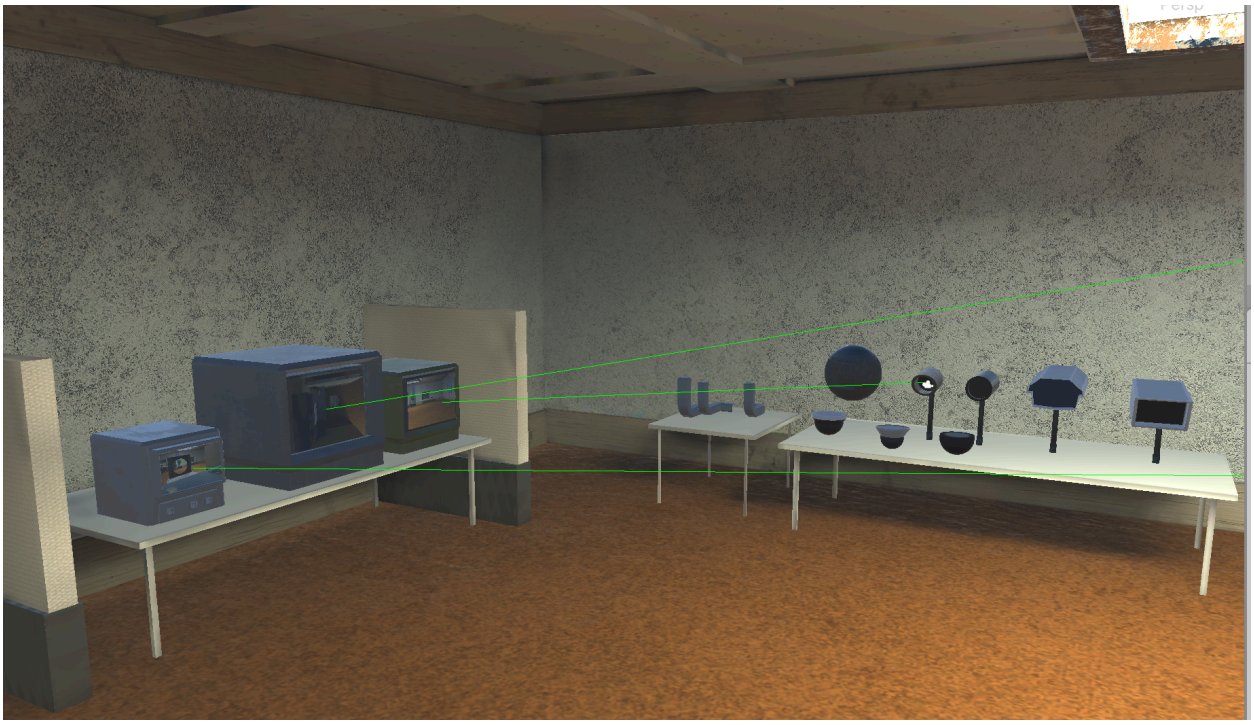
CCTV Camera System



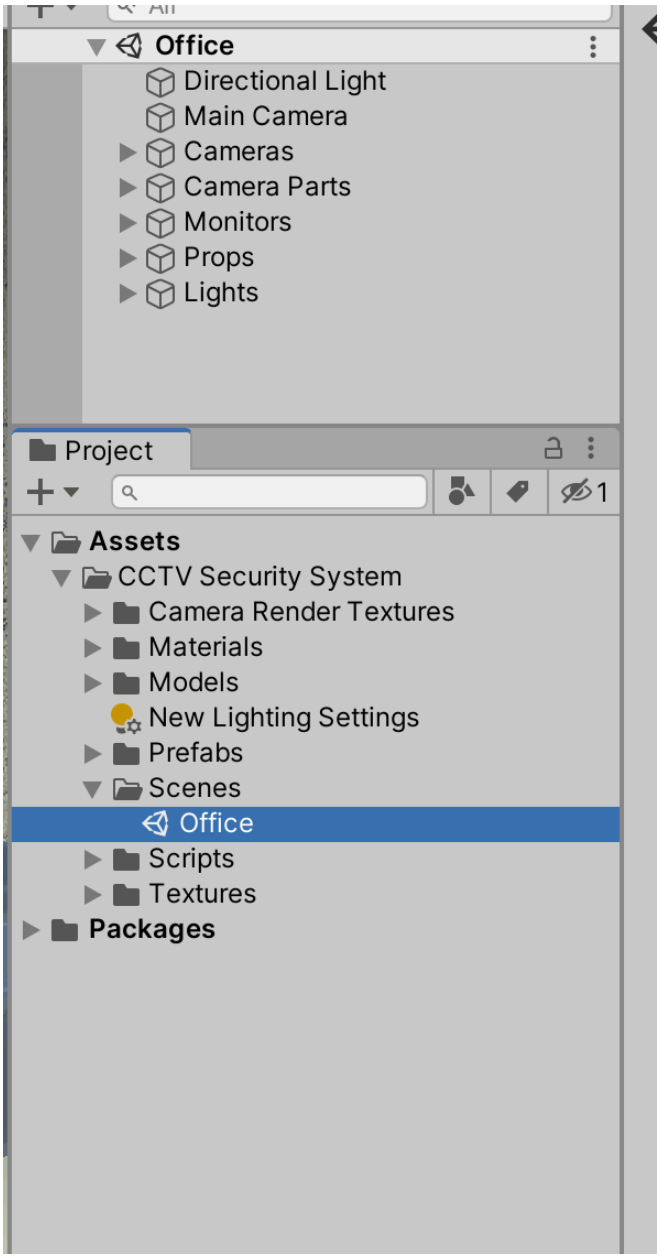
Hello and welcome!

