

Multi Tracking Camera Manual

Hello, thanks very much for purchasing my pack. I'll keep this document updated with inquiries from previous customers. Feel free to drop me a line on my support mail if you need something or you want to ask me for anything specific.

Online version of this manual:

<https://docs.google.com/document/d/1jkyj6VUzmA4ooDdfAIQPldS3YKxSFMW3mrS1xMXK6w8/edit?usp=sharing>

What's Multitrack Camera?

It's a system that allows you to frame multiple elements in the camera view and interpolate between positions when the objects move keeping them framed and in view.

What for?

- Strategy Games where you're top down view and you want to keep units in view.
- Fighting games where you want both characters in view.
- Platformer games where you want to focus / defocus elements.
- RPG for instance to focus when you start talking to an NPC and keep both characters in screen.

There are lots of uses for this kind of system.

How does it works?

To setup the camera system is quite easy.

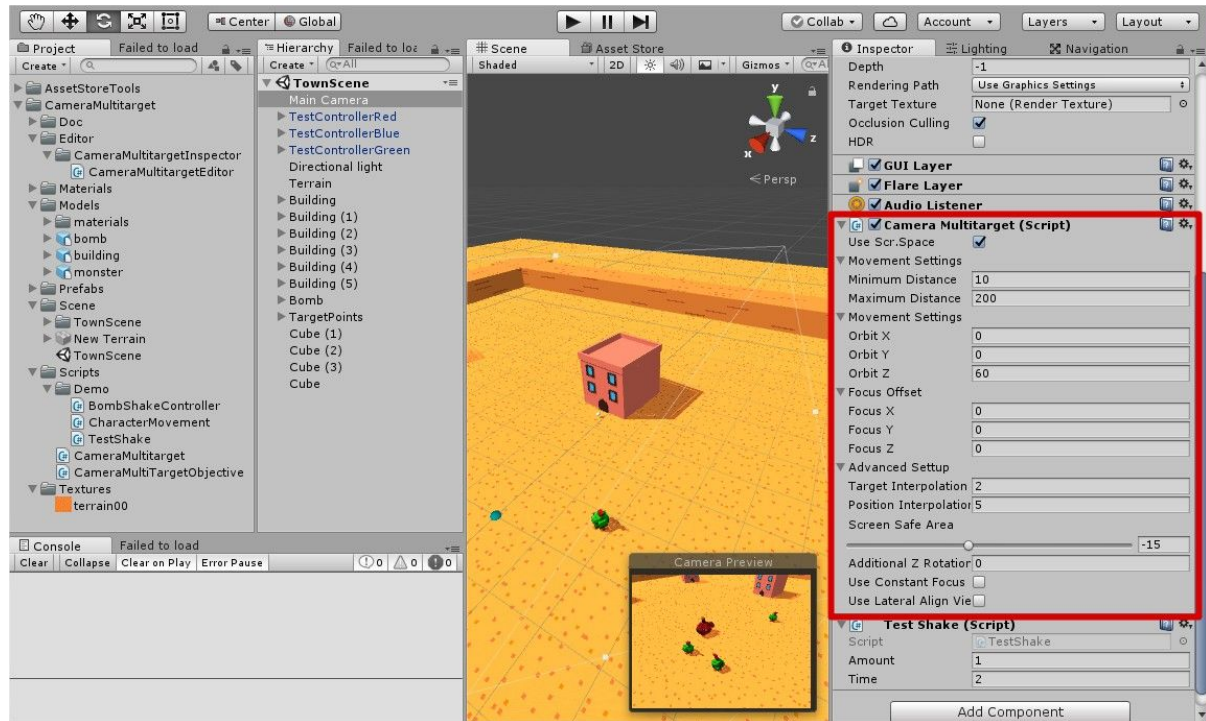
- Attach the "CameraMultitarget.cs" behavior to your camera.
- Attach the "CameraMultitargetObjective" to the objects you want to track.
 - Keep in mind that the object you attach to should have a renderer to compute image inside screen. so make sure you attach it to your "model" or "renderable" element inside your prefab.

After this the camera will take care of focusing and framing the objects targetted.

