GAME1017 – Game Fundamentals II

Assignment 1

*** Check Brightspace for due date ***

Assignment: 'Completing' Top-Down Shooter Game

You can do this individually, or still in pairs for assignments only this semester.

The goal of this assignment is to expand a simple 2D shooter to include a singleton main controller, different scenes, UI buttons, and some additional gameplay. This assignment, worth 20% total, won't be broken down into different milestones but you'll have several headings going over each criterion due.

Source Project(s)

If you've taken GAME1007 when it changed to Unity, you can use your GAME1007 assignment. Your gameplay should be perfect as that will be part of the criteria. If you aren't confident in your GAME1007 assignment or don't have one, I will provide a minimal starter project that may even be missing some minor criteria. See the Gameplay criteria at the end of this document.

If you use my GAME1007 start project, you MUST replace the sprites and audio with your own to add your own flavor. If you do not replace the sprites, you will get 0. It will just take a little time to find them. I will post sprites sites on D2L.

Requirements

Here's a summary of the new requirements for this assignment. You'll need to:

- Create a main controller script that will be a singleton and hold other managers that will consist of the following:
    - A sound manager – and yes, it can be the one from the labs
    - A UI manager – to be explained later in this document
- Several scenes
    - Title scene, game scene, game over scene
- Gameplay additions

Main Controller Singleton

Okay bub here's the rub. The main/master controller singleton isn't going to do much besides be a parent for a couple of managers. It will, however, become a gateway to these managers, especially the sound manager that will still work like the version we created in the labs. Here's what accessing the sound manager should look like:
MainController.Instance.SoundManager.<any public functions>;

There should be two public properties to reference the SoundManager and a new UIManager and the Awake of the MainController should set these properties using GetComponentInChildren<>() as you should have GameObjects for these that are children of the MainController GameObject that should already be in the scene. That is, the title scene.

Also, make sure to set your MainController object to DontDestroyOnLoad so that it persists throughout all scenes.

SoundManager Criteria:

- All the functionality we've done so far, including that in Lab Exercise 1. The sliders will be part of a GameObject panel that will be controlled by the UIManager which will be explained shortly.
- For this assignment, there won't be any additions to this manager.

UIManager Criteria:

- The UIManager will be responsible for displaying and hiding an options panel that will pause the game when it is brought up, and for this assignment, only the sound sliders need to be part of this pane.
    - The pane should be a child of the canvas so that when you pause the game using Time.timeScale = 0f; then the UI elements will still work.
    - The amount of screen space the pane should cover will be up to you, but it shouldn't be too small, but it also shouldn't cover the entire Unity window.
    - I'd like the pane background to be translucent so I can see the paused game underneath.
- There should also be a persistent button that pauses and toggles the panel, as well as all the elements from the 1007 game, including score and some sort of health system.
- For this assignment, there will also be a high score requirement, and we'll go over some options when we go over saving and loading of data.

Scenes

I’ll briefly go over each scene and the requirements in point-form for each. If I don’t mention anything, then please assume that you can have freedom to make your own decision for it. The pause screen must work in all the following scenes:

A Title scene will:

- Be the initial scene.
- Display an image for the name of the game with size and name of your choice. Keep it profanity free. It can just be a bitmap made in Paint, but there will be aesthetic marks, so it must be decent and fit your game theme. Use your judgement to determine the size on screen.
- Contain a small button with the text ‘Start Game’ on it. When you click the button, the game will start.
- The button that brings up the pause menu will also be displayed as you can bring up the sound options.
- Yes, there should be a music track for the title scene.

The Game scene will:

- Run the main game.
- See the heading below for additions to the gameplay.
- The game scene should be paused when the pause menu is displayed as I mentioned above with the time scale.

The Game Over scene will:

- Be triggered when the player collides with any of the hazards of the game, including enemies and the enemy bullets.
- Display a button with ‘Main Menu’ text on it. When you click button, you’ll go back to the title scene.
- You can display anything else you want in this screen but keep it clean.

Gameplay Additions

For the gameplay additions, there will be the following.

