Lebanese American University
School of Arts and Sciences
Department of Computer Science and Mathematics

CSC458– Game Programming - Project 1 (a group project)

Date assigned: Monday 17th of October 2022 – 11:00 PM
Date Due: Monday 7th of November 2022 at 11:00 PM.

Objectives:

To test students' knowledge of the following Unity topics:

Movement and rotation via any method[1]	Using the Terrain tool in Unity/ using Textures
Collisions and Triggers.	Populating the game with beautiful assets from online stores such as the Unity Asset Store or other places + correct citations
Instantiating and Destroying game objects.	Particle Systems
UI in Unity	Animations
Serialization of variables & classes.	Scene management
Camera follow - 1st person perspective[2]	Coroutines
Sounds Effects (SFX), Music	Raycasting
User Input from the Keyboard and the Mouse.	Game Design Document & Technical Design Document

The game submitted must be a 3D game using the Unity 3D template.

General Rules

- Late submissions are not allowed. This is **a group assessment**. Maximum number of students forming a group is **3**. You can do the project alone or with 1 or 2 students if you like. In all cases, whether alone or with other students, you must fill in your full name and the full names of your team group members in the Excel sheet shared and you must reserve a group number.
- **Cheating or copying** other students' work will get all the involved students a zero grade. I do not care if someone copied your work or who copied from whom. The project assignment should be unique. It is statistically impossible that two groups in the class would end up doing the same project or have the same or similar code.
- **Any form of plagiarism or academic misconduct is prohibited. I have a zero-tolerance policy concerning this**. Copying code found elsewhere or changing it a bit and then claiming it to be yours will get you a zero grade immediately. **Using anything which you did not create yourself, MUST be cited and/or credited adequately** in a separate MS Document which you include in your submitted Unity project zip file. Using a game asset, any form of code (small or big in size), pseudocode, ideas, techniques, from any public or private source, online or in any other form, from a video on YouTube or from a Udemy/Coursera/Skillshare course or similar platforms or from a book or article etc… without citing the

[1] Either via the transform component, Physics Rigidbody component, character controller component or Navigation Mesh Agent or using ready-made standard character controller such as Unity Standard Asset: First Person Character Controller

[2] Using any technique, you find suitable (Cinemachine, Constraint components, offset technique, via C# code etc…) or ready-made standard character controller such as **Unity Standard Asset: First Person Character Controller**.

exact source adequately in your accompanying report **will get a zero on project 1 for all group members.** This is literally the definition of plagiarism. **Make sure your citations are not vague** such as saying "Unity Forum" or Unity Scripting API" or saying: "I did a Google Search and found X". If you do that you will get a zero. By definition, a citation must be specific and precise.

- The **Unity code/project submitted should not appear elsewhere**, **nor should be a previous submission to a different or to the same course in any previous semester or in any other university or academic institution,** nor should be a project done elsewhere in the industry, nor done for a client, nor made before for any other reasons that I did not state. If we discover this is the case, you will get a zero grade immediately and an expulsion from the entire course on first offence.
- Please refer to the student code of conduct in the syllabus.

What do I need to submit?

Each group member MUST submit the following: (please include the group number in all the MS Word documents):

1. **Citations and Contributions (as an MS Word document)**: In this document, you must include all the references of anything & everything that is not your own creation. There must be a contributions section in this document, which should be written by each group member. In this section, you must state exactly what is your contribution and that of each group member. Make sure that the contributions are relatively equal since this will affect your grade if you do not contribute much.
2. A **Game Design Document (as MS Word Document)**: must be submitted. It is a document that explains game mechanics, levels, genre, theme and other characteristics. It should not be more than 2 pages.
3. A **Technical Design Document (as MS Word Document)** containing flowcharts or sequence diagrams and important class diagrams with explanation of the most pertinent code.
4. **A Unity project with all essential files & folders (Assets, ProjectSettings & Packages). You can delete the "Library", "UserSettings", "Logs", "Obj" folders**. Please only keep these 3 important folders (Assets, ProjectSettings & Packages) after you finish completely your Unity game. Unity engine should be closed when you delete these folders. Why? these deleted folders i.e. "Library", "UserSettings, "Logs", "Obj" are auto generated by the Unity Editor when you open the project. This is in order to make sure that the folder size of your Unity project remains small. Always make sure you backup your game on a regular basis.

Put all the above in one single folder and create a zip file with **your full name the way it appears on Banner** (FirstName-LastName.zip) and upload it to LAU blackboard. If you are doing the project alone: you still need to submit all the documents & the Unity project files as detailed in the points above but there no need of course to include the contributions section in the Citations MS Word Document since you are doing the project alone. In this case, just include the citations in the citations MS Word document.

