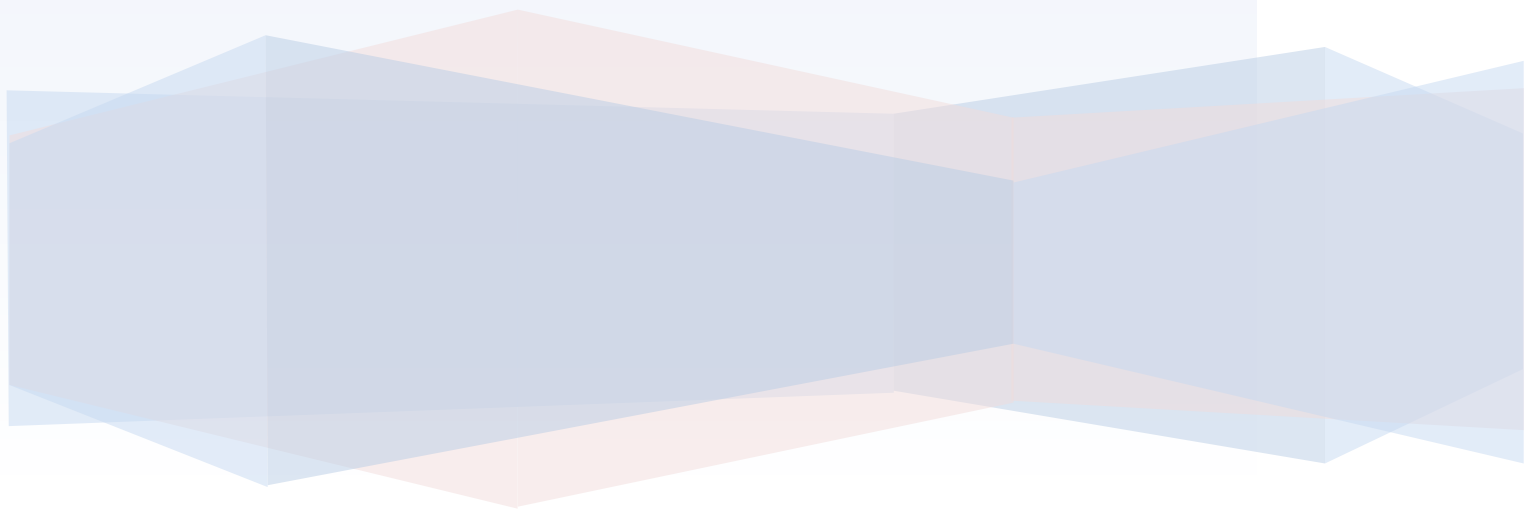


COS30031 Games Programming

Learning Summary Report
Sam Huffer (101633177)



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)			Y		

Self-assessment Statement

	Included? (tick)
Learning Summary Report.	Y
Time-boxed Demonstration Activity (Lab Test) in Doubtfire.	Y
Complete Pass ("core") task work, approved in Doubtfire.	Y

Minimum Pass Checklist

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

Credit Checklist, in addition to Pass Checklist

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

Distinction Checklist, in addition to Credit Checklist

	Included? (tick)
Custom Project "HD" level documents in Doubtfire, to document the program(s) (structure chart etc) including links to repository areas.	N

Low High Distinction Checklist, in addition to Distinction Checklist

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

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

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

This section outlines the pieces that I have included in my portfolio:

