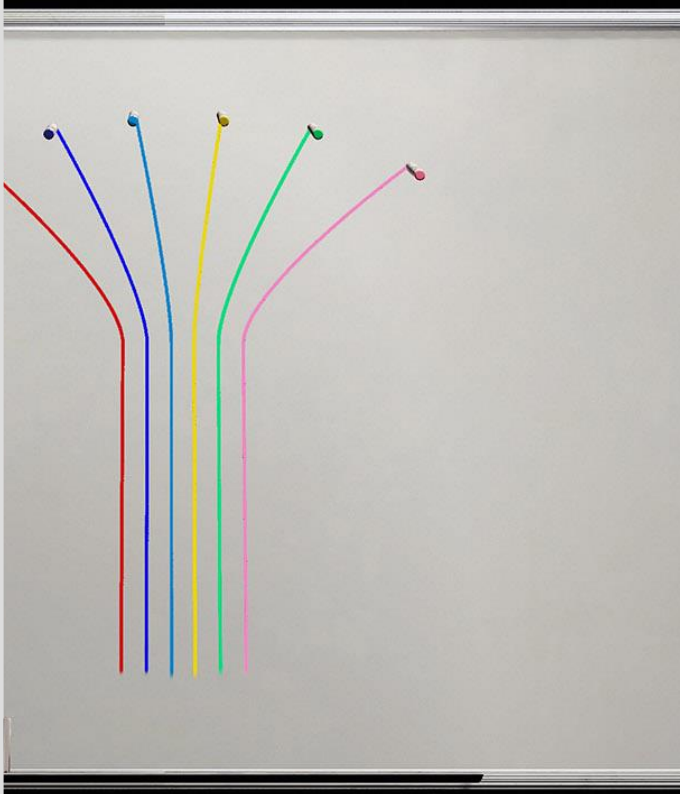
DRAW SHADER

V1.1.1



Thank you for purchasing my Draw Shader asset! This guide will show you how to set up the shader for your Unity scenes, how the shader works, and additional recommendations.

If you end up using this asset in a project, feel free to Tweet me a screenshot @SuggoCreations, I'd love to see what you do with it!

Please consider leaving a review at the store page. Your feedback is instrumental to the future of this package!

