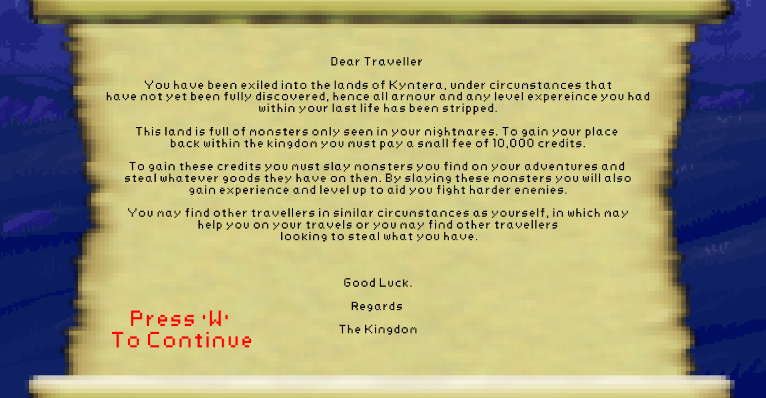
## Software Design Patterns

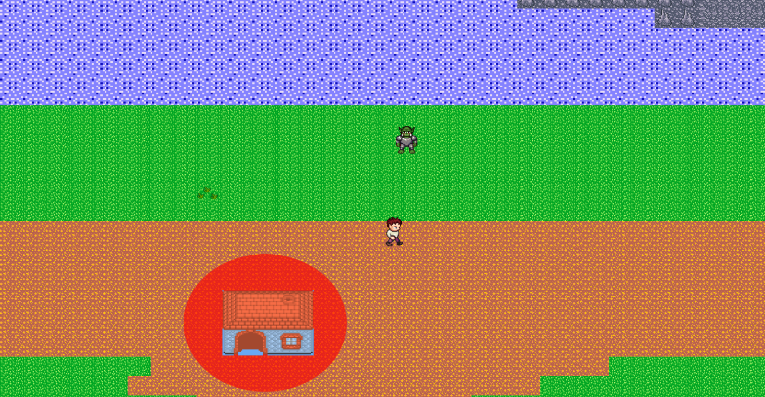
### Singleton Design Pattern

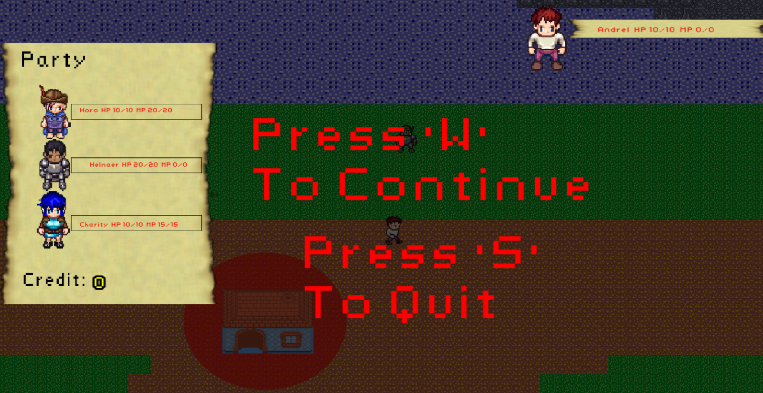
The Singleton pattern was used to create the Audio Manager and Input Manager. This ensured that only one instance of those classes was created and allowed us to handle the input and audio from one source respectively.   


# Game Description

## Screenshots







# Evaluation

# Summary