V1.2

Introduction

Thanks for the purchase and support! We are a community of VR devs, working together to create games, experiences, development tools, and tutorials in an effort to empower emerging VR developers worldwide. Join us here: https://www.youtube.com/nurfacegames/

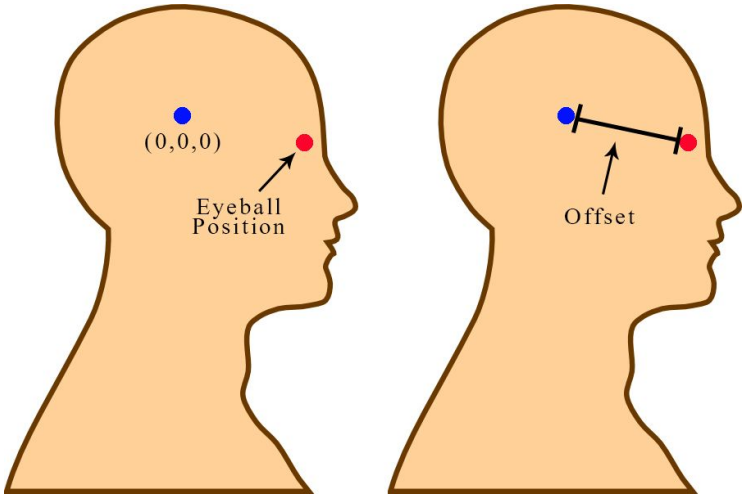
Video Tutorial

http://www.youtube.com/watch?v=dY5xMaYuFbE

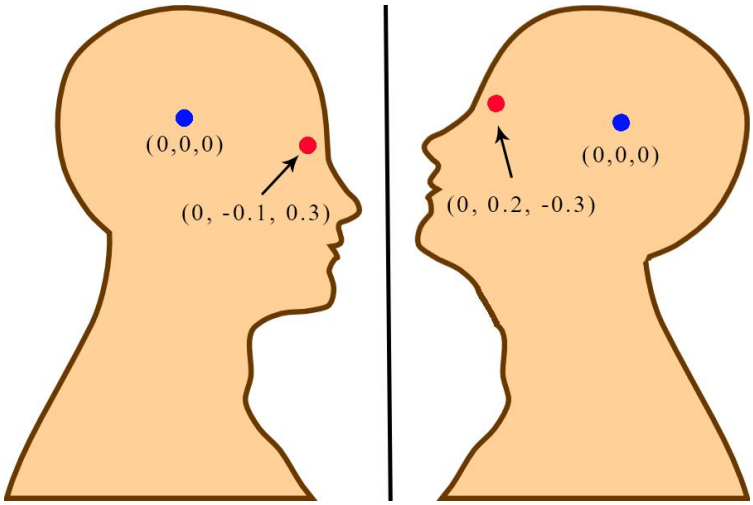
What is VR Body?

Add a virtual body / avatar to your VR project! This is for mobile VR such as Gear VR, Google Cardboard, Daydream, Idealens, etc. This is not for ‘room-scale’.

This tool does 2 things:
- Moves and rotates a virtual body with the camera so the body position matches the actual user’s body movements in real life.
- Moves the camera’s position based on where the user is looking. Our eyeballs do not remain at “center-of-skull” when we rotate our head. If 0,0,0 was the top of our spine, our eyeballs are offset by some distance from 0,0,0, perhaps 2 or 3 inches.



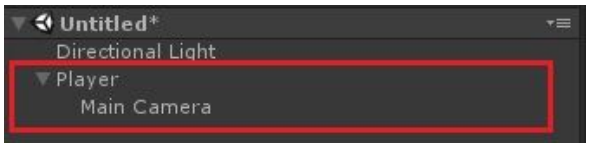
Notice in the following diagram that the user has not have their body position from 0,0,0 and they have only rotated 180 degrees, and the position of the eyeball has moved.



