Camera Script Instruction Manual

AndrewDoesUnityStuff
Unity User
andrewwinkelman@isd110.org

In this document you will learn the ins and outs of my camera script, this manual will cover version 2.5. The first thing you will want to do is leave the editor alone, you will not need to apply the editor script to anything, the editor script will find the camera script on its own. Next you must add your new camera script to your camera (If necessary, you can open the sample scene given in the camera file to see what you should expect). Put your camera inside of your Game-Object (your player). From there (unlike version 1.0) the script will find your Game-Object no matter the name or tag.

Now you have your script set up entirely, now you can use the GUI inside of the Inspector to edit any variable inside of the script, all variables are organized and will have a title of some sort. In version 2.5, the first line you will see is *Active Camera* which follows is the *Sensitivity, and Camera Settings*. *Sensitivity* is a slider which displays a changeable float variable that can be changed before and while the game is running either via the number or slider (The slider will always round to whole numbers unless your optimization settings are different), you can also change the minimum and maximum values of your *sensitivity* slider. Next is a *Foldout* variable named *Extra Settings* which holds *Lock Cursor, Debug Mode, Sensitivity Constraints:, Offset:, and Y Constraints:.* When *Extra Settings* is clicked it will drop down a menu of changeable bools and floats, both work before and while the game is active.

Warning – A warning is displayed when your Camera does not have the *MainCamera* tag or your script is not inside of your camera, this warning does not change the script's functionality.

Camera Active – Activates and deactivates the camera, this variable is now independent as of version 2.5.

Sensitivity – Is a slider that determines the rotation speed of your camera.

1st Person –1st Person is set up with only one parent Game-Object. Must be set up before game start.

3rd Person – 3rd Person is set up with a secondary parent Game-Object. Must be set up before game start.

Notification – This notification will automatically show up in the inspector when you have 3rd person enabled, if you did not set another parent Game-Object this notification will stay around but if you have set it up properly then when you start the game it will go away.

Lock Cursor –Lock Cursor has no effect on performance or usability of the camera, this will only lock the cursor. Works before start and during update depending on your optimization settings.

Debug Mode – This bool runs through every value in your script and displays them in your debug log.

Min Sensitivity – Minimum value that the Sensitivity slider can go.

Max Sensitivity – Maximum value that the Sensitivity slider can go.

Y Rot Offset – This value will offset your camera's position on the Y scale.

Z Rot Offset – This value will offset your camera's position on the Z scale.

Min Y Constraint – Minimum Y value that you can look up (Makes more sense in 3rd person).

Max Y Constraint – Maximum Y value that you can look down (Makes more sense in 3rd person).

After all of these settings you will get to a new drop-down called Optimize Modes, in version 2.5 you can customize how optimized you want or need your script to be. You can choose from *Not Optimized* to *Moderately Optimized* to *Extremely Optimized* to *Custom*. Then finally there is a Reset Values button which resets everything old and new.

