Arron Healy – G00346883

Mobile Applications Development 3

Developer Diary

Introduction:

This Developer Diary is to be submitted along with my GitHub repository as a requirement for 4th year module Mobile applications development 3. We are required to build a 2D game for one of our classmates using Unity and Visual Studio. I received Kevin Niland's design for a game which he named The Pixel Wizard. This game will be a 2D side-scrolling platform game, where the user can control the character to attack enemies and move through the levels.

Entry 1: 01/10/19

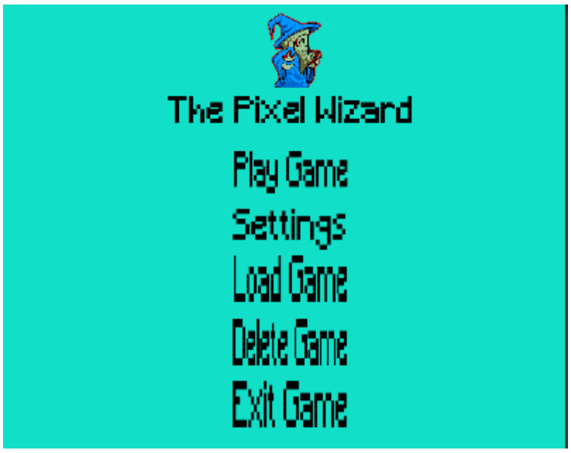
I have received the sprites from Kevin. I have read over the design document and decided to deal with designing the game menus and loading scenes/levels first. I feel this may be a good place to start as it will get the initial gameplay going.

I will be able to implement a game loop from the start menu to level 1 and then to game over which will then allow me to focus on entirely on level design. I have created my GitHub repository online and have it linked to my initial project in Unity which for now has only a basic start menu set up.

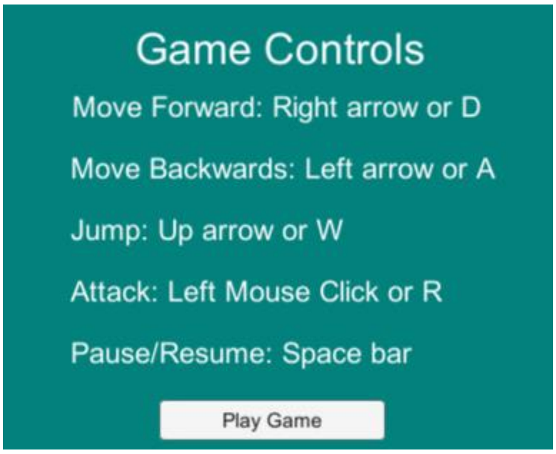
Entry 2: 05/10/19

I have worked a little on the project. I have added a scene loader script to manage loading scenes. I have added 4 scenes to the project so far, with them being a start menu, instructions scene, settings scene and the initial layout for level 1. Attached are images of progress so far.

Start menu:



Instructions:


Game Controls

Move Forward: Right arrow or D

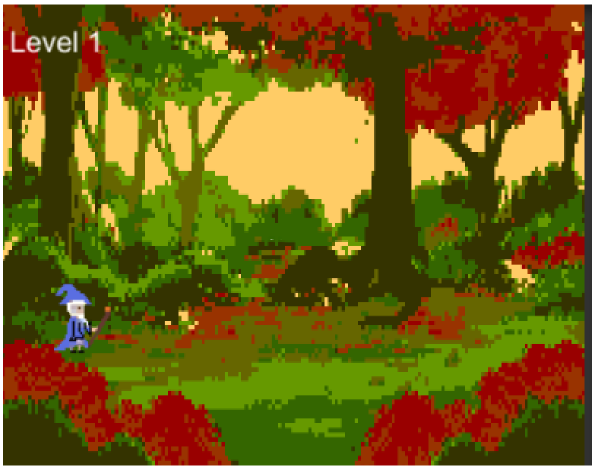
Move Backwards: Left arrow or A

Jump: Up arrow or W

Attack: Left Mouse Click or R

Pause/Resume: Space bar

Level 1:



Entry 3: 15/10/19

I have load level flow control from start to game over menus. The next steps I have implemented is basic move and jump methods to allow the player control over the sprite by using the get axis method in conjunction with the time.delta method. The player is now able to move left and right and jump to a certain height. The player is still only an image sprite as I am tasked with making a walk animation from the sprites provided. To stop the player from being able to move off screen in left or right positions I have added a method that stops the player from being able to move off screen using Mathf.Clamp to clamp everything to the camera view.

Entry 4: 15/10/19

I have added player shooting functionality to the game. The fireball sprite instantiates beside the player sprite and fires along the x axis. The shooting is implemented using a Coroutine as this will allow the player to fire without having to keep pressing the fire button. I have also added two game object shredders to destroy the fired projectiles once the have left the game view for both sides of the screen. This is implemented using a trigger detection function.