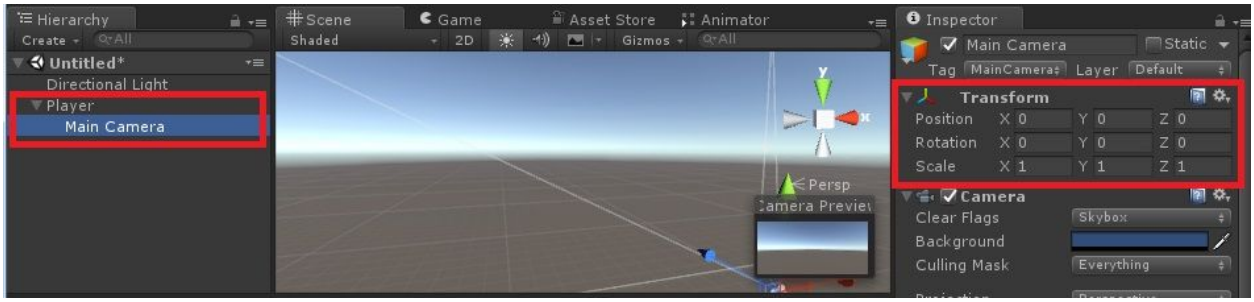
The basic VR camera is set to 0,0,0 and only rotates. It does not move as our real eyeballs move, at some small offset from 0,0,0. Perhaps not accounting for this eyeball movement contributes to VR sickness?

How To Use

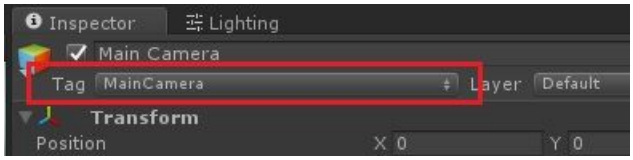
1. In order to move Main Camera, it must be a child. Create a “Player” GameObject and make Main Camera a child of this.



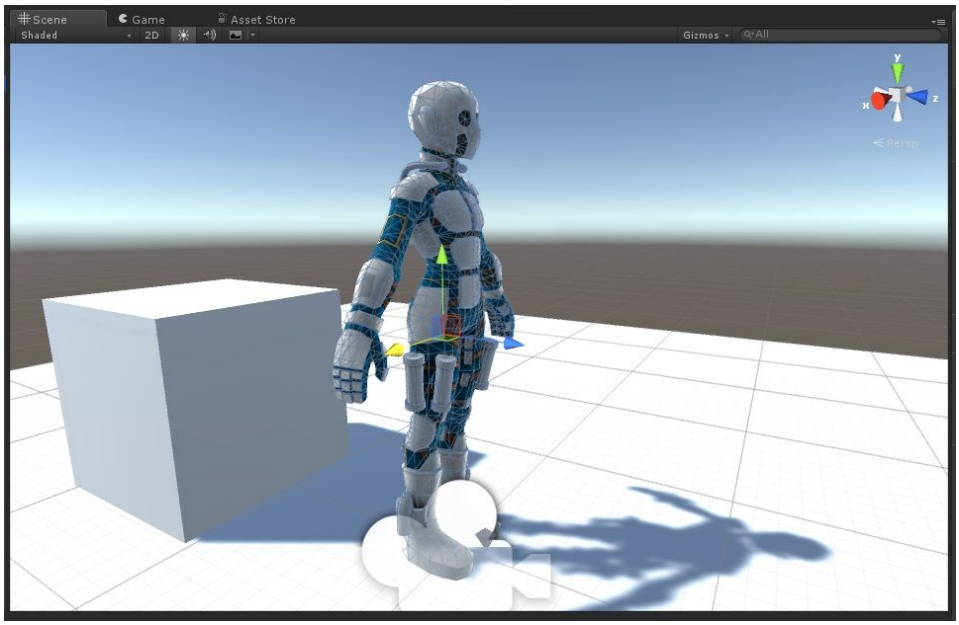
2. Set Main Camera’s position and rotation to 0,0,0.