Version 2.5

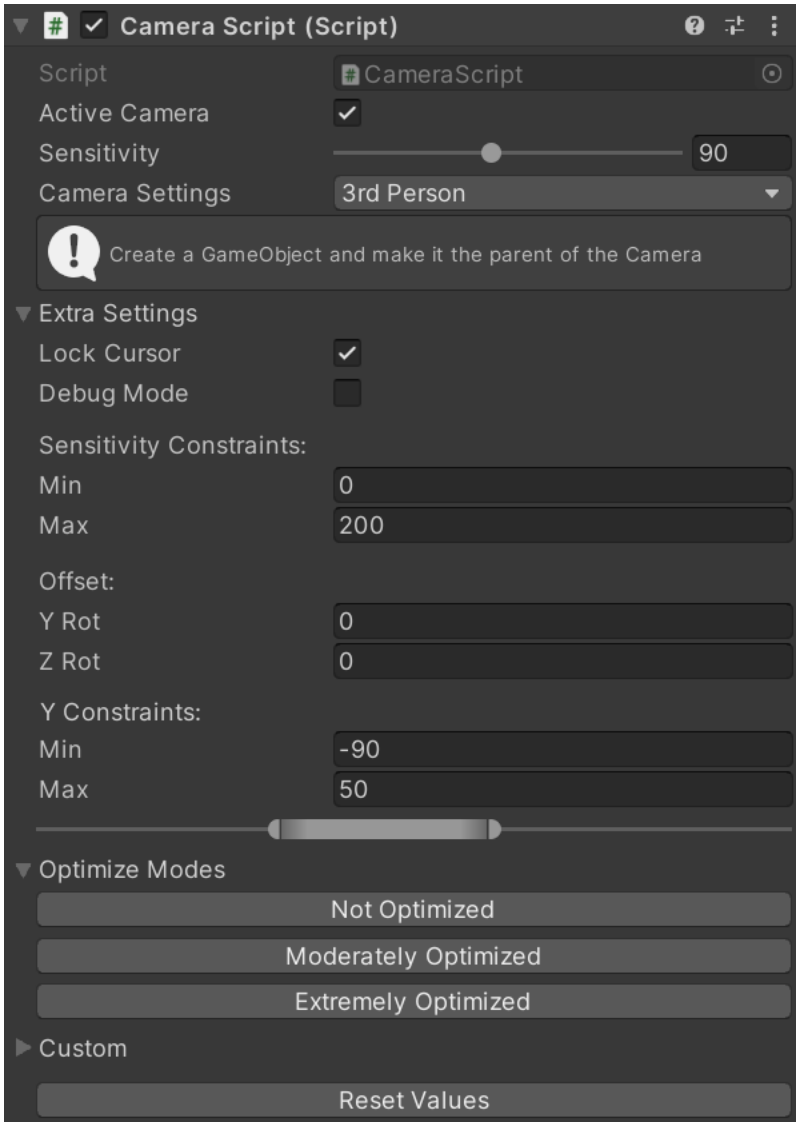
If you do not have the script inside of your camera or your camera does not have the MainCamera tag there will be a warning pop up.

Camera Settings will let you change between 1st person and 3rd person.

First in *Extra Settings* is *Lock Cursor* which locks the cursor, then *Debug Mode* which runs through all of your script's vanilla values and displays them in your debug log.

Then there is *Offset* which is the same from 2.0.

Inside of *Optimize Modes* there are three buttons for levels of optimization you want inside of your script this will cost you your easy usability.



This is a finalized image of the camera script inside of the inspector, completed in a single day.

Sticking to the original Idea, we start out with the Active Camera, which activates accessibility to the rest of the script. Next is the Sensitivity Slider which remains the same.

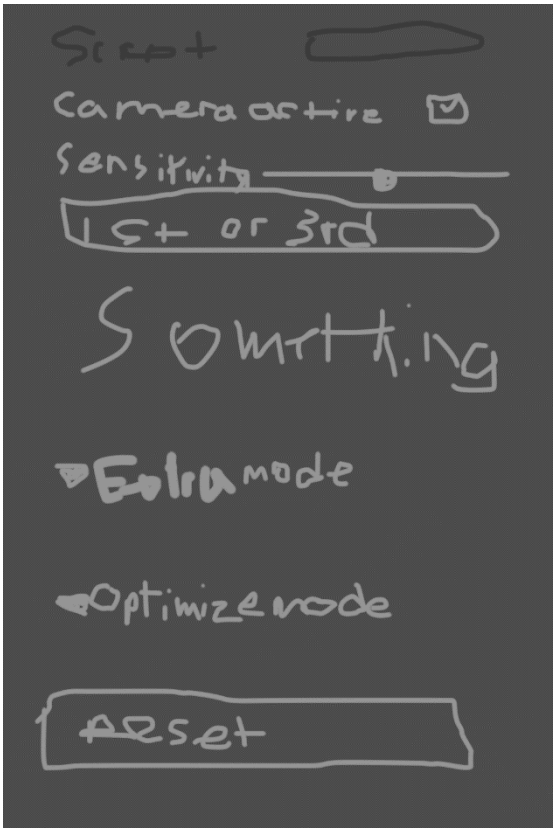
A notification will pop up when you activate 3rd person, will remain there if you do not have two parent objects when you hit play.

