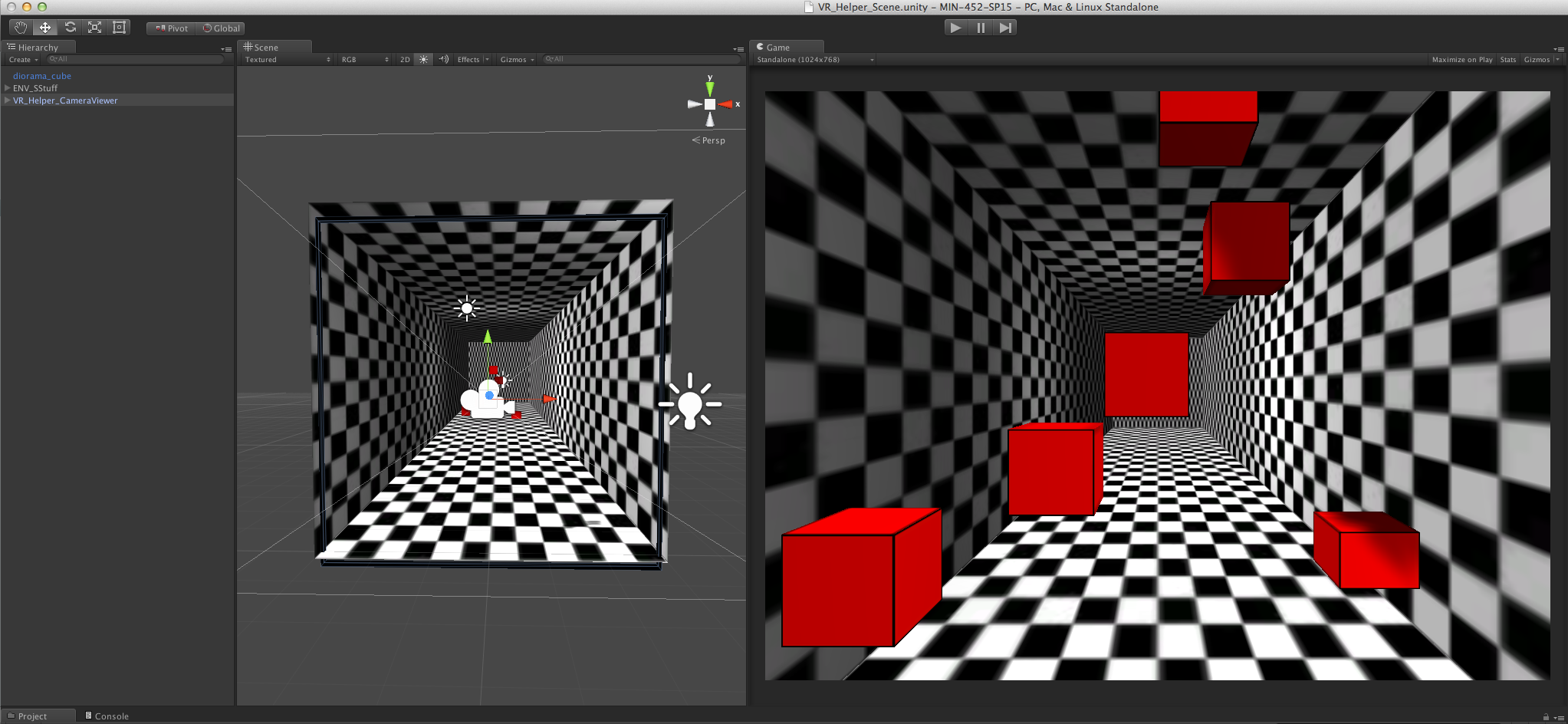
VR Helper 1.0

Copyright RHZ Development, LLC, 2015



**Scripts included:**

1. **Z\_VRCam**, C# Script – This is the script that will take your camera and the VRHead object and turn the position of the VRHead into a VR Camera Projection for a Unity camera. This will also create a look-at projection, creating a virtual VR tracking of an object. The VRHead object will serve as the point from which the projection is generated, which can then be controlled by any tracking element desired, including a face-tracking api, Kinect, or WiiMote IR Point Tracking. You can easily use this projection to do Chang-style Wiimote tracking by using the calculations to drive the position of VRHead.
2. **Z\_TrackMouse**, C# Script – This is a simple script to use the mouse to simulate the effect of the VR Perspective Projection.

**Prefabs Included:**

1. **VR\_Helper\_CameraViewer** – This prefab has a parent node named VR\_Viewer, which contains the VRHead node (a Unity primitive cube) and a Unity camera to make the projection. The camera movements are controlled by the Z\_VRCam script, and are able to be offset in the X, Y, and Z axes based on the VR\_Viewer parent node.

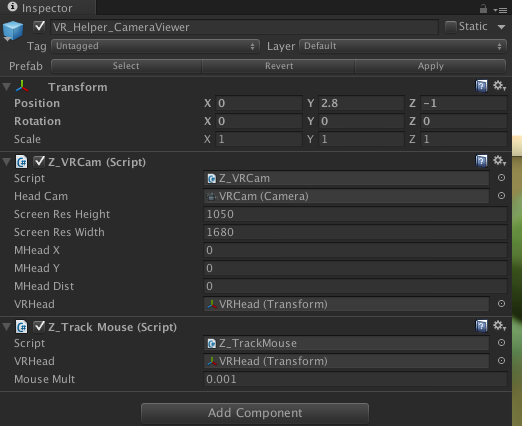
**Unity Scenes Included:**

1. **VR\_Helper\_Scene –** This scene contains some elements that assist the optical illusion and projection matrix in being seen to test out the perspective.

Current Limitations:

* Rotations of the camera parent aren’t supported yet (in progress for version 2.0.
* Head tracking does not exist with this plugin script, it only exposes the VRHead node for you to track with some other method (Wiimote, faceAPI, Kinect)

**Instructions:**



Drop the VR\_Helper\_CameraViewer object into your scene. It should create the VR perspective automatically for you. You can open the scene file VR\_Helper\_Scene and see the settings in the scripts, which reside on the VR\_Viewer parent object.

**Variables of Z\_VRCam.cs:**

1. **Head Cam –** This is the camera you wish to set the VR projection onto. It’s automatically set to the camera it belongs to.
2. **Screen Res Height –** The height in pixel resolution of the viewing screen (must be correct in order to achieve correct perspective).
3. **Screen Res Width -** The width in pixel resolution of the viewing screen (must be correct in order to achieve correct perspective).
4. **MHeadX –** X position in metersof head from screen (used for debugging).
5. **MHeadY –** Y position in metersof head from screen (used for debugging).
6. **MHead Dist –** Z position of head in mm from screen (-1 is 1 meter away).
7. **VRHead -**  The Transform Node to control the projection (set to the VRHead node in the prefab).

**Variables of Z\_TrackMouse.cs:**

1. **VRHead –** Transform to move with the mouse**.**
2. **Mouse Mult –** Transform multiplier for mouse movement.