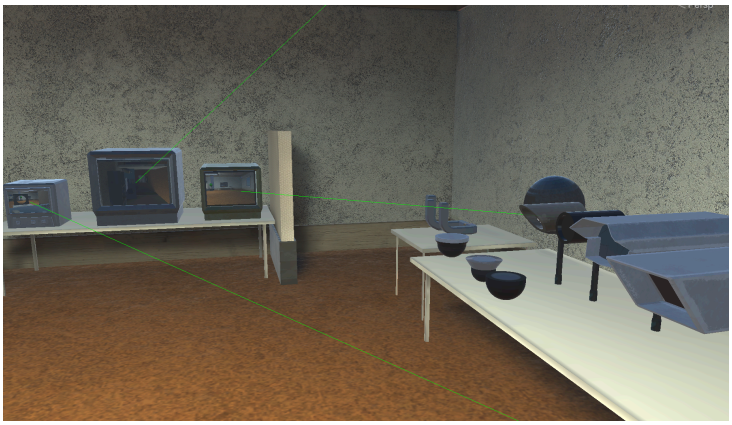
First, I want to thank you for purchasing this asset. I really appreciate the support!

If you have any questions that this README does not answer, please don't hesitate to reach out to me at: Alan@AlanOToole.com

What's Included?

This asset includes an office scene that demonstrates a CCTV Camera System based on Unity render textures.

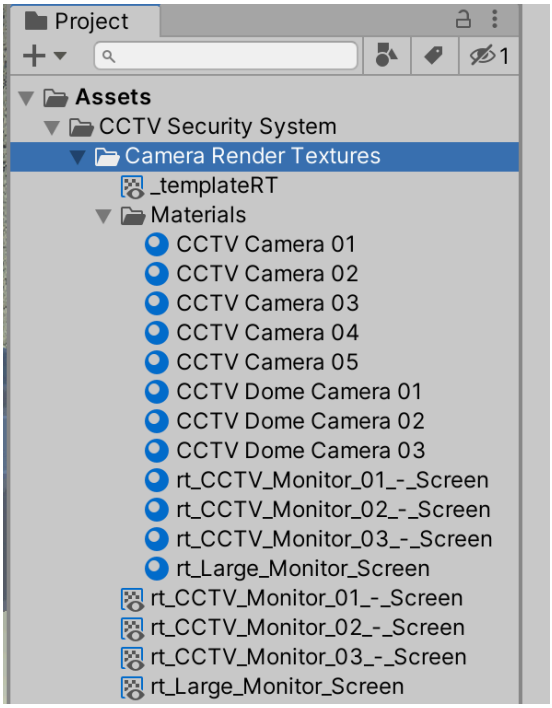




Folders

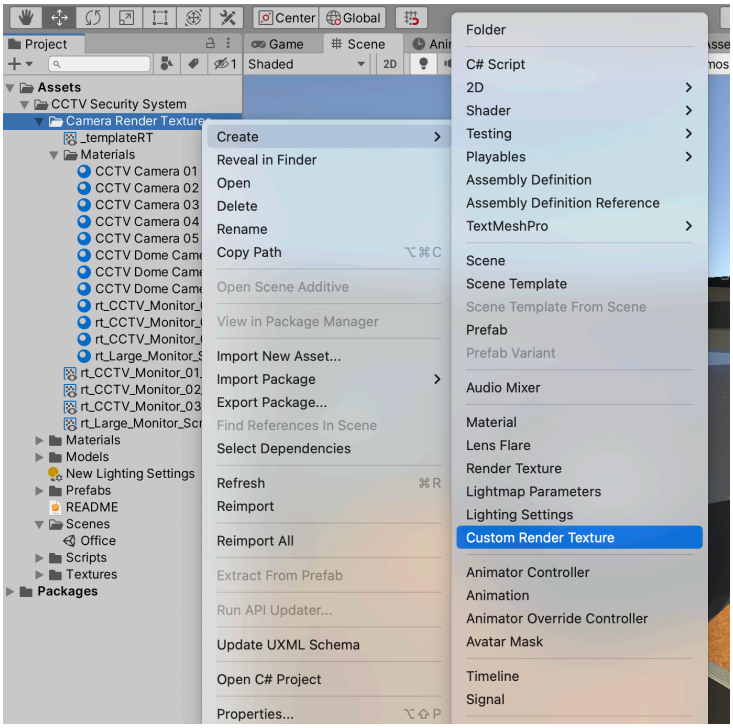
- Camera Render Textures
  - This folder contains the Render Textures that are used from a camera in the Scene
- Materials
  - This folder contains the materials for the models included
- Models
  - This folder contains the models of CCTV Cameras, monitors, and office equipment
- Prefabs
  - The prefabs for each model included
- Scripts
  - This contains scripts included with CCTC Camera System
- Textures
  - All textures for the models included are in this folder

Camera Render Textures



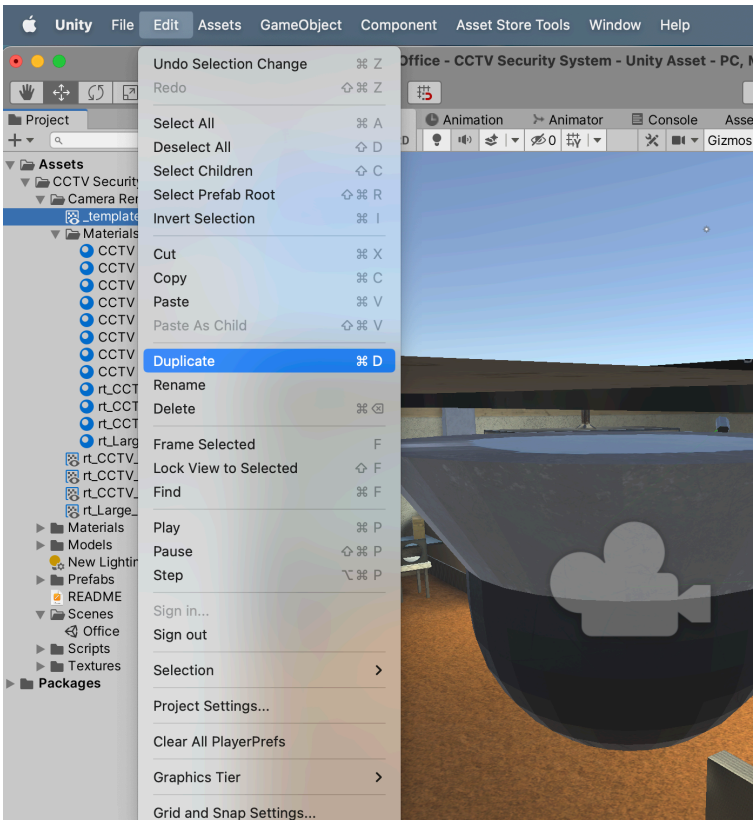
Camera Render Textures are how we render (draw) what the CCTV Camera shows on a Monitor.