MADE FOR THE UNITY ASSET STORE

SUGGOCREATIONS.COM

INDEX

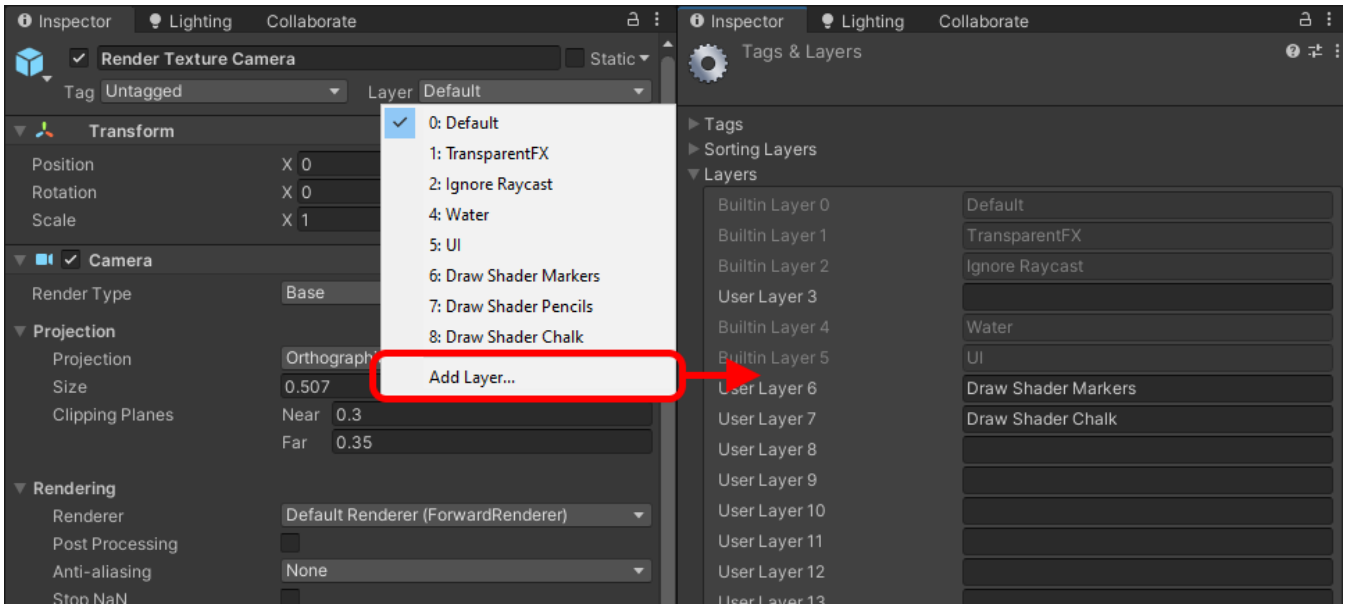
FIRST TIME SET-UP

Currently, there is no consistent way to import Layers with an Asset Store package. This means the first time set-up is a manual process of a few steps:

Step 1. Creating and Adding Layers

Drag in any the prefab you wish to use (Whiteboard, Chalkboard, etc.)

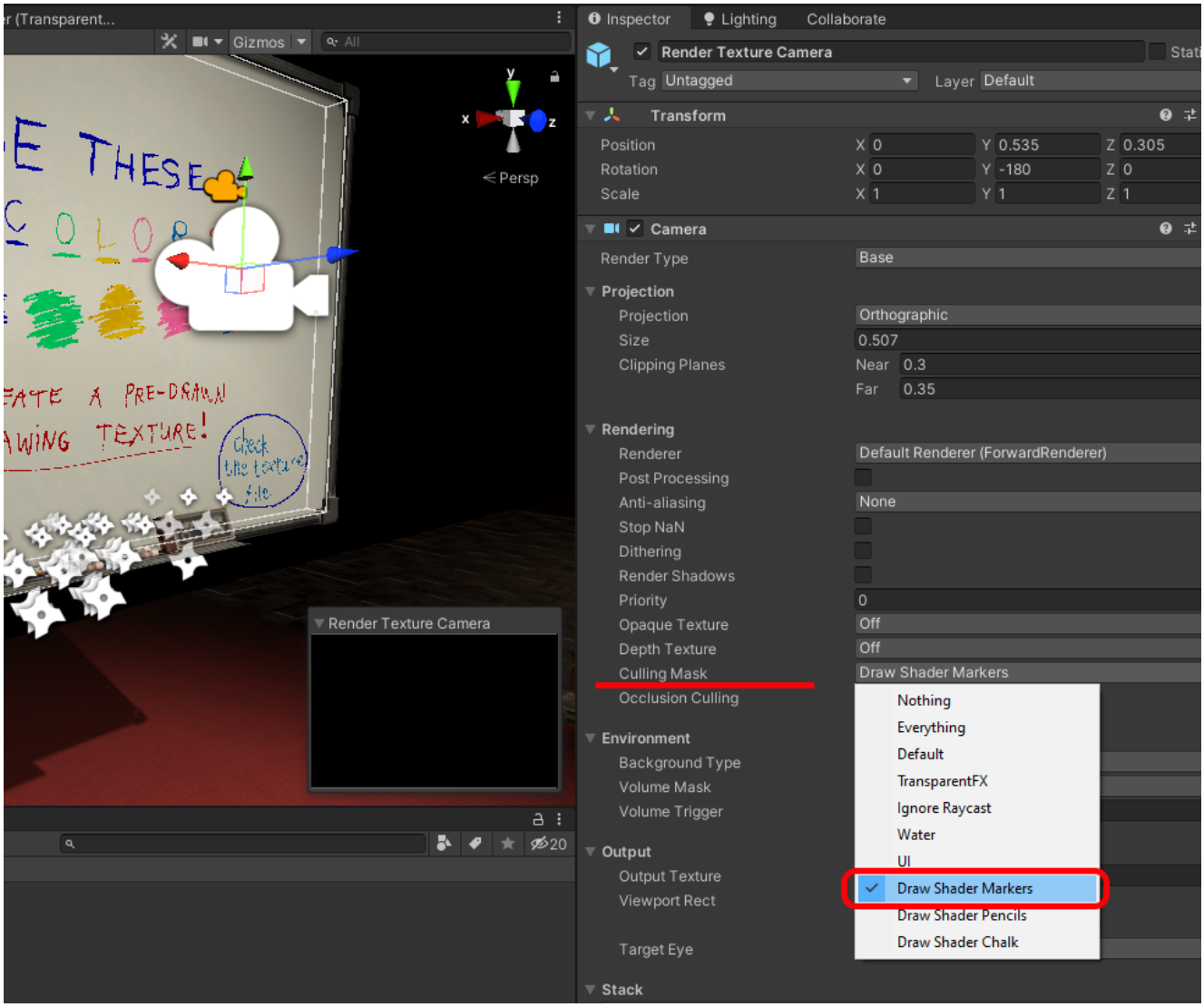
In any Inspector window, find the Layer drop-down and click on “Add Layer…”. To make a new Layer, rename it to something like “Draw Shader Markers”.