If you want to get more insights on details continue reading. But if you get these previous steps correctly you should get it working.

I've included sample scene where you'll find a system already setup with the basic features of the camera working and running.

The MultitrackCamera Script



The script contains a lot of control values in order to help you get the camera running.

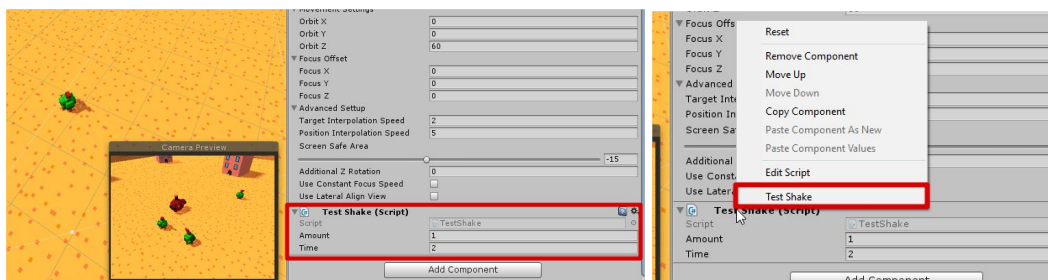
Parameters:

- **Use Scr.Space:**
 - Will make the camera to take the focus based on screen view instead of world position. It will change camera orientation in order to fit the objects in screen space.
- **Minimum Distance:**
 - The minimum distance the camera will get from the objects in order to focus them.
- **Maximum Distance:**
 - This value tells the camera how far you want it to go in order to keep objects framed.
- **Orbit (X, Y, Z)**
 - This tells the camera to rotate around the center of the bounding box containing all focused objects.
 - Use this to rotate around your objects. basically the Y value will give you a nice orbit feature.
- **Focus Offset (X, Y, Z)**
 - Allows you to give the camera an offset from the center bounding box of your items.

- **Target Interpolation & Focus Interpolation:**
 - Allows you to control how smooth you want the camera to move target or position.
- **Screen Safe Area:**
 - How fitted you want the bounding box in screen view. Negative values will allow objects to get outside screen, while positive values will keep your objects more in the central part of the screen. A recommended value is about -15 to 15.
- **Use Constant Focus Speed:**
 - This value will interpolate constantly instead of lerping to object. The camera will move at constant speed (you'll need to modify the interpolation values because this feature will require different parameters there).
- **Use lateral align view:**
 - This is mainly focused to fighting games.
 - It takes the first 2 objects tracked (it's designed for 2 objects). and keep the camera perpendicular to them like old fighting games where characters can rotate but the camera always keep in the lateral view.

Shaking Feature:

Multitrack camera comes with a built in shaking feature. In order to test it you can use the script I've attached to the camera so you can modulate the parameters to see the result. Hit play and right click on ThestShake script to test written values.



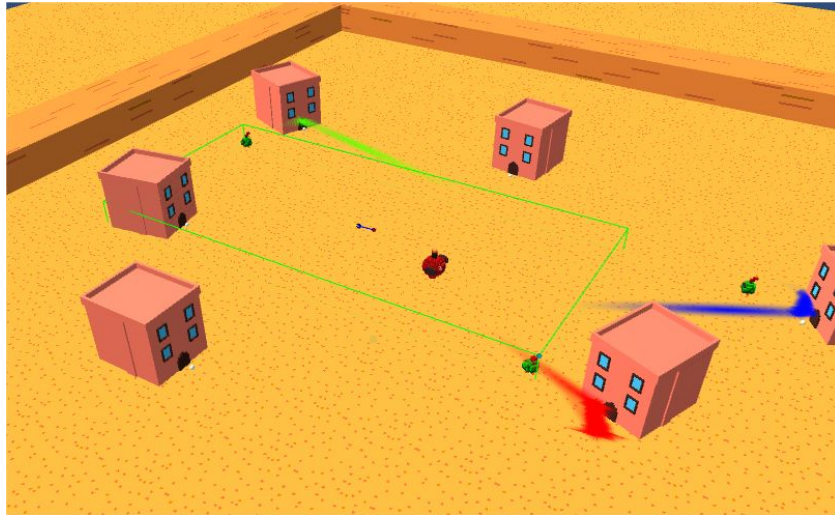
In order to call the shaking feature you can use a simple code:

```
Camera.main.GetComponent<CameraMultitarget>().Shake(amount, time);
```