The Render Texture can be created in two ways.



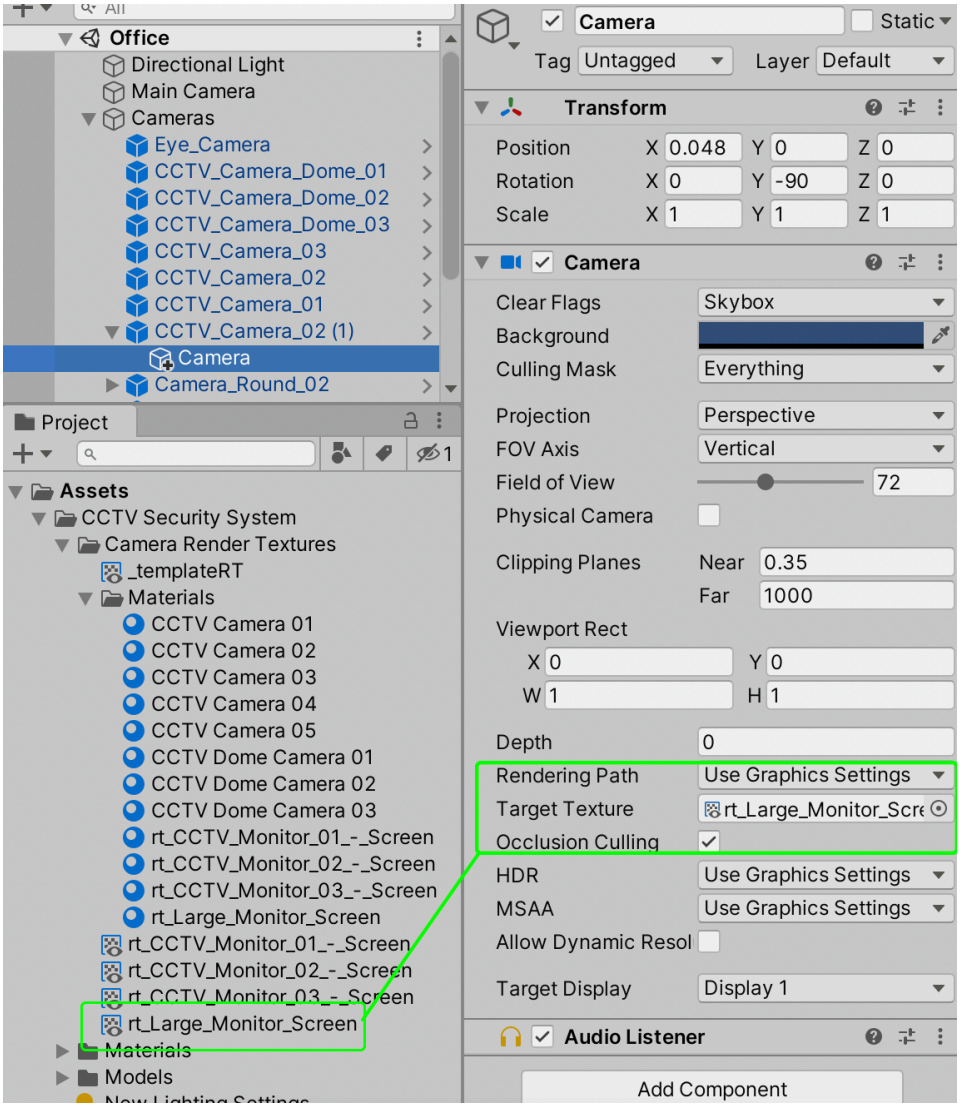
The first method is to right click on the **Camera Render Textures** folder, and click **Custom Render Texture**.

The second method is to click on **_templateRT** and go to **Edit -> Duplicate**



How to use the Render Texture?

When you create a new **Render Texture**, you'll need to assign it to a **Camera**.



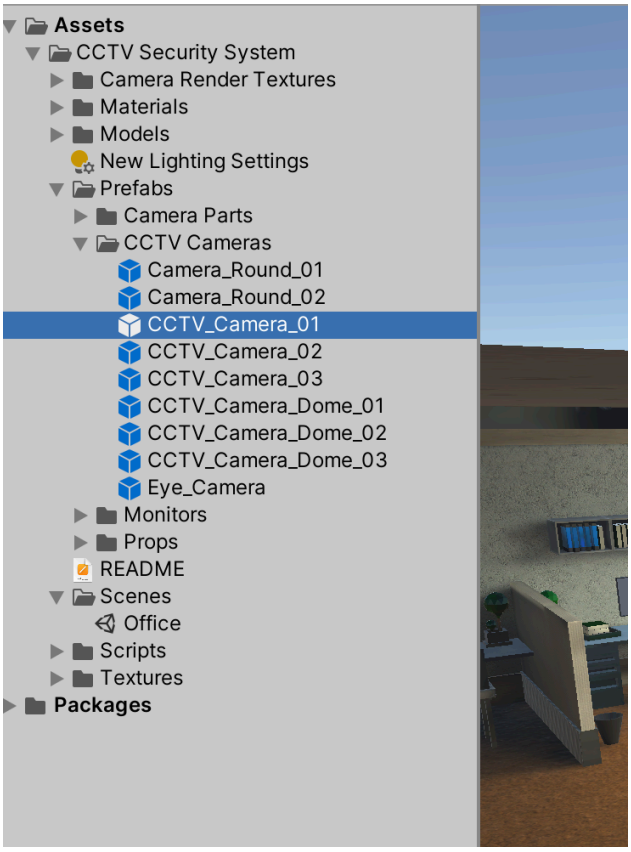
The first step is to find your **CCTV Camera** in the scene.

