**TEMASEK POLYTECHNIC**

**SCHOOL OF INFORMATICS & IT**

**DIPLOMA IN GAME DESIGN AND DEVELOPMENT**

**ASSIGNMENT BRIEF FOR PROGRAMMING WITH GAME ENGINES (CGE2C19)**

# Assignment 1: Practical Assignment - Source Codes (30%)

**Submission Date: 30 November 2023 - 2100 hrs.**

Download the Unity project from your LMS. The package comprises the assets and the base code required to start your assignment. This project consists of what you have done as class worksheets from Week 1 to Week 5. Open the project in Unity, click Play and see the behaviour. Get yourself familiarised with the project content.

Your tasks for this assignment are as follows:

## Q1: Camera Repositioning due to Object(s) Within Line of Sight (10 marks)

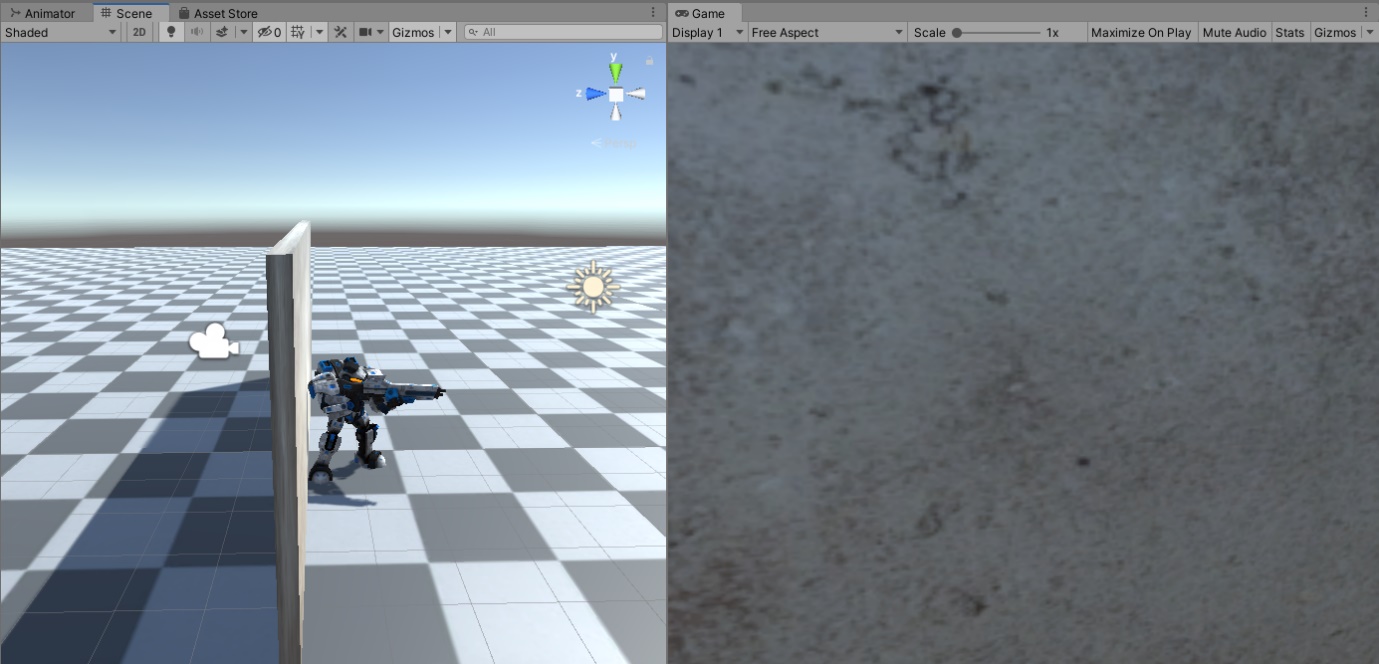
This programming task will assess your general game programming skills. For this question, you will write codes to ensure that other objects within the scene that fall between the camera and the Player do not block the Third-Person camera.

