Reflective Essay – Final Project

By Zackary Direen

Word Count: 1,537

[Year]

**Table of Contents**

**Introduction 2**

**Technical Struggles 2**

**Variance from Schedule? 2**

**Variance on Functionality? 3**

**How well does EasyQuest work? 3**

**Were there any changes to resource usage? 4**

**Any additional features? 4**

**What went right? 5**

**Lessons learned 5**

**Conclusion 6**

**Introduction**EasyQuest was developed in 2019 as part of my final project, in which I had 10 weeks in order to develop it, though this project has been planned since the start of the year, where all the project details where outlined and identified for the purposes of making sure that the scope of the project was very clear, thus making it easier to develop my project. During the course of the project development however, I was faced with some successes and challenges along the development of EasyQuest. In this reflective essay, I shall be talking about all the issues I encountered with this project, with lessons of what I will be able to take with me within future projects, and I will also be discussing what went right during the project.

**Technical Issues**

Most of the technical issues that I encountered were version control related, as the initial commit and push/pull to my work computers was incredibly long, halting my ability to be able to work on my project, especially when I had to work on new computers. I also had to reset my GitHub repository around 3 times due to project corruptions that occurred when I was doing file management.

I also had to encounter the issue of my personal computer and the computers at AIE Watson containing different versions of Unreal Engine. (The project was started on 4.23.0, whereas the other computers had version 4.2.2) This meant that I was not able to run my project on the AIE computers and ended up losing a day’s worth of work due to not being able to work on the project while at AIE. I had to solve this with me bringing my own copy of Unreal on a hard drive for future visits to AIE, and after this was done, this issue was resolved.

**Variance from Schedule?**

Initially, the project was very successful in following the schedule that was detailed in the plans for the project, I however, faced some personal issues that affected my mental ability to work, therefore, I ceased work on product for about 4 weeks due to being mentally and emotionally unwell during a rough period of time that I had never experience before. This caused deadline dates to be changes around and an overall faster pace of production from the project. I however, always ensured at the end of the week that internal bug testing was being performed and had all major components finished by week 8 of development, with non-game breaking bugs being sorted and fixed during week 9 of production.

**Variance on Functionality?**

As far as functionality was concerned for the quest system, the project met all the requirements and specifications in terms of functionality. I was able to not only initialise all the 4 different quest types, But I was able to simplify the quest states while still being able to preform quest chains and pre-requisite quests/level in order to access a certain quest.

I however, was not aware that in order to implement some quest types, I need to incorporate at least basic other systems to be able to fully utilise my quest system (Inventory System – Collecting items for ‘collect’ quest requirements, Levelling System – Quest EXP rewards, Health System – Killing NPC’s for ‘kill’ requirements.) Therefore, this needed to be implemented in some capacity in order to show off EasyQuest.

**How well does EasyQuest work?**

In its final build, EasyQuest works extremely well in what it was designed to do. Quests are able to be easily created from making a child class of the quest blueprint and outlining its objectives and quest name and dialog, then you are able to assign it to an NPC as a quest holder that the player will interact with in order to pick up the quest. It tracks objectives based upon when an action being done a call objective command will be fired, this allows the players quest log to track dynamically when a quest has been completed. It also correctly gives the player an amount of experience that is set on the quest. It is also able to set a requirement quest that must be completed in order to be able to accept a quest, as well as set a prerequisite level that the player must reach before being able to accept the quest. I was also finding that it was difficult for the player to know that an NPC had a quest for them. To combat this, I added quest objectives that are bound to the location of the quest. The developer can move its location based on their NPC model’s size and will show –

* “!” – Has a quest you are able to actively pick up.
* “?” – Has a quest that you are able to turn in and complete.

**Were there any changes to resource usage?**

As far as resource usage was concerned, due to the issues that I faced as outline both in my technical issues and my variance from schedule, This meant that I had had less time in order to do the same amount of work I was scheduled to do and required a lot of extra hours being put in during the period I was developing in order to ensure that my new deadlines were being met, as a failure to put in more hours towards development would ultimately have led towards the project failing.

The only other changes to resources was in the computers I was working on, as the computer that I had at AIE had lower tier hardware compared to my PC at home. However, this caused no issues in development, as the lower tiered hardware was able to build and run the EasyQuest demonstration example smoothly.

I had a conflict in Unreal Engine versions as stated within the “Technical Issues” section, I however, ensured that my project was always developed using the Unreal Engine version 4.23.0.

**Any additional features?**

As highlighted within “**How well does EasyQuest work?**” The main additional feature that I worked on was my quest markers, as I felt that this was a requirement that I missed during my initial planning process of EasyQuest, as it just makes it some much easier for players of the game to be able to track quests, as well as showing where to pick them up and where to complete them. Though, I was also able to get quest chains working as well, which was an additional feature that was mentioned as something I would work on should I get additional time to be able to add them. I was able to squeeze these additional features both in time and they both work flawlessly and as required.

**What went right?**

The learning process for how to use blueprints within Unreal was incredibly easy for me to understand and pick up, which made the start of initial development very easy.

I was also able to actively seek out advice and help in developing my system though the use of online sources in order for me to better understand how certain systems should work within EasyQuest.

I was also able to complete the project successfully, even with all the issues, both technical and personal that I faced, while also being able to add additional features that helped improve EasyQuest as a tool for developers to use and implement within their own game.

The quests within easy quest are very easy to add, customise and put within a level, and will just automatically work, should the Player have a QuestLog, Inventory and Levelling component.

Objectives are incredibly easy to assign, as they are based upon an actor’s class.

**Lessons learned**

Some of the lessons I have been able to take away from this project are

* Setting development goals and objectives within your project help keep stress at a minimum and leads to a steady development time.
* Cuts in production time have a dramatic impact on the amount of work you are able to produce. So, the more production time you have, the more you are able to not only produce, but also be able to debug and test your project.
* Any uncertainty’s that you may have should be discussed with someone who is able to help you, rather then delving on the problem, as production time can often suffer because of this.
* Should anything personal that affects your ability to work on a project should be immediately bought to the attention of your project manager, as they will be able to give you advice on how you should go about your project.
* Don’t give up.

**Conclusion**

In conclusion, this essay shows some of the issues that I faced during the development of this project, with some of them being serious and how I was able to persevere in order to not only complete EasyQuest with all its functionality, but also was able to add additional functionality to EasyQuest in order to improve it while also still meeting the requirements of the initial scope. It also highlighted some the lessons that I learnt that I wouldn’t have otherwise learnt had the issues I faced not occurred or that I didn’t undertake the project of EasyQuest. It has also been able to expose me to a new game engine that I now feel confident in and am able to use my newly learned skills towards other projects.