Step 2. Culling Masks

With the Layers in place, go to the Render Texture Camera on any prefab and view its Inspector. Make sure only the Layer that you want to use is selected.

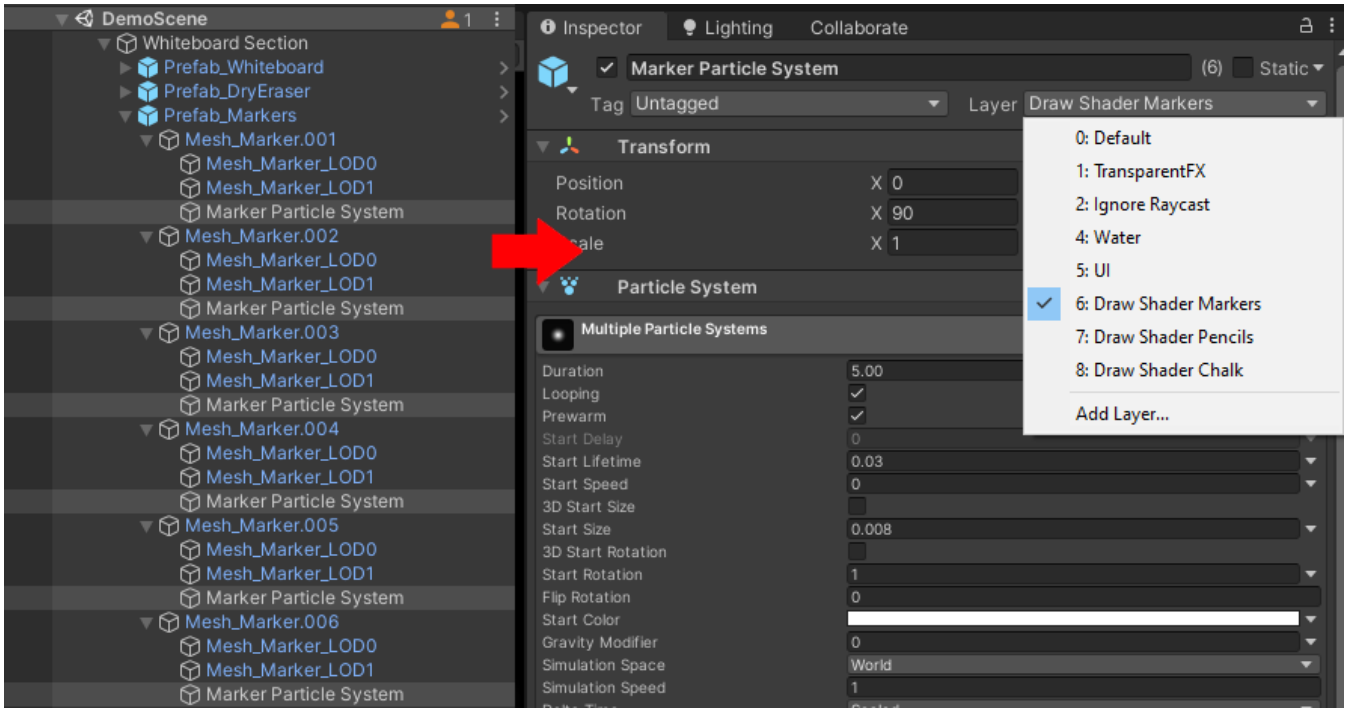
In this case, I only want to render the Layer I use for the Whiteboard Markers.