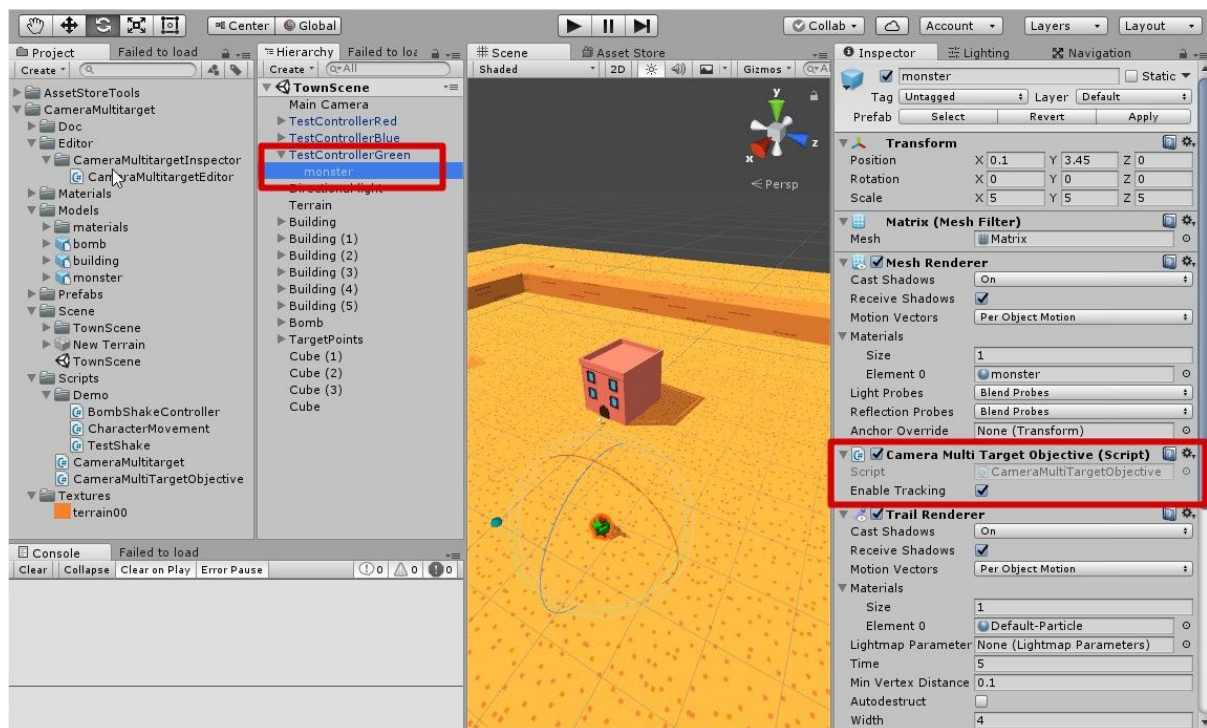
Replace Camera.main by your working camera in case your main camera is not focusing objects.

Gizmos

Enable gizmos to preview where the bounding box is and where the camera is looking at. It will give you additional information of what's happening.



The MultitrackCameraObjective Script



Attach this script to the mesh object so it will be tracked. In the case you (as me) keep models inside other hierarchy add the script to the object with the mesh. It's needed this way in order to compute the viewable area and keep it inside the screen.

