College of Engineering, Construction & Living Sciences
Bachelor of Information Technology
ID511001: Programming 2
Level 5, Credits 15

Classroom Tasks: Unity Game Engine Research

Assessment Overview

In this assessment, you will develop two projects using the **Unity** game engine as well answer questions pertaining to different concepts in the **Unity** game engine.

Learning Outcomes

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

1. Build interactive, event-driven GUI applications using pre-built components.
2. Declare & implement user-defined classes using encapsulation, inheritance & polymorphism.

Assessments

Assessment	Weighting	Due Date	Learning Outcomes
Project 1 (C# Console App): Learner Gradebook	25%	26-04-2023 (Wednesday at 4.59 PM)	1 & 2
Project 2 (C# Windows Forms App): Pong	35%	14-06-2023 (Wednesday at 4.59 PM)	1 & 2
Theory Examination	30%	21-06-2023 (Wednesday at 4.45 PM)	1 & 2
Classroom Tasks	10%	07-06-2023 (Wednesday at 4.59 PM)	1 & 2

Conditions of Assessment

You will complete this assessment during your learner-managed time. However, there will be time during class to discuss the requirements & your progress on this assessment. This assessment will need to be completed by **Wednesday, 07 June 2023** at **4.59 PM**.

Pass Criteria

This assessment is criterion-referenced (CRA) with a cumulative pass mark of **50%** over all assessments in **ID511001: Programming 2**.

Authenticity

All parts of your submitted assessment **must** be completely your work. If you use code snippets from **GitHub**, **StackOverflow** or other online resources, you **must** reference it appropriately using **APA 7th edition**. Provide your references in the **README.md** file in your repository. Failure to do this will result in a mark of **zero** for this assessment.

Policy on Submissions, Extensions, Resubmissions & Resits

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

Submission

You **must** submit all app files via **GitHub Classroom**. Here is the URL to the repository you will use for your submission – https://classroom.github.com/a/YOLzkboo. Create a **.gitignore** & add the ignored files in this resource - https://raw.githubusercontent.com/github/gitignore/main/Unity.gitignore. The latest app files in the **classroom-tasks-unity** 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. Contact the course lecturer before the due date if you need an extension. If you require more than a week's extension, you will need to provide a medical certificate or support letter from your manager.

Resubmissions

Learners may be requested to resubmit an assessment following a rework of part/s of the original assessment. Resubmissions are to be completed within a negotiable short time frame & usually **must** be completed within the timing of the course to which the assessment relates. Resubmissions will be available to learners who have made a genuine attempt at the first assessment opportunity & achieved a **D grade (40-49%)**. The maximum grade awarded for resubmission will be **C-**.

Resits

Resits & reassessments **are not** applicable in **ID511001: Programming 2**.

Instructions

You will need to submit an app & documentation that meet the following requirements:

Functionality - Learning Outcomes 1 & 2 (50%)

Complete the two projects as outlined in the **classroom-tasks-unity.md** file.

Questions - Learning Outcomes 1 & 2 (50%)

- Provide answers to the following in your repository **README.md** file:
  - Describe some best practices for organising your Unity project.
  - Explain the concept of **GameObjects** & **Components**, & how they are related.
  - Explain the concept of **Prefabs**. What are they, & how do they contribute to the efficiency & reusability?
  - Explain the concept of game object hierarchy. How can you organise & structure game objects using parent-child relationships? Provide an example of a situation where using hierarchy can be beneficial.
  - In the game **Pong**, how would you implement the collision detection between the ball & the paddles using colliders?