Step 3. Selecting Layers on Particle Systems

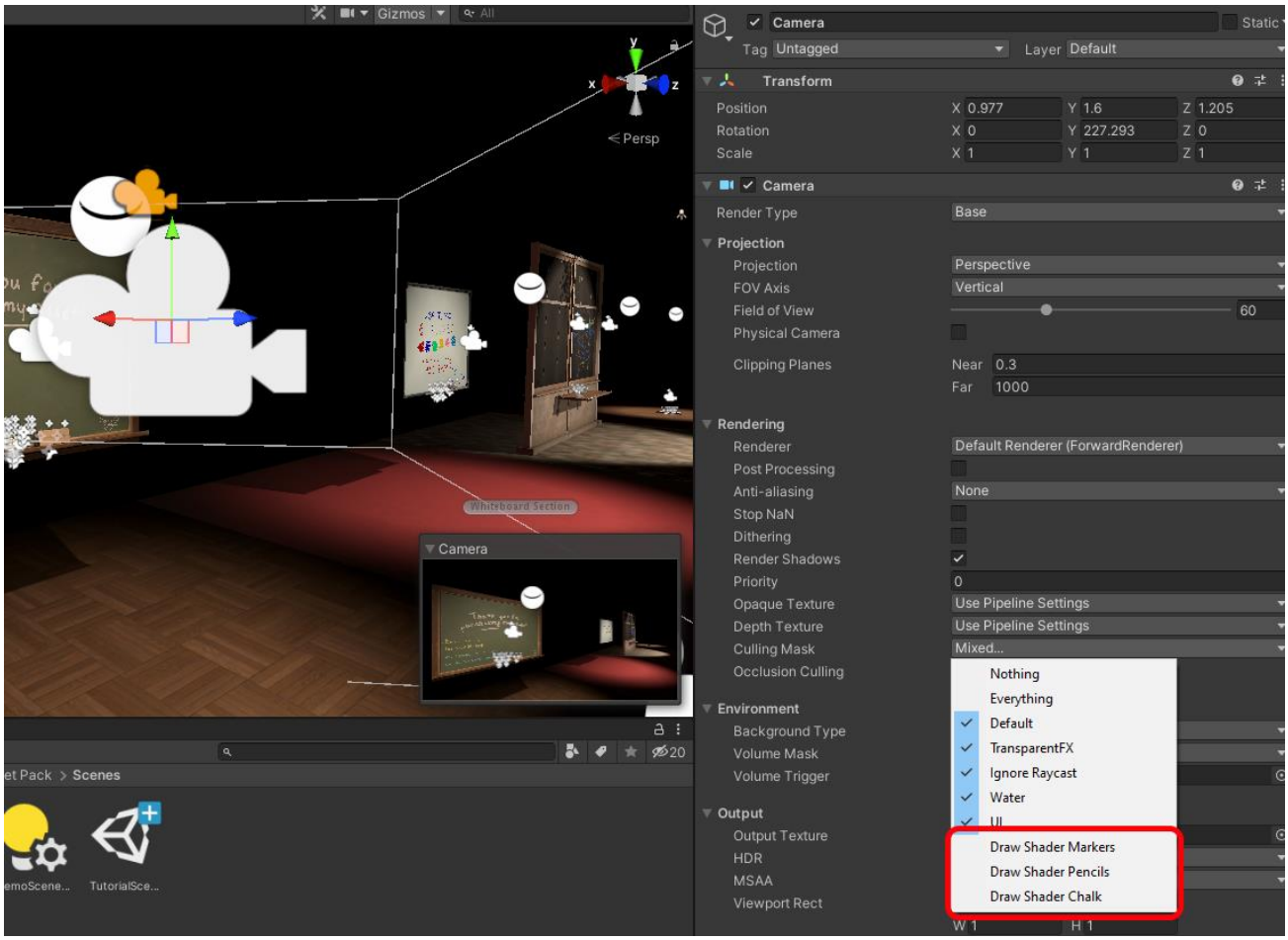
Select all the Marker Particle Systems (using ctrl+click) within the Markers Prefab.

