Technical Report for Assignment 1

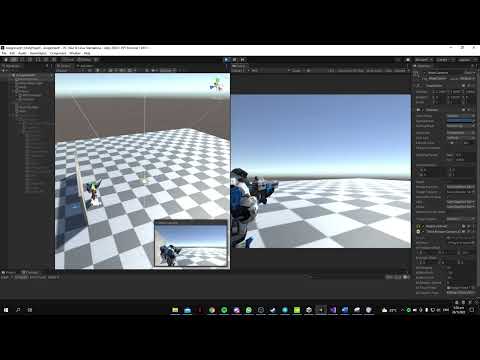
**Practical Assignment - Source Codes**

|  |  |
| --- | --- |
| Name | Lee Chow-Yi |
| ID | 2002697G |
| Class | P03 |

# Github link

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

## Video of your solution

[](https://www.youtube.com/embed/NDTPB5h0tP0?feature=oembed)

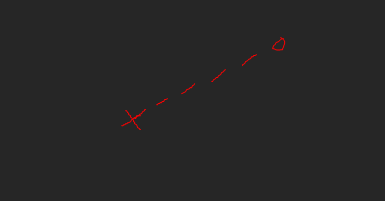
## Describe how you implemented your solution

Vector3 dollyDir = mCameraTransform.localPosition.normalized;

^ This is used to find the direction the camera is facing, however it’s normalized in order for calculations later on.

Vector3 desiredCameraPos = mCameraTransform.TransformPoint(dollyDir \* maxDistance);

^ The dollyDir is multiplied by maxDistance in order to get the desired camera position



^x is camera position, o is desired position

RaycastHit hit;

if (Physics.Linecast(mCameraTransform.transform.position, desiredCameraPos, out hit))

{

//MathF.Clamp() used to ensure that camera will never go past maxDistance or minDistance

distance = Mathf.Clamp(hit.distance, minDistance, maxDistance);

}

else

{

distance = maxDistance;

}

^Using raycasts to determine if an object is in the way of the camera.

//lerp used to make sure shift is smooth, same with deltatime

//deltatime is multiplied by 4 to ensure that the camera wil not phase through objects

mCameraTransform.localPosition = Vector3.Lerp(mCameraTransform.localPosition, dollyDir \* distance, Time.deltaTime \* 4);

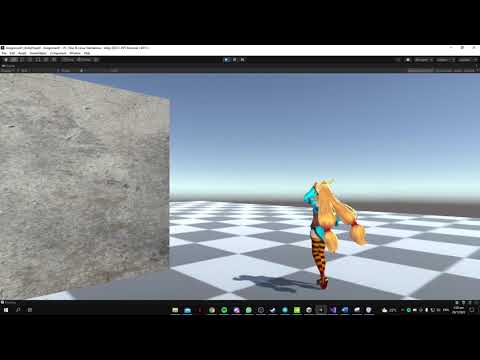
^Updating the camera transform

## Reflect this learning experience

I have learnt more about raycasting and the importance of understanding theories behind why certain things work in certain ways. To help with this assignment, I watched a few videos that helped me better understand how raycasting works and different ways it can be implemented, which helped me decide on what to do and how to do it for this assignment. Overall it was still quite a pain to implement as there was still clipping between the player and the wall which needed some time to debug as I did not really know where to start.

# Q2: Configure a new Character for the Player

## Video of your solution

[](https://www.youtube.com/embed/vvd87PoChfs?feature=oembed)

## Describe how you implemented your solution

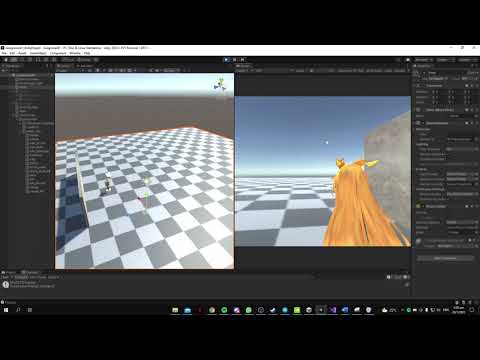
Originally, I made a character controller script from scratch to fit more animations. However due to time constraints (due to my bad planning), I ended up repurposing the original player scripts. Firstly, I added animations to a new blend state tree with parameters posX and posY which would update when the player moves, causing the animations to update as well. I used one of the default poses as an idle animation. Next, I added a new state in the base layer for jumping. I also added a trigger parameter for the jump. When the spacebar is pressed, the jump parameter would be triggered, allowing for a transition to the jumping state, therefore playing the jumping animation. Then, I added a new state for Attack1. I also added a Boolean parameter for the attack called Attack1. When mouse1 is pressed, the attack animation would then play.

## Reflect this learning experience

In this section I feel that I have learnt a lot more about animators and how to navigate and control different animations. However, I am aware that I still only have a surface level understanding of how to use animations in Unity and therefore I will have to work harder to learn and make up for mistakes made during this project.

# Q3: Implement Step Sounds

## Video of your solution

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

## Describe how you implemented your solution

First, referring to a video made by Jason Weimann on YouTube, I added events to the walking forwards and backwards animations, syncing them to the footsteps. These events would call a “Step” function. Then, I created a script to store the audio clips in arrays according to where they were (seperate arrays for clips for concrete, dirt etc). Using raycasts pointed downwards, I then proceded to get the layer values for any hits. If the raycast is 9 for example, that would mean the floor is made of concrete, therefore the audio clips for concrete would play. Along with this, I also added a randomizer for not only which of the audio clips would play, I also added the randomizer for the loudness of the clips.

## Reflect this learning experience

I really enjoyed doing this question as not only was it somewhat more simplistic than the questions before, I also thought that this was very informative for me as this method of using events tied to timings for animations could be used to create very intricate set pieces for example during a cutscene or during a chain of events which can play out, such as triggering a certain particle effect to play while a door is being opened, or healing the player if he was walking into a hospital (This would delay the time wasted by the player for waiting to be healed as we would not need to bring the player into the hospital and load new scenes etc, and just heal the player on the spot).