Developer Diary

By Conor Raftery (G00274094)

Table of Contents

Phase 1 – Design

Week 1

This week I began to learn how the unity engine works. I done this by looking at online tutorials and using the knowledge I gained during lab time. I also examined similar games that I have been tasked with creating. This gave me insight into how the game should be designed.

Week 2

I discussed in depth with the customer what exactly the customer wants. This included the main features he wanted to be implemented, along with the layout of the game. We designed the artwork for the game together.

Phase 2 – Implementation

Week 3 – Game Layout

I began the implementation by adding the layout of the game. This included the boundary walls, background, and the player. I also incorporated a basic platform for visualization purposes. The player sprite I created is just a 2D Square.

Week 4 – Player Movement

Player movement was the focus this week. I first started by making the player be able to move left and right. An issue I discovered was that the player was not kept within the boundary walls. I fixed this issue by 'clamping' the player along the x axis, and not allowing the player to stray further than the boundary walls.

Another issue I found was that the player was moving too fast and looked 'unnatural'. This was resolved by creating a variable that is accessible within Unity, in which I could test certain values and find one that suited.

I also need the player to jump. I done this by using 'Input.GetButtonDown' within my PlayerBehaviour script. This created an issue. The issue was that, although the player could jump, he could jump constantly (without being grounded). Meaning the player doesn't need the platforms to traverse up the level. To resolve this, I added a tag on the platform and the ground labeled 'Ground'. I then checked if the player was on the 'Ground' within my script.

I also used the above method for the wall jump, which is a functionality within the games design spec.

Week 5 & 6 – Unlimited Scrolling

To do the unlimited scrolling, I initially thought about having the background scrolling, along with the platforms and the boundary walls. But this was very tricky to accomplish, so I decided to go with another method.

This method was adapted from an in-lab game we were working on, in which there is point spawners to spawn in game objects. How I used this was that I created point spawners for the boundary walls,

the platforms and the background. These spawners are a constant distance away from the player, and out of the view of the main camera. These spawners create the game objects needed to give the effect of a constant scrolling game. The background and boundary walls were relatively easy to spawn, as they are always created at a relative position, but the platforms were quite tricky.

The reason for this is that the platforms had to be created at random positions, in which the player can reach with a single jump and the platforms don't exceed the boundary walls along the x axis. This was done by finding out the distance between each boundary wall and generating the platform within this distance.

I initially had the camera following the player, but this is no use as the player has to have a challenge. I overcame this by starting the camera at a very slow speed, and this gets incremented as the player plays the game, increasing in difficulty. This is on par with the customer's request.

Week 7 – Menus

Now that I have the core mechanics of the game done, I need to add a start menu, pause menu and death menu for the game.

I started by creating the Start Menu. To do this, I created a new scene. This scene is the first scene that opens when the user plays the game. The customer has told me he wanted to have a 'how to play' option within this menu, so I created this by having a separate canvas. I also took the initiative to add a slider within the options menu which controls the sound. Within the Start Menu, there is an option to quit the game which will quit the application if it is deployed to Android but displays "quit" in the console for demonstration purposes. When the "Play" option is selected, it will launch into the game.

The Pause Menu was surprisingly easy to implement. How this was done was by creating a canvas, which is deactivated by default, and only activated when the user presses the esc button on their keyboard. In return, it is toggled off when the user presses the esc button again or clicks the resume option. To get it to pause the game was difficult. I had to pause the players sprite, along with stopping the camera from scrolling. See the PauseMenu script. Within this menu, there is an option to return to the main menu. This was done by using "SceneManager".

The Death menu was the hardest to implement. The problem that arose was trying to determine when the player actually dies. My solution was to figure out when the player left the bottom of the main camera. I achieved this by getting the position of the top of the players sprite and getting the position of the bottom of the main camera and then checking if the player is less than the main camera. It took quite a lot of trial and error. When the player dies, the death menu is activated (similar to Pause Menu) and the player has the option to return to the main menu. The problem I could not resolve was stopping the camera from scrolling, this is an issue as the score can still be increased even when the player dies. I will fix this later.