Then go to the top right in the Inspector of the Particle Systems and make sure only your Draw Shader Markers Layer is checked.



Step 4. Removing Particle Trail from Game View

Go to the Camera you use to render your game view, and make sure the Layers the Draw Shader uses remain unchecked. This will remove the particle trail from the ends of drawing pens.

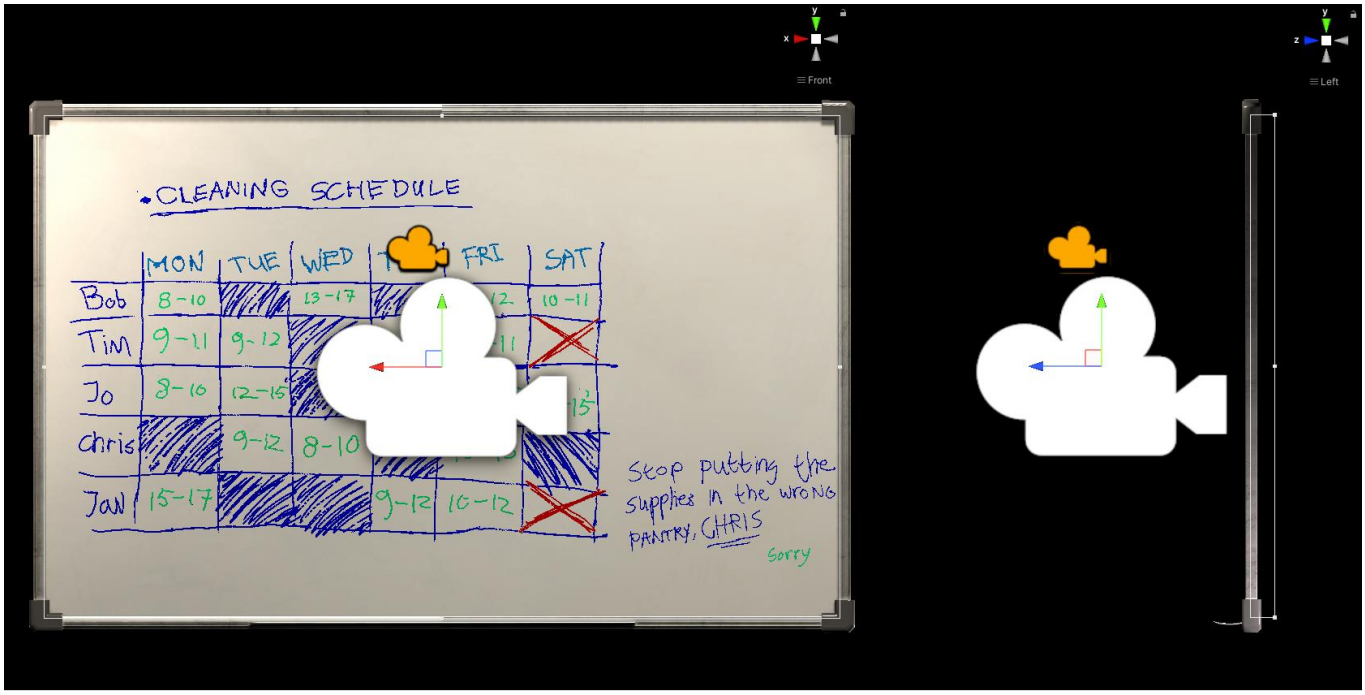


Your prefab should now be all set up and ready to go!