- Health pickups. Upon destroying an enemy, they will have a low chance (10-20% at your discretion) of dropping a health pickup that will restore some health or a life, depending on your health system.
- Cougar Storm Counter-Swarm superweapon. Couscous for short.
    - Ammo for the Couscous superweapon is another pickup with the same chance to drop from enemies. What this will be is ultimately up to you, but I suggest a few options.
    - Option 1: A nearest-enemy seeking missile using a Seek algorithm that, upon hitting the enemy, has a wide explosion radius and if your enemies are close enough, it will destroy all enemies in that radius.

    - Option 2: When fired, a Couscous missile will fire towards the center of the screen (still using a Seek) and it will destroy all enemies currently on screen.
    - So long as multiple enemies can be destroyed and you use some type of Seek, you can implement Couscous as you like. Even something like a chain attack that starts at one enemy and another missile spreads out to the rest like a chain reaction. That'd be cool.
    - How you fire Couscous is up to you, but I suggest the biggest key on the keyboard and that is Space. Oh, and when Couscous is fired, you MUST play the sound of a roaring cat.
- High score. In addition to the normal score in the UI, you must keep track of a high score between games. We will start to go through some options, including the most ideal for this simple stat, in Week 3.

Extra Sounds (optional)

Sounds were part of the requirements for the GAME1007 assignment and you need those, but you can add any more that you like.

Misc. Notes

- You can work on this in pairs. Both partners MUST submit the same GitHub repo separately.
    - I should not have to say this, but if you work on it in pairs, both partners must do equal work.

Git/GitHub Requirement

You will be required to create a PRIVATE Git repo for this and use GitHub. Git will be required for both assignments. You must provide the instructor access and submit your GitHub link as your assignment submission on Blackboard.

You must also have at least six different commits all at different times.

A1 Marks: 20% of course grade

Task	Marks	Description
Main Controller / UI	4	You created a main controller GameObject and script properly – 1 The main controller starts in title scene and persists through all scenes - 1 You added a SoundManager and can access it properly -1 You added a UIManager that works properly – 1
Scenes	4	You created all the requirements for the title scene properly – 1 You created all the requirements for the game scene properly – 1 You created all the requirements for the game over scene properly – 1 The pause screen with sound options works in every scene - 1
Gameplay	4	All these requirements from your GAME1007 project are working: You have sprites for scrolling bg, player, player bullet and enemies – 0.25 The collisions between player and enemies and enemy bullets works – 0.25 All your game objects properly de-spawn when going off-screen – 0.25 You added sfx for enemy bullets and all collisions – 0.25 The health and superweapon pickups work properly – 1 The superweapon destroys multiple enemies and plays the proper sound – 1 The high score UI element works properly between games - 1
Aesthetics & Polish	4	Your game looks clean with proper transparency on sprites if you have it – 1 Your game plays smoothly and no game objects are too fast or slow - 1 Your game doesn't throw any runtime errors – 1 Sounds play at a proper volume, including the music – 1
GitHub & Commits	4	You have submitted a proper working GitHub link with proper name – 1 You have at least six commits at different times – 3
Total:	**20**	

Assignment 1 Submission:

You MUST use GitHub and do not submit any files on Blackboard. Add arichard6@georgebrown.ca as a collaborator.

Name the repo: **GAME1017_A1_LastnameFirstname**

or if working in pairs: **GAME1017_A1_Lastname1Lastname2**

There is no video submission requirement this time.

Penalties:

- Late penalty: 10% per day late up to 50%, then not accepted at all
- You use any of my assets (sprites or audio), and have not changed the ones from the start project: 0%
- Your game doesn't compile and run after downloading your repo: 0%
- Your game uses code that is too different from what we did in 1007 and in 1017, in other words it is suspiciously done by someone else: 0%
- Your game is not done in C# and Unity and the above applies: 0%
- You have just submitted what we did in a lab and/or from a start project without adding anything of your own: 0% (yes, I have to specify this!)

- Potentially more as I think of them. Check back frequently. <3 penalties.