# MS Advanced Camera Controller

**Created by:** Marcos Ismael Schultz
**Age:** 22
**Fórum:** www.schultzgames.com
**Email:** marcos11-24@hotmail.com

'MS Advanced Camera Controller' is a camera controller, which has several camera options to suit the most diverse situations. The cameras included in the code are:

- **LookAt The Player:** A camera that stands still, but always rotates toward the object containing the script.
- **FirstPerson:** A first-person camera that allows 360-degree rotation without moving the player. Ideal for FPS.
- **FollowPlayer:** A camera that follows the object that contains the script smoothly, and avoids obstacles, getting in front of them. This type of camera also makes a smooth rotation towards the object containing the code.
- **Orbital:** A simple orbital camera with configurable zoom, distance and motion speed options, and also contains a function that avoids obstacles and detects collisions.
- **Stop:** A totally dead camera with no action at all.
- **StraightStop:** A camera that stands still in its position, but keeps the horizon always straight.
- **OrbitalThatFollow:** A junction of the '**FollowPlayer**' and '**Orbital**' cameras.
- **ETS_StyleCamera:** A '**FistPerson**' camera, with an additional sliding option when the player looks to the left, allowing the camera to move slightly, and come back in case the player looks to the right.
- **Fly_Camera:** A free camera, which allows you to walk the scenery freely, without collisions or limitations.

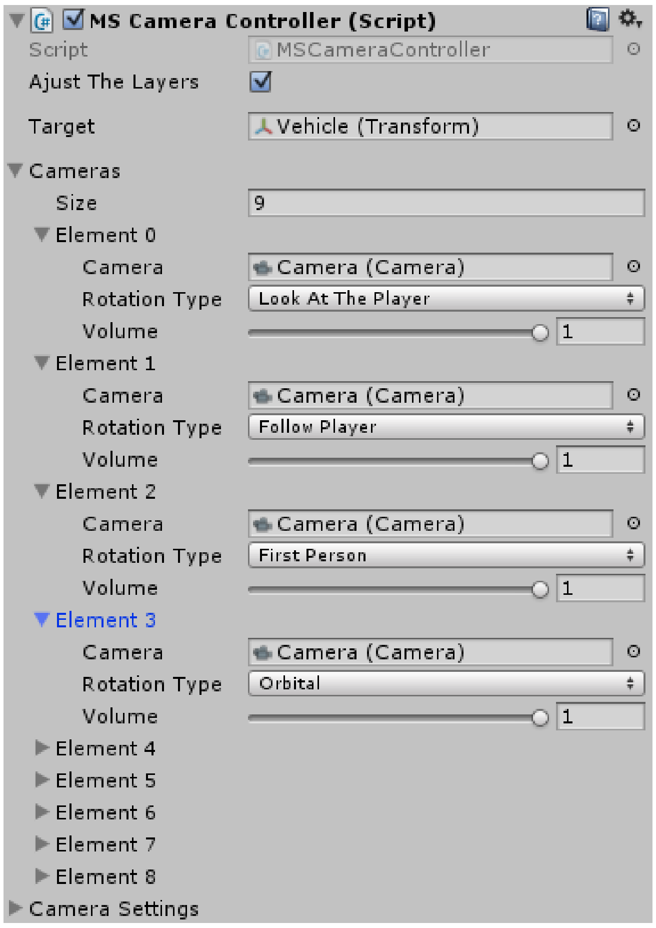
'MS Advanced Camera Controller' also includes a demo scene, with all kinds of cameras listed above, to demonstrate the operation.

## How to use:

To use this feature, simply place the 'MSCameraController' script on any object in your scene. It can be an empty object or even the player itself. Having done this, you must associate the player with the variable 'Target', or the target the cameras will follow. If you do not associate any objects in the 'Target' variable, the cameras will follow the very object on which the script is allocated. After that, just associate the cameras with their variables and set everything up.