HOW IT WORKS

Render Texture

Using a **Camera in Orthographic mode**, we can render a thin sliver as a Render Texture. Making sure the **Camera aligns perfectly with the Drawing Board**'s surface, we can then feed this Render Texture as an input into the Draw Shader.

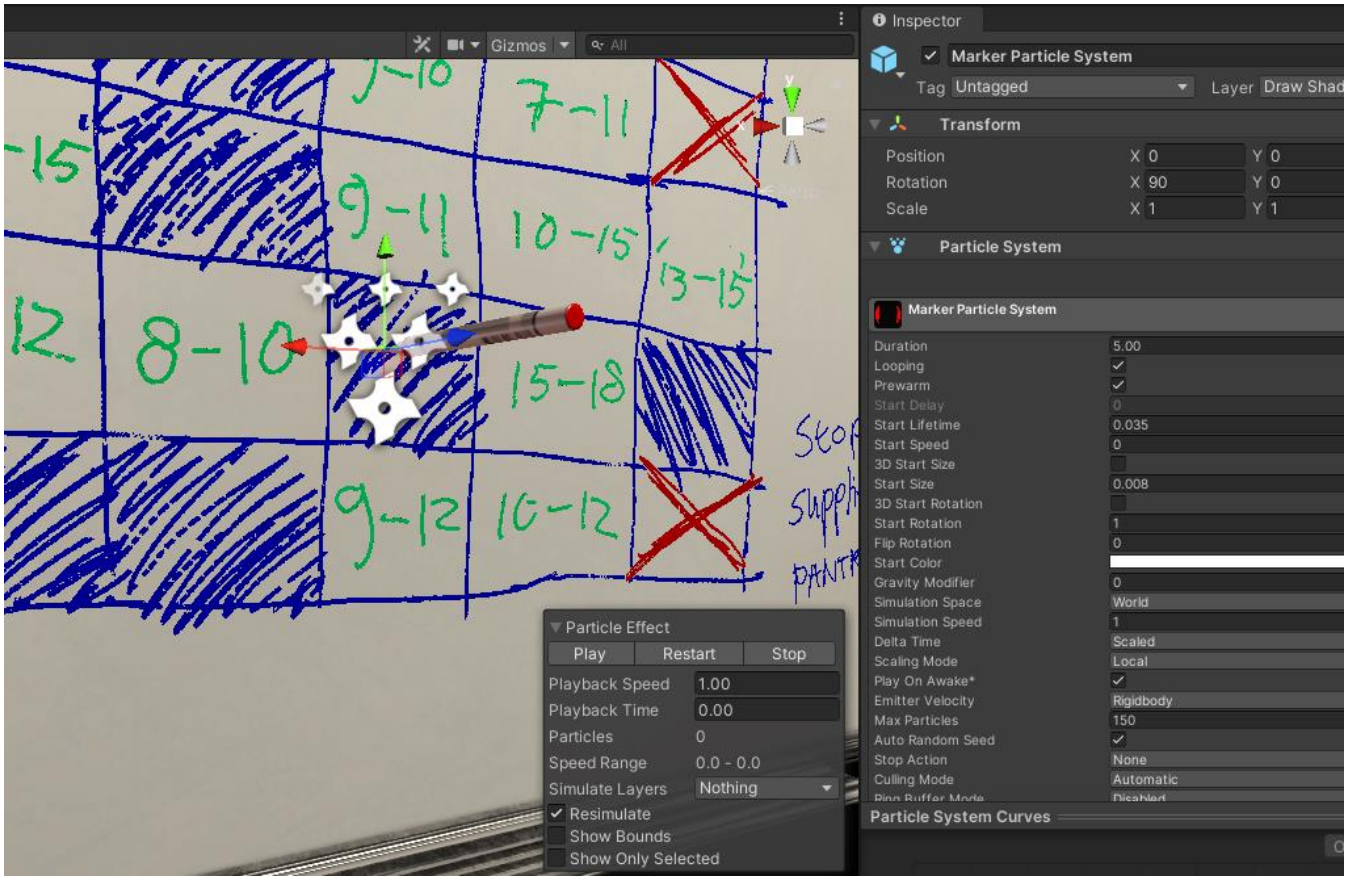


The Camera in Orthographic mode aligning perfectly with the plane of the drawable surface.

