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. |
| 2 | Lab | C++ for Programmers | Y | Demonstrates basic C++ implementation. |
| 3 | Spike | Gridworld | Y |  |
| 4 | Extension | Gridworld Multi-threaded | N | Not completed. |
| 5 | Spike | IDE Familiarity | Y |  |
| 6 | Lab | Debugging | Y |  |
| 7 | Lab | Data Structure Basics | Y |  |
| 8 | Spike | Performance Measurement | Y |  |
| 9 | Spike | Game State Management | Y |  |
| 10 | Spike | Game Data Structures | Y |  |
| 11 | Lab | File Input Output | Y |  |
| 12 | Spike | Game Graphs from Data | Y |  |
| 13 | Lab | Attributes from Components | N | Not completed. Task sheet unavailable. |
| 14 | Spike | Command Pattern | Y |  |
| 15 | Spike | Composite and Component Patterns | Y |  |
| 16 | Extension | Configuration Files | Y |  |
| 17 | Lab | Message Systems | N | Not completed. Task sheet unavailable. |
| 18 | Spike | Announcements and Blackboards | Y |  |
| 19 | Extension | Messaging Comparison | Y |  |
| 20 | Test | Test | Y |  |
| 21 | Test | Test Response | N | Not completed. |
| 22 | Lab | SDL2 Concepts | N | Not completed. Task sheet unavailable? |
| 23 | Spike | Sound Board | Y |  |
| 24 | Spike | Sprites & Graphics | Y |  |
| 25 | Extension | Control Mapping | N | Not completed. |
| 26 | Spike | Collisions | Y |  |
| 27 | Extension | Collisions Extended | N | Not completed. |
| 28 | Spike | Profiling, Performance and Optimisation | N | Not completed. Task sheet unavailable? |
| 29 | Credit | Spike Extension Report | Y |  |
| 30 | Credit | Custom Project Plan | Y |  |
| 31 | Distinction | Custom Project D Level | Y |  |
| 32 | Credit | Research Plan | Y |  |
| 33 | High Distinction | Custom Project HD Level | Y |  |
| 34 | High Distinction | Research Report | Y |  |
| 35 | LSR | Learning Summary Report | Y | Reiterates my understanding and competency regarding each ILO for each task. |
| N/A | Other | Get the Fog Out | Y | Basis of my custom project report, demonstrates implementation competency. |

# 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 alibility 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 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### Spike Task 12: Game Graphs from Data

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### Spike Task 14: Command Pattern

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### Spike Task 15: Composite and Component Patterns

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### Spike Task 18: Announcements and Blackboards

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### Pass Task 20: Test

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### Spike Task 23: Sound Board

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### Spike Task 24: Sprites and Graphics

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### Spike Task 26: Collisions

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### 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.

### Extension Task 16: Configuration Files

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### Extension Task 19: Messaging Comparison

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### Credit Task 29: Spike Extension Report

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### Credit Task 30: Custom Project Plan

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### Credit Task 32: Research Plan

* **ILO 1: Design:** this task shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

## 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 shows
* **ILO 2: Implementation:** this task shows
* **ILO 3: Performance:** this task shows
* **ILO 4: Maintenance:** this task shows

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

### 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 leant:

* 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:

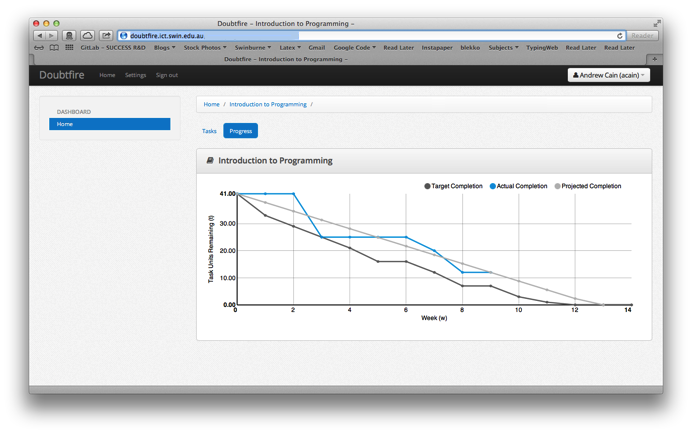
* C++: it was a lot of work to learn to use it well, but I reckon I got the hang of it pretty well by the end of the semester. I certainly found myself focusing less on the “how” of general C++ and more on the “how” of the specific tasks and patterns, and the specialised libraries required for each task. 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 …:

*Include a screenshot of your progress graph from DoubtFire, and comment on what happened from your perspective… what does the graph say about how you approached the unit? (Login to Doubtfire to get your graph* [*https://doubtfire.ict.swin.edu.au*](https://doubtfire.ict.swin.edu.au)*)*



## This unit will help me in the future:

*How will the things you learnt relate to the rest of your studies, and career. What have you learnt that will be valuable for you in the future?*

## 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?*

## Other…:

*Add any other reflections you think help you demonstrate your learning*

# Conclusion

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

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