COMP397 – Web Game Programming

Assignment 2 – 3D Mobile Game - Part 1

Due: Week #9 at midnight

Value 5%

3D Mobile Game **Maximum Mark: 100**

Overview: This is the first part of a three-part Assignment. Using the Unity Game Engine and the C# Programming Language you will work in a small group (alone or with a partner) to create a 3D Game with a **single polished level**. The game must also include a **Menu Screen, Options Screen,** and a **Game-Over Screen**.

For this first project your **target platform(s)** will be Android and/or iOS.

In this part, you will complete the following tasks

1. Convert your game from Assignment 1 (part 1, 2 and 3) to a **Mobile Game**.
2. **Touch** - create and / or acquire assets to enable mobile touch.
3. **Load a Saved Game** - Your Menu Screen should allow the player to Load a saved game
4. Include the **Factory Method Design Pattern** to create various configurable and reusable Game Objects.
5. Create an update of your Game Design Document (GDD) from Assignment 1 to include elements for mobile View Ports.
6. Update your GitHub repository for your game
7. Update your Project Management Tool to track your Game's progress.
8. Build and Test your Game on an Emulator (**LDPlayer** Recommended)

Part 1. Project Description:

You will continue to build one of three possible games Types for Mobile Platforms:

1. Platform Runner / Shooter
2. Isometric Tower Defense
3. Sandbox Crafting Survival Game

Please see the Design requirements from Assignment 1 – Part 1 if clarification is required.

Part 2. Assignment Deliverables:

Common Requirements / 60

1. This version of your application will have the following characteristics
    - **Mobile User Interface Assets** – Acquire (and or create) UI Elements to provide a virtual game controller and any buttons required for touch screens. These assets should be positioned on the screen inside of the "safe area"
    - **Install the Device Simulator in Unity** – to promote testing our game on Mobile Devices, enable experimental features in Unity to display the Device Simulator on the Game Screen.
    - **Enable Touch** – Write code or import appropriate assets to enable touch controls for your Game. You may also include code that enables a standard Game Controller to connect with your mobile device if required.
    - A **Menu Screen (**the Game Start State**)** – that will allow the user to get ready and displaying at least 4 options: **New Game**, **Load Game**, **Options** or **Exit.** When the **New Game option** is selected the game will transition to the Play Scene. When the **Load Game option** is selected the game will display a **UI panel** (or other UI elements) that will be used to load a saved game. This feature will be **completely functional** at this point. When the user selects the **Option button**, the game will transition to the **Options screen**. When the user selects the **Exit option**, the game Application will **Quit**.
    - Your game must include an **Inventory System**. This feature will remain a placeholder for this release and **will not** be functional.
    - Your game will include **enemies** with the AI behaviour detailed above (**sensing suite**, **pathfinding** and **steering behaviours**)**.** In this release the Enemies will still not include steering behaviours but rely on the **Sensing Suite** and **Pathfinding**.
    - Your Game Play Level must include a User Interface that allows the player to **Pause** the Game, **Save** the Game State and **Load** the game from a specific game state snapshot. The Pause / Save / Load panel must be **activatable** by a key press or other type of player input (e.g., Game controller key press). Pausing, Saving and Loading the game will be **fully operational** at this point. You should be able to "deserialize" the game state from a saved file so that the game will be restored to a previously saved state.
    - Include the **Factory Method Design Pattern** to spawn enemies, bullets, voxel tiles or other items in the game. The **Factory Method Design Pattern** is a creational design pattern that will streamline your code base. The **Factory Method Design Pattern** may optionally be created as a **Singleton**.

Game Design Document / 20

2. Include a fourth draft of the **Game Design Document (GDD)** for your game that includes:
    - **A Tile page with Company Logo, Game Name, Authors Name(s) and Student ID(s)**
    - **Table of Contents**

- **Version History – ensure you include an update here that the features you have enabled with your code**
- **MDA –** Update your Game's **Mechanics, Dynamics** and **Aesthetics** based on what you have learned during this development cycle.
- **Touch Controls**. Include how your game will be controlled using presses, taps, gestures and/or swipes
- **Level and UI Sketches** – update the **wireframes** of each of your game screens (Menu Screen, Options Screen, Game-Play Screen, Game Over Screen) to reflect how your game will appear on mobile View Ports. Ensure you take the "Safe Area" of your mobile device into account when developing these wireframes. Please use an appropriate tool to create these. Hand-drawn sketches **will not be accepted**. This is an important part of the planning process for your game.
- **Screen Captures** – Include at least **4 new screen shots** for your game to reflect the mobile View Port Size: 1 for your Start Screen, 1 for your Gameplay Screen, 1 for your Game-End Screen and 1 that captures your Health System, 1 that captures your Inventory System, 1 that captures your Pausing/Saving/Loading feature.

Internal Documentation / 5

3. Include **Internal Documentation** for your program
    - Ensure you include a program header for each module of your game that indicates: The Source file name, Author's name, Student Number, Date last Modified, Program description and Revision History
    - Ensure you include a header for all your functions and classes
    - Ensure your program uses contextual variable names that help make the program human-readable
    - Ensure you include inline comments that describe your code.

Version Control / 5

4. Share your files on **GitHub** to demonstrate Version Control Best Practices
    - Your repository must include **your code** and be well structured
    - Your repository must include **commits** that demonstrates the project being updated at different stages of development – each time a major change is implemented

Demo Video / 10

5. Create a **Short Video** presentation with your favourite screen capture and streaming tool (OBS Recommended) and upload it to your Learning Management System. You must also include a short PowerPoint (or Google Slides) Slide Deck that includes a **single slide** to start your video
    - The first (and only) Slide of your Slide Deck must include a **current image of you (and your partner)** (no avatars allowed) that is displayed appropriately on the page.

You must also include your Full Name(s), Student ID(s), the Course Code, Course Name, and your Assignment information.

- You will **demonstrate** your game's Screens on the Device Simulator. Ensure you include a simple mechanism to switch Screens. Your UI must be clearly visible
- You will **describe** the design for your Game
- Sound for your Video must at an appropriate level so that your voice may be clearly heard. Your Screen should be clearly visible
- Your Short Video should run no more than 5 minutes

Note: Your project will not be accepted without your video demo

SUBMITTING YOUR WORK

Your submission should include:

1. An external Game Design Document (MS Word or PDF). You should use the example document provided as a template. This will be your first draft.
2. A working link to your project files on GitHub. Ensure that the repo is appropriately named.
3. Your project files zipped and submitted to your Learning Management System. Rar files **will not** be accepted.
4. Your short Video Demo link uploaded to your Learning Management System.
5. Indicate in your submission which agile project management tool you will use to track your progress (e.g., Trello, Jira)
6. **Important:** Ensure your BUILD YOUR GAME for a Mobile Game platform on an Emulator such as **LDPlayer** or **Genymotion**.

This assignment is weighted **5%** of your total mark for this course.

Late submissions:

- 20% deducted for each additional day.

External code (e.g., from the internet or other sources) can be used for student submissions within the following parameters:

1. The code source (i.e., where you got the code and who wrote it) must be cited in your internal documentation.
2. It encompasses a maximum of 10% of your code (any more will be considered cheating).
3. You must understand any code you use and include documentation (comments) around the code that explains its function.
4. You must get written approval from me via email.