No.	Type	Name	Included?	Rationale
1	Lab	Bitbucket Setup	Y	Demonstrates the basic ability to use a repository to maintain a software project (ILO 1).
2	Lab	C++ for Programmers	Y	Demonstrates basic C++ implementation (ILO 2).
3	Spike	Gridworld	Y	Demonstrates basic planning capability (ILO 1) and basic C++ implementation (ILO 2).
4	Extension	Gridworld Multi-threaded	N	Not completed due to time constraints.
5	Spike	IDE Familiarity	Y	Demonstrates ability to use Visual Studio to implement (ILO 2), debug and maintain (ILO 4) software projects.
6	Lab	Debugging	Y	Demonstrates ability to view performance-related information (ILO 3), analyse, debug and maintain existing code (ILO 4), and re-implement it (ILO 2).
7	Lab	Data Structure Basics	Y	Demonstrates ability to analyse code to debug and maintain it (ILO 4), to research, understand and account for performance concerns (ILO 3), and with understanding why code does or doesn't work and how it should be coded to work (ILO 2).
8	Spike	Performance Measurement	Y	Demonstrates my ability to use timer code to assess the performance of other code (ILO 2, 3), and to have Visual Studio optimise compiled code (ILO 3).
9	Spike	Game State Management	Y	Demonstrates my ability to produce designs for a state pattern (ILO 1), and to implement those designs with a stable multi-file class structure in C++ (ILO 2, ILO 4).
10	Spike	Game Data Structures	Y	Demonstrates my ability to incorporate assessments of data structure suitability into designs (ILO 1), including regarding performance considerations (ILO 3), and to implement player inventories and commands (ILO 2).
11	Lab	File Input Output	Y	Demonstrates my ability to read from and write to files, and my understanding regarding such code's parameters and considerations (ILO 2).
12	Spike	Game Graphs from Data	Y	Demonstrates designing of game world text file specification format (ILO 1), implementation of the reconstruction of worlds / entities from said text files, and of the move command and direction aliases (ILO 2), and debugging of text files and reading code and reorganisation of look command code (ILO 4).
13	Lab	Attributes from Components	N	Not completed. Task removed from Canvas and Doubtfire.
14	Spike	Command Pattern	Y	Design and implementation of the command pattern from a textual specification (ILO 1, ILO 2), adapting existing command methods to the new structure to ensure consistency (ILO 4).
15	Spike	Composite and Component Patterns	Y	Design and implementation of the composite and component patterns and how to incorporate them with existing code, and the redesign and implementation of world text files and the reconstruction of game worlds and entities to work with the new code (ILO 1, ILO 2).
16	Extension	Configuration Files	Y	Design and implementation of new components for restricting and unlocking commands (ILO 1, ILO 2), reorganisation of error messages to enhance

				maintainability, and maintaining existing code by identifying and fixing bugs and deficiencies (ILO 4).
17	Lab	Message Systems	N	Not completed. Task removed from Canvas and Doubtfire.
18	Spike	Announcements and Blackboards	Y	Design and implementation of a dispatcher-style, observer pattern-based messaging system (ILO 1, ILO 2), debugging and maintenance of command unlocking code from Task 16, and discussion of future uses of messages beyond their current implementation in this task (ILO 4).
19	Extension	Messaging Comparison	Y	Design and implementation of message broadcasting, filtering and scheduling functionality, and components to demonstrate said functionality (ILO 1, ILO 2).
20	Pass	Test	Y	Discussion of uses of patterns and rationale behind choices, and of choices of data structures and their impact on the design of a player inventory system (ILO 1), and their performance and maintenance considerations (ILO 3, ILO 4).
21	Credit	Test Response	N	Not completed.
22	Lab	SDL2 Concepts	N	Not completed. Task removed from Canvas and Doubtfire.
23	Spike	Sound Board	Y	Implementation of a soundboard in C++ using SDL2 and SDL Mixer (ILO 2), and noting of separation of asset loading and management (ILO 4).
24	Spike	Sprites & Graphics	Y	Implementation of loading and managing sprites and sprite maps (ILO 2), the latter maximising performance (ILO 3), and implementation of foundational content from the tutorial series I was following such that this and future tasks could build upon it if necessary (ILO 4).
25	Extension	Control Mapping	N	Not completed due to time constraints.
26	Spike	Collisions	Y	Implementation of circle-circle and axis-aligned box-box collision testing (ILO 1), and identification and separation of input-, update-, and rendering-related code into separate, more maintainable methods (ILO 4).
27	Extension	Collisions Extended	N	Not completed due to time constraints.
28	Spike	Profiling, Performance and Optimisation	N	Not completed. Task removed from Canvas and Doubtfire.
29	Credit	Spike Extension Report	Y	Includes discussion of further additions to completed extensions and how I might tackle uncompleted extensions (ILO 1), and reiterates my implementation of the deliverables of completed extensions (ILO 2).
30	Credit	Custom Project Plan	Y	Includes planning for what my Distinction report would be and how I would analyse my capstone project (ILO 1), noting my contributions to said capstone project (ILO 2).
31	Distinction	Custom Project D Level	Y	Extensive redesigns of <i>Get the Fog Out's</i> code to better address performance and maintainability concerns discussed (ILO 1, ILO 3, ILO 4), and reiterates my implementation skills, showing both code and the final results of my contributions to <i>Get the Fog Out</i> (ILO 2).
32	Credit	Research Plan	Y	Includes planning for what my research report would be (ILO 1), notes my past attempts to use Unity's ECS, and the implementation, performance and maintenance concerns regarding the use of ECS's (ILO 2, ILO 3, ILO 4).
33	High Distinction	Custom Project HD Level	N	Not completed. Not going for a HD.