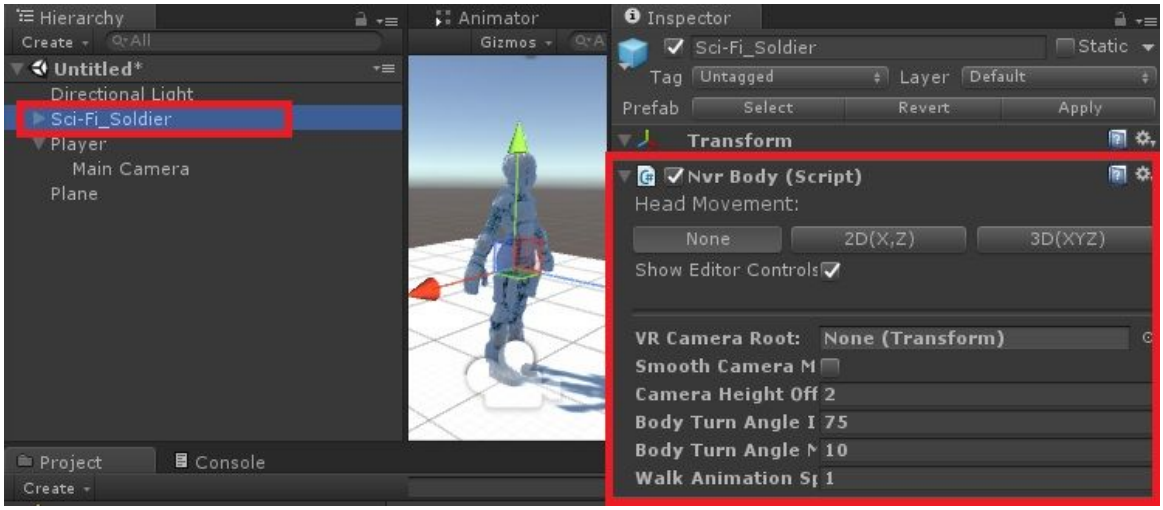
3. Make sure Main Camera has been tagged with “MainCamera”.



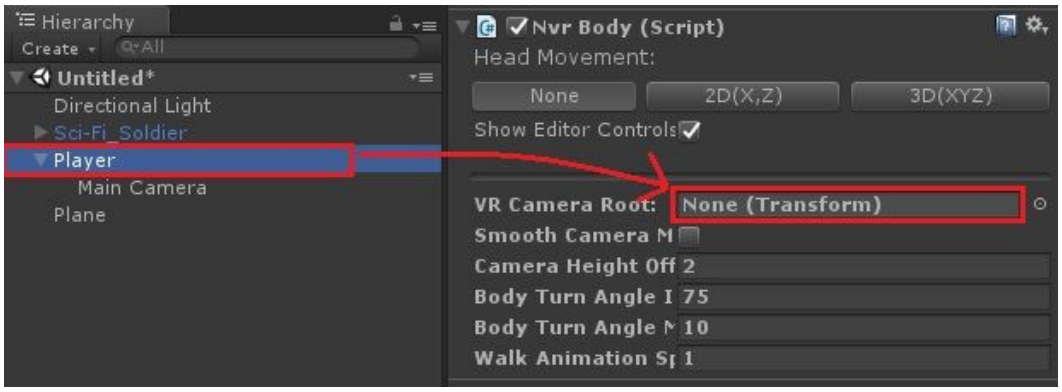
4. Add a 3d Model of a body to your scene. A human should be about 2 units/meters tall.