If you do not have one, you can drag and drop one from the **Prefabs** folder.

Once you have the **Camera** created, find the **Camera** component and drag and drop your new **Render Texture** on the **Target Texture** variable.

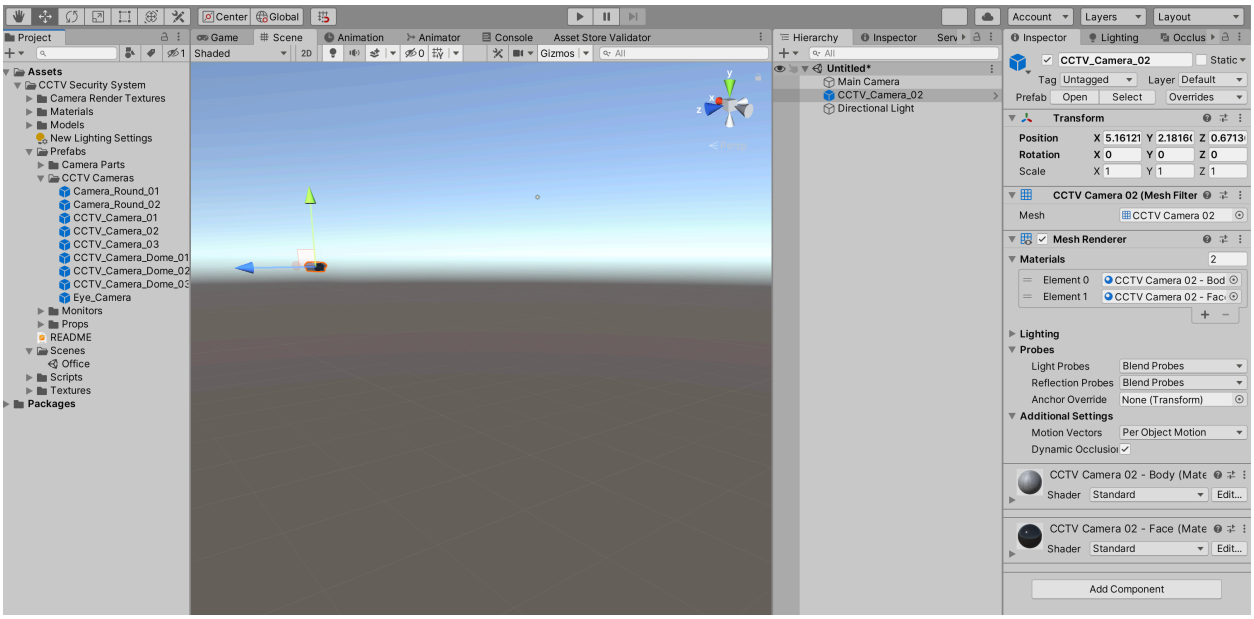
The screenshot on the left will show you this connection.

How to create a new CCTV Camera?

