

Mirror Toggles with UI

<https://www.youtube.com/embed/m02WrMnseVE?t=1s>

Let's get to the moment most of you have been waiting for, adding a mirror! In the project tab, search for mirror. Drag VRMirror into your scene, rename it to HQ Mirror, then position, rotate, and scale how you want it. If you imported VRWorldToolkit (like you should have in part 1) then you will see a button that says Show Players/World. Click on it, and then under Reflect Layers also enable Pickup and Walkthrough. Now disable the mirror.

OK, we have a mirror, but how do we toggle it on and off? This is where UI comes in. We want to have a toggle we can click, so in the Hierarchy, right click, UI>Toggle. This will spawn a UI toggle parented to a canvas, which is a necessary component to display UI elements; however, we need to fix a few things first. In the Hierarchy, click Canvas, then change its Render Mode to World Space. Under Rect Transform, change the XYZ scale to .001. Then change the layer from UI to Default. Add two components, a box collider and a VRC Ui Shape. Now position it against a wall; turn off progrid so you can position it slightly off of the wall, not inside of it.

In the hierarchy, under Canvas click toggle and set its width to 700 and height to 120 and untick Is On.

In the hierarchy, under toggle click label. Make the text "HQ Mirror Toggle", change the color to white and increase the font size. You can change the font, and you can even import custom fonts by dragging in a ttf font file from C:/Windows/Fonts. You can also add a shadow by clicking Add Component and searching for it.

Click Background in the hierarchy and change the width and height to 100. Then change the offsets to 50 and -50 to position the checkbox correctly. Then click Checkmark in the hierarchy and change the width and height of that to 100 as well. Finally select the label again and set left to at least 100 (you can use the gizmos to translate UI elements the same as other objects).

Now under toggle, set navigation to None. Click the plus icon at the bottom of the component where it says On Value Changed and drag in the mirror from the hierarchy. Click the dropdown that says No Function and select GameObject>SetActive under dynamic bool.

Duplicate your HQ Mirror, rename it to LQ Mirror, click show players only, and duplicate the button we just created. In the toggle drag in the LQ Mirror in place of the HQ one and change the label to LQ Mirror Toggle.

To ensure that only one mirror is on at a time, add a new slot, drag in the HQ Mirror Toggle, and under the dropdown select Toggle>isOn under static bool and leave the box unchecked. Do the same for the HQ Mirror Toggle, with the LQ Mirror Toggle in the slot. However, if you have a mirror on and want to select a different one, you have to click the button twice, once to turn the current mirror off and another time to turn the desired mirror on.

Press the play button to test this out by using CyanEmu (which you also should have imported in part 1).

If you want to make this UI but using Udon you can check out Vowgan's video!

<https://youtu.be/E0D9Z8-HVBI>

Move one of the mirrors slightly in front of the other to avoid z-fighting.

If you have a larger world, note that the LQ mirror will render avatars behind it, even if they are through walls. For the sake of performance and privacy it would be wise to position mirrors where they are not going to do this, or even design your spaces where mirrors will go with this in mind.

By the way you'll probably want to organize your hierarchy at this point. You can essentially use empty gameobjects as folders, but make sure you reset their transform first or else it will mess with the position of the child gameobjects which will mess with progrid's! If you forget and don't reset the transform, drag out all of the child gameobjects, reset the transform and then drag them back in.

A bit of a bonus, I had a question asked in the comments about how to do neon lighting. First add in an object you want to give neon to, so I'll go into the probuilder window and add a skinny cylinder with no height segments. I'll extrude it to form it how I want. I can create extrusions and then rotate with a center pivot to create turns in the neon tube. After you finish modeling, create a new material and check the emissive box and set it to baked. Apply it to your object and select the color you want it to emit. Bake and you're done!

A few tips with this: one, you can enable bezier curves (and booleans!) in probuilder by going to Edit>Preferences>Probuilder>Experimental Features Enabled. However, neither of these work great in probuilder so I would recommend doing them in Blender instead.

Two, make sure the lighting fits your scene. If you are going for a warm or natural look you will want to limit yourself to the colors on the kelvin spectrum. If you are going for an artificial look you can use colors on the full visible light spectrum. This is covered in the Blender Guru lighting course I recommended at the beginning of lesson 4.

That's it for this lesson. The next one will be a continuation of this one, where I'll show you how to toggle objects between the default and walkable layers by using UI with Udon!

Revision #1

Created 2022-02-19 22:52:31 UTC by Iconoclass

Updated 2022-02-19 22:53:30 UTC by Iconoclass