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 2.0. 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 2.0, the first line you will see is *Main Settings:* which will hold *Sensitivity* and *Editing*. *Sensitivity* is a slider which displays a changeable float variable that can be changed before in while the game is running either via the number or slider (The slider will always round to whole numbers). *Editing* is a *Foldout* variable which holds *Camera Active*, *1st Person*, *Cursor Locked*, and the following controller buttons for those variables. When *Editing* is clicked it will drop down a menu of changeable bools, checking the bools and clicking the button both work before and while the game is active.

Camera Active – Activates the camera, will become relevant on its own when a 3rd person update is released. Works before start and during update.

1st Person – On top of Camera Active, 1st Person also has to be checked for the camera to work. Works before start and during update.

Cursor Locked – Cursor Locked has no effect on performance or usability of the camera unlike the other two bools, this will only lock the cursor. Works before start and during update.

The following buttons correlates with the bools listed above:

Activate Camera – Works the same as clicking the bool, pressing the button will send a Debug.Log notifying you if the bool is true or false.

1st Person – Works the same as clicking the bool, pressing the button will send a Debug.Log notifying you if the bool is true or false.

Lock Cursor – Works the same as clicking the bool, pressing the button will send a Debug.Log notifying you if the bool is true or false.

After you have navigated across the *Editing* menu you will see the *Technical Settings:* tag above more variables. Inside of *Technical Settings:* is *Offset:, Y Rot, Z Rot, Constraints:, Min, Max, MinMaxSlider*.

Offset: holds *Y Rot*, and *Z Rot*. *Y Rot* will offset your camera's current direction and start direction left and right. *Z Rot* will offset your camera's current direction and start direction by rotating left and right.

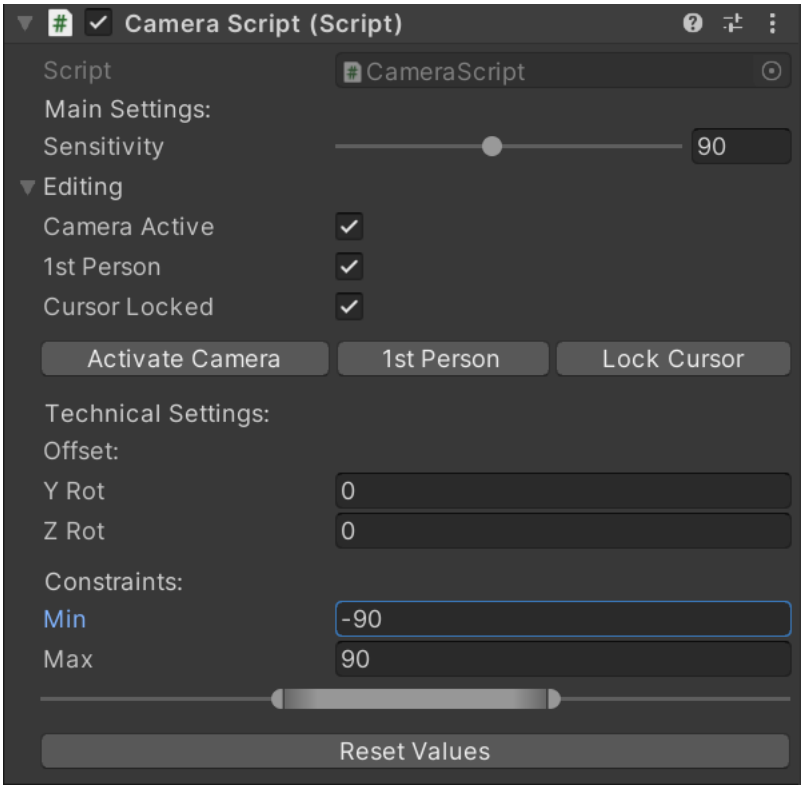
Constraints: holds *Min*, *Max*, and its following *MinMaxSlider*. The *Min* and *Max* float values cannot be edited by selecting them, you must use the *MinMaxSlider* under them to change both values.

Min = Y distance you can look up.

Max = Y distance you can look down.

The *MinMaxSlider* will always round to a whole number as to not become too complicated (The rounding system for the slider is hidden in the editor script and can be changed). Below the *Constraints:* menu, you will find a singular button named *Reset Values*, this button, when activated, will fully reset the script's values, once pressed, all *Constraints:, Technical Settings:,* and *Main Settings:* will be reset completely, the only thing that will have a value will be Sensitivity which will intentionally revert to 90. Notice: after you press the *Reset Values* button you will need to repress *Camera Active* and *1st Person* to reactivate the camera.