Next is *Sensitivity Constraints*, new from 2.0 will let you edit your sensitivity constraints through the inspector rather than in your script.

Then *Y Constraints* which is the same as 2.0.

You can customize what optimization you want along with the classic *Reset Values* button.

Version 2.5 Idea



This is a visualized first idea for 2.5.

This Idea includes the idea of the camera active bool to activate the whole rest of the script and remains priority at the top of the inspector.

Next is the classic *sensitivity slider* then a dropdown including first or *third person* settings.

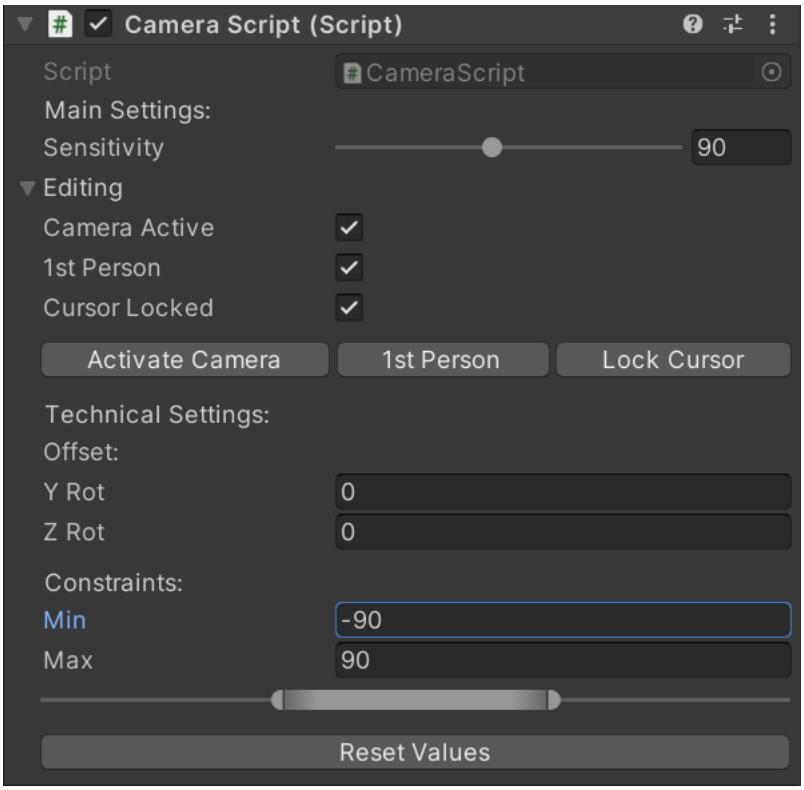
Next there was room for general improvement in the near future and for anyone actually reading this, most likely a *camera shake* feature.

Then an *Extra Mode*, including all extra settings.

Then an *Optimization Mode* that will optimize all of the code.

Then the classic reset last.

Version 2.0



There are more convenient Headers for each part, and *Sensitivity* has a slider.

