College of Engineering, Construction and Living Sciences
Bachelor of Information Technology
ID630151: Introduction to Algorithmic Problem Solving
Level 6, Credits 15

Portfolio

Assessment Overview

In this **individual** assessment, you will develop **four** games using **C#** and **Unity**. You will build the core mechanics for each game in class using the provided **lecture notes**. You will be given **assessment tasks** to implement in your learner-managed time. These tasks are at a beginner to intermediate level. In addition, marks will be allocated for code quality and best practices, and documentation.

Learning Outcomes

At the successful completion of this course, learners will be able to:

1. Design and build usable, attractive games using various introductory algorithms following an appropriate software development methodology.

Assessments

Assessment	Weighting	Due Date	Learning Outcomes
Portfolio	100%	21-06-2024 (Friday at 4.59 PM)	1

Conditions of Assessment

You will complete this assessment during your learner-managed time. However, there will be time during class to discuss the requirements and your progress on this assessment. This assessment will need to be completed by **Friday, 21 June 2024** at **4.59 PM**.

Pass Criteria

This assessment is criterion-referenced (CRA) with a cumulative pass mark of **50%** over all assessments in **ID630151: Introduction to Algorithmic Problem Solving**.

Authenticity

All parts of your submitted assessment **must** be completely your work and any references **must** be cited appropriately including, externally-sourced graphic elements using **APA 7th edition**. Provide your references in a **README.md** file. All media **must** be royalty free (or legally purchased) for educational use. Failure to do this will result in a mark of **zero** for this assessment.

Policy on Submissions, Extensions, Resubmissions and Resits

The school's process concerning submissions, extensions, resubmissions and resits complies with **Otago Polytechnic** policies. Learners can view policies on the **Otago Polytechnic** website located at https://www.op.ac.nz/about-us/governance-and-management/policies.

Submission

You **must** submit all application files via **GitHub Classroom**. Here is the URL to the repository you will use for your submission – https://classroom.github.com/a/-8gjb_cN. If you do not have not one, create a **.gitignore** and add the ignored files in this resource - https://raw.githubusercontent.com/github/gitignore/main/Unity.gitignore. The latest application files in the **main** branch will be used to mark against the **Functionality** criterion. Please test before you submit. Partial marks **will not** be given for incomplete functionality. Late submissions will incur a **10% penalty per day**, rolling over at **5:00 PM**.

Extensions

Familiarise yourself with the assessment due date. Extensions will **only** be granted if you are unable to complete the assessment by the due date because of **unforeseen circumstances outside your control**. The length of the extension granted will depend on the circumstances and **must** be negotiated with the course lecturer before the assessment due date. A medical certificate or support letter may be needed. Extensions will not be granted for poor time management or pressure of other assessments.

Resits

Resits and reassessments **are not** applicable in **ID630151: Introduction to Algorithmic Problem Solving**.

Instructions

Functionality - Learning Outcome 1 (50%)

- Application **must** open without code or file structure modification in **Unity**.
- The four games you will create are:
  - Introduction to Unity scripting - Sheep Saving (25%)
  - Game mechanics - Tower Defence (25%)
  - Maze generation - 3D Dungeon Crawler (25%)
  - AI strategy - Chess (25%)
- In the course materials repository on **GitHub**, you will find the following directories:
  - 01-introduction-to-unity-scripting
  - 02-game-mechanics
  - 03-maze-generation
  - 04-ai-strategy

- In each of these directories, you will find additional directories - **lecture notes** and **assessment tasks**.
  - The **lecture notes** consist of detailed step-by-step tasks that will help you develop skills and knowledge in **Unity** while building a simple game. In addition, you will be introduced to commonly used algorithms in games.
  - The **assessment tasks** consist of step-by-step tasks that will help you extend the functionality of your game. However, these tasks are not as detailed as the **lecture notes**. The **advanced assessment tasks** consist of **independent research** tasks that will help you extend the functionality of your game to an **intermediate level**.

Code Quality and Best Practices - Learning Outcome 1 (45%)

- A **Unity .gitignore** file is used.
- Appropriate naming of files, variables, methods and classes.
- Idiomatic use of values, control flow, data structures and in-built functions.
- Efficient algorithmic approach.
- Sufficient modularity.
- Each file has an **XML documentation comment** located at the top of the file. In the **root** directory of the **course materials** repository, you will find an **XML documentation comment** example in the **xml-documentation-comment.txt** file.
- Formatted code.
- No dead or unused code.

Documentation - Learning Outcome 1 (5%)

- A **GitHub** project board to help you organise and prioritise your development work. The course lecturer needs to see consistent use of the **GitHub** project board for the duration of the assessment.

Additional Information

- **Do not** rewrite your **Git** history. It is important that the course lecturer can see how you worked on your assessment over time.
- You need to show the course lecturer the initial **GitHub** project board before you start your development work. Following this, you need to show the course lecturer your **GitHub** project board at the end of each week.