You will implement your C# code in the function **RepositionCamera**. I have provided hints within the function as comments. The idea is to check every frame to see if any opaque objects occlude the camera. You can achieve this by:

1. creating a ray that connects the camera position to the player position,
2. doing an intersection between this ray and the game objects in the scene, and
3. setting the camera's position to that intersected point (add a slight offset as well) to reposition the camera.

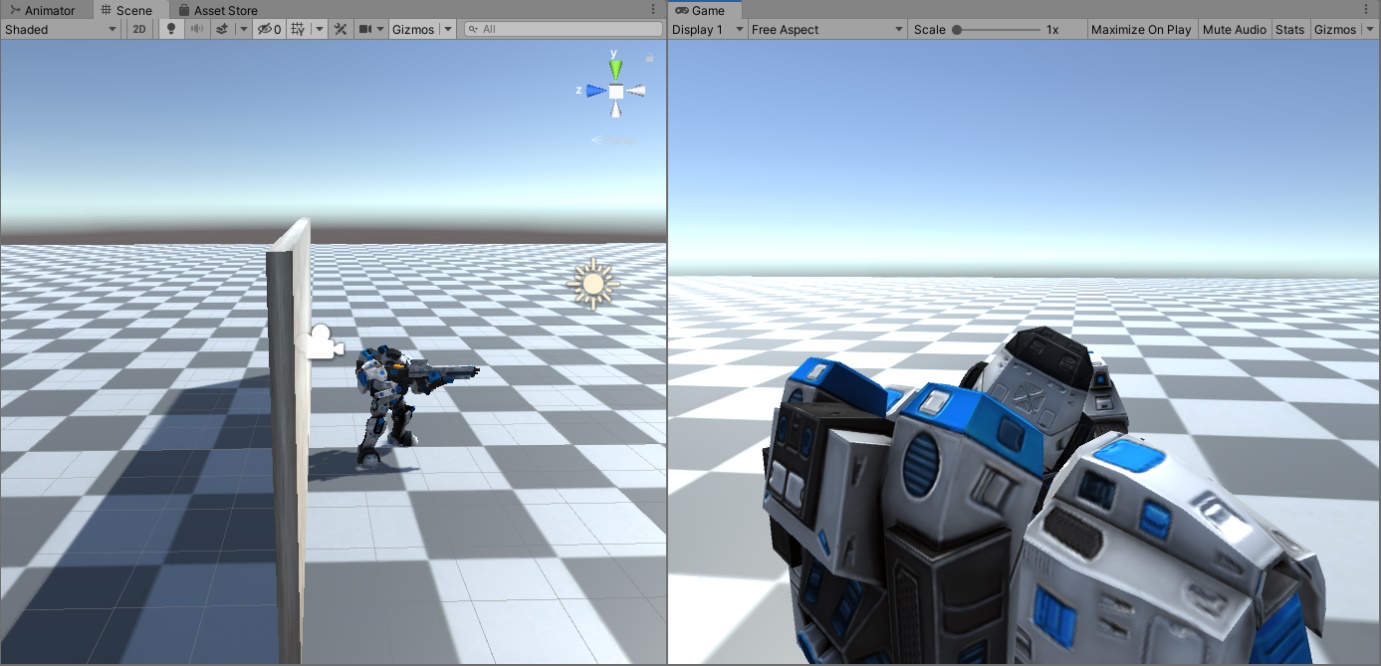
**References and hints on implementation**

The image below shows the camera's behaviour without RepositionCamera due to objects within line of sight. The left side of the picture shows the **Scene** view, and the right side shows the **Game** view. You can see the camera behind the wall from the scene view in the image below. Consequently, the camera is getting blocked by the wall in the game view.



**Figure 1:** Without **RepositionCamera** implementation.

The figure below shows the behaviour with a correct **RepositionCamera** implementation. You can see that the camera is in front of the wall this time. The view from the camera does not get blocked by the wall.