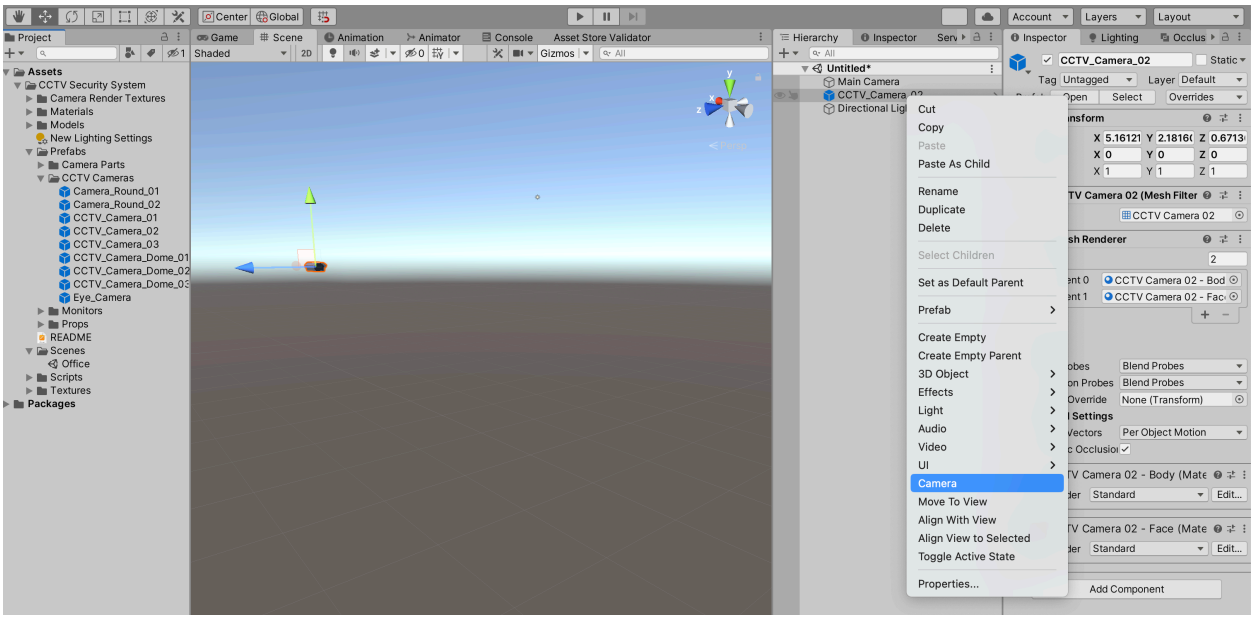


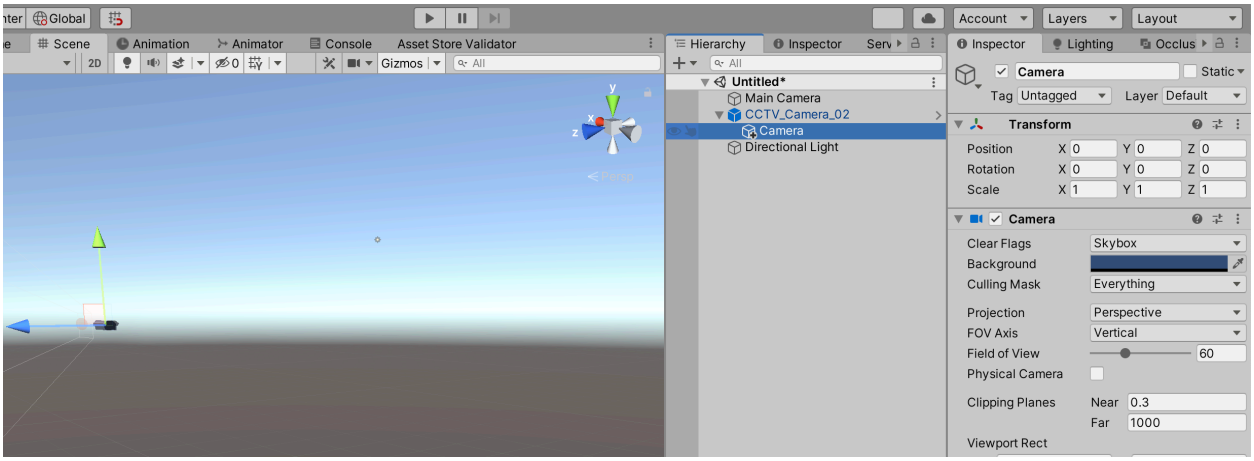
Find a **CCTV Camera** model you like in the **Prefabs** folder

Drag and drop this **Prefab** in your Scene

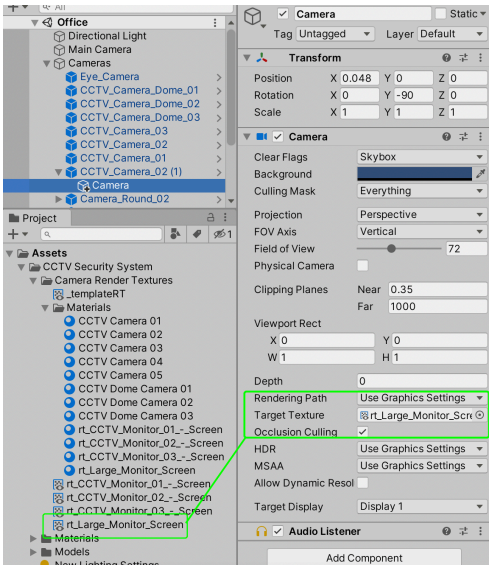


Once in your scene, right on the new **GameObject** and add a **Camera**



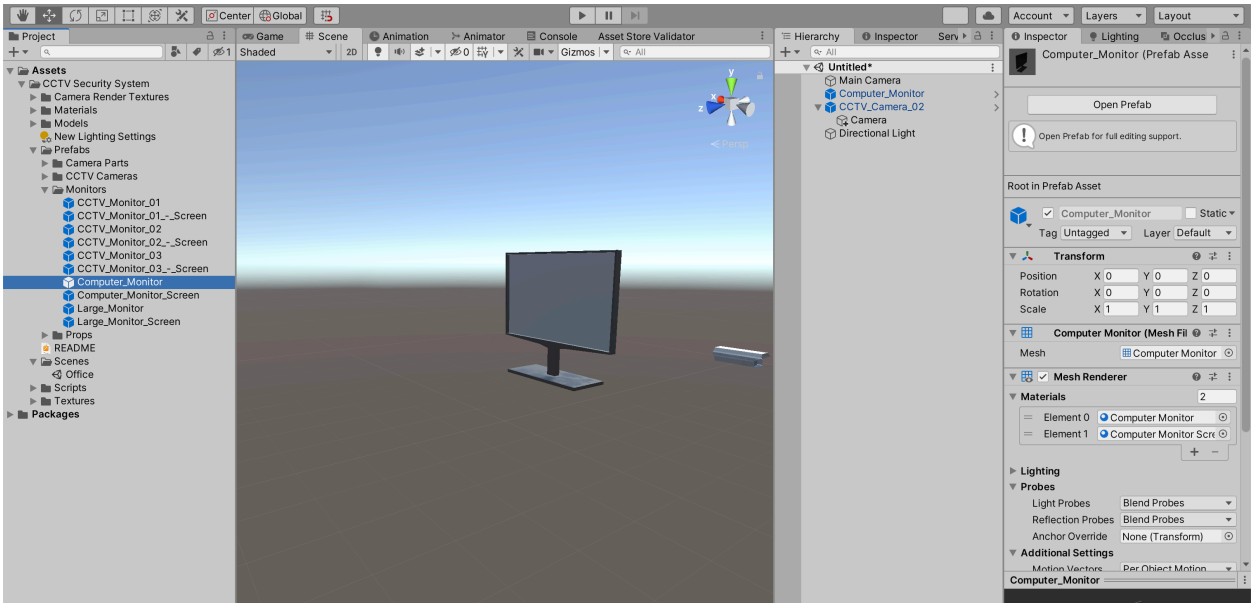


Once the **Camera** has been added, you can now assign your **Render Texture** to the **Target Texture**



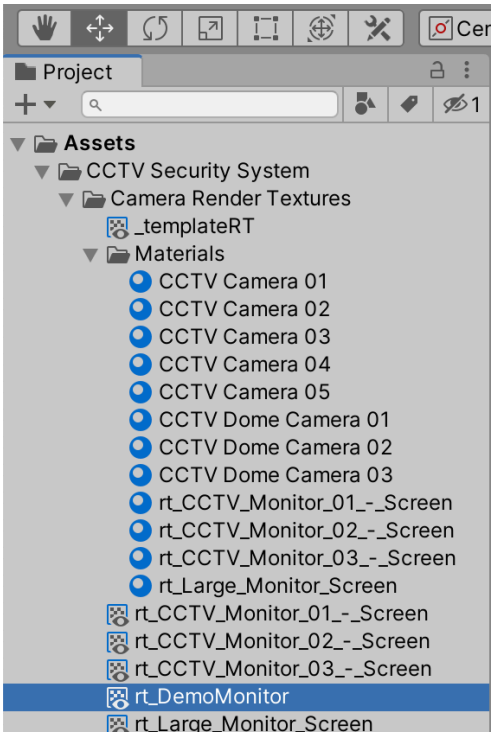
How to create a Monitor?

