COS30031 Games Programming

Learning Summary Report

Thomas Horsley - 103071494

Self-Assessment Details

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Pass (P) | Credit (C) | Distinction (D) | High Distinction (Low HD) | (High HD) | |
| Self-Assessment (please tick) |  |  |  | X | X |

*Self-assessment Statement*

|  |  |
| --- | --- |
|  | Included? (tick) |
| Learning Summary Report | X |
| Complete Pass (“core”) task work | X |

*Minimum Pass Checklist*

|  |  |
| --- | --- |
|  | Included? (tick) |
| Additional non-core task work (or equivalent) in a private repository and accessible to staff account. |  |
| Spike Extension Report (for spike extensions) |  |
| Custom Project plan (for D and/or low HD), and/or High HD Research Plan document (optional) | X |

*Credit Checklist, in addition to Pass Checklist*

|  |  |
| --- | --- |
|  | Included? (tick) |
| Custom Project Distinction Plan document, approved |  |
| All associated work (code, data etc.) available to staff (private repository), for non-trivial custom program(s) of own design | X |
| Custom Project “D” level documents, to document the program(s) (structure chart etc) including links to repository areas | X |

*Distinction Checklist, in addition to Credit Checklist*

|  |  |
| --- | --- |
|  | Included? (tick) |
| Custom Project “HD” level documents, to document the program(s) (structure chart etc) including links to repository areas | X |

*Low High Distinction Checklist, in addition to Distinction Checklist*

|  |  |
| --- | --- |
|  | Included? (tick) |
| High Distinction Plan document, approved |  |
| High Distinction Report document, which includes links to repository assets | X |
| All associated work (code, data etc.) available to staff (private repository) for your research work | X |

*High High Distinction (Research) Checklist, in addition to D/Low HD Checklist*

# 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: Thomas Horsley

# Introduction

This report summarises what I learnt in COS30031 Games Programming. It includes a self-assessment against the criteria described in the unit outline, a justification of the pieces included details of the coverage of the unit’s intended learning outcomes, and a reflection on my learning.

# Overview of Pieces Included

***Core Work:***

1. Spike - Git Repository Setup
2. Spike - C++ for Programmers
3. Spike - Gridworld
4. Lab - Debugging
5. Lab - Data Structure Basics
6. Spike - Performance Measurement
7. Spike - Game State Management
8. Spike - Game Data Structures
9. Lab - File Input Output
10. Spike - Game Graphs from Data
11. Spike - Command Pattern
12. Spike - Composite and Component Pattern
13. Lab - SDL2 Concepts
14. Spike - Soundboard
15. Spike - Sprites and Graphics
16. Spike - Messaging and Announcements
17. Spike - Collisions
18. Spike - Profiling Performance and Optimizations

***Extension Work:***

1. ECS Extended?
2. Custom Project Plan (HD)
3. Custom Project
4. Research Plan
5. Research Report (HD)

# Coverage of the Intended Learning Outcomes

This section outlines how the pieces I have included demonstrate the depth of my understanding in relation to each of the unit’s intended learning outcomes.

## ILO 1: Design

*Discuss game engine components including architectures of components, selection of components for a particular game specification, the role and purpose of specific game engine components, and the relationship of components with underlying technologies.*

*Engine architecture and design was prevalent during the second half of the unit. The first example of engine design was the “Composite and Component Patterns” Spike. As this task was heavily overengineered, I’ve come to realize in later learning that the patterns and ideas implemented here are used extensively within game engine design.*

*“Spike 19 – Messaging Systems” related to events and dispatching messages to the relevant components within the ECS implemented in the “Composite and Component Patterns” task. Polling and event structures were also seen in the SDL2 related Spikes, specifically “Sprites and Graphics” where input was used to randomly instance textures. Additionally, “Sprites and Graphics” introduced me to how a graphics renderer function and process data before presentation.  
  
