Programming Journal

08/02/2022:

For my first package, I wanted to create an instant form of damage/projectile which I discovered was with raycasting. So today I created a package for shooting a ray cast which dealt a knockback force to the target object and eventually destroyed it with an integrated health system. This worked rather well, and by that, I mean intended and it worked in general, I am not sure what my other packages will be or if it will clash with this one but if so I would like to be using raycasts in some way.

15/02/2022:

Today I took the time to create a fire visual effect with a VFX graph and shader graph to create a really cool and procedural fire effect. I know this isn't coding/programming related however I am still getting used to Unity and its library of a registry and I wanted to try making something from this. The outcome of the fireball is going to be included in my RCR mech demo and will be built upon and doing this has actually given me an interest in visual effects within Unity. I was already really excited about the particle system but this is icing on the cake.

22/02/2022:

For this package, I wanted to use raycasting in some form of interaction system. Following the tutorial I found, it would cycle through different colours on a cube that had the script component attached. The issue I had with this script is that it was camera-based and would not work effectively and as intended with the third-person, however, it worked but only when the player was directly next to the cube. I decided to keep this to fix later down the line.

08/03/2022:

For my RCR mechanical demo, I needed a movement script of some kind so I decided to follow a tutorial on how to create one. It is a really simple to attach package and also allows customizable camera positioning with the target object. This makes it a really versatile third-person character controller for any future game I decide to create.

15/03/2022:

I decided I wanted to create a multi-purpose interaction system which uses multiple scripts. This interaction system uses raycasting to do something either when the player is hovering on a target object, stopped looking at a target object or interacting with the target object. Within my package, I have made the interaction destroy as this is a simple and easy way of depicting that the interaction has occurred and works as intended. I also have game.logs in the console for the hovering start, hovering end and interaction all happen so behind the scene I or anyone who uses the package can tell it works. However, this interaction system is the groundwork to be capable of interacting with in-game NPCs, collecting objects and anything the designer can think of they just need to rewrite the terms of the code for that. So it is rather useful and I hope some people can make use of it.

26/04/2022:

Today I wanted to get everything remaining completed. Creating my interaction system completely made my previously made interaction system redundant so I decided to remove that from my list of 4 packages, as well as the raycasting knockback system due to the new interaction system destroying the object. This required me to make an additional 2 packages, one of which I had already started on for my specialism module.

My radar was already near completion and felt about right to use this in the package as it was essentially a third-person shooter I was creating with these packages so being able to track the position of the enemies made sense. The other script was something I had been trying to do for a while but couldn't get to work which was camera shake. I had tried other methods in the previous week for my cinematic to add some dramatic effect but shaking through the cinemachine impulse method was not working. I found a tutorial which easily laid out what to do and sure enough, it worked really well.

All I had to do now was converge all these 4 packages together to be able to work together. The camera shake was going to be this effect when destroying an enemy object and it also wasn't that hard to reprogram into the main interaction script, just calling upon the coroutine for the shake when destroying an object.

I feel as if this works really well together and shows my programming knowledge has improved significantly. I still can't quite write multiple lines of code from memory but that is something I need to learn to do after enough exposure and practice. That being said, with my programming specialism class on top of all this really has assisted with troubleshooting issues I have with my scripts. Being able to figure out what is wrong with my code by myself has been a noticeable change this semester and it makes me happy being able to reflect on my progress.