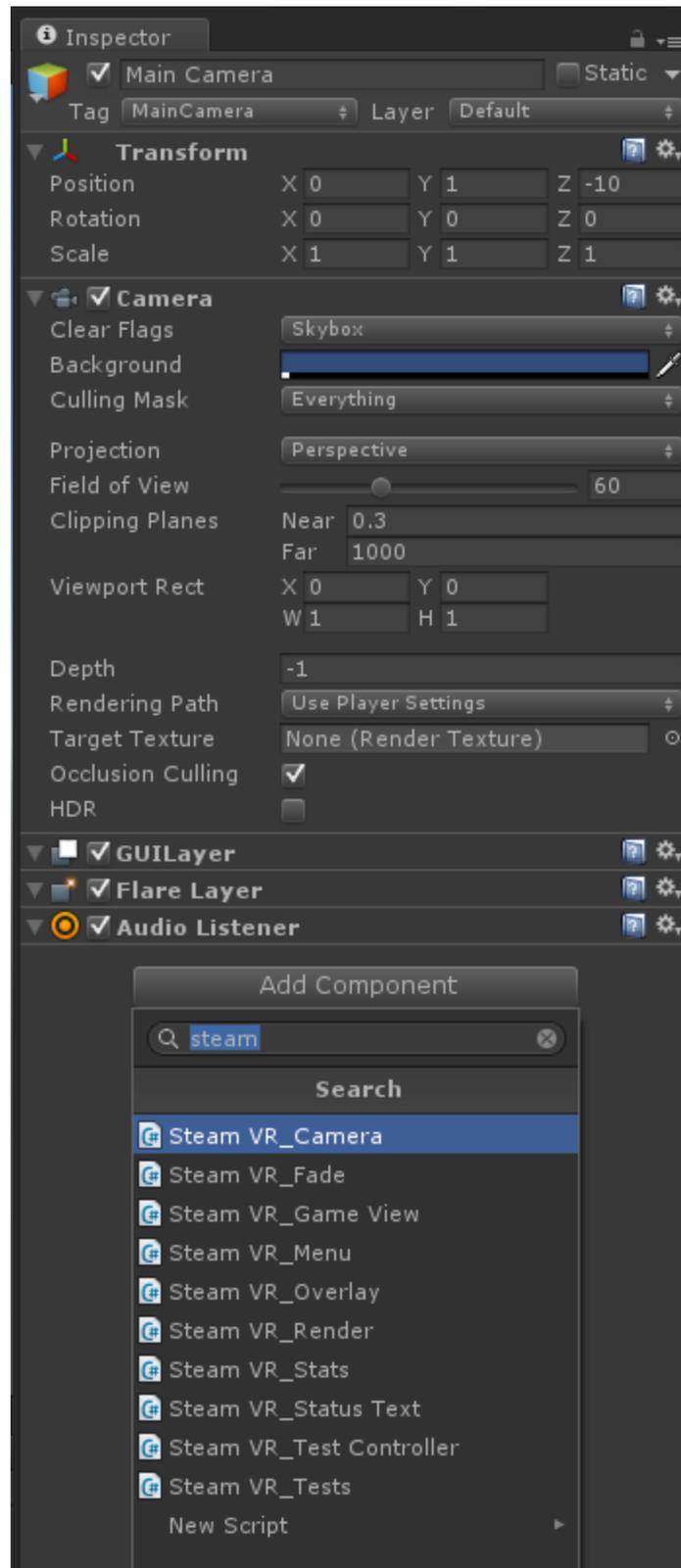




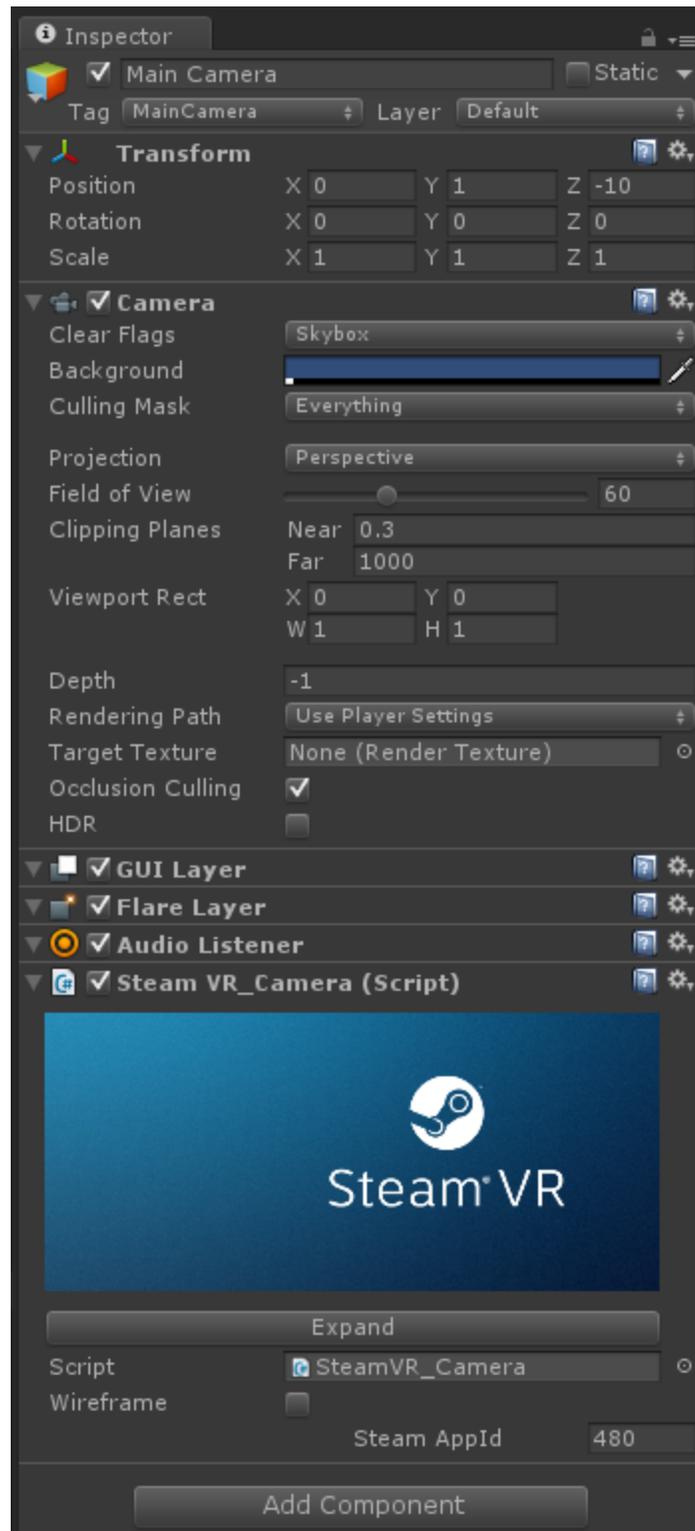
1) Select camera(s) to add VR support:



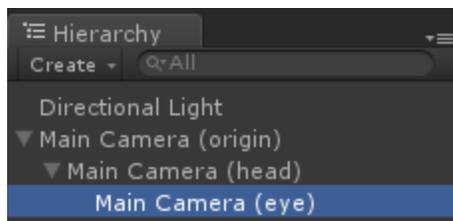
2) Click Add Component and add SteamVR\_Camera script:



3) Click the Expand button:



This expands your camera rig creating head and origin parents, leaving your original camera as the eye. This allows you to position the *origin* to control where your tracking volume lines up in the virtual world and attach items that should follow the *head* motion.



The Expand button changes to Collapse which can be used to reverse this operation. If you leave your SteamVR\_Camera collapsed, it will automatically get expanded on startup.