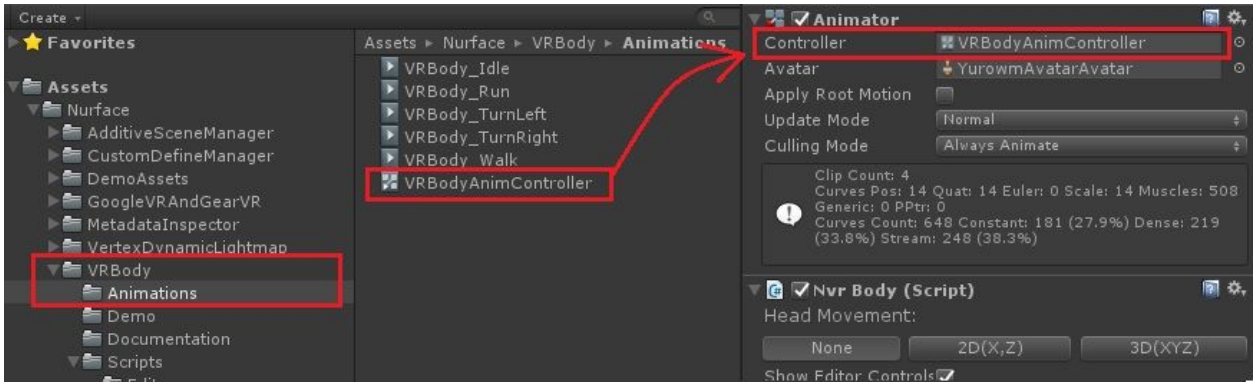
5. Import this package, and add the script ‘NvrBody.cs’ to the character’s root gameobject.



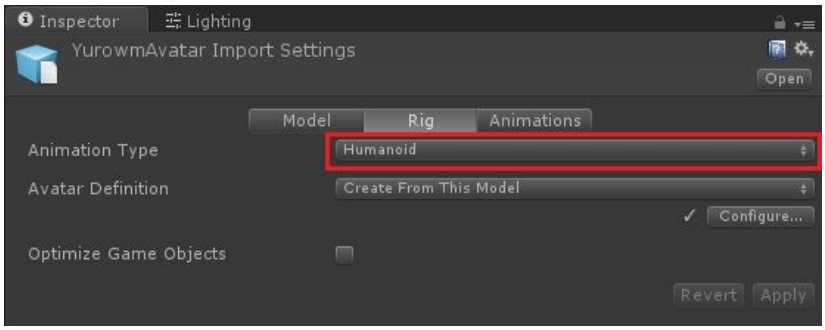
6. Assign the GameObject ‘Player’ (Main Camera’s Root Parent) to var “VR Camera Root”.



7. Assign the Animation Controller **VRBodyAnimController** located at *Nurface\VRBody\Animations* to the Character’s Animator. Add an Animator if necessary.



8. If your model does not animate, it may need a humanoid avatar configured for Mecanim.

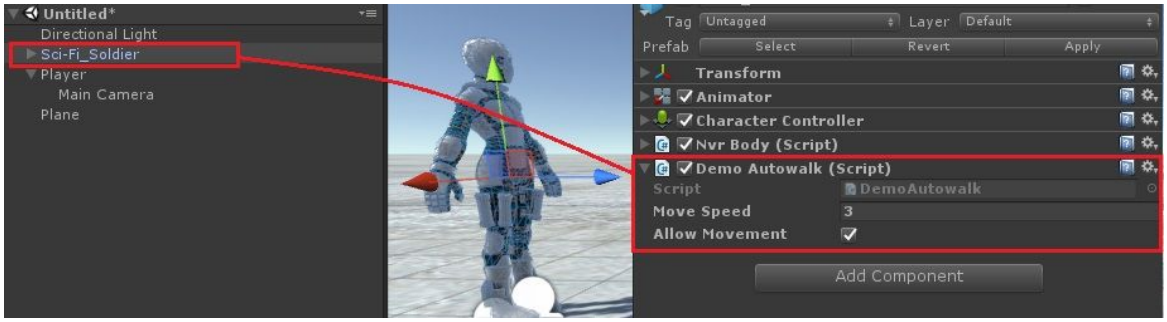


9. The basic setup is complete. Please see the following tips and proceed to configure the script to your needs, covered in the following section.

