College of Engineering, Construction and Living Sciences
Bachelor of Information Technology
ID737001: Game Development
Level 7, Credits 15

Assignment

Assessment Overview

In this **group** assessment, you will design and develop a game in collaboration with **Bachelor of Design (Communication)** learners using **Unity** and **C#**. You will be assessed on your ability to demonstrate technical and professional proficiency, and robust documentation and reflective practices.

Assessments

Assessment	Weighting	Due Date	Learning Outcome
Assignment	30%	07-06-2024 (Friday at 4.59 PM)	1
Project: Game Development + Demo	70%	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, 07 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 **ID737001: Game Development**.

Authenticity

All parts of your submitted assessment **must** be completely your work. Do your best to complete this assessment without using an **AI generative tool**. You need to demonstrate to the course lecturer that you can meet the learning outcome(s) for this assessment.

However, if you get stuck, you can use an **AI generative tool** to help you get unstuck, permitting you to acknowledge that you have used it. In the assessment's repository **README.md** file, please include what prompt(s) you provided to the **AI generative tool** and how you used the response(s) to help you with your work. It also applies to code snippets retrieved from **StackOverflow** and **GitHub**.

Failure to do this may 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/UgQiawTf. 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 **ID737001: Game Development**.

Instructions

Functionality - Learning Outcome 1 (50%)

- Application **must** open without code or file structure modification in **Unity**.
- Gather client requirements and deconstruct them into user stories.
- Contribute a meaningful amount of code to the client project.
- Perform other jobs such as code reviews, testing and project management.

- Communicate within your team, i.e., client, **Bachelor of Information Technology** and **Bachelor of Design (Communication)** learners, to maintain sustainable productivity. Forms of communication include but are not limited to face-to-face or online team meetings, instant messaging and in-class discussions.

Code Quality and Best Practices - Learning Outcome 1 (30%)

- 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 and Git Usage - Learning Outcome 1 (20%)

- **GitHub** issues and a project board to help you organise and prioritise your development work. The course lecturer needs to see consistent use of **GitHub** issues and the project board for the duration of the assessment.
- In a **Microsoft Word** document called **game-document**, include the following:
  - Core concept
  - Design pillars
  - Main features and mechanics
  - Target platform and audience
  - Interface and controls
  - Basic story
  - Visual style
  - Audio style
- In a **Microsoft Word** document called **reflections**, include the following:
  - Select two interesting game mechanics that you implemented in your game and explain the following:
    - What did you implement?
    - What did you research during the implementation? Provide a link to the resources you used.
    - What did you try? What worked? What did not work?
    - What did you learn?
    - How can you apply what you learned to future games?
  - During the development of your game, what did you find most challenging professionally? How did you overcome it?
- Correct spelling and grammar.
- Your **Git commit messages** should:
  - Reflect the context of each functional requirement change.
  - Be formatted using an appropriate naming convention style.

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.