Reign (Game Demo)

Lewis Brown

CST-451 Capstone Project Requirements Document

Grand Canyon University

Instructor: Professor Mark Reha

Revision: 2

Date: 10/4/2019

**ABSTRACT**

Reign is a hack and slash video game where the Player will fight an enemy leading up to their inevitable demise. The player will have the ability to traverse the map, attack the enemy, and block an enemy’s attack. Once the enemy is defeated the player has the choice to leave the game or start again. At the start of the demo the player will start off with a sword and shield. The player will have the ability to attack, block, run, and jump in a completely immersive 3D environment.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
| 4/5/2020 | Lewis Brown | Revised abstract |
| 4/5/2020 | Lewis Brown | Revised Functional Requirements |
| 4/5/2020 | Lewis Brown | Revised Technical Requirements |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**TABLE OF CONTENTS**

Functional Requirements 4

Non-Functional Requirements 5

Technical Requirements 6

Logical System Design 7

User Interface Design 8

Reports Design 9

Functional Requirements

**Use Cases**

Use Case 1: Main Menu selection:

Main Flow:

1. The user will move their mouse over play game and left click to start the game
2. The user will move their mouse over Exit game and left click to close the application

Use Case 2: Game Play:

Main Flow:

1. The user can walk forward by pressing and holding down the ‘W’ key
2. The user can walk backward by pressing the and holding down the ‘S’ key
3. The user can walk to the left by pressing the and holding down the ‘A’ key
4. The user can walk to the right by pressing the and holding down the ‘D’ key
5. The user can attack the enemy by pressing the left mouse button
6. The user can block an enemy attack by pressing the Right mouse button
7. The user can jump by pressing the Space Bar
8. The user can run by pressing and holding down the Shift key and holding down the ‘W’ key
9. The user can crouch by pressing the Ctrl key
10. The user can look around by using the mouse
11. The user can bring up the pause menu by pressing the ‘Esc’ key
12. The Engine will handle the enemies attacking the user
13. The Engine will handle collision detection so that the user and enemies will not run through solid objects
14. The Engine handles damage calculation so that the user or enemies will slowly take damage after a direct hit see how much damage the enemy or play took then take it out of their health
15. The engine will handle enemy intelligence by having them know the player location and attacking them
16. The Engine will handle Collision detection on enemies and players so the cannot run through each other
17. The engine will handle collision detection for the floor so that the user or enemies wont fall through the floor

Use Case 3: Pause Menu:

Main Flow:

1. The user will go over with their mouse and left click on the main menu button to go back to the main menu
2. The user can select continue game by going over it with their mouse to get out of the pause menu and continue playing the game
3. The user can select Exit Game by going over it with the mouse and that will close down the application
4. The user can also press the ‘Esc’ key to exit out of the pause menu and continue playing the game.

Use Case 4: Controls Menu:

Main Flow:

1. The user can view all the controls in the game
2. The user can go over ‘return’ with the mouse and left click to go back to the pause menu

Use Case 5: Game Over Menu:

Main Menu:

1. The user can go over with their mouse and selecting ‘Try Again?’ to start the game over from the beginning
2. The user can go over with their mouse and selecting ‘Exit Game’ to go back to the main menu selection screen

Non-Functional Requirements

Performance:

* The Engine will need to load all game assets into memory in less than a minute. This is the loading time to get into the game as fast as possible
* The game must run at 60 frames per second to make the game play as smooth as possible
* Low-end standard definition graphics so that the game can run on most low-end computers
* The game should not use more than 3 GB of RAM while it is running
* All damage calculations must happen as soon as the player or enemy gets hit
* The game must work on at least a dual core CPU
* The game must work on a graphics card that uses 2 GB of VRAM

Technical Requirements

Unity: 2018.4.8f1

The Unity game engine is capable of rendering full 3D environments, handling complex animations, and uses C# as a programming language. Also has a store for various 3d models. This engine will be responsible for the following task:

* Rendering the 3d world the player will walk on
* Rendering the character that the player will play as
* Rendering the Enemies
* Rendering the equipment
* Mapping actions to buttons such as walking around by using the WASD keys
* Collision of objects and player
* Handling all in game rules

Visual studio 2019: 16.2

Microsoft’s IDE that is able to create classes that are written in C#. It also supports integration to Unity. This IDE will be used to create these components:

- Used for combat such as hit points and damage

- Calculate damage

- Basic ai for the enemies

- Collision between two objects

- Calculate how many enemies are on the map alive/defeated

Blender: 2.8

Blender is a 3dmodeling software that does 3d modeling and animation. Blender such as unity has its own 3d model store that may be used for this project, but 3d models will most likely be self-made. Animation taken form the store my include models of equipment and character prefabs. Blender will be used to complete these tasks:

- Walking animation

- Attacking animation

- Blocking animation

- 3d models of equipment such as sword, and shield

- 3d rendering of the entire world the player can explore

- 3d model of character the player plays as

- 3d model of Enemies

Make Human: 1.1.1

Make Human is an open source program that can make and render 3d humanoid models. Which also has a sizeable community that uploads different assets to Make Human’s online repository. Make Human will be responsible for completing these tasks:

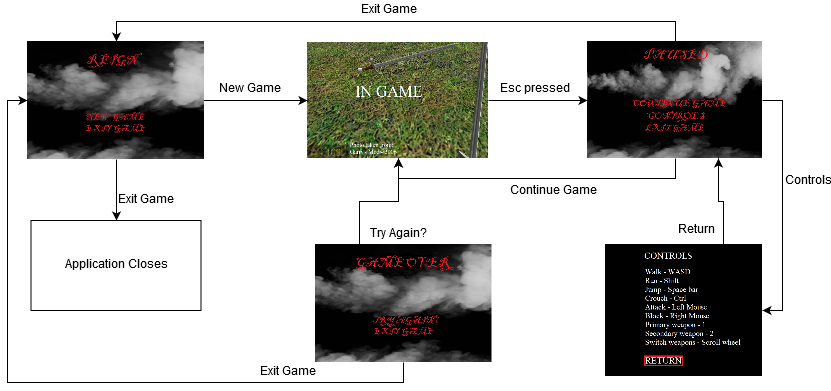
* 3D Character

Adobe mixamo:

An open source program with a large repository of free 3d models and you can upload your own 3d model and it will rig the model for you. Adobe mixamo will be used for:

* Rigging 3d characters

Logical System Design

****

User Interface Design

Main Menu:

New game - starts the game

Exit game – closes the application



Pause menu:

Continue game – continues the game

Controls – brings up the in-game controls

Exit game – brings you back to the main menu



Controls Menu:

Return – brings you back to the pause menu



Game Over Menu:

Try Again? - starts the game over from the beginning

Exit Game - takes you to the main menu



3D models in game