**Figure 2:** With a correct **RepositionCamera** implementation.

You can also view the videos with **RepositionCamera** and without **RepositionCamera** implementations from the links below.

**Third-Person Camera without RepositionCamera implemented:**

<https://youtu.be/3ZNsDJW637M>

**Third-Person Camera with RepositionCamera correctly implemented:** <https://youtu.be/WCBpk3LpZgM>

Do read about the following for your implementation:

* **LayerMask**: <https://docs.unity3d.com/ScriptReference/LayerMask.html>
* **Physics.SphereCast**: <https://docs.unity3d.com/ScriptReference/Physics.SphereCast.html>
* **Physics.CapsuleCast:** <https://docs.unity3d.com/ScriptReference/Physics.CapsuleCast.html>
* **Physics.RayCast:** <https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>

Write a brief reflection on your learning experience on this topic. Take a video that shows the camera working with one wall, multiple walls, a narrow corridor, a roof etc.

## Q2: Configure a new Character for the Player (10 marks)

This assignment will configure a new Player and replace it with the current SciFi Player in the given Unity project.

To configure your new Player, you will have to find a 3D character from Asset Store that comes with at least Idle, Walk (or/and Run), Attack and Reload (or recharge) animations.

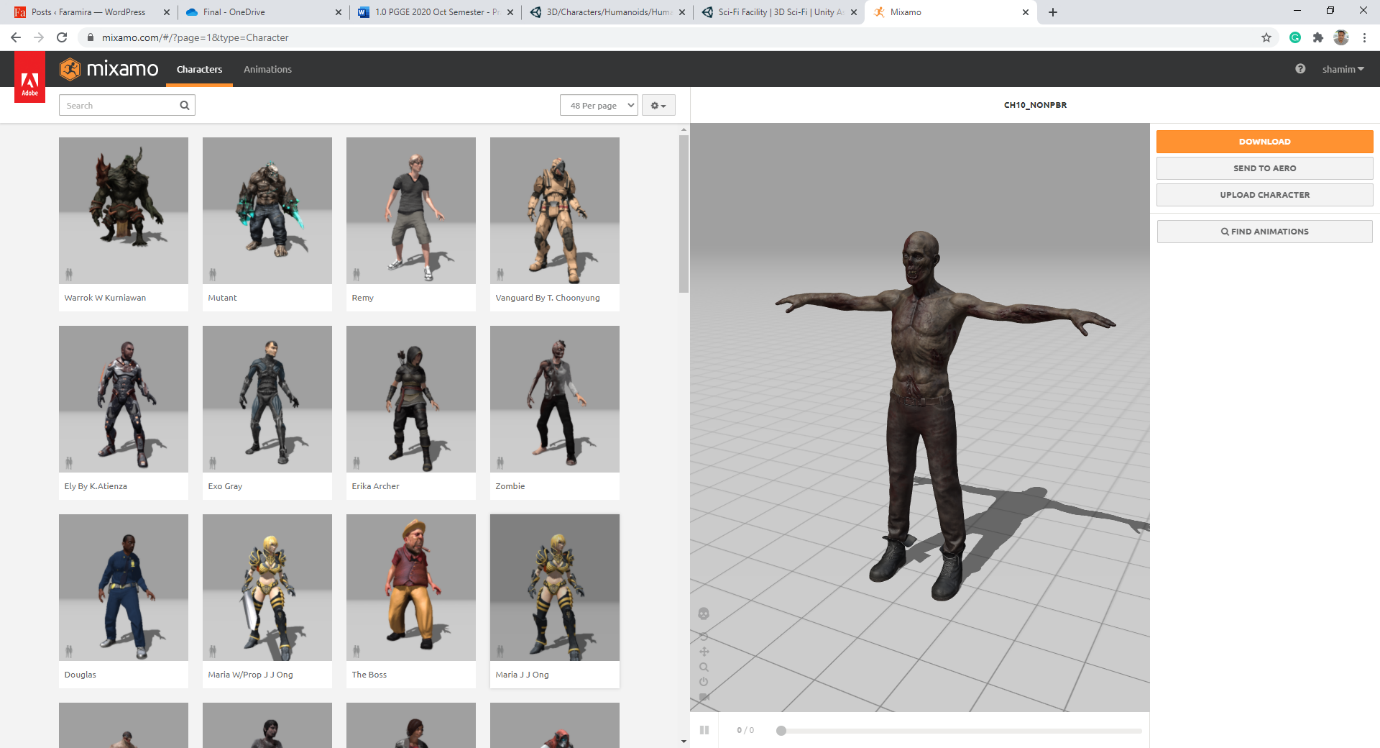
There are several freely available characters from the Unity Asset Store.