34	High Distinction	Research Report	N	Not completed. Not going for a HD.
35	Pass	Learning Summary Report	Y	Reiterates my understanding and competency regarding ILOs 1, 2, 3 and 4 for each task.
N/A	Distinction / Other	Get the Fog Out	Y	Basis of my custom project report, demonstrates implementation competency (ILO 2) and allows for discussion of performance and maintenance-affecting decisions and future designs (ILOs 3, 4 and 1 respectively).

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.

Intended Learning Outcomes

For each ILO, describe what you have included in your portfolio that demonstrates your ability in relation to each outcome:

- **Pass:** Working implementations / demonstrations from the spike work.
- **Credit:** Additional work that extends the core spike work features.
- **Distinction, High Distinction:** relate to your project and/or research.

The ILOs are:

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

Pass-Level Tasks

For each task I have included in this section, I shall describe how it demonstrates my understanding and ability in relation to each ILO at a Pass level.

Lab Task 1: Bitbucket Setup

- **ILO 4: Maintenance:** this task shows that I can set up a repository that can assist in the maintenance of a software project, demonstrating my ability regarding this ILO at a Pass level.

Lab Task 2: C++ for Programmers

- **ILO 2: Implementation:** this task shows that I can use C++ to create a basic program, thereby demonstrating my ability regarding this ILO at a Pass level.

Spike Task 3: Gridworld

- **ILO 1: Design:** this task shows that I can create a basic design for a program from a written outline, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 2: Implementation:** this task shows that I can create a basic program in C++, thereby demonstrating my ability regarding this ILO at a Pass level.

Spike Task 5: IDE Familiarity

- **ILO 2: Implementation:** this task shows my familiarity with Visual Studio and my ability to use it to create a Visual Studio solution for implementing a software project, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 4: Maintenance:** this task shows my familiarity with Visual Studio and my ability to use it to debug and therefore maintain the code of a software project, thereby demonstrating my ability regarding this ILO at a Pass level.

Lab Task 6: Debugging

- **ILO 2: Implementation:** this task shows my ability to understand, debug and re-implement existing code, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 3: Performance:** this task shows my ability to display the memory occupied by different variables, information that can be used to inform performance-affecting decisions, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 4: Maintenance:** this task shows my ability to analyse, understand, debug and maintain existing code, thereby demonstrating my ability regarding this ILO at a Pass level.

Lab Task 7: Data Structure Basics

- **ILO 2: Implementation:** this task shows my understanding of why particular lines of code do and don't work, and therefore how something should be coded to get it to work, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 3: Performance:** this task shows my ability to research, understand and account for performance-related concerns when programming, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 4: Maintenance:** this task shows my ability to analyse and understand code to debug and maintain it, thereby demonstrating my ability regarding this ILO at a Pass level.

Spike Task 8: Performance Measurement

- **ILO 1: Design:** this task shows
- **ILO 2: Implementation:** this task shows my ability to analyse, understand, and re-implement code for the purpose of timing other code under a variety of conditions, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 3: Performance:** this task shows my ability to the execution of code under a variety of conditions, to record, display and analyse the results, and to configure Visual Studio to optimise code at compilation time, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 4: Maintenance:** this task shows

Spike Task 9: Game State Management

- **ILO 1: Design:** this task shows my ability to produce UML class diagram designs for state-pattern-based software according to provided specifications, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 2: Implementation:** this task shows my ability: to use .h files, #includes statements, and forward declarations to implement a stable class structure across multiple files; and to implement the singleton and state patterns, the latter incorporating the casting child classes as their parent class. As such, it demonstrates my ability regarding this ILO at a Pass level.
- **ILO 4: Maintenance:** this task shows my ability to implement a stable class structure across multiple files as noted above, a structure that is much more maintainable than a shopping list of anything and everything all in one file, thereby demonstrating my ability to produce maintainable code, and my ability regarding this ILO at a Pass level.

Spike Task 10: Game Data Structures

- **ILO 1: Design:** this task includes my assessment of the suitability of various data structures for a given task in the research report, and shows the incorporation of my findings in the implementation of a player inventory, demonstrating my ability to incorporate evidence in my (formal or otherwise) design process, and thereby my ability regarding this ILO at a Pass level.
- **ILO 2: Implementation:** this task shows my ability to implement a player inventory system on top of a suitable data structure, as well as my ability to implement commands that can be adapted to the command pattern proper, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 3: Performance:** this task includes my assessment of the suitability of various data structures for a given task in this task's research report. That assessment included considering performance-affecting characteristics of said data structures, thereby demonstrating my understanding and ability regarding this ILO at a Pass level.