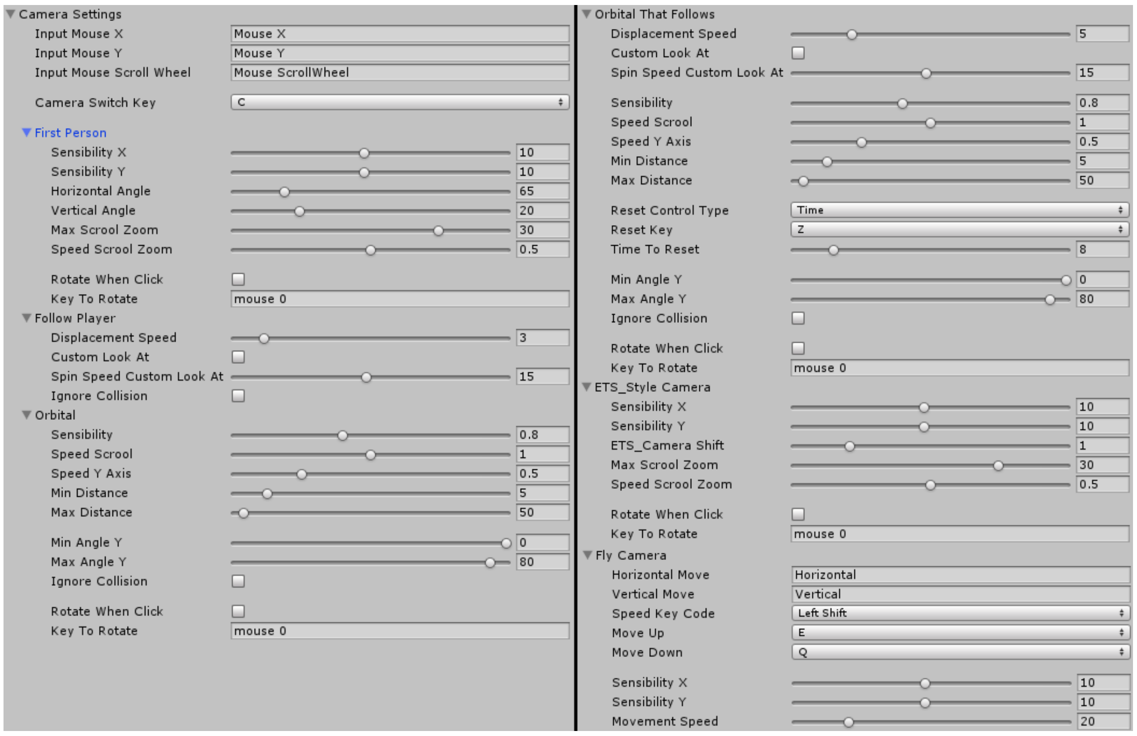
In the 'Cameras' array, you must define the number of cameras you have and press the 'Enter' key to initialize the array of cameras.

Once you've done that, simply associate each camera with an index and set the type of rotation or movement that each camera should have, as the image below illustrates.



```
MS Camera Controller (Script)
Script                MSCameraController
Ajust The Layers      ✓
Target                Vehicle (Transform)
Cameras
  Size                9
  Element 0
    Camera            Camera (Camera)
    Rotation Type     Look At The Player
    Volume            1
  Element 1
    Camera            Camera (Camera)
    Rotation Type     Follow Player
    Volume            1
  Element 2
    Camera            Camera (Camera)
    Rotation Type     First Person
    Volume            1
  Element 3
    Camera            Camera (Camera)
    Rotation Type     Orbital
    Volume            1
  Element 4
  Element 5
  Element 6
  Element 7
  Element 8
Camera Settings
```

You can configure each type of Camera as well, by setting the limits of movement, speeds, rotations, among other things, as the image below demonstrates.

