Bryce Standley

33046367

GAD210 Game Report

# How did I go about creating my game? In what order did you need to make things? How did you go about finding out how to write the different code systems that are needed?

With my previous experience working inside Unity, I find it extremely beneficial to build the project file structure as the first thing and move directly into getting some sort of player input into the game. I built a test level that I could load up and test ideas and mechanics without making whole levels.

Once I got some player input working on a cube, I started to change some of the art to some from the asset store to get a good idea on how the character will look and models of the enemy’s in the game. With some previous experience in Unity and with C#, a big portion of my code was written with my own knowledge. Although a lot of systems in the game were built by myself like the menu system, save games and enemy AI, there were a handful of systems I either learned by following tutorials on YouTube and forums or using the Unity Standard assets.

# What the main scripting systems are in your game – where did you need to spend your time?

The largest system in the game is probably the UI system that I built to work seamlessly across all the levels. I spent a large amount of time, programming and debugging the UI of the game to not only get the look I was after but to get the functionality working correctly. Although I spent a lot of time working on the UI, one thing I really enjoyed was trying out new things for the UI like changing the screen mode in the options menu. I did follow a few tutorials on getting custom key bindings working in the game to allow the player to re-bind their keys without using the Unity menu when the game first loads but I decided to scrap that idea as it was going to be to much work to get it working properly with the assignment deadline closing in.

Apart from the UI system, everything else in the game wasn’t to script heavy apart from handling collisions for triggers etc. I did my best to make manager scripts that would handle a lot of the leg work like updating player UI or keeping track of the score, that way when a object like a star collectable is collected by the player, its easy to grab a reference from the manager and to update the UI and score tracking though the score manager with only a single line of code due to all the extra code is sitting in a different script.

# What were the challenges in making the game? What kind of bugs frustrated you? What did you learn?

As having some previous experience inside of Unity, I wanted to challenge my self and do a sort of game I haven’t messed with before, that’s why I decided to go with a Captain Toad style 3D platformer without the ability to jump. I’ve never really done much with 3rd person camera and character controls so I was a great learning experience to get these parts of the game working the way I want.

The first major bug I found in the game was when resetting the level, the issue arises when either reloading or moving the players character back to a starting point of the level. Although I’ve worked with rigidbodys before, I’ve never seen a issue where the player would continue to fall out of the level like the way my character was. With lots of asking around and trial and error I was able to squish the bug and got the player to not fall though the floor by changing my player controller from a custom controller I learned though a YouTube tutorial, to the Unity standard assets 3rd person controller. After changing the controller, some tweaking to my level end script was needed to get the player to be moved back to the spawn point of the level. I wish I was able to get the whole level to reset but with time constraints, moving the player back to the spawn point was the best solution.

Throughout making the remainder of the game, most bugs were small and didn’t cause much hassle to fix like issues with the main menu UI animations acting strange and odd errors with enemy AI and models.

The most significant thing I leaned is to keep most variables private and create methods to be able to access and modify these variables in a safe manner. The coding standard was a great help in keeping my scripts a bit easier to debug as you can easily identify the public, private and local variables in the script. I also leaned a lot about how movement works with animations and rigidbodys as well as how a lot of the built-in functions are used to make life easier so I wouldn’t have to rewrite code and reinvent the wheel.