Lab Task 11: File Input Output

- **ILO 2: Implementation:** this task shows my ability to use basic file input and output code to read from and write to files, and understanding regarding various parameters and considerations to be accounted for when doing so, thereby demonstrating my ability and understanding regarding this ILP at a Pass level.

Spike Task 12: Game Graphs from Data

- **ILO 1: Design:** this task shows my ability to design a format for representing game worlds and entities in a text file that can be read into a program and have the game worlds and entities reconstructed from its details, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 2: Implementation:** this task shows my ability to implement the reading of aforementioned text files and reconstruction of game worlds and entities based on what the text file specifies, to implement a command for moving between locations within created game worlds, and for providing directions

with aliases that they can be referred to by when the movement command is invoked, thereby demonstrating my ability regarding this ILO at a Pass level.

- **ILO 4: Maintenance:** this task shows my ability to debug and maintain world text files and the code for parsing them to reconstruct game world from them, and to reorganise code for looking around in that game world such that that code is not duplicated repeatedly and therefore more maintainable, thereby demonstrating my ability regarding this ILO at a Pass level.

Spike Task 14: Command Pattern

- **ILO 1: Design:** this task shows my ability to design an implementation of the command pattern from a textual specification, also incorporating existing commands to be adapted to the command pattern proper. This demonstrates my ability regarding this ILO at a Pass level.
- **ILO 2: Implementation:** this task shows my ability to implement the aforementioned designs and creating the command pattern's class structure and new commands from scratch, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 4: Maintenance:** this task shows my ability to reorganise existing command methods into their own command classes, enhancing their maintainability and ensuring consistency with the new commands, and to address warnings of mismatches between signed and unsigned integers to prevent future errors, thereby demonstrating my ability regarding this ILO at a Pass level.

Spike Task 15: Composite and Component Patterns

- **ILO 1: Design:** this task shows my ability to design an implementation of the composite and component patterns based on a written brief, and to adjust the text file specification format for worlds and game entities to match the patterns to be implemented, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 2: Implementation:** this task shows my ability to implement the composite and component patterns based on the aforementioned designs, adding new component classes not just based on existing functionality but new content derived from my designs, and to adjust the reconstruction of worlds and game entities from text files to match the new text file specification format, thereby demonstrating my ability regarding this ILO.

Spike Task 18: Announcements and Blackboards

- **ILO 1: Design:** this task shows my ability to design a dispatcher-style messaging system that takes advantage of the observer pattern, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 2: Implementation:** this task shows my ability to implement a dispatcher-style, observer pattern-based messaging system based on my aforementioned designs, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 4: Maintenance:** this task shows my ability to debug and maintain my Task 16 implementation of the unlocking of commands under various conditions, and my understanding of maintainability advantages of messages via my discussion of their possible further uses beyond what I implemented in this task, thereby demonstrating my ability and understanding regarding this ILO at a Pass level.

Pass Task 20: Test

- **ILO 1: Design:** this task includes my discussion of choices of data structures and the impact of different choices when designing a player inventory system, as well as my discussion of where I've used particular patterns in my spike work and why, thereby showing my ability and understanding regarding this ILO at a Pass level.
- **ILO 3: Performance:** this task includes my discussion of performance considerations of choosing particular data structures to build player inventories on top of, thereby demonstrating my ability and understanding regarding this ILO at a Pass level.
- **ILO 4: Maintenance:** this task includes my discussion of maintenance considerations when choosing particular data structures to build player inventories on top of, thereby demonstrating my ability and understanding regarding this ILO at a Pass level.

Spike Task 23: Sound Board

- **ILO 2: Implementation:** this task shows my use of the SDL2 and SDL Mixer libraries to create a sound board in C++, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 4: Maintenance:** this task includes my noting of the beneficence of separating asset management tasks for maintainability, thereby demonstrating my understanding regarding this ILO at a Pass level.

Spike Task 24: Sprites and Graphics

- **ILO 2: Implementation:** this task shows my ability to implement the loading and management of sprites and sprite maps using the SDL2 and SDL Image libraries, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 3: Performance:** this task shows my use of sprite maps to maximise the performance of programs that make extensive use of images, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 4: Maintenance:** this task includes my updating of code from Task 23 to be in-line with previous tutorials in the tutorial series I was following for this task in case I needed that previous functionality to properly implement sprite loading and management, and in case future tasks needed to be built upon that earlier functionality as well, thereby demonstrating my awareness, understanding and ability regarding this ILO at a Pass level.

