Advanced Application Development

Michael Little – 1103677

**King of Tokyo**

**Reflective Essay**

1. **Introduction**

The objective of this coursework was to create a digital representation of a board game and document its development in a professional manner similar to how it’s practiced in professional software development. During the course of its development a lot has been learnt about the process of developing software and how it can help improve the overall quality of the final product.

Initially the documentation was thought of as a minor part of the development process, more for recollection and review rather than to help the project progress. However after having to plan out the project on paper it made me have to think about some of the problems I would encounter before I even started coding.

1. **Reflective**

In Software Engineering the Software Requirement Specifications (SRS) and Design Documentation are used to help allow the clients and management make more detailed decisions about the development and viability of the program before any development begins, this is very useful for making sure companies don’t waste any unnecessary time on projects that way prove to be unfeasible or poorly managed.

Creating the SRS document allowed for the initial rules of the game to be put to paper, this helped provide a basic framework that the program as a whole would be based on. As programmers we are naturally inclined to dive straight into the programing, so taking a step back and planning everything out provided a refreshing alternative that I fell helped the development in the long run.

The Design Documentation was the hardest of all of the documentation, partially due to its in depth analysis of the different classes and how their functions worked, but also due to how we had to state how we were going to code the application. As a programmer this was the first time I had to state what coding standards I would use and my naming conventions for functions and variables, I found this rather tedious as I was stating what was already known to me, but that’s not who this document was for, so in reflection I can see how it’s important to state these things in a professional environment. The main benefit from the design document is that it forces you to take a more in depth look at your development process, this I felt helped lead to an improved final product as I had to plan out the various functions that would essentially make the game.

In reflection I feel that the Design Document was responsible for the gameplay class which acts as a mediator between the Player/Dice classes and the main class. This was because when writing the document we had to think up the various functions that resembled the real world mechanics of the actual game. Normally when programming a game, a problem is not tackled until it comes up, and in this case a huge problem would have been how to process different results of the dice roll, but thanks to the documentation I was able to carefully take my time and plan ahead accordingly, this allowed the program to come together very smoothly near the end.

The testing stage was also something that at first felt tedious and unnecessary, usually in pass projects the idea was not to see how well something worked, but rather to just have it work. One example of this would be the DiceRoll() function, in previous projects just having 6 different random numbers generated at the click of a button would be enough, but thanks to the Testing specifications, time was taken to ensure that the numbers were completely random and that the odds of getting one number were the same as the rest, which not only improved the quality of the application but its resemblance to the physical game.

As for the actual application itself, in reflection there would be a few things that I would have done differently. The first would have been to start the documentation earlier, I feel this was left a little too late as development of the application started shortly after due to an approaching deadline and other course responsibilities. As a result I felt that this made development of the project feel rushed and not as thought out as it should have been. Another problem was that the capabilities of the development library SFML (Simple and Fast Media Library) were still being discovered, this and the inclusion of a gameplay class was left too late, because of this allot of the games functions and logic are handled within the main class. In coding circles this is considered bad practice, and in reflection I do wish that the gameplay class was implemented earlier as this would have led to a much less cluttered feeling final product. And finally it was only near the end of development was it found that the application doesn’t scale appropriately to all computers, If I was aware of this at the beginning I would have used SFML’s sf::VideoMode::GetDesktopMode() to scale all of the entities by the size of the desktop. Unfortunately there wasn’t enough time remaining to do this so the project windows were rescaled and positioned so that the fitted on the university’s computers better.

1. **Conclusion**

In conclusion, the documenting the development of this project has helped benefit the development of the application and the project as a whole. Although at the beginning I was sceptical that it would provide any real benefit to the project or be anything more that tedious, it has proven to not only help develop a better product, but help me be a better programmer. My only regret is that development should have been started earlier, I feel that is was left as late as it was due to a lack of urgency on the deadline, perhaps if the SRS had an earlier deadline that the rest of the project it may have helped development of the project as a whole. That said thanks to the lessons learned during this project I feel that I am now more experienced to my requirements as a professional programmer and the development process as a whole.