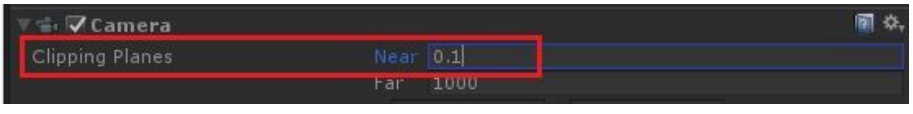
Tips:

- To rotate the camera in Unity Editor, add the script **EditorMouseLook.cs** located at *Nurface\DemoAssets\Scripts* to the Main Camera.
- Hold **ALT** to rotate head, hold **CTRL** to tilt the head.

- Add the script **DemoAutowalk.cs** located at *Nurface\DemoAssets\Scripts* to the Character Model to make them walk. Click/tap to walk and click to stop walking.



- Adjust the Camera's Clipping Planes, Near, down to about 0.1 so you can see the body.

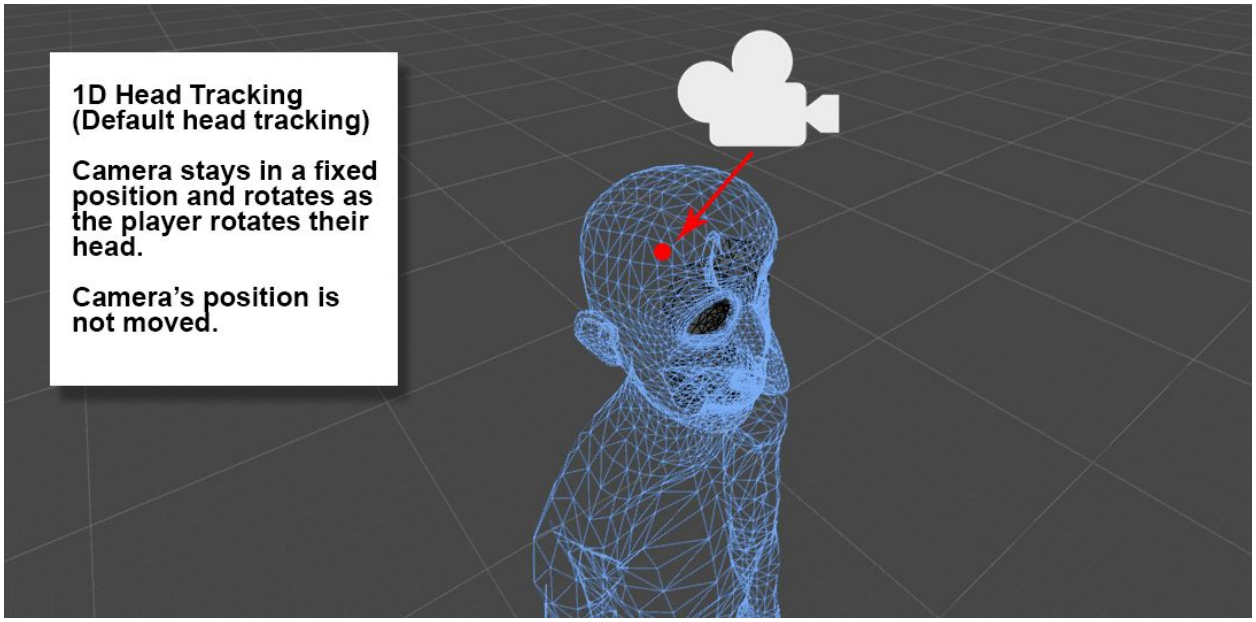


NvrBody.cs

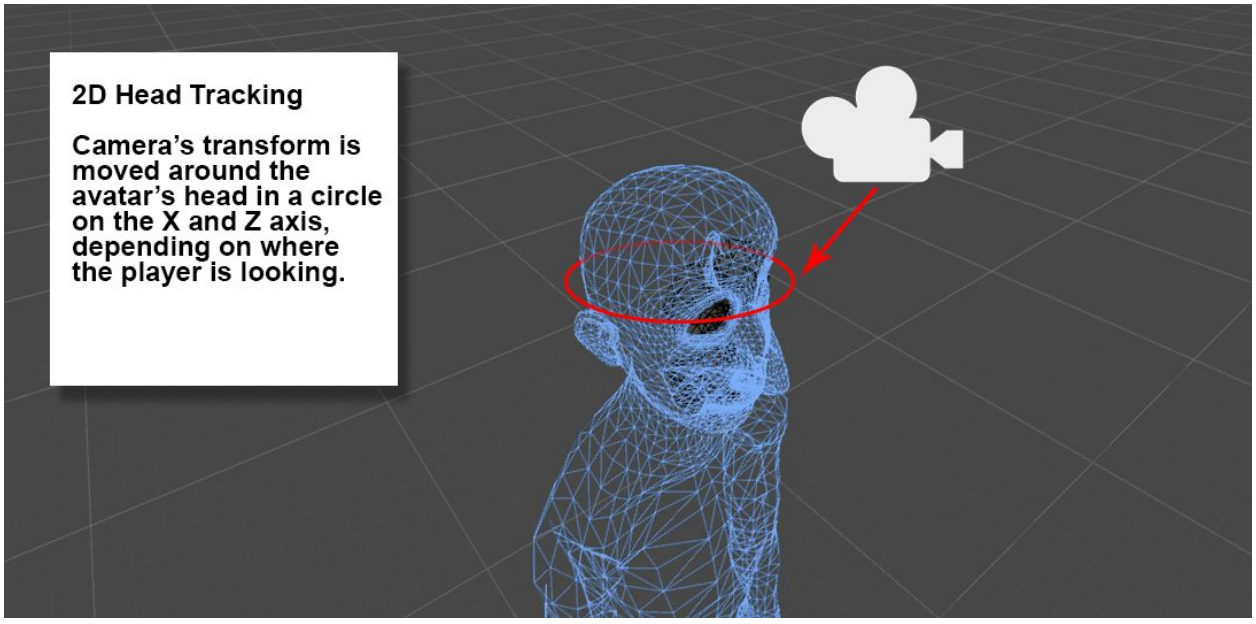