Spike Task 26: Collisions

- **ILO 2: Implementation:** this task shows my ability to implement circle-circle and axis-aligned box-box collision testing, thereby demonstrating my ability regarding this ILO at a Pass level.
- **ILO 4: Maintenance:** this task shows my ability to identify and separate input-, update- and rendering-related code into separate methods, enhancing maintainability and thereby demonstrating my ability regarding this ILO at a Pass level.

Pass Task 35: Learning Summary Report

- **ILO 1: Design:** this learning summary report reiterates my understanding and competency regarding software design, my discussions of the decision-making underlying the designs of specific tasks and projects included in my portfolio, and how they relate to this learning outcome, thereby exhibiting my understanding and competency regarding software design.
- **ILO 2: Implementation:** this learning summary report reiterates my understanding and capability regarding software implementation, my discussions of the decision-making underlying how I implemented specific tasks and projects included in my portfolio, and how that relates to this learning outcome, thereby exhibiting my understanding and competency regarding software implementation.
- **ILO 3: Performance:** this learning summary report reiterates my understanding and capability regarding software optimisation for performance, my discussions regarding the effects on performance of various decisions made while working on the tasks and projects included in my portfolio, and how that relates to this learning outcome, thereby exhibiting my understanding and competency regarding optimising software's performance.
- **ILO 4: Maintenance:** this learning summary report reiterates my understanding and capability regarding software optimisation for maintainability, my discussions regarding the effects on maintainability of various decisions made while working on the tasks and projects included in my portfolio, and how that relates to this learning outcome, thereby exhibiting my understanding and competency regarding optimising software's maintainability.

Credit-Level Tasks

For each task I have included in this section, I shall describe how it demonstrates my understanding and ability in relation to each ILO at a Credit level.

Spike Task 12: Game Graphs from Data

- **ILO 1: Design:** this task shows my ability to design a format for representing game worlds and entities (i.e. items, etc.) in a text file that can be read into a program and have the game worlds and entities reconstructed from its details, thereby demonstrating my ability regarding this ILO at a Pass level. Unbeknownst to me when I completed this task, representation of such entities in world text files is listed as an option for Task 16: Configuration files, an Extension task and therefore Credit level. As such, this task also demonstrates my ability regarding this ILO at a Credit level.
- **ILO 2: Implementation:** this task shows my ability to implement the reading of aforementioned text files and reconstruction of game worlds and entities based on what the text file specifies, to implement a command for moving between locations within created game worlds, and for providing directions with aliases that they can be referred to by when the movement command is invoked, thereby demonstrating my ability regarding this ILO at a Pass level. Unbeknownst to me when I completed this task, construction of such entities from world text files is listed as an option for Task 16: Configuration files, an Extension task and therefore Credit level. As such, this task also demonstrates my ability regarding this ILO at a Credit level.

Spike Task 15: Composite and Component Patterns

- **ILO 1: Design:** this task shows my ability to design an implementation of the composite and component patterns based on a written brief, and to adjust the text file specification format for worlds and game entities (i.e. items, components, etc.) to match the patterns to be implemented, thereby demonstrating my ability regarding this ILO at a Pass level. Unbeknownst to me when I completed this task, representation of such entities in world text files is listed as an option for Task 16: Configuration files, an Extension task and therefore Credit level. As such, this task also demonstrates my ability regarding this ILO at a Credit level.
- **ILO 2: Implementation:** this task shows my ability to implement the composite and component patterns based on the aforementioned designs, adding new component classes not just based on existing functionality but new content derived from my designs, and to adjust the reconstruction of worlds and game entities from text files to match the new text file specification format, thereby demonstrating my ability regarding this ILO. Unbeknownst to me when I completed this task, construction of such entities from world text files is listed as an option for Task 16: Configuration files, an Extension task and therefore Credit level. As such, this task also demonstrates my ability regarding this ILO at a Credit level.

Extension Task 16: Configuration Files

- **ILO 1: Design:** this task shows my ability to design new components to be added to the program, and to adjust the text file specification of game objects and components to handle them, and to enable the restriction and unlocking of commands, thereby demonstrating my ability regarding this ILO at a Credit level.
- **ILO 2: Implementation:** this task shows my ability to implement new components to enable the restriction and unlocking of commands in line with the aforementioned design, thereby demonstrating my ability regarding this ILO at a Credit level.
- **ILO 4: Maintenance:** this task shows my ability to reorganise error message printing to reuse error messages where possible and thus enhance maintainability, and to maintain existing code by identifying and removing bugs and deficiencies, thereby demonstrating my ability regarding this ILO at a Credit level.