NB: If project 1 final file size is still large (after you have removed the Library, UserSettings, Logs, obj folders) & you cannot upload it to BB, please upload it to OneDrive or Google Drive or similar services and share the link only with the lecturer and/or with the TA before the deadline. The **link should remain active and accessible** till the end of the course with no changes in the files nor in their metadata. If we discover any change even a minor change, all the group members will have zeros. Any change in the project after submission is a serious academic offence and will have disciplinary actions especially if done to fool the lecturer or the TAs in the demos and especially when the lecturer and the TA were not notified of the changes made. Don't leave submission till the last few hours. We live in an unstable country with an unstable internet and electricity so ideally you should finish early if you have good time management. You have a lot of time to finish this project so use the time very wisely.

All group members MUST demo their project in front of the lecturer or the TA or both otherwise you will get a zero grade as I have to check that it is actually your own work not someone else work. If you do not understand your code or some feature in your game, this means that you did not write it or this could mean that you did not do the project or part of it. Don't let me think this way. You can learn how to do things from the web. Nobody reinvents the wheel but do not use something in your project that you do not understand. There are no pedagogical benefits in doing this anyway. If you do this, it will negatively affect your grade. **Penalty of not understanding code could reach deduction of 45% of the total grade**. Bear in mind that a true creator always knows well his/her own creation.

The **project is an opportunity for you to learn new things so please impress me**! Make sure you do a good project so you can impress **prospective employers** and so that you can put it in **your GitHub/bitbucket profile** or **itch.io portfolio**.

Please read the requirements slowly and carefully. Your project 1 MUST adhere to the following requirements to get the full grade – each point is assigned a certain weight of the total grade. **NB:** I did not specify dimensions nor quantities of ammo or similar concepts when I mentioned requirement concerning game features and things like that. Please use what you find convenient in terms of quantities and dimensions.

Mandatory Requirements

- **Game Genre**: the genre for your game will be **a First-Person Shooter**. This is a very classical genre with plenty of material available online. We covered and we will cover topics pertaining to this genre. Please pay attention and make sure you do not commit academic misconduct or plagiarism. No need to plagiarize and thus take zero.
- **Game Theme**: you have the freedom to pick a theme from a list of themes that I will give to the class. The theme will govern literally the assets chosen in your game world. Example: choosing the Medieval Europe theme means that maybe you will use a bow and arrow as a shooting mechanism and the environment could a castle in a forest or a medieval town etc… The theme will open new ideas, elements, and gameplay mechanics. Themes could be Zombie apocalyptic, medieval Europe, modern, Word War 2 or World War 1, Alternate Reality, futuristic, sci-fi, fantasy, Ancient Arabia, Ancient Babylon, Ancient Egypt, Roman Empire, outer space, cyberpunk, Aztec culture (forests & Pyramids) etc... Please check the full list of themes shared on BB. I need to see different themes chosen by different groups (no repetitions allowed) so please include the theme in the shared Excel Sheet and avoid repetitions as much as possible. I will provide you with a list of themes to choose from.
- **Game Level(s)/ Level Design**: The game will have only **one complete meaningful level** in project 1. Make sure you follow good practices in **game level design**. Make sure you choose your assets separately in a different project since some of them might contain scripts that are buggy, or they might have problems with shaders and other stuff. Advice: Do what is known as whiteboxing/greyboxing initially (either on paper or inside Unity itself) – this would be a brainstorming session with the group - then program the components then start populating your actual Unity project with heavy assets after your test them. Advice2: Only include the assets you are indeed using to minimize the file size of the submission. You can add more meaningful complete levels to get bonus grades. Please check the Bonus Grades section.
- **Environment/ Assets**: The map of the world should be created using **the terrain tool in Unity**. You should change the terrain and make it beautiful. You can also include buildings, mountains, forests, vegetation, ponds etc... The map of the world should contain monuments and features that are pertinent to your theme. Esthetically pleasing terrains with different terrain features or painted vegetation or using different textures (desert, Rocky Mountains etc…) will get the full grade on this requirement. Please make sure you decrease the size of the Terrain the way we did in the lecture and please make an environment that is suitable for your game. Your game must be beautiful but not necessarily big in size (remember

Simplicity is beauty (a saying by Plato, Leonardo Da Vinci & many others). Actually having big terrains need a lot of draw calls and will make your FPS very low and thus your game becomes unresponsive especially if you have a slow laptop. Models and other assets can be acquired from free resources online as long as you mention their sources.

