Take a Stroll Around My Brain Why Don't You?

COMP160 - CPD Report

1601016

March 25, 2017

Introduction

My career goal hasn't changed a lot since the last report as I still would ideally like to work for an indie games company. My specialism is more open than before, since I have added AI and procedural generation to the possibilities alongside the existing interest in technical art. This semester I have noticed that I haven't been informing my team enough of what I'm doing and my time allocation has been skewed too much towards the group project. Other notable problems have been establishing an early pipeline, abstract prototyping, and the sole use of Blueprints in unreal.

1 Informing the team of my work in progress

At the start of the group project I found myself frequently in the scenario in which I was working on things without informing the team of what I was doing. Communication was lacking both verbally and on the Trello task board. In next years project I will add all the tasks I plan on doing to the task board backlog and put them in progress when I am working on them. I will also count up the number of tasks that I have done and make sure that number is in line with the number of tasks that are completed or in review on the board.

2 Allocating time proportionally between team and personal tasks

This semester I have noticed a disproportional amount of time going into the team project, with my personal tasks being sidelined and being done in a more condensed time period than the credit split would suggest. At the start of next year, I will get the credit percentage of each module and divide each project into the appropriate time ratio. I will then use this as a guideline each week to determine which days I spend working on which tasks. There may be some exceptions to this, especially when deadlines approach, but if I keep the work division consistent then I should have allocated enough time by the time deadlines approach to have completed the necessary work for submission.

3 Establishing a clear pipeline early

It took several weeks of the group project before any assets were put into engine, and the first models that were eventually added were already high poly. For the next group project I will request simple low poly test assets from each of the artists and animators, this will require everyone to get an organised pipeline so that the assets can be put into the game engine. Once this is done, these assets can be iterated upon for more polished ones without the need for programming intervention. This should be simpler in the next project since I have already had experience with putting assets into the unreal engine, although some of the team may be new to the engine which I will need to take into account.

4 Prototyping with minimal assets

For the AI, I was able to program a basic version with the starter content from the Unreal Engine FPS template, however I struggled to program the melee combat system without having the weapon animations in the game. This meant that the player wasn't able to attack in any form until about a 2 weeks before the trailer deadline. Now, with more experience in unreal, I can already better visualise how code is going to work without seeing the assets, although I can still further improve on this. Over the summer I am going to work on a personal Unreal game project which will consist of starter content. This will force me to code functionality without specific assets in place, thus hopefully heightening my ability to code more abstractly with limited resources. I can then see if I have successfully improved when it comes to the second year project when I will do the same thing.

5 Using C++ and reducing reliance on blueprints

Partially because of my experience with having several C++ builds fail on me in the studio, I have not used C++ since the "Mandlebrot Set" worksheet and have solely relied on blueprints when programming in unreal. This means that my code suffers from efficiency and maintainability problems. My personal summer game project will be a practice in coding with C++ in unreal. I plan on implementing the majority of the game's functionality in C++ which will improve my understanding of how the language coincides with the engine. Once again, I can use the second year project to see if I have managed to take in what I have applied, since I plan on implementing most of my team project next year in C++ as well.

Conclusion

Over summer I will embark on a personal Unreal game project in which I will use mostly C++ and try to code functionality with only starter content assets. The other three goals will wait until the second year when I have another go with working in a team. Before the team project properly starts, I will need to work out my time and how I'm going to split it between the modules. When the team project begins I will ensure all my tasks are in the backlog and get updated every time I start or finish one. I will also request simple assets from the artists and animators to establish a pipeline early on, in order to help with the general flow of the project.