

# 3<sup>rd</sup> Person Camera Controller script

by

Joe Censored Games LLC

This script is intended to rotate a camera around a player, such as a player vehicle. The player then holds a mouse button and moves the mouse to rotate the camera around the player, and uses the mouse scroll wheel or similar functionality to zoom in and out.

If you need help with this script, you can contact me on the Unity forums via PM at Joe-Censored. I'm on there all the time, so it is likely the fastest way. You can also email me at [jocensoredgames@gmail.com](mailto:jocensoredgames@gmail.com)

## Instructions:

Attach the JCGCameraControl.cs script to your camera. Set Target to the player's gameobject, or whatever object you'd like the camera to follow. You're done!

Press play and it should be working. Use the scroll wheel to zoom in and out, and by default press the right mouse button and move the mouse to rotate the camera around the player gameobject.

**EnableControls** – When set to true it enables rotation and zoom functions, set to false to disable those controls. There's certain times when you may want to disable these controls, such as when interacting with UI windows.

**Target** – You have to set this for the script to work. Set this to the player gameobject, or whatever object you'd like the camera to center on. If you would like the player to appear lower in the screen, rather than in the center, consider placing an empty gameobject above the player and setting that object as the Target.

**MaxDistance** – The maximum distance the player will be allowed to zoom out.

**MinDistance** – The Minimum distance the player will be allowed to zoom in.

**StartDistance** – The distance the camera will start from the Target.

**RotationSpeed** – Higher numbers mean faster camera rotation, lower numbers slow down rotation.

**ScrollSpeed** – Higher numbers increase the speed at which the camera zooms in and out, lower numbers reduce this speed.

**MaxAngle** – Controls the maximum angle that the camera can rotate above the Target. You shouldn't set this above 89.

**MinAngle** – Controls the lowest angle the camera can rotate below the Target. For games where the player is moving on something like a terrain, such as a character that walks around, you will generally want to keep this setting as a positive number, such as 10 or so. For games without that, like a space game or a flying game, you can go into negative numbers so the camera can rotate below the object and look up to it. You shouldn't set this below -89.

**MouseButton** – Set this to the mouse button number you would like to enable camera rotation when pressed. By default this is set to button 1, which typically is the right mouse button. On most machines button 0 is the left mouse button and button 2 is the middle mouse button.

**ReverseScrollDirection** – Enabling this will reverse the direction of zoom.

**AlwaysRotateWithMouseMovement** – Enabling this will no longer require a mouse button to be held down to rotate the camera. The camera will rotate with all mouse movements without pressing a mouse button. This setting still respects the EnableControls bool.

**UseLateUpdate** – Enabling this will move the camera during LateUpdate instead of Update (original version of this script moved the camera during Update). Set to false to move the camera during Update. The benefit of moving during LateUpdate is you can move the target during Update and ensure the camera is properly positioned to center on it when the camera moves afterward in LateUpdate.

**UseRotationMultiplier1 & 2** – To allow customization of the rate of rotation in an options menu, you can now set static public floats RotationMultiplier1 & 2 in your own scripts. Since these are static they apply to all instances of the JCGCameraControl script. Setting a value of 1f will result in no rotation speed change. Higher values increase rotation rate, lower values decrease. Set UseRotationMultiplier1 or 2 to true to apply that rotation multiplier to this instance of the script. These are intended for use where you have multiple instances of this script in your game, and you want to use different rotation rates and especially when you want them customizable. If you just have a single instance, you can just set RotationSpeed instead of using this feature. Enabling both 1 and 2 will just result in RotationMultiplier1 being applied.

**DisableLogging** – Enabling this disables all log messages from this script. If the target is not provided then the script will write a log message to warn you of the issue, but there may be instances where you intentionally are not providing a target. If so, enable DisableLogging. When you don't provide a target of course this script will not be able to position the camera.

**CallAfterPositionUpdate** – This is not an inspector option. This is a delegate you can set to a method you want called after the camera is repositioned each frame. For example, if you have UI elements which hover over other characters in the game, those generally need to be positioned after the camera is moved. A simple example of how you set the delegate is below, followed further down by how you'd declare the method being called. Research how delegates work in C# for more information.

```
JCGCameraControl jcgcc = Camera.main.gameObject.GetComponent<JCGCameraControl>();  
if (jcgcc != null)  
{  
    jcgcc.CallAfterPositionUpdate = AfterCameraPositionUpdate;  
}
```

```
//Example method to call after camera position update  
public void AfterCameraPositionUpdate()  
{  
    //Move your UI elements after the camera has moved here  
}
```