Week 8 – Score System

For the scores, I initially decided to increment a point every time the player reaches a platform. This was going to be done by placing a 'token' on the platform that was invisible to the player. That was very difficult as I could not determine where the next platform was going to be generated as the position is generated at random. Due to this, I came up with the solution of adding a point every time a new platform is generated. To display the scores, I just created a canvas which is constantly in view and I placed the current score in the top left corner via a 'Text' GameObject which is referenced within my ScoreManager script. The highscore is displayed the same way. The highscore is saved in PlayerPrefs and is checked against the current score to see if a new highscore is achieved.

Phase 3 – Demo Prep

Week 9 – Fixing Known Issues

This week I had to prepare the game for an in-class demonstration. I had to do extensive testing of the basic functionality for this. One issue I found was that the players movement was still very 'jumpy'. This was fixed my restructuring my PlayerBehaviour script.

Another issue I found was that the player was getting 'stuck' against the side of the platform. This was due to the players sprite having friction. I resolved this by creating a player material which had 0 friction.

Phase 4 – Finalization

Week 10 - Animation

As we were not allowed to import pre-made animations and sprites from the asset store, I decided to create my own using Paint.net. I discussed this with my customer and we decided upon making a simple but elegant sprite for the player by creating a stick figure. I first created the stick figure, then copied and pasted it multiple times with slight changes to posture. I then imported this as a sprite sheet into Unity.

I split up the sprite sheet into 3 animations; idle, moving and jumping. I then assigned these animations to the players sprite. I got a handle to these animations in my PlayerBehaviour script by using 'GetComponent<Animator>()'. I then assigned the animation to each movement accordingly using Unity's Animator.

Week 11 – Sounds

This week I added sounds. The first sound I added was the background music. This was done simply by creating a game object in the scene and attaching the music to that. The tricky bit was to control the sound across multiple scenes (Main Menu & the Game). I done this by getting the value from the slider in the options menu and assigning it to a variable in 'PlayerPrefs'. Then I could easily just read in that value in other scenes and set the volume of the music.

I achieved the player movement sounds my attaching a script to the player which plays a sound depending on what action the player does. E.g. jumping, dying. The script gets a handle on the sounds in game via 'AudioSource'.

Phase 5 – Deploy and Submission

Week 12 – Testing

This week was spent cleaning up any known bugs. Along with combing through the scripts removing any unnecessary code blocks while also adding additional comments. I attempted to deploy this game to the play store but the apk version of the game would not work due to lack of touch controls.

My final testing strategy included:

- Unit Testing: Testing a lot of different aspects of the game separately to ensure everything was working correctly.
- Integration Testing: Worked through the entire game lifecycle, cycling through each menu, starting the game, pausing the game, going back to the main menu, starting the game again, when the player dies, and finally returning to the main menu. I done this type of testing to ensure that the entire game ran smoothly.
- Beta Testing: I allowed multiple people to play the to make sure they test the game vigorously and to find any underlying bugs.
- Acceptance Testing: I sat down with the customer to make sure he was happy with the final product, and to see if he would like to implement any more changes.

Features

- Scoring System
- Animations
- Sounds
- Vertical Unlimited scroller
- Randomly generated platforms
- Wall bounce
- In-Game menus
- Main menu
- Volume controller
- Incremented scrolling speed for difficulty

Conclusion / Areas for improvement

I feel I have met the customers requirements and some cases, exceeded them. An example of this would be the Menu system. The user experience is an important factor in this game. It is addictive and challenging at the same time. I was considering adding enemies and other obstacles, but the customer decided against this. The functionality of the game has been implemented fully to the customers specifications.

Some areas for improvement contain:

- The wall bounce feature is buggy, sometimes it works, sometimes it doesn't.
- I feel that there should be some other obstacles in the game, such as enemies.
- I could include pickup perks for the player to obtain.
- The camera does not stop scrolling when the player dies.
- There is a bug with the 'death sound' in which sometimes it plays and sometimes it doesn't.

Overall, I am happy with the game, and more importantly so is the customer.