This (along with the “Soundboard” Spike), allowed me to design a rudimentary, code-based engine environment for my custom project. This engine uses an ECS designed for turn-based strategy styles games and supports rendering ASCII environments. Additionally, input systems have been developed using the concepts discussing “Spike 19 – Messaging Systems”.*

## ILO 2: Implementation

*Create games that utilise and demonstrate game engine component functionality, including the implementation of components that encapsulate specific low-level APIs.*

*Custom Project.*

*The second milestone for the unit was making a Zork-esque CLI game (named Zorkish) utilizing the concepts and ideas discussed in the first portion of the semester. My version of Zorkish utilizes an ECS (Entity Component System) and input polling system implemented in the “Composite and Component Patterns” and “Messaging Systems” Spikes respectively. The ECS implemented within my Zorkish project was built such that it’s extensible and performant enough to support hundreds of game entities containing components. The SDL2 Spikes (“Soundboard” and “Sprites and Graphics”) served as an introduction to low level graphics API’s, allowing me the tools to interact with GPUs for parallel computing during a game / engines runtime.*

*Both the SDL2 knowledge and ECS were implemented within my custom project. TOTOTOTOTOOTTOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTOTO*

## ILO 3: Performance

*Explain and illustrate the role of data structures and patterns in game programming and rationalise the selection of these for the development of a specified game scenario.*

*Describe what you have included in your portfolio that demonstrates your ability in relation to this outcome.*

* *Pass: Identify where in your lab tests and/or core spike work where you have explained and illustrated this ILO.*
* *Credit: Evidence of depth in the portfolio work and explanations provided.*
* *Distinction, High Distinction: relate to your project and/or research*

## ILO 4: Maintenance

*Explain and illustrate the role of data structures and patterns in game programming and rationalise the selection of these for the development of a specified game scenario.*

*Describe what you have included in your portfolio that demonstrates your ability in relation to this outcome.*

* *Pass Identify where in your lab tests and/or core spike work where you have explained and illustrated this ILO.*
* *Credit: Evidence of depth in the portfolio work and explanations provided.*
* *Distinction, High Distinction: relate to your project and/or research.*

# Reflection

## The most important things I leant:

*Think about topics covered, but also other general things you may have learnt. Think about what you have learnt in this subject and reflect on what you think were key learning points, or incidents. Did you learn what you wanted/expected to learn?*

*“Spikes are throwaway code”.*

## The things that helped me most were:

*The Spike structure. Given a pointer in the right direction and enough resources to get started researching. I found the overall process of a Spike was great for expanding all domains of my programming knowledge, rather than traditional Labs which have generally been more focused and narrower.*

## I found the following topics particularly challenging:

*Combining ideas from multiple previous Spikes into one project was something I hadn’t handled before and proved incredibly challenging on my first approaches. This was experienced during the “Command Pattern” Spike, where Zorkish had to be functional enough to implement the command pattern idea involving the merging of Tasks 8 – 11. In this case the Spike itself proved trivial opposed to the implementation of the game-world.  
  
Once this issue was resolved however, I found myself taking the lessons learned into the next Spike (Composite and Component Patterns) and having a substantially easier time. Given this, the “Composite and Component Patterns” Spike was mismanaged regarding time. I didn’t account for underlying datatype and structure limitations forcing multiple refactors of the ECS before completion. The combination of these tasks took two weeks to complete due to the challenges faced and proved a major set-back for the desired outcome of my custom project.*

## I found the following topics particularly interesting:

*Entity Component Systems and overall game engine architecture have become quite*

## I feel I learnt these topics, concepts, and/or tools really well:

*List and explain – if none explain why, refer to your pieces for evidence to support your claims*

## I still need to work on the following areas:

*List and explain – if none explain why, refer to your pieces.*

## My progress in this unit was …:

*Comment on how you progressed in this unit.   
Did you consistently submit work and engage with your tutor? If not, why?  
How do you think your rate of progress affected your overall learning and final grade?  
What have you learned that you could take into future units or topics you want to learn?*

## This unit will help me in the future:

*As of the time of writing, I plan to move into the game’s development field. Whether it be engine design / development or games programming and design it depends, but in either case the knowledge gained from this unit has given me a look into the foundations of games / engine programming, the data structures and patterns behind them which have been established and used since their earliest iterations.*

## If I did this unit again, I would do the following things differently:

*List and explain, how will you approach learning in the future? What things worked well, but what could you change to make sure you did better next time?*

*Given my general understanding of games / engine programming has improved substantially since the start of the unit, I feel as though I would do better structuring the order of tasks I would accomplish during the latter half of the semester. I felt this to be a non-issue during the first portion of the unit however once I started moving into unfamiliar territory, I would often drift down rabbit-holes of programming wizardry and fail to see the forest for the trees. This caused each task to take at least a day or two longer than the last as the amount that I knew I could know grew and by the end of semester I found myself in a time crunch.*

# Conclusion

In summary, I believe that I have clearly demonstrate that my portfolio is sufficient to be awarded a HD grade.

*Add more points if you wish, but don’t add anything you haven’t already mentioned in an earlier section*