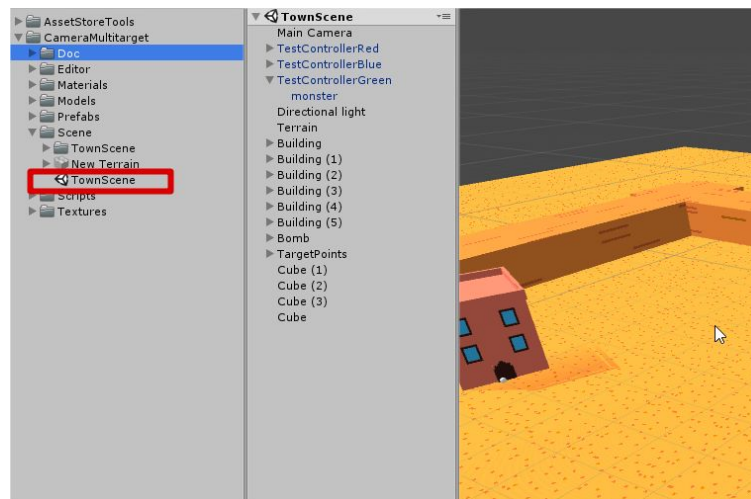
- Enable Tracking:
 - Activating / Deactivating this value will tell the camera to track / untrack this object.

If you make prefabs with this script it should be tracked once you spawn them.

Sample Scene

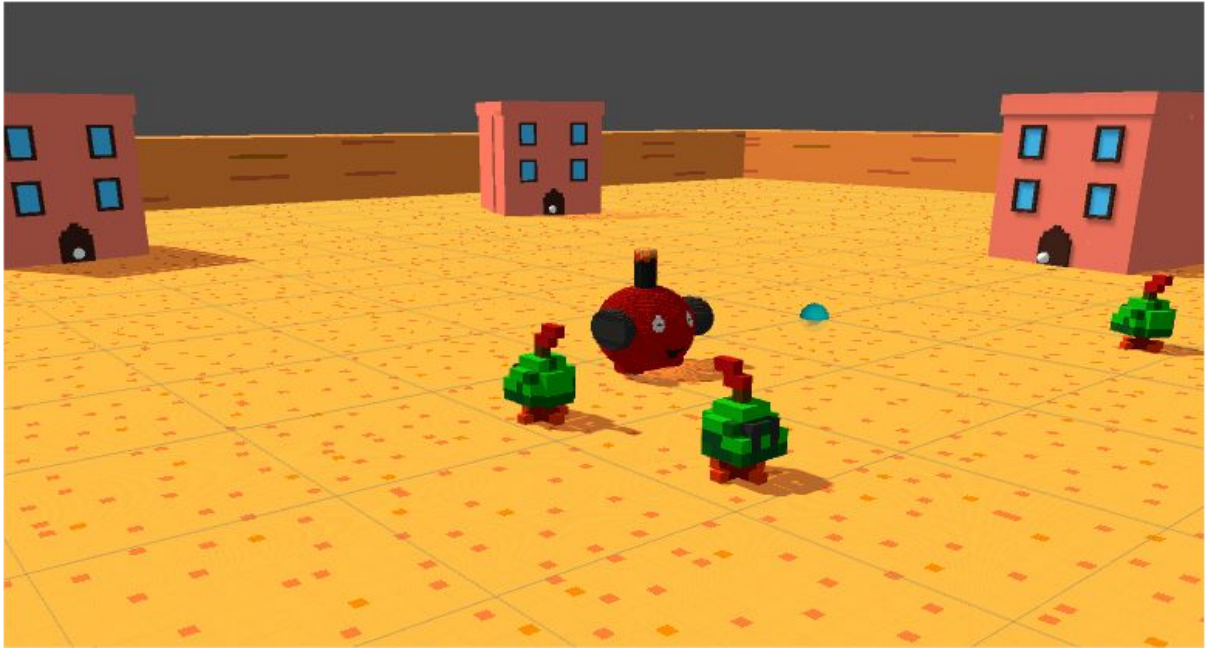
I've included a working sample scene, where I've preconfigured working values for the camera and where you can play around with values to come up with something you want.

You'll find the sample scene inside the Scene Folder in the Asset folder structure.



Inside the scene you'll find TestControllerRed / Blue / Green that are the small objects with a basic AI that moves them around the map. basically visiting the doors of the houses.

Also you'll find a big monster that jumps from time to time making the camera shake. I've included an script that makes the shake take into account the distance so it's more intense as you get closer for the sake of sampling the shaking system.



Thanks very much for your input and for your purchase. Don't hesitate to contacting me for any need.