Extension Task 19: Messaging Comparison

- **ILO 1: Design:** this task shows my ability to design message broadcasting, filtering and scheduling functionality to add to my existing messaging system, and to design components to demonstrate said functionality, including how such components would be represented in world text files, thereby demonstrating my ability regarding this ILO at a Credit level.
- **ILO 2: Implementation:** this task shows my ability to implement message broadcasting, filtering and scheduling in line with the aforementioned designs, as well as the components required to demonstrate that functionality, including updating the text file specification format and code for reading those world text files to handle the new components, thereby demonstrating my ability regarding this ILO at a Credit level.

Credit Task 29: Spike Extension Report

- **ILO 1: Design:** this task includes my discussion of further possible additions I could make to some extension tasks that I completed, and how I might go about addressing extension tasks that I did not complete, thereby demonstrating my understanding and ability regarding this ILO at a Credit level.
- **ILO 2: Implementation:** this task reiterates my implementation of the deliverable of the extension tasks that I completed, reiterating my understanding and ability regarding this ILO at a Credit level.

Credit Task 30: Custom Project Plan

- **ILO 1: Design:** this task shows my ability to plan and set goals for a report analysing my capstone project in terms of this unit's content and what that report would cover, thereby demonstrating my awareness and understanding regarding this ILO at a Credit level. Note: my description of what the report would cover nods at discussing performance and maintenance concerns under the banner of data structures' and data patterns' suitability for my capstone project, but doesn't explicitly mention them.
- **ILO 2: Implementation:** this task notes my contributions to my capstone project, where much of this unit's content would be applicable in implementing various features of the project, thereby demonstrating my awareness and understanding regarding this ILO at a Credit level.

Credit Task 32: Research Plan

- **ILO 1: Design:** this task includes my planning and goal setting for a research report investigating and comparing game engines' entity-component systems, thereby demonstrating my understanding and ability regarding this ILO at a Credit level.
- **ILO 2: Implementation:** this task notes my past attempts at using Unity's entity-component system and that I had difficulty there, and shows my awareness of implementation concerns regarding using entity-component systems, thereby demonstrating my awareness and understanding regarding this ILO at a Credit level.
- **ILO 3: Performance:** this task shows my awareness of performance concerns and that entity-component systems are a potential tool for addressing them, thereby demonstrating my awareness and understanding regarding this ILO at a Credit level.
- **ILO 4: Maintenance:** this task shows my awareness of maintainability concerns regarding the use of entity-component systems, thereby demonstrating my awareness and understanding regarding this ILO at a Credit level.

Distinction-Level Tasks and Pieces

For each task or piece I have included in this section, I shall describe how it demonstrates my understanding and ability in relation to each ILO at a Distinction level.

Distinction Task 31: Custom Project

- **ILO 1: Design:** this task extensively shows my ability to produce designs that rework an existing software artefact to better address performance and maintenance concerns, thereby demonstrating my understanding and ability regarding this ILO at a Distinction level.
- **ILO 2: Implementation:** this task reiterates my ability to implement a design to a Distinction level, showcasing the results of my work and that of my teammates through screenshots, and includes screenshots of code that I have contributed to the project under discussion, thereby demonstrating my ability regarding this ILO at a Distinction level.
- **ILO 3: Performance:** this task extensively shows my understanding and awareness of performance concerns through my discussion and comparison of the performance impacts of using various data structures, patterns, and implementations, as well as the performance of previous and existing code, thereby demonstrating my awareness, understanding and ability regarding this ILO at a Distinction level.
- **ILO 4: Maintenance:** this task extensively shows my understanding and awareness of maintenance concerns through my discussion and comparison of the maintainability impacts of using various data structures, patterns, and implementations, as well as the maintainability of existing code, thereby demonstrating my awareness, understanding and ability regarding this ILO at a Distinction level.

Other Piece: Custom Project

- **ILO 1: Design:** *Get the Fog Out* serves as an artefact for the discussion of its design and the rationales behind various decisions, particularly as they relate to performance and maintenance concerns, and what changes I would make in future.
- **ILO 2: Implementation:** *Get the Fog Out* demonstrates to a Distinction level my ability to implement a design, being the largest, most fully-formed, and most polished project that I have worked on to date. It also serves as an artefact for discussion of how it was implemented and why I made the decisions that I did regarding its implementation.
- **ILO 3: Performance:** *Get the Fog Out* serves as an artefact for the discussion of design and implementation decisions affecting performance, and what changes I would make in future.
- **ILO 4: Maintenance:** *Get the Fog Out* serves as an artefact for the discussion of design and implementation decisions affecting performance, and what changes I would make in future.