Version 2.0



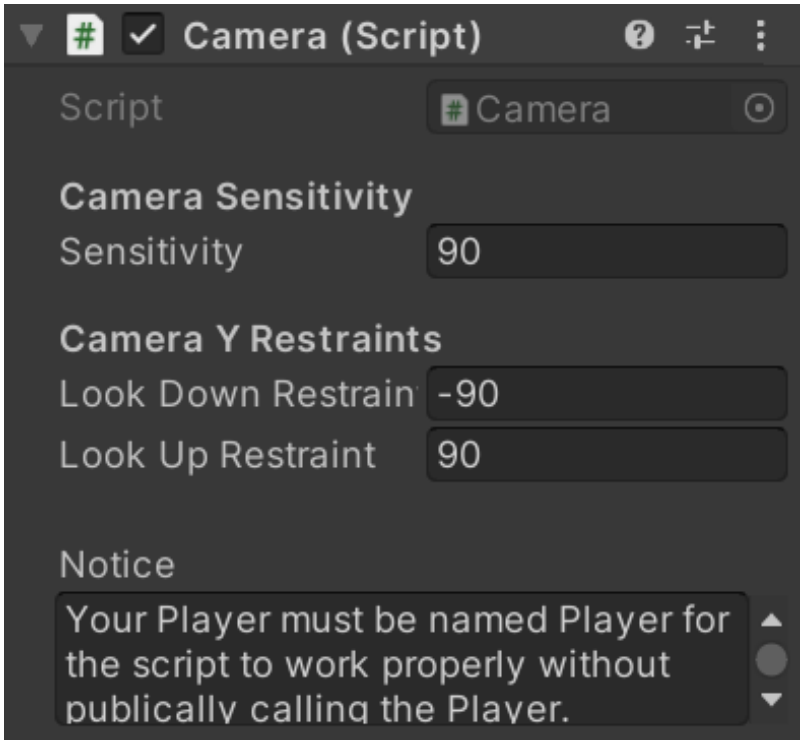
There are more convenient Headers for each part, and *Sensitivity* has a slider.

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.

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.

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, *CameraEditor* should only be manipulated by someone who is familiar with 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). Everything important inside of the script will have a tag above it explaining what it does. On *Start, IsLockCursor* is determined, it can be deleted without any code breaking concequence, may have issues though. Under the *IsLockCursor* statement, the player 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 would change it to GameObject.Find("nameOfObject").

When updated the script will respond to the *IsLockCursor* then move to the input calculation part of the script, if *IsActive* and *Is1stPerson* bools are true then it will start calculating your mouse input and putting it into the output where your camera moves inside the scene.

Outside of *void Update()* 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 changing it after initially receiving it 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:

Right away before the class, the target script is called by the *CustomEditor,* Before the override void, there is a bool and string for the *Editing* menu. Afterward there are two floats to limit the *MinMaxSlider,* these values can be changed freely without serious breaks.

Inside of the override void is where all customization happens. *GUILayout.Label* is the editor's title, you will see this used on *Main Settings:, Technical Settings:, Offset:,* and *Constraints:.* Under the editor note is our first variable, *Sensitivity*. *Sensitivity* inside of the editor is directly grabbed from the *CameraScript*. After *Sensitivity* is the *Editing Foldout*, inside of the *Foldout* is all of the bools and buttons that show up inside the Inspector in order.

After the *Foldout* ends there are two *GUILayout.Label's* before moving to the rounding part of the editor, this part rounds your offset numbers, afterward is the actual rotation variables. Now you have reached the *Constraints:* part of the editor. Here there are two floats, these two floats are also put inside of the *MinMaxSlider* to be edited inside of the Inspector tab. After the *MinMaxSlider* there is the *Restart Button* which will have the same function type as the other buttons.

Conclusion

Days of development has led to the release of an entirely new interface that users can have an easier and more comfortable time using and major improvements have been made since 1.0 was released. Everything down to the code inside of the scripts were made to be as user friendly as possible as to not confuse people who may be beginners, this paper has all of the information needed to be covered to fully understand 2.0, all you need to do is put your camera script inside of your camera and put your camera into any Game-Object and now you have fully set up your script and are now ready to edit your script to your liking.

AndrewDoesUnityStuff, "Simple Camera Script Version 2.0", December 4, 2021