**Abstract – 150 Words**

*Brief Summary of content and conclusions*

The aim of this project is to use the 2D functionality within Unity to create a simple platforming game. It will show off a range of skills such as animation, game design and software engineering.

**Introduction – 300 Words**

*Project and its aims.*

This is a 2D game which is inspired by Braid – taking the main game mechanic and adapting it into this project. One of the main game mechanics is to have the ability to travel your character back 20 seconds. I designed two different levels for the game – the game ends after the second level has been completed.

The main character is an adventurer, who is fully animated – with the ability to jump, climb, and run.

I did not create the original assets; sources are listed below. I meant this project to be a display of my skills within Unity and game creation.

The aim is to create a similar game in Unity, using 2D physics and tile maps.

**Game Design – 1 page**

*What makes this project a game? What sort of game is it? What will motivate someone to play this game? What will they get out of this? Describe the primary mechanic. How the mechanics fit together and make the user play?*

This is a side scrolling, platformer game. It revolves around an adventurer who must traverse ladders and platforms to reach the end goal. The camera follows the player - keeping it centred. This means that the focus on the player and allows for a complex and bigger screen.

People will play this game because it has a nice aesthetic, challenging levels and a variety of obstacles.

This project is game because it has a win and lose state. The win state is when the player reaches the flag – there is one per level. There are several obstacles such as falling rocks and spikes – they cause the player to die – ending the game.

The game also has a graphic user interface, in which you can restart and quit a level. The initial design of that is shown below.

I went through a few stages of designing levels – the basic level design is here though it needed a lot more testing when during the implementation stage.

**Software Design – 4 pages**

*Describe the principle components of the design. Describe how these components contribute to fulfilling the specification. Describe how the components fit together and contribute to the whole. Use UML diagrams to explain key points. What alternative designs do you consider (or try) and what are the pros and cons of these different choices?*

Game features are:

* Platforms
* Spikes
* Falling Rocks
* Time Travel
* Ladder
* Goal

Platforms have a sprite which can be varied by adding different decorations. They exist on the “PlatformLayer” which is used for player collision. That is checked by the player which changes ENUM state when it collided with the platform.

I also used a similar layer to limit the playing screen.

**Testing – 1 page**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Description | Expected Outcome | Outcome | Pass/Fail |
| Player moves right on the horizontal axis. | Player moves when ‘D’ and right arrow are pressed. | Player moves right on key press. |  | Pass |
| Player moves left on the horizontal axis. | Player moves when ‘A’ and left arrow are pressed. | Player moves left on key press. |  | Pass |
| Right running animation plays when moved on Horizonal Axis | Player animation plays when ‘D’ and right arrow are pressed. | Right running animation plays when key pressed down. |  | Pass |
| Left running animation plays when moved on Horizonal Axis | Player animation plays when ‘A’ and left arrow are pressed. | Left running animation plays when key pressed down. |  |  |
| Player jumps correctly on the Y axis. | Player triggers the JUMP state and moves on the Y axis when the space bar is pressed. | Player moves on the Y axis appropriately and triggers JUMP state. |  | Pass |
| Jump animation plays when Player jumps. | Player jump animation plays when space bar is pressed. | Player changes to jump animation when space key pressed. |  | Pass |
| Player dies when it touches spikes. | Player triggers GAMEOVER state and ends game when colliding with spikes. | Player triggers GAMEOVER state and ends game when colliding with spikes. |  |  |
| Game over animation plays when Player dies. | Player game over animation plays when GAMOVER state triggered. | Player changes to game over animation when GAMEOVER state triggered. |  | Pass |
| Player dies when it touches rocks. | Player triggers GAMEOVER state and ends game when colliding with rocks. | Player triggers GAMEOVER state and ends game when colliding with rocks. |  | Fail |
| Player wins the level when touching the flag. | Player triggers WIN state and ends level when colliding with flag. | Player triggers WIN state and ends game when colliding with the flag. |  | Fail |
| Player goes to next level when touching the flag. | New scene opens and Player starts new level if not currently at the last level. |  |  | Fail |

**Discussion and Reflection – 1 page**

*This section should answer the following questions: What are the primary strengths of your project? What are its weakness? What have you learned during this project? What would you do differently next time? If during self reflection you have identified an issue, e.g. time management, what actual steps could you take to address this issue?*

References:

* [*https://rvros.itch.io/animated-pixel-hero?download*](https://rvros.itch.io/animated-pixel-hero?download)
* *https://trixelized.itch.io/starstring-fields*