Find a **Prefab** of a Monitor you like and drag and drop it in your scene

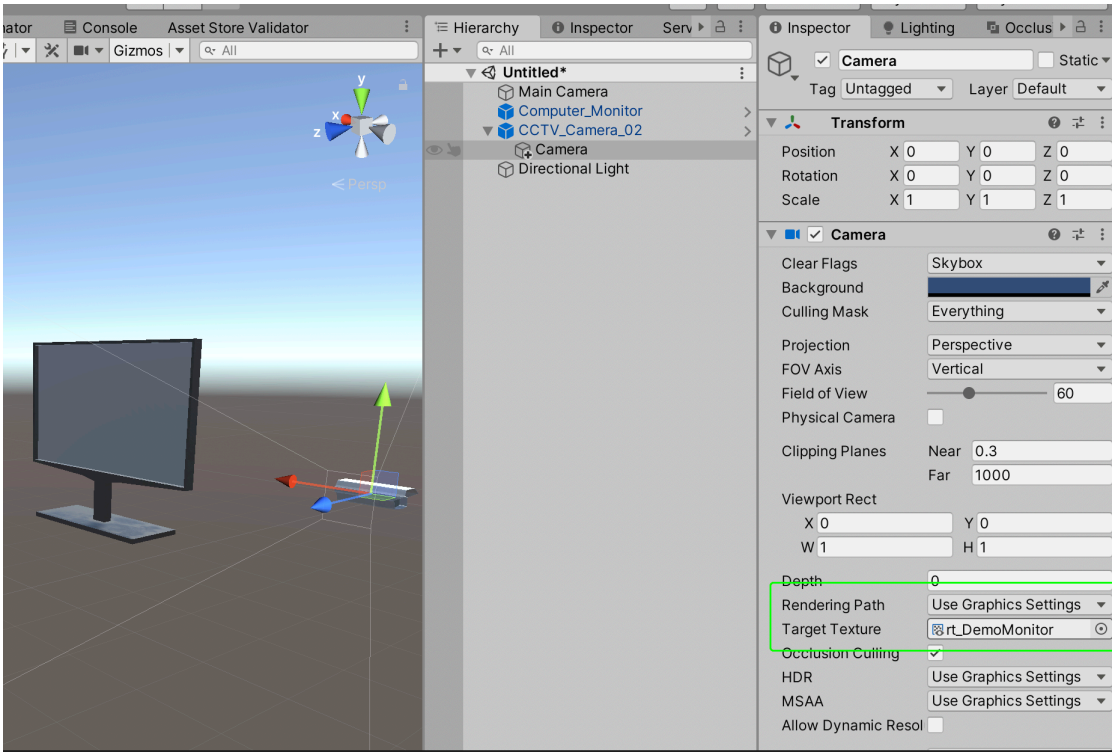


Once in your scene, find your **Render Texture** you want to display on this new monitor. If you do not have one, you can create one with the instructions in this document.

In this example, I created **rt_DemoMonitor**.

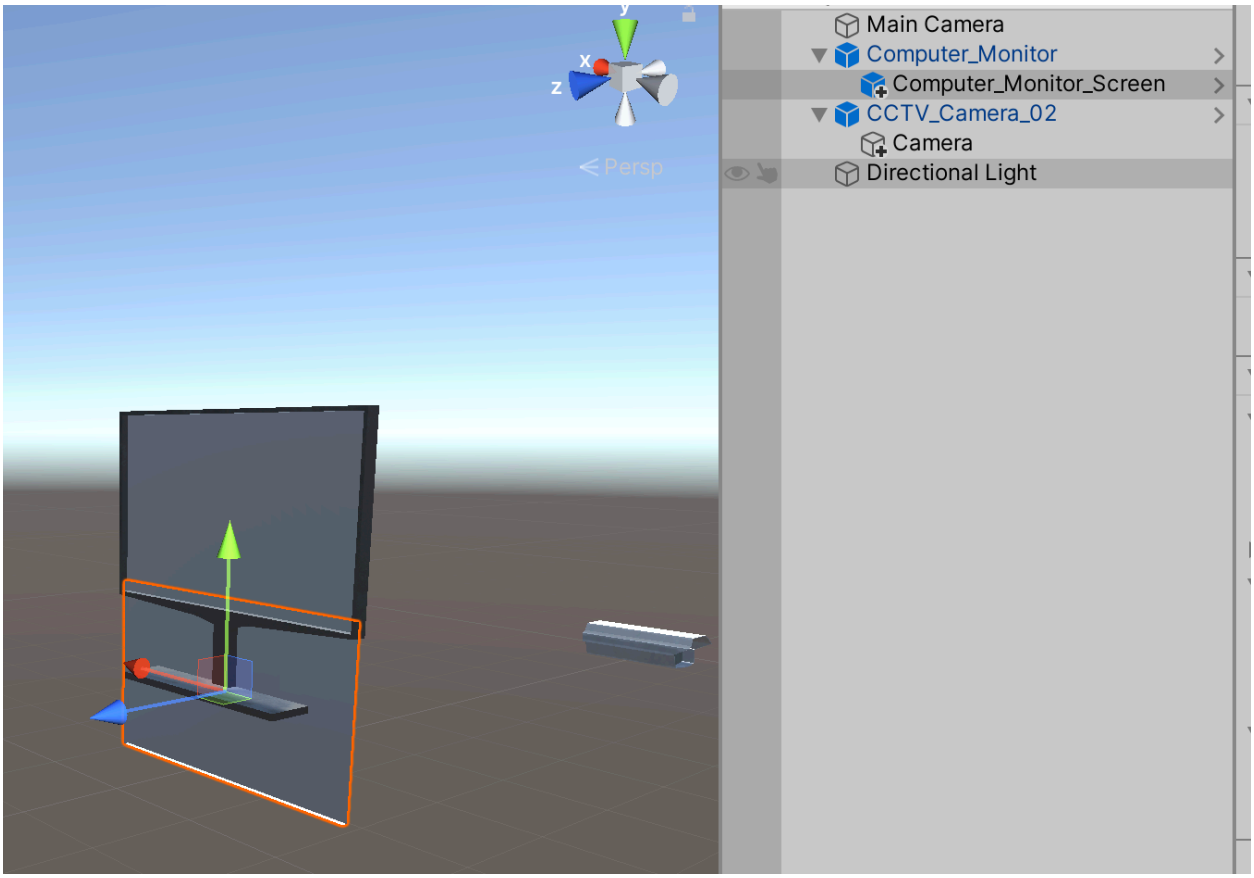


Once created, drag and drop this to your **Camera** you created over the **Target Texture** variable.