Reflection

The most important things I learnt:

- I learnt how to program in C++, particularly regarding how to split classes into header (.h) and implementation (.cpp) files, and how to include other header files at the start of a .h or .cpp file; this unit was the first time I have used C++.
- I learnt about a variety of data patterns and usability patterns, some of which I'd heard of before, others that I hadn't, each being discussed in substantial detail, and some being implemented in spike tasks.
- I learnt a lot more about a variety of data structures, including some that I hadn't heard of or encountered before, and with a particular focus on their pros and cons when used in C++.

The things that helped me most were:

- The lectures and my lecture notes, which covered the major concepts, the rationale behind them, and how they might be implemented.
- Online forums and examples to further explain how to implement a particular component in C++ or do a specific thing in C++. The spike instructions detailed a lot of the "what" and the labs (when available) detailed some of the "how", but for the nitty-gritty "how" or the "how" not covered by some of the later labs that weren't made available, Google was my friend.
- Visual Studio's pch.h header file; including every header file in this header file, including it in every other header file, and just forward-declaring classes where necessary helped resolve so much of the headaches relating to including header files in other classes.
- Tien. Asking him questions regarding the particulars of this or that task was very helpful in making sure I understood what each task was asking for.

I found the following topics particularly challenging:

- C++. Like I said, this unit was the first time I've used C++, so I had to learn it from scratch this semester. The early labs did help some, but there was still a lot that they didn't cover, so figuring out C++ on the fly was half the work, particularly for the earlier tasks.
- Including header files. This was a nightmare and a half until I caved in and started using Visual Studio's pch.h header file.

I found the following topics particularly interesting:

- Data patterns: it was interesting seeing what patterns were available for use, learning more about the theory and implementation of ones that I'd heard about, and discovering others that I hadn't. Particularly, it was interesting noticing ones that I'd implemented in my capstone project or that might be of benefit there.

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

- C++: it was a lot of work to learn to use it well, but I reckon I got the hang of it pretty well by the end of the semester. I certainly found myself focusing less on the "how" of general C++ and more on the "how" of the specific tasks and patterns, and the specialised libraries required for each task. See my spike and extension work, especially the later tasks for Zorkish Adventures and the SDL2 tasks.
- Representing game worlds and entities in text files, and reading them into the game to reconstruct those worlds and entities. See tasks 11, 12, 15, 16 and 19; I'm pretty happy with my specification for representing game entities in a text file and the code to read those text files and reconstruct their detailed entities, and where it got to by the end of the Zorkish tasks.
- The state, command, composite, component, message and factory patterns: I feel I got pretty comfortable with these patterns and how to use them, and made at least small-scale demonstrations of each in their associated tasks (9, 12, 14, 15, 18 and 19).

I still need to work on the following areas:

- Looking at the tasks I didn't complete, I did not cover and could therefore improve upon the following skills:
 - Multi-threading: this was covered in Task 4: Gridworld Multi-threaded, which I did not complete due to time constraints. As I haven't managed to get multi-threading working in other units, this is an area that I could improve in and would benefit from improving in.
 - Runtime control remapping: this was covered in Task 25: Control Mapping, which I did not complete due to time constraints. I understand the concept of remapping controls at runtime, but haven't had much practice in it beyond providing aliases for commands in command

pattern-related tasks. As such, I could do with some practice in implementing the remap-ability of controls.

- Collision detection: this was covered in Task 26: Collisions and Task 27: Collisions Extended. The former only covered circle-circle collisions and axis-aligned box-box collisions, and did not cover more complex box-box collision tests, nor collision tests for differing or complex polygons, while I did not complete the latter due to time constraints. As such, I could do with some further practice here to reinforce my understanding of how collision detection is implemented.
- Profiling code, assessing its performance, and optimising it: this was the focus of the task of Task 28: Profiling, Performance and Optimisation, which I did not complete as it was rendered optional and I had limited time to complete tasks due to other units. Any tips and tricks this task covers, I am possibly not aware of. As such, I could further develop my ability to assess the performance of and optimise code by completing this task.