All of our core scripts contain the prefix “Nvr” so it can be identified as a ‘Nurface VR script’.

This script has 3 operating modes: None, 2D, and 3D.

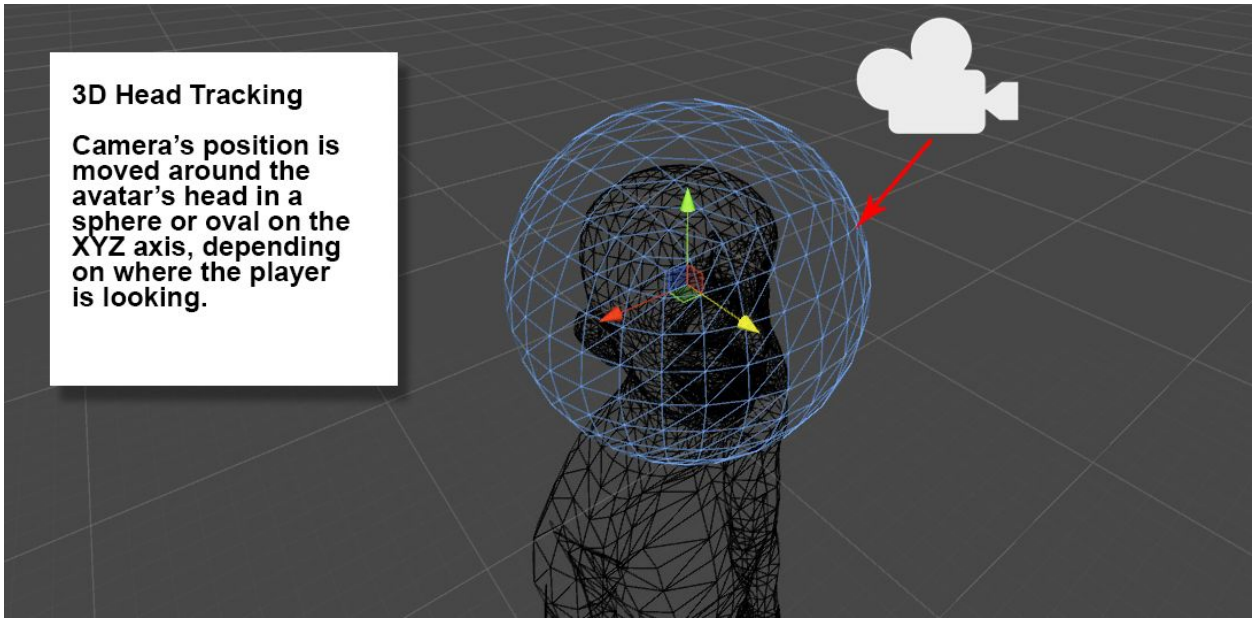
None: Default VR head rotation. The camera stays in a fixed position (0,0,0) at the head’s location. This is problematic because when I look to the left, my eyeballs should be over my left shoulder and when I look right my eyeballs should be over my right shoulder.
Without moving the camera to where my physical eyeballs are located, showing a VR body from the correct perspective is impossible.