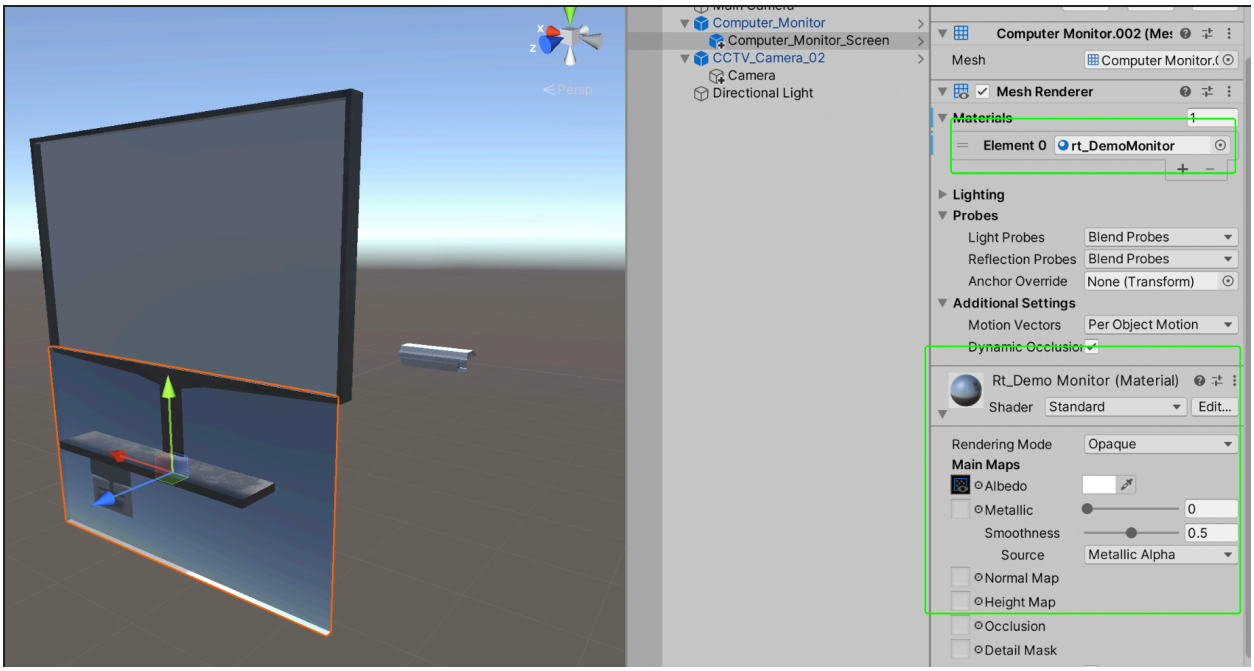


This **Camera** will now draw what it “see’s” on this new **Render Texture**.

The next step is to create the **Monitor Screen**. In the **Monitors Prefab** folder, you’ll see a **Screen** for each monitor. Drag and drop the appropriate one on your new **Monitor**.