My progress in this unit was ...:

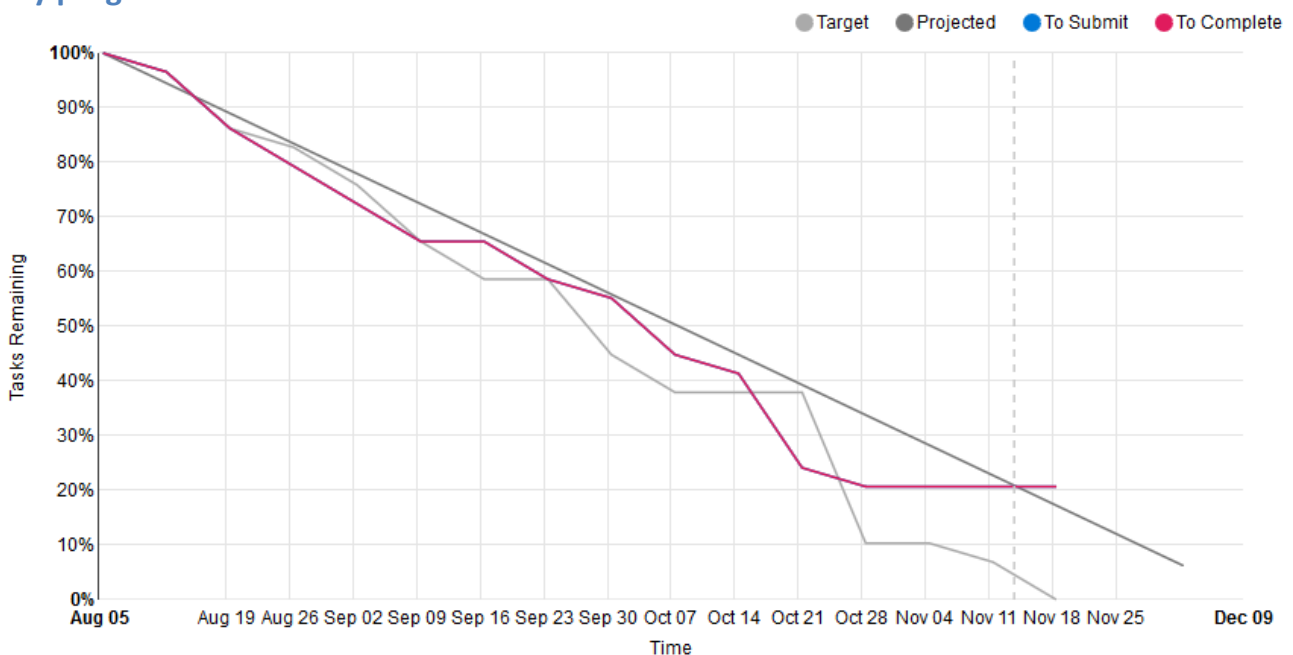


Figure 1: the burndown chart of my progress through the semester.

During the semester, I aimed to keep up with the due dates of tasks on Doubtfire, as well as the task spreadsheet on Canvas, finishing tasks by the end of the week that they were due to be started and having all Pass and Credit work signed off by week 11 or 12. My task progress was consistent and reasonably close to the target line on Doubtfire. I was, on and off, about a week behind my goal of finishing tasks the same week they were due to be started, but I did submit most if not all tasks within the recommended time frame of the task spreadsheet, and I had all tasks besides this learning summary report and my distinction report completed and signed off in week 11. I would have liked to have met my goal of starting and finishing tasks in the same week they were due to be started, even gotten a bit ahead, and managed to complete all tasks, but due to commitments in other units, I didn't have enough time to devote to that.

This unit will help me in the future:

- I now am able to code programs using C++, both in terms of using C++ code and splitting classes amongst multiple .h and .cpp files and including in .h and .cpp files the .h files and libraries required for their operation. This will be quite useful, as C++ is a common development language for software and game development, and thus my studies and career will benefit from me being able to work with it.
- I now have a more complete theoretical and practical knowledge of software and usability patterns and the pros and cons of various data structures. I will be able to apply this understanding to solve problems in future projects that I contribute to, both as a student and as a non-student game developer post university.

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

- Ideally, if I had the time, I would want to attempt the extension tasks that I opted not to this semester due to time constraints. Similarly, I would also want to attempt a High Distinction research report and to develop my Distinction report to a High Distinction level.
- If I did this unit again a second time, I would want to be able to use C++ from the start rather than have to learn it as I go. That would have made the first few tasks a lot quicker to work through, and given me more time to spread out the tasks focusing on data structures and patterns, perhaps affording me the time required to do the tasks I opted not to. Certainly, it would have saved me most or all of the time that I spent learning how to work with it rather than having it work against me.

Conclusion

I have completed all of the required Pass-level tasks, all of the Credit-level document-based tasks and a good number of the extension tasks, and a Distinction-level report analysing, discussing, and redesigning my contributions to my capstone project. As such, I believe that I have clearly demonstrated through the code that I have implemented, the accompanying reports, and other discussion here, that my portfolio is of sufficient quality to be awarded a Distinction grade.