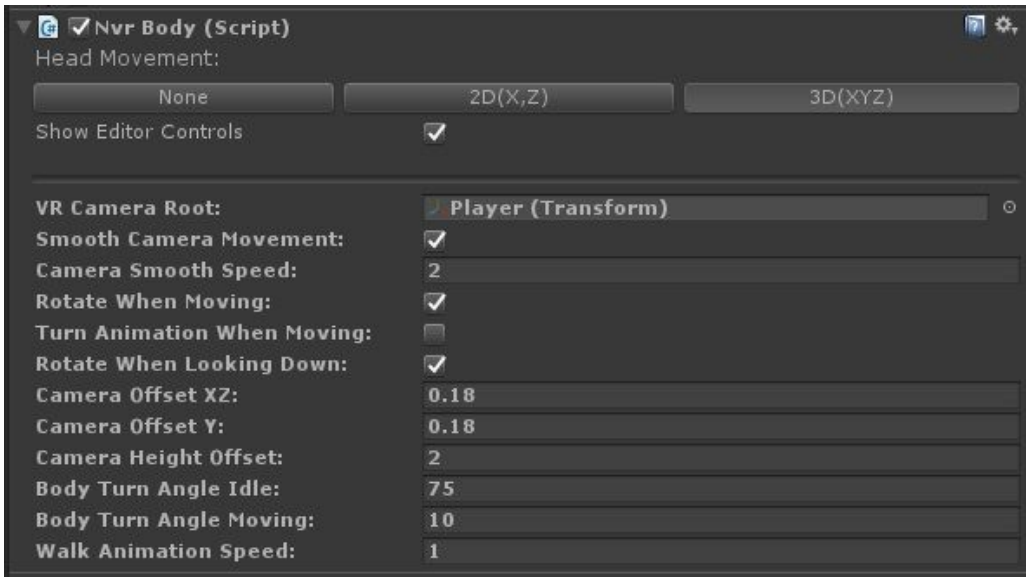
2D Movement: Camera moves on the X and Z axis only, in a perfect circle around the head. This method moves the camera away from 0,0,0 at a defined offset value on the X and Z axis. However the Y position remains constant.



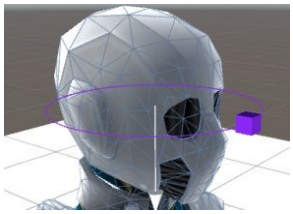
3D Movement: Camera moves on the X,Y,Z axis to try to simulate the most realistic head and eye movements. When looking up and to the left, the camera will move into a position also up and to the left, depending on XZ and Y offset values.



