SIT102

Introduction to Programming

Learning Summary Report

Ben Marriner
220253518

Self-Assessment Details

The following checklists provide an overview of my self-assessment for this unit.

	Pass (D)	Credit (C)	Distinction (B)	High Distinction (A)
Self-Assessment				✓

Self-Assessment Statement

Declaration

I declare that this portfolio is my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person.

Signature:

Portfolio Overview

Initially, I came into this unit having programmed in Swift. I had also used Unreal Engine 4's visual scripting system called Blueprints which is essentially C++ but written using node graphs. I have learned C++ for 5 years through this visual scripting system without really realizing it until I noticed how closely linked the concepts that I learned through Introduction to Programming are to what I had learned when using Blueprints. This unit has certainly taught me many new things, notably pass-by-references, vectors, pointers and overall how to code properly in C++. With all the knowledge I have gained from this unit, it has made me reflect on my own projects that I have done in the past in terms of what I could have done differently. Going forward, I can now say that I have a firm grasp of the basics of C++ as well as programming in general. I intend to use these skills to take on projects in the I.T industry that involve writing code in C++ as well as exploring the endless possibilities of programming.

Throughout Introduction to Programming, I learned how to read and write code in C++ as well as the underlying principles behind it. It was tasks such Arrays and Structs and the game tasks that really encouraged me to read the code and come up with ways to write it as well. After learning all the concepts and completing all the tasks in this unit, I decided to go and create my own program for the custom program and something awesome tasks. I wanted to find a way to use C++ and conjunction Splashkit to replicate the famous gun barrel sequence from the James Bond films. This required me to use all the principles and coding concepts that I had learned in order to build a system that one can use to make animations with the Splashkit library and C++.

Reflections

I do think that learning how to read somebody else's code is very important, especially in a situation where coding conventions and proper coding practices were not used. It is also very important to be able to read your own code and debug it when something is not right. Knowing the proper terminology is also useful when reading error messages in an IDE as you will be able to understand what it is trying to tell you that is wrong with your code. Finally, being able to write your own code, particularly from scratch is also important when creating a custom program as building a program in itself is an open-ended problem that requires a creative solution.

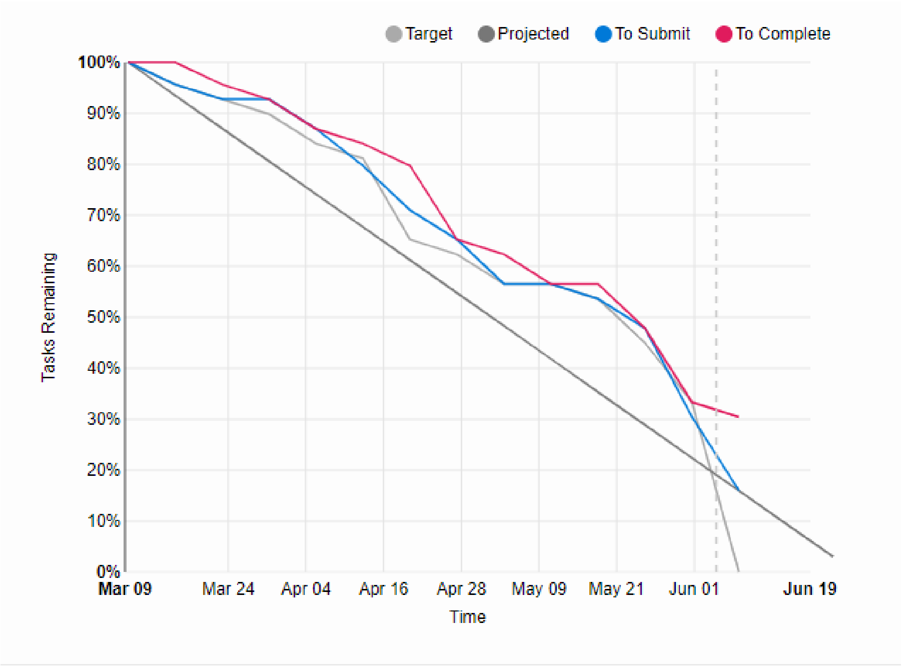
I am really confident in being able to solve computer problems using C++ since I am now able to express my solution in terms of code. I am now also better able to read other code not just in C++ but in other languages that I might not be as familiar with as well as get a better understanding as to how much more complex program work.

The game task (particularly part 2) was challenging as it required me to think outside the box in order to add more functionality to the game I was trying to make. The linked lists task was also challenging as I had never been exposed to linked lists as a concept in all the programming I had done up until I encountered that task. However, like with all the programming concepts, I managed to understand how it worked in the end and I was able to create functions and procedures that could manipulate these linked lists.

I personally found the entire unit interesting because I thought I already knew a lot about programming to begin with and yet the unit taught me so much more that I did not know about. However, I would say that one of the most interesting things that I learned in this unit is just how you can build a program from the ground up and watch it become more and more complex as you write more and more functions and procedures for it that take advantage of more small functions and procedures.

I shall continue to explore more and more areas and possibilities of programming. I will take what I have learned in this unit and apply it to much bigger projects in the future, where I will be working with teams of people to create software that does many things. I think that the more I program, the more ideas I can come up with in terms of what I could apply my skills to.

I found that the videos for each task were the most useful as they explained not only what the task was about but how the underlying concepts they were trying to teach worked.



I always tried to keep on top of the tasks by doing them week-by-week. For each task it was as simple as watching the videos for that task and then following the task sheet, doing what it told you to do and then preparing the task to be sent onto OnTrack. I never really had any issues with that, so I always found it easy to keep on top of.

Towards the end of the semester, it was difficult to keep up because of the other units. I think that what I could do differently next time is to structure my schedule properly so that this unit does not get affected by other units. If I did this, I could everything in this unit as well as the other units more effectively and understand the content better.

I believe that continuing to explore the many uses of C++ outside of its use with Splashkit will be very beneficial to my career in I.T as there are so many applications for it. Accomplishing as many things as possible with it as well as with other languages will help me grow as a programmer and as an I.T professional and therefore, make me more employable as well.