It is not necessary to choose a character with a gun. You can download any character that has an attack animation, could be other forms of attacks like sword attack, kick fight etc.

Some notable ones are:

* <https://assetstore.unity.com/packages/3d/characters/humanoids/fantasy-monster-skeleton-35635>
* <https://assetstore.unity.com/packages/3d/characters/humanoids/fantasy/oriental-fantasy-character-berserker-grade-0-171147>
* <https://assetstore.unity.com/packages/3d/characters/unity-chan-model-18705>

Alternately, you can create a free account in Mixamo (<https://www.mixamo.com/#/>) and download a character and the necessary animations and import them in Unity.



Follow the tutorials from <https://www.youtube.com/watch?v=xxM6D6coDAU> to learn how to import from Mixamo to Unity.

Once you have imported the character and the animations, you will need to create the animation controller in Unity's Animator editor. You can follow the steps that you have done to make the SciFi player in your Week 2 worksheet. For smooth movement, you will need to have a Blend state. Again, you can follow the steps that we took to implement the SciFi player.

After that, you will need to replace the SciFi character with this new Player and integrate smoothly the

* Player movement,
* The Reload/Recharge, and
* Attack animations

Remember to attach the third-person camera to this new Player.

Write a brief reflection on your learning experience on this topic. Take a video that shows the new Player walking, running, attacking, reloading/recharging and other animations. Provide clear instructions on how to run the application.

## Q3: Implement Steps Sound (10 marks)

In this assignment, you will program the walking and running sounds of a Player. You have learned how to add **AudioSource** and **AudioClips** to generate sounds in your game.

You will now use this learning experience to implement the **Player's** step sounds. I have given a sample video here below that implements step sound.

[](https://www.youtube.com/embed/ETJvfidZGXs?feature=oembed)

There are many ways to implement walking and running sounds. One easy way to implement will be to play a looping sound whenever the Player is walking or running. However, implementing this way will only give you a passing grade. To score a good grade, you will have to implement better quality walking and running sounds.

I have provided sounds for walking on various types of ground. You will need to use a number of these sounds to have a better effect. You might also want to randomise a list of sounds with different volumes and pitch settings to give a more realistic sound effect.

I will leave it to your judgement and creativity.

Your submission should include the source codes, the technical writeup and a video demonstration of the sound effect by making the Player move around the scene. To record your video, you may use the new Player that you created for Question 2.

## Penalty on Late Submission, Incomplete Submission & Plagiarism

All late submissions will be penalised following the three levels of lateness as stated below. Note that plagiarism is a severe academic offence ([See TP Plagiarism Policy](http://www.tp.edu.sg/staticfiles/TP/files/studentportal/Plagiarism%20Policy.pdf)). All submitted works should be genuine and originated from you.

TP Plagiarism Policy Source:

<http://www.tp.edu.sg/staticfiles/TP/files/studentportal/Plagiarism%20Policy.pdf>

|  |  |
| --- | --- |
| Late < 1 day | 10% deduction from absolute mark. |
| Late >= 1 and < 2 days | 20% deduction from absolute mark. |
| Late >= 2 days | No marks will be awarded. |

#End of Document