COMP397 – Web Game Programming

Assignment 2 – 3D Mobile Game - Part 3

Due: Week #13 at midnight

Value 20%

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

Overview: This is the third 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. **Quest System** – design and build a fully functional Quest System for your game
2. **In-Game Tutorial** – use the Quest System to build a tutorial for the user.
3. **Achievement System** – design and build a fully functional achievement system for your game.
4. **Implement the Observer Pattern** – this will enable both the Quest System and the Achievement System to function properly.
5. **Memory Profiling and Performance optimization** – use Unity's Profiler to improve mobile performance.
6. **Polish your Game** – add / remove assets as required to improve the visual appeal and the overall aesthetics of your game.
7. Create an update of your Game Design Document (GDD)
8. Update your GitHub repository for your game
9. Update your Project Management Tool to track your Game's progress.
10. Build and Test your Game on an Emulator (**LDPlayer** Recommended)

Part 1. Assignment Deliverables:

Common Requirements / 70

1. This version of your application will have the following characteristics

- Design and Build A **Quest System** for your Game. Acquire (or create) User Interface elements to show quest items to the user. These quest items may be check boxes in a UI panel that give the user directions to complete several tasks.
- Use the Quest System to build an **In-Game Tutorial** for the first part of your level. Direct the user to use their character controller appropriately (e.g., confirm that they know how to run, jump, duck, attack, pick up items etc.). Use the In-Game Tutorial to have the user perform common mechanics (e.g., perform a melee attack, ranged attack, collect a resource, etc.).
- **Achievement System** – create a simple achievement system that rewards the player when they complete various quest items or reach important game milestones. Acquire (or create) appropriate User Interface elements to display each achievement to the player.
- **Implement the Observer Design Pattern** – the Observer Pattern is a Behavioural Design pattern that creates Publisher and Subscribers and Observable objects. The Publisher notifies each Subscriber that an Observable object has changed in some way. This pattern ties into and is required for both the **Quest System** and the **Achievement System**.
- You must include **memory and performance profiling** in your game. You will use the Unity Profiler to generate a report for both Memory Allocation and CPU utilization. Your report should show before and after screenshots. Include this report as a separate PDF for your submission.
- **Polish your game** – Acquire (or create) additional assets as required to improve the visual appeal and overall aesthetics for your game. This may include additional sprites, animations, sounds and music. You may also use dynamic sound to respond to climactic events in your games, such as a boss battle or rare achievements.

Game Design Document / 10

2. Include a sixth and final 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.
    - **Screen Captures** – Include updated screen captures that include the Quest System and Achievement System for your game.

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 **20%** 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.