COMP397 – Web Game Programming

Assignment 2 – 3D Mobile Game - Part 2

Due: Week #11 (Friday April 02, 2021) @ midnight

Value 10%

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

Overview: This is the second 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. **Inventory System** – fully functional and connected to your Game Type Resource System
2. **Full Functioning Enemy AI** – all enemies will have access to their Sensing Suite, Pathfinding and Steering Behaviours.
3. Include the **Object Pooling Optimization Pattern** to create various pools of reusable game objects to reduce memory fragmentation and improve mobile performance.
4. Create an update of your Game Design Document (GDD)
5. Update your GitHub repository for your game
6. Update your Project Management Tool to track your Game's progress.
7. 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
   - Your game must include an **Inventory System**. This feature will be **fully functional** at this point. You will enable drag and drop functionality to work for this feature. A user should be able to select an Item from their inventory and drag it either into the game in real-time or into an **active menu** where it can be consumed in some way. The Inventory system will be available in different capacities depending on the selected Game Type:
     - **Platform Runner / Shooter** – the Inventory system will allow the player to store special items found in the level such as weapons, keys and / or ammunition.
     - **Isometric Tower Defense** - the Inventory system will allow the player to gather resources and place them into inventory for future use. This may include

environmental resources and resource points in the form of in-game currency items.

- **Sandbox Crafting Survival Game –** the Inventory system will allow the player to gather resources from the destructible voxel tiles in the game. The player will then be able to "Craft" simple structures and upgraded items by combining elements from their inventory in some way. This may include a requirement of a in-game crafting table, anvil or other contrivance.

- Your game will include **enemies** with the AI behaviour detailed above (**sensing suite**, **pathfinding** and **steering behaviours**)**.** In this release, **Enemy AI** will be fully functional.
- Include the **Object Pool Optimization Pattern** to create Bullet Managers, Enemy Managers, etc. Each **Object Pool** may optionally be created as a **Singleton**.

Game Design Document / 10

2. Include a fifth 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 Inventory 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 **10%** 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.