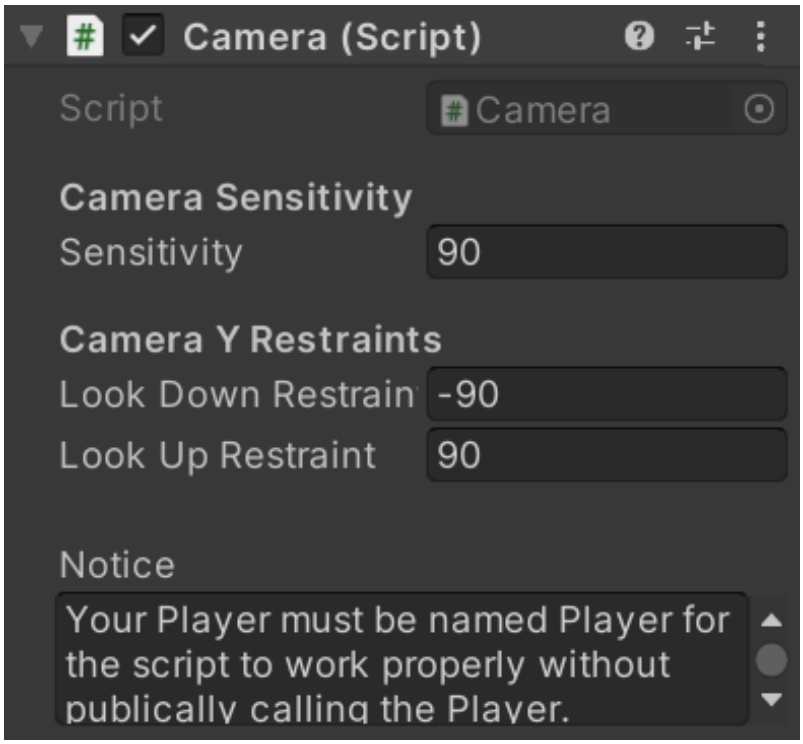
There is a drop down with changeable bools and multiple buttons.

There is an offset and constraints for the camera.

There is a *MinMaxSlider* for the constraints and there is a reset button.

This is a finalized view of version 2.0 done in multiple days.

Version 1.0



Camera Sensitivity was not a Slider and there were only headers for organization.

There are only two Y restrictions and there is a handwritten notice that the script can only find the player if it were called player.

No details on version 1.0.

Scripting:

If you are to ever change *CameraScript* or *CameraEditor* you will want to get used to the interface, *CameraScript* is made as simple as possible, whereas *CameraEditor* should only be manipulated by someone who is familiar with the interface of Unity C# editor scripts.

CameraScript:

When you open the script for the first time there is a lot going on, everything is spaced out and properly organized, if you ever wanted to disable the *CameraEditor* script you can delete all *[HideInInspector]* inside of *CameraScript* to show the raw variables (Or if you're troubleshooting) then delete *CameraEditor*. Everything important inside of the script will have a tag above it explaining what it does. On *Start,* Person view is determined then *IsLockCursor* is determined, it can be deleted without any code breaking consequences but may cause issues. Under the person view and *IsLockCursor* statement, the player/playerplayer will be determined via the parent of the camera, this part of the script does not need to be changed unless you want to specify which Game-Object you want in which you want the parent or parent parent it to be via GameObject.Find("nameOfObject").

When updated the script will respond to the *IsLockCursor* depending on optimization settings then moves to the input calculation part of the script, if *IsActive* and *Is1stPerson* bools are true then it will update a void calculating your mouse input and putting it into the output where your camera moves inside the scene. If you are in 3rd person the same will apply just in a different void

Outside of *void LateUpdate()* are the voids that are called by the editor script, changing the names of the voids without changing them in *CameraEditor* will result in the breaking of both codes. You are able to change what the *CameraEditor* buttons can do through these voids, warning, if you do change the intended function of the Inspector buttons, they will no longer do their original task. Any breaking of code made by nonvanilla changes will not be the responsibility of the owner. Make sure you know what you're doing before you make unnecessary changes to the code.

CameraEditor:

The *CameraEditor* will look inside of the script as it does inside of the Inspector and any deleting of editor code can break the inspector, if you do not delete editor code properly you can cause a code break.

Conclusion

A long day of development has led to the release of a newer interface where users can freely edit further in the inspector along with a little automatic help from script system itself like warnings and debug mode. There is much hope for a better more ranging camera script allowing you to do anything through the inspector alone in the future.

AndrewDoesUnityStuff, “Simple Camera Script Version 2.0”, December 4, 2021