In implementing the coroutine I have found a bug. The customer wishes for the game to use either the R button or mouse click to fire projectiles. However, if it is implemented and you click both buttons at the same time the coroutine enters an endless loop of firing as it is not able to call it’s stop method once this error occurs. For now, I will only have the player be able to fire from the keyboard and come back to this bug later.

Entry 5: 23/10/19

I've added two Game Object shredders to the level both to the left and right off the screen. This is to manage the fired projectiles once they have left the view of the game to destroy them and help manage memory.

I've started to work on the enemy sprites also, I've decided I want to have some enemies move on a path towards the player. I've been able to implement it using a list of waypoints to move an enemy on a path. Following some tutorials online I've added a wave config scriptable object file to manage spawning the enemy and moving it along a path.

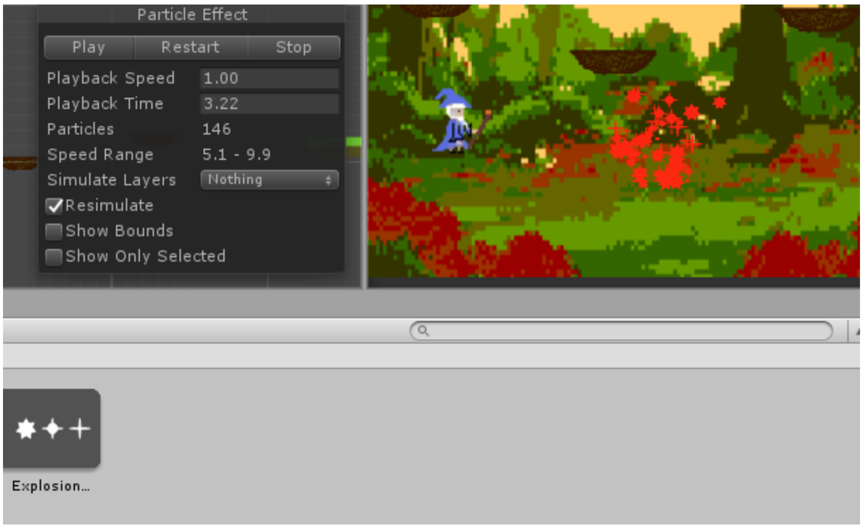
Entry 6: 24/10/19

I have enemies spawning and some moving on a path through a scriptable object file. I have now added a damage dealer class to handle taking damage whether that be for the enemy or player. I'm now able to deal damage and destroy game objects once health has fallen below zero and have enemy firing projectiles.

A small bug to note is that when enemy sprites were firing projectiles, they were inflicting damage on each other. I was able to solve this issue through setting the layer collision matrix in unity project settings to ignore certain sprites coming into contact with each other.

Entry 7: 05/11/19

After implementing the game mechanics to inflict damage and have the enemies or player die, I felt that it was good but a bit bland as the prefabs just disappear without any affect. To get around this I found lots of tutorials on how to implement a particle system. With just a few different shapes it can give the effect of an explosion taking place.



Playback Speed 1.00
Playback Time 3.22
Particles 146
Speed Range 5.1 - 9.9
Simulate Layers Nothing
☑ Resimulate
□ Show Bounds
□ Show Only Selected

I've also added the scripts to handle the players health and score. The player score is not required in game spec but may incorporate it or remove it in future commits. As a final touch to get the feeling of the level being complete, I have added a function to spawn the boss in the level once all enemy sprites have been spawned.

Particle System Tutorials:

https://www.youtube.com/watch?v=gjlh4YrMaHY

https://www.youtube.com/watch?v=d7McyujCHc4

Entry 8: 06/11/19

A quick entry today, I just have the player health and score stats displaying on game screen. Also added a game over scene to load once player score has dropped below zero and the players game object is destroyed.

I've also added a pause menu to the levels so that once space bar is hit the game pauses. I was able to do this through a panel being set to active or not active. This was done from the following tutorial:

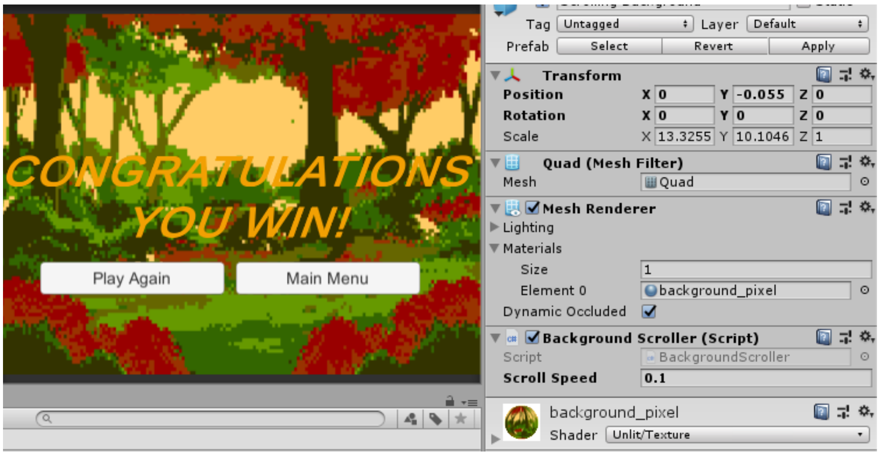
https://www.youtube.com/watch?v=JivuXdrIHK0

Entry 9: 13/11/19

As stated in Entry 6, I have the boss spawning once the all lower enemies have spawned. As of now they are all spawning within the camera view but will change as to expand the levels. I have duplicated level one into three levels as per the designer spec and will expand them to increase difficulty and give a sense of level design also.

The sprites I received from Kevin included only one background which I have attached to the canvas which is fine for the moment but may not work as I try to expand the levels. To overcome this, I've decided to use the background provided and implement scrolling functionality using a Quad UI type provided by unity. A quad allows you to attach a image to it and it acts a flat plane that can cycle over the image repeatedly.

To test it out how it might work in gameplay I've implemented it in the game over scene and it works perfectly. I now have the functionality to add it to the game play later.



Tag Untagged Layer Default
Prefab Select Revert Apply

Transform
Position X 0 Y -0.055 Z 0
Rotation X 0 Y 0 Z 0
Scale X 13.3255 Y 10.1046 Z 1

Quad (Mesh Filter)
Mesh Quad

☑ Mesh Renderer
Lighting
Materials
Size 1
Element 0 background_pixel
Dynamic Occluded ☑

☑ Background Scroller (Script)
Script BackgroundScroller
Scroll Speed 0.1

background_pixel
Shader Unlit/Texture

Quad tutorial:

https://www.youtube.com/watch?v=HrDxnMI7pCc

Entry 9: 30/11/19

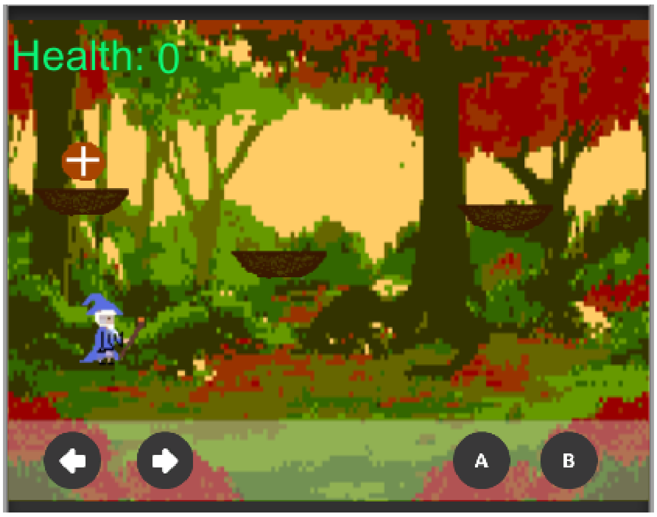
With this entry I have majority of game play functionality implemented. The goal now was to expand the levels and have add more depth to levels. The first issue I had to overcome was having the camera follow the player which was simple enough as I only had to fix the camera to follow the player sprites transform position.

From the previous entry I added the scrolling background to the win and lose screens. I only had to make the quad a child of the camera and will a small change in the code I was able to have the background scroll as the player moved and this gave the illusion of ever-changing scene.

With the deadline approaching and the module being to develop a game for a mobile application I have gotten some button images from the web and added a panel to the scene for mobile UI control buttons. I tried to follow the adapter pattern and add a class that would call the player keyboard controls via the on pointer down functionality. It must be some setting in unity editor that is messing it up.

However, this is not working for me as of now even though the code is logically sound. I've researched a lot online trying to find why the panel buttons won't work but have not been able to solve it.

The script and panel are still in project, but I will shut them off and hopefully have them sorted by submission date otherwise cannot upload game to play store.



Entry 10: 11/12/19

As a final entry to diary before submission, I have added a music player to the game to add to the sense of play and it helps with drowning out the noise of all the shooting going on in the game. I've added some new enemies to the level to increase difficulty and tidied up the level layout a bit. I was not able to solve the issue with the mobile control buttons but I intend to add a screencast to my GitHub as a walkthrough of the game in the unity developer addressing this issue and showing a little gameplay to anyone who may visit my repository.

Music Refence:

https://opengameart.org/content/heroic-demise-updated-version