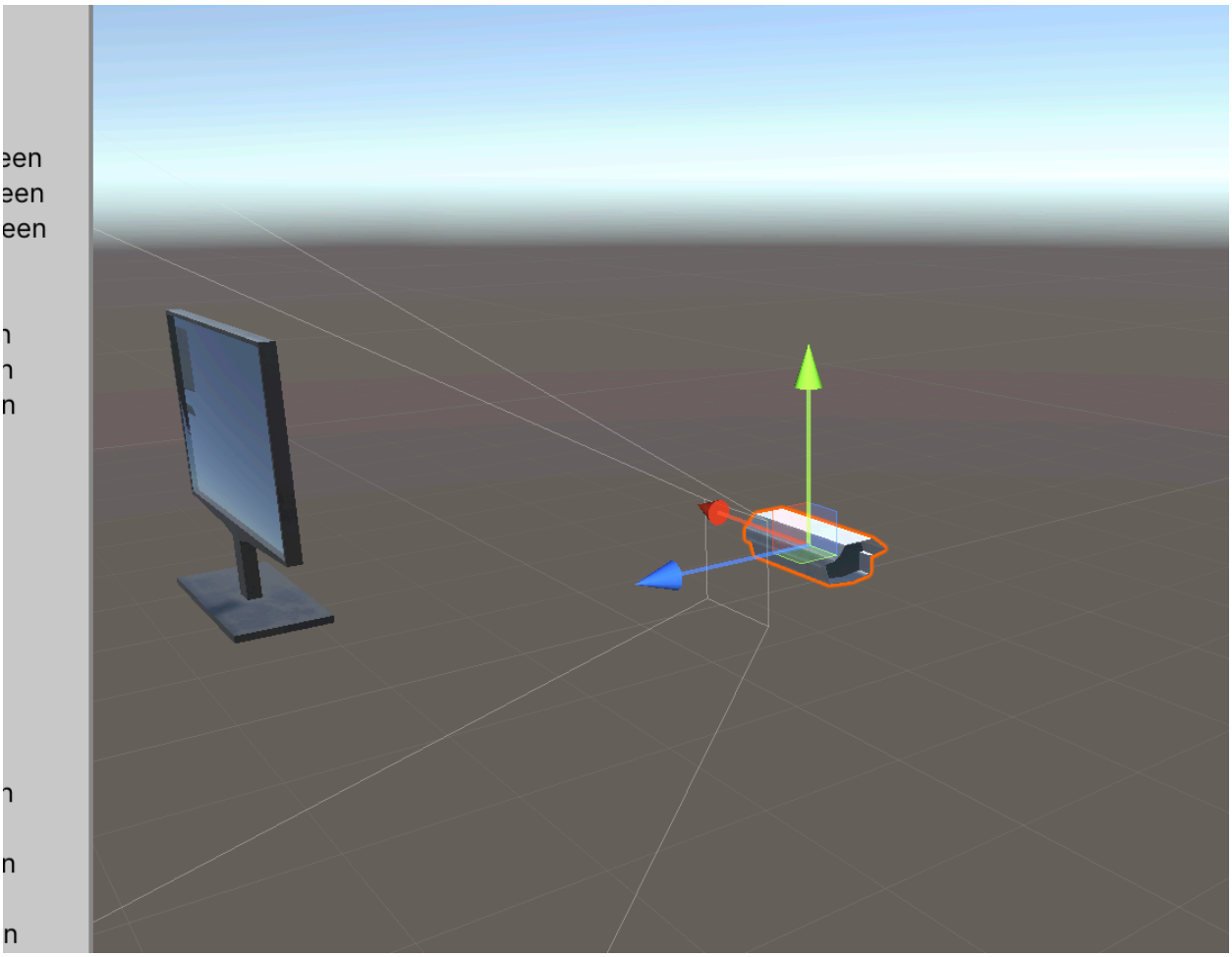


Now you have the **Screen** created, find your **Render Texture** and drag and drop it on this new **Screen**.



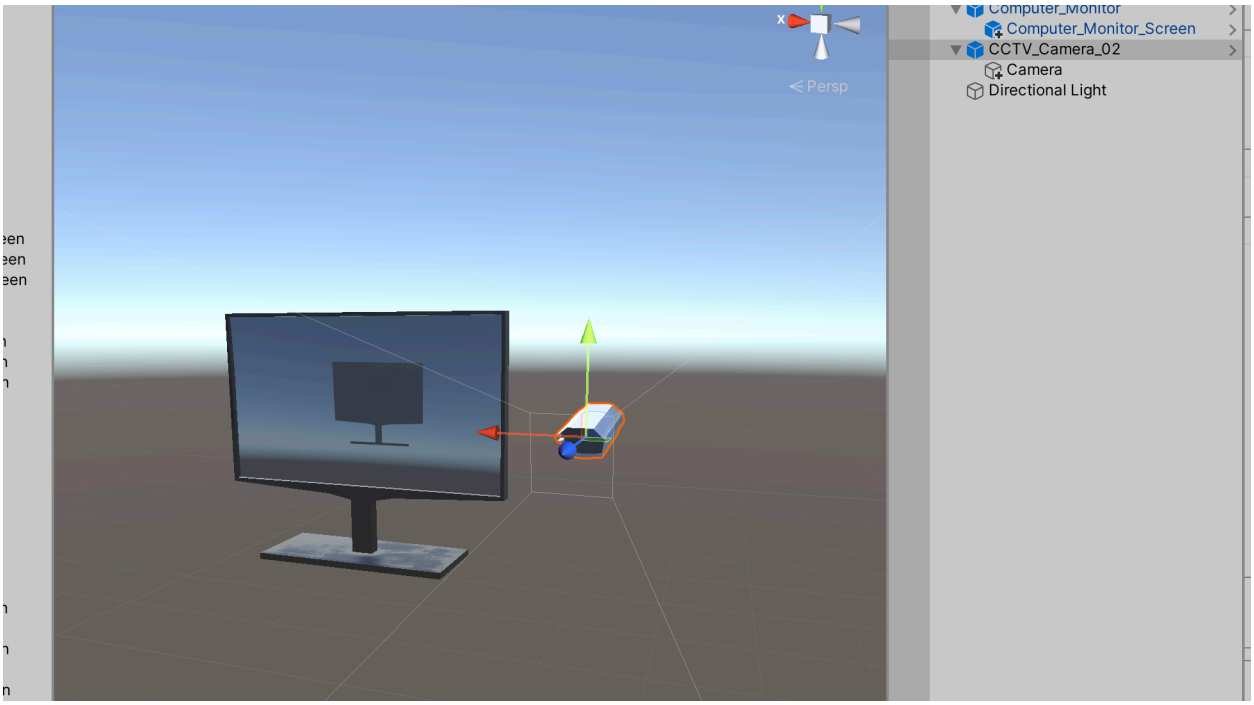
You'll notice that the **Material** updates for the new **Screen**. You may need to **rotate and move** the screen to see the **Render Texture** and align it with the new **Monitor**.

The same is true for the **CCTV Camera** too. You may need to **rotate and move** the **Camera** you created



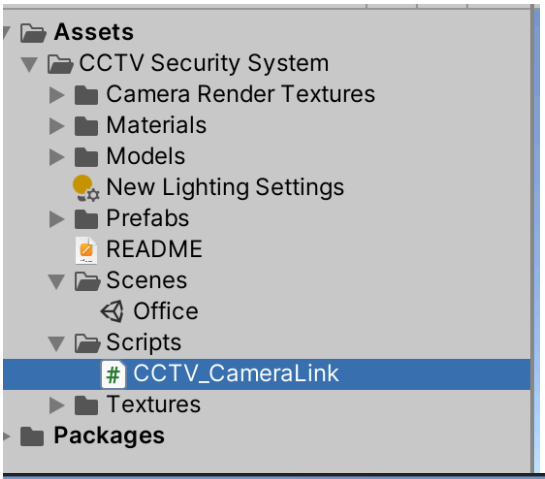
You can see the camera is facing down the **Blue Z Axis** while the camera is facing down another. Simply **rotate** the Camera to match your own needs.

You can see that the **Camera** is now **Rendering** what it can see on the **Monitor** we created



How to use the CCTV_CameraLink script

The **CCTV_CameraLink** script allows you to **visual** the link between a Camera and Monitor.



Simply, drag and drop the **CCTV_CameraLink** script on a **Monitor**

Once on your **Monitor**, you'll see you can add a **CCTV_Camera** game object.

When you drag and drop your **Camera** on this variable, it will draw a green line showing the connection.