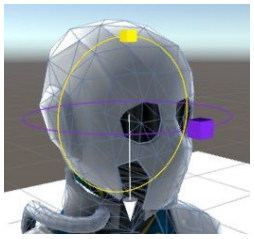
Parameters



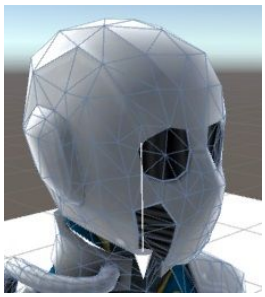
1. **Show Editor Controls:** Show the in-scene controls for manipulating the *Camera Offset XZ*, *Camera Offset Y*, and *Camera Height Offset* values.
2. **VR Camera Root:** The 'Player' Transform. This is the root parent of the Main Camera.
3. **Smooth Camera Movement:** Smooth the camera's movements via Lerp.
4. **Camera Smooth Speed:** How fast the camera moves during a smooth movement.
5. **Rotate When Moving:** Should the body rotate when walking / moving? If you are using a waypoint movement system, you may not want to the body to rotate when it's moving between waypoints.
6. **Turn Animation When Moving:** Should the '*turn left*' and '*turn right*' animations play when the player is moving? It does not look good to play to the turn animations while the walking animation is playing.
7. **Rotate When Looking Down:** Should the body rotate when looking down? If the camera is looking straight down (or upside down), do not rotate the player's body. This is useful is you want to show a menu when looking straight down.
8. **Camera Offset XZ:** This is the offset on the XZ axis between 'center of the head' (0,0,0) and the camera position. The position is drawn in purple in the editor and you can adjust it by clicking the purple box. Because purple is a mix of red and blue :)



9. **Camera Offset Y:** This is the offset on the Y axis between ‘center of head’ and camera position. This adjusts how far the camera will move up and down when the head rotates up/down. This value is drawn in yellow and can be adjusted by clicking the yellow box.



10. **Camera Height Offset:** This is how far from the Character’s Feet to move the camera UP, so that it is in the correct position. This should be the height of the eyeballs and it is shown in the editor as a White Arrow. You can adjust the height with the white arrow.



11. **Body Turn Angle Idle:** When the character is idle and not moving, at which angle should the body turn to match the camera’s position? Normally when we stand and look around in VR, our head turns some distance to the left and right before we pick up our feet and turn. This is probably about 90 degrees, because after a 90 degree turn of the head, we must really stretch to look behind ourselves.
12. **Body Turn Angle Moving:** Same as before but when we are moving we will tend to walk looking straight ahead of us, so this angle should be reduced a lot.
13. **Walk Animation Speed:** How fast to play the walk animation. You can adjust this value so the feet will not look like they are ‘sliding’ on the ground when walking.

Additional Notes and Tips

Unfortunately, in this situation beheading is a good thing.

To make this quick and easy you can use a tool like QuickEdit from ProCore:
https://www.assetstore.unity3d.com/en/#!/content/4464

Motion Sickness
Maybe this tool can help reduce motion sickness (more accurate camera-to-eyeball movements). Maybe it can make motion sickness worse if the camera is moving too much. I'm not affected by motion sickness, but when the XZ and Y Offset Values go too high, and the camera moves far, it does not look good.

I separated the Y and XZ offsets because I think it's possible the most accurate movements will be more of an OVAL shape, because when I look left and right my eyes seems to move further than when I move my head up and down. In other words, on the Y axis, I believe the eyes stay a lot closer to 0,0,0 than they do on the X and Z axis.

Help And Support

For a video tutorial related to this asset, please click here:
http://www.youtube.com/watch?v=dY5xMaYuFbE

Join our VR community here for VR tutorials and videos:
https://www.youtube.com/nurfacegames/

For any questions or support, please email:
nurfacegames@gmail.com