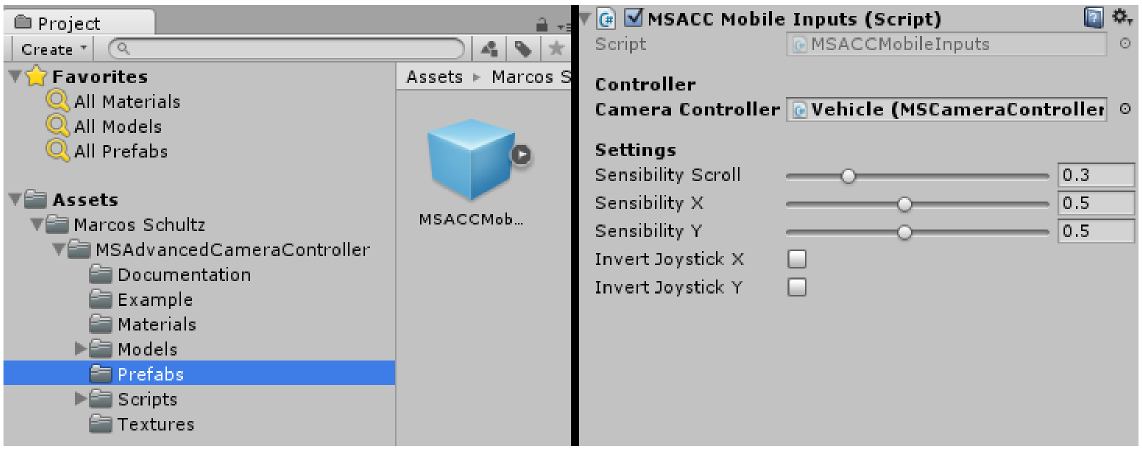


| | |
|---|---|
| **Camera Settings** | |
| Input Mouse X | Mouse X |
| Input Mouse Y | Mouse Y |
| Input Mouse Scroll Wheel | Mouse ScrollWheel |
| Camera Switch Key | C |
| **First Person** | |
| Sensibility X | 10 |
| Sensibility Y | 10 |
| Horizontal Angle | 65 |
| Vertical Angle | 20 |
| Max Scrool Zoom | 30 |
| Speed Scrool Zoom | 0.5 |
| Rotate When Click | ☐ |
| Key To Rotate | mouse 0 |
| **Follow Player** | |
| Displacement Speed | 3 |
| Custom Look At | ☐ |
| Spin Speed Custom Look At | 15 |
| Ignore Collision | ☐ |
| **Orbital** | |
| Sensibility | 0.8 |
| Speed Scrool | 1 |
| Speed Y Axis | 0.5 |
| Min Distance | 5 |
| Max Distance | 50 |
| Min Angle Y | 0 |
| Max Angle Y | 80 |
| Ignore Collision | ☐ |
| Rotate When Click | ☐ |
| Key To Rotate | mouse 0 |
| **Orbital That Follows** | |
| Displacement Speed | 5 |
| Custom Look At | ☐ |
| Spin Speed Custom Look At | 15 |
| Sensibility | 0.8 |
| Speed Scrool | 1 |
| Speed Y Axis | 0.5 |
| Min Distance | 5 |
| Max Distance | 50 |
| Reset Control Type | Time |
| Reset Key | Z |
| Time To Reset | 8 |
| Min Angle Y | 0 |
| Max Angle Y | 80 |
| Ignore Collision | ☐ |
| Rotate When Click | ☐ |
| Key To Rotate | mouse 0 |
| **ETS_Style Camera** | |
| Sensibility X | 10 |
| Sensibility Y | 10 |
| ETS_Camera Shift | 1 |
| Max Scrool Zoom | 30 |
| Speed Scrool Zoom | 0.5 |
| Rotate When Click | ☐ |
| Key To Rotate | mouse 0 |
| **Fly Camera** | |
| Horizontal Move | Horizontal |
| Vertical Move | Vertical |
| Speed Key Code | Left Shift |
| Move Up | E |
| Move Down | Q |
| Sensibility X | 10 |
| Sensibility Y | 10 |
| Movement Speed | 20 |

The entire system has 'Tooltips' in the variables, making the system easy to implement, and helping to avoid errors. These Tooltips provide warnings that help the user to implement the system. To see the warnings, just rest your mouse over some variable, and the warning will appear.

## Mobile inputs:

In the 'MarcosSchultz> MSAdvancedCameraController>Prefabs' folder there is a 'prefab' named 'MSACCMobileInputs (new)'. Just place this object in your scene and configure it.



**MSACC Mobile Inputs (Script)**

| | |
|---|---|
| Script | MSACCMobileInputs |
| **Controller** | |
| Camera Controller | Vehicle (MSCameraController |
| **Settings** | |
| Sensibility Scroll | 0.3 |
| Sensibility X | 0.5 |
| Sensibility Y | 0.5 |
| Invert Joystick X | ☐ |
| Invert Joystick Y | ☐ |