- **Pickups/Obstacles**: The game world map will feature several elements such as ammo/manna/potions etc... as pickups. It should feature monuments, obstacles and others structures according to the theme chosen. Esthetically pleasing & functioning pickups will get the full grade of this requirement.
- **Movement/Jumping/Rotation of the Main Character**: Movement and rotation of your main character should be in the **First-person perspective only**. You can use any movement/rotation method that you want including using components such as transform, rigidbody or character controller. You are even allowed to use readymade first person character controllers found online such as the one that is available in Unity Asset store: Starter Assets - First Person Character Controller (Unity Technologies). Input should be only via Keyboard and mouse. The camera the player should follow smoothly and in a natural way.
- **Collisions & Triggers**: Your game must contain good amount of collision logic and triggers. Triggers can be used to set hidden game objects which when you pass though trigger something happens. Examples: turn on/off lights or set an explosion or sound an alarm. It is up to your imagination!
- **Instantiation/Destruction of GameObjects**: You need to instantiate and destroy objects in your game programmatically. This could be things you shoot at in order to blow them and make them disappear or this could be enemies you kill or anything else.
- **Sound Effects (SFX)**: When you shoot, there should be a small SFX (sound effect) played according to your theme. You might have SFX for other stuff. You enter a room and you might hear an explosion etc.. You might be shooting arrows, spears, bullets, lazer, Dartguns, poison darts etc… Grab some good SFX from online stores and use them (after citation). I need to see you control SFX via C# code. You can also have a background music. Make sure that any music is not noisy or distracting. Always cite.
- **UI**: You need to have an **intuitive player-friendly UI** that shows the score, health bar, ammo, manna or anything else depending on your chosen theme. **You must have more than 1 weapon** and **be able to change the weapons easily via certain keyboard keys or a mouse scroll**. Each weapon must have a type of ammo and an initial quantity of ammo. Ammo boxes of each type of weapon should be scattered in the environment. You player can pick these pickup boxes and thus increase the ammo count of the corresponding weapon. Hint: Check Lectures & check Brakeys Health bar YouTube video that explains how to create a health bar. Change it a bit to suit your own theme. This would flex your muscles in UI design in Unity.
- **Raycasting**: I need to see you use **Raycasting** to implement **an effective shooting mechanism**. You are allowed to use **Raycasting** for other tasks as well (seeing obstacles as an example), but the shooting mechanic should use raycasting (not prefab shooting or other methods).
- **Enemies/NPC/Animations on Enemis**: You need to create enemies of some sort depending on the theme you choose. Suppose if the theme is apocalyptic, you can have your enemies look like zombies that will attack you when you are in their attack range and when you shoot at them directly. Have a simple AI no need for anything complicated. We will cover AI later in the course. **Enemies must decrease your health when they attack you**. Add an attack animation to your zombies so that the attack looks realistic.
- **Killing Enemies, Health Changes, UI changes**: Killing an enemy increases the score. When your health reaches 0 you die. On death, a UI menu should appear asking you whether to play again or to quit. These should be as buttons. Pressing the UI button "Play Again" reloads your scene, pressing the "Quit" quits the whole game. If you kill all your enemies a similar UI menu should appear which should congratulate you, show you your score and ask you whether you want to play again or quit. You add more UI if you like.

- **Particle Systems**: Your game should contain particle system(s) in a meaningful way. Ideas: You can create a nice small effect particle system that appear briefly when bullets or laser hit an object at the location of the hit or create a particle system for explosions or a water fountain or anything really. Please I need to see that you are controlling the particle system via C# code in order to get the full grade of this requirement.
- **Coroutines**: I need to see that you have used coroutines at least once in meaningful way to get the full grade of this requirement. Refer to the lecture on Coroutines or external sources that they are used to do tasks that spread over time or across different frames such as dimming slowly a light or fading slowly a color or moving an enemy slowly. There are many places where you can use these in your FPS game.
- **Serialized Fields**: I need to see a good number of serialized variables so that your game can be tweaked from the Unity Editor (via inspectors) by other game designers or developers.
- **Coding Style:** Good coding style/Good usage of comments/ DRY (Do not Repeat Yourself) code/Good Overall program hygiene. Also you should follow established OO Principles such as SOLID and GRASP and show me that you did that in your code. You should not graduate any CS degree these days without knowing these OO principles since they are asked frequently in all prestigious Software/Web/Game/Mobile development jobs' interviews. Kindly add to the serialized variables in the inspector some tooltips (recall the attribute Tooltip), Headers (recall attribute Header), and other clarifying attributes to explain briefly what these variables change so that the TA or Lecturer would understand what your serialized fields do in the Inspector.

Remember a project such as this aims to teach you things that might not yet be covered in class. Every descent assignment or project in any descent university that is worth its salt involves **a research component on your part**. You should develop the skills of looking for solutions online for your encountered development problems. You should develop the skills to do research on how to implement a certain feature. Although we will try to cover most of topics, we might not cover everything.

Please build your game as executable if you have time (optional) in order to speed up the demos: check the lecture where we have explained how to do that in detail.

Bonuses: You can still get a 100 without doing any of the bonuses but they help in case you missed a mandatory requirement.

2 or more Design Patterns

5% - if you use more than 2 design patterns whenever needed. You will learn few **design patterns for games** later.

Many Levels:

- 5% - Adding 2 or more complete meaningful levels (meaningful to the theme chosen). Here you will utilize your knowledge of scene management API which was covered in class.
- 5% - Adding background music that last through more than 1 level. There are many techniques in Unity to achieve this. One classical technique is to use the **Singleton Design Pattern** and the method **DontDestroyOnLoad**. There are other techniques of course. Please always cite what you learn online. We will cover them in class later anyway.

Developing your own assets:

Developing own assets (models, animations, textures/sprites, sounds) in Blender, Maya, Photoshop, GIMP, Audacity or any other external tool and using them in your game will get you a 5% bonus.

Good Luck!