

Adventures in Personal Development

COMP130 - CPD Report

1604281

May 6, 2018

1 Introduction

This year has been highly educational and interesting. Although I am still unsure of what career I may eventually pursue, working in a team has shown me how fun (and how stressful) group development can be. After collaborating with artists and designers, I certainly think that I would enjoy working in similar environments in the future. I would however like to take a more active role in the design and concepting of games, as I enjoy the creative side of games development as much as I do the technical, programmatic side.

2 Maintaining Enthusiasm - Affective

While I feel motivated and enthusiastic about projects, I am driven to work on them often and can focus well. However, long projects especially ones that I do not have a great deal of interest in can cause me to lose enthusiasm. The ability to maintain a level of investment and enthusiasm for a project is necessary in the games industry, where projects can take years to complete, but a high quality of work is still required.

The concept for our group game project didn't inspire me as much as I would have hoped, and the lack of concept art and storyboards left me with little to get excited about. This, in addition to having little knowledge of using Unreal leading to simple tasks taking very long reduced my overall enthusiasm for the project, which in turn hindered my motivation to put in extra work outside of our team sessions. I feel that having a group playthroughs of the game with the whole team would have helped to boost the motivation and positive spirit of the whole team, myself included. To this end, on our next project, I will push for weekly play-tests to try and showcase what we have accomplished so far.

3 Taking Non-Programming Responsibilities - Interpersonal

Although as a programmer I am expected to mainly focus on the coding side of games development, there were often times when other members of the team deferred to me on non-programming issues. Although I took on many other roles and responsibilities in our group project such as creating cinematics and voice acting I always tried to reduce other team members' dependency on my opinions, trying to encourage them to communicate with other artists or designers. This was largely because I felt that I did not have a good enough understanding of these areas to advise in a helpful manner, however I realise now that being a more active presence in non-programming areas (art direction, themes, gameplay) could have helped to bridge communication between our team.

In games development, delivery is the ultimate goal, so stepping outside of your area of knowledge and taking greater responsibility for other areas of the project may be necessary to ensure that release deadlines are met. In our next project, I will take a more active role in issues outside of programming, specifically if there is no-one acting as a creative director to ensure that a consistent style is maintained throughout the project. I will also try and be more open to making judgements on art, design and story

elements, as it is often more important that decisions are actually made, rather than trying to get someone more suitable to make the decisions.

4 Estimating Task Times - Dispositional

One of the most important skills in the games industry is the ability to accurately estimate how much time a task will take you finish. This allows the development team to make realistic sprint goals, release dates and development schedules as well as helping to give other team members a good idea of where the project is at. During my own projects this year, I have had difficulty in estimating how long tasks will take me which usually lead to me taking much longer than planned.

I feel this is largely because of a lack of experience, but also due to underestimating the complexity of certain tasks, something that is closely tied to my lack of architectural planning (which I will talk about later). This is a difficult skill to improve, as experience and practice seems to be the largest factor in good estimation. However, I will record how long significant programming tasks have taken me by using my project Trello boards and Github logs, so that I can use them as a baseline when making future estimates.

5 Use of Pointers vs References - Cognitive

One of the concepts that I have struggled with this year is when pointers should be used in place of references and why. Obviously, methods of passing data are critical in any programming context, but memory issues associated with pointers could be particularly damaging in games where memory usage and framerate is highly important.

Because of my shaky knowledge of how to use pointers, I have mainly defaulted to using references (when not passing by value) and only utilised pointers when certain functions or types require it. This appears to have worked fairly well for me, however I feel like I am guessing at when to use pointers without knowing why I'm using them. I

plan to go over the lecture slides regarding pointers, as well as reading the documentation on them, to try and improve my knowledge of how they are used. Aside from this, I will make a note of when I use pointers over references, and then try to research why they were necessary in this situation.

6 Planning Program Architecture - Procedural

While working on the group game project this year, there was very little in the way of planning for the programs structure. I didnt feel comfortable enough with the engine to suggest architectural designs, however this is a very important part of the iterative plan, do, test, review cycle. As I have already experienced, lack of planning in the programming side of game development can cause major delays and issues later, especially redundant work if multiple developers create code to do similar tasks. The ability to design and managed system architecture is also a useful skill for any software career, as good high-level design can often make the whole project run more smoothly, as there is less confusion over how different features should interact.

To overcome this, I will endeavour to create UML class diagrams for each of my coding projects, both in the next university year, and for any of the projects I undertake during the summer. Specifically, I will focus on planning the interactions between these classes, as knowing which areas of the program will need to communicate with each other is incredibly useful when working with multiple developers.

7 Conclusion

Overall, many of the skills I need to work on simply require time and practice. Despite that, actively working on them will help me to improve faster and hopefully benefit me in the next year of university. Working in a team as introduced me to many new challenges and made me examine the new skills that need to be applied when working with others,

especially with multiple developers.

Improving my general programming skills and knowledge will also help me in all these areas, as confidence with the tasks I am doing allows me to focus more on other areas planning, time management, elegant code, etc rather than simply worrying about how to implement a single feature, class or function.

References