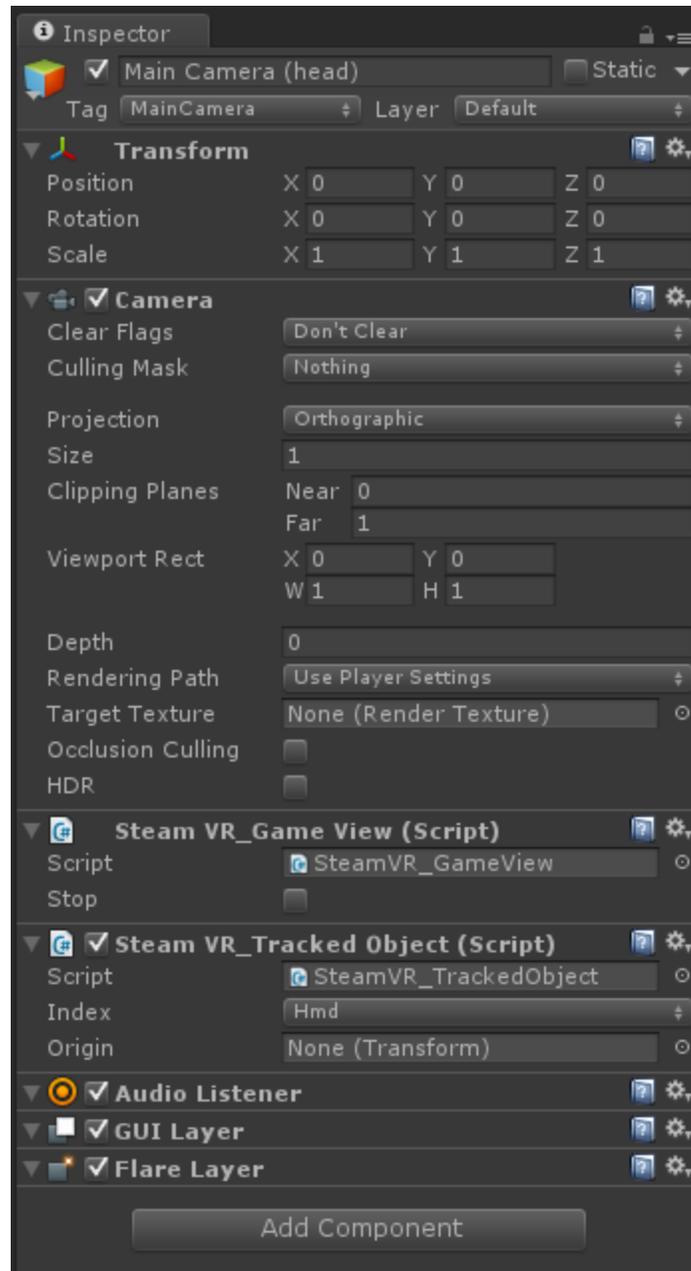


## Room-Scale tracking

The SteamVR Unity plugin defaults to standing tracking space, which requires configuring your tracking space using the SteamVR calibration tool. The origin will be placed on the floor, and content should be built accordingly. However, prior to calibration, the origin will be at the camera or primary base station's height.

If you are creating a seated experience, you will want to add the [SteamVR] prefab to your scene and set its tracking space to seated. This will use the configured seated zero pose set using the calibration tool as the origin. Prior to calibration, the origin will default to the first valid pose on startup, mimicking the auto-zeroing functionality of previous versions.

One final note, when you expand your SteamVR\_Camera rig, the Audio Listener, GUI Layer and Flare Layer components are automatically moved to the head object. This transform actually represents the midpoint between both eyes, and follows the Hmd's motion via the SteamVR\_TrackedObject component.



You will also notice that this object has its own camera. This is used to render the companion window on the primary monitor. It is also the object returned by Camera.main (assuming your original camera was tagged MainCamera).

Check out the example scene in Assets/SteamVR/Scenes to see examples of additional features like using SteamVR\_Fade (fullscreen color fade in/out) and the [Status] prefab which uses SteamVR\_Overlay (display 2D content), SteamVR\_Menu (OnGUI example), SteamVR\_Stats and SteamVR\_StatusText (GUIText example and notifications).

Refer to the included readme.txt for more details.