After that, we'll use *Unity's layer system* to use in the Camera's *Culling Mask*. This makes sure we **only** render the Drawing Pen's Particles onto the Render Texture. We will cover how to set up Camera *Culling Masks* using *Unity's layer system* in the next few segments.

The Particles/Drawing Pens

The Drawing Pens (for example *Prefab_Markers* or *Prefab_Chalk*) each have a **Particle System** at the tip/ends of the pen. These **particles get picked up by the Camera and rendered to the Render Texture**. This way we can change the shape of a particle, giving the pen's brushstrokes (the trail it leaves behind) a unique shape.



The Marker Particle System on the tip of a Marker.

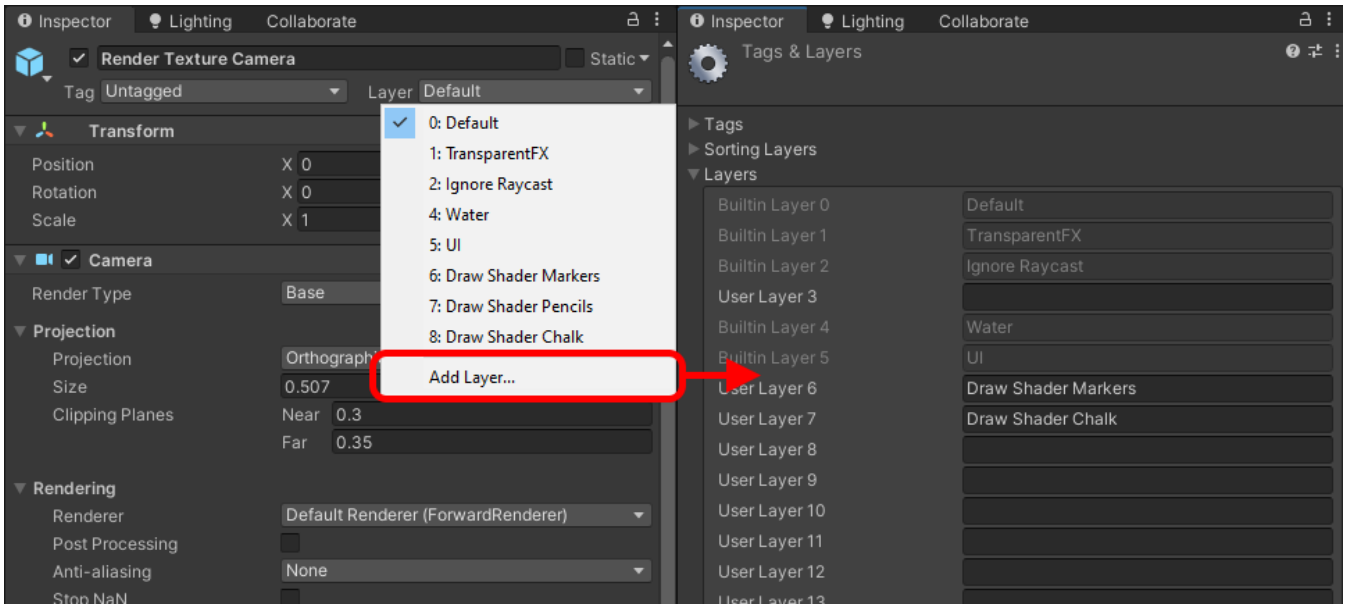
This Particle System is divided up by 6 Material Colors.

- Red, Dark Blue, Green, Yellow, Pink, Light Blue and White.

These Material Colors must remain unchanged, as they make sure the shader is able to separate between each color. With the exception of the White **Material Color**, which is used for **Erasing**. For more information on how the colors get separated, how erasing works, and how they get rendered onto the 3D object, check the Shadergraph. It's full of notes explaining the innerworkings of the shader! The Particle System also should be on a designated **Draw Shader Layer**, which will be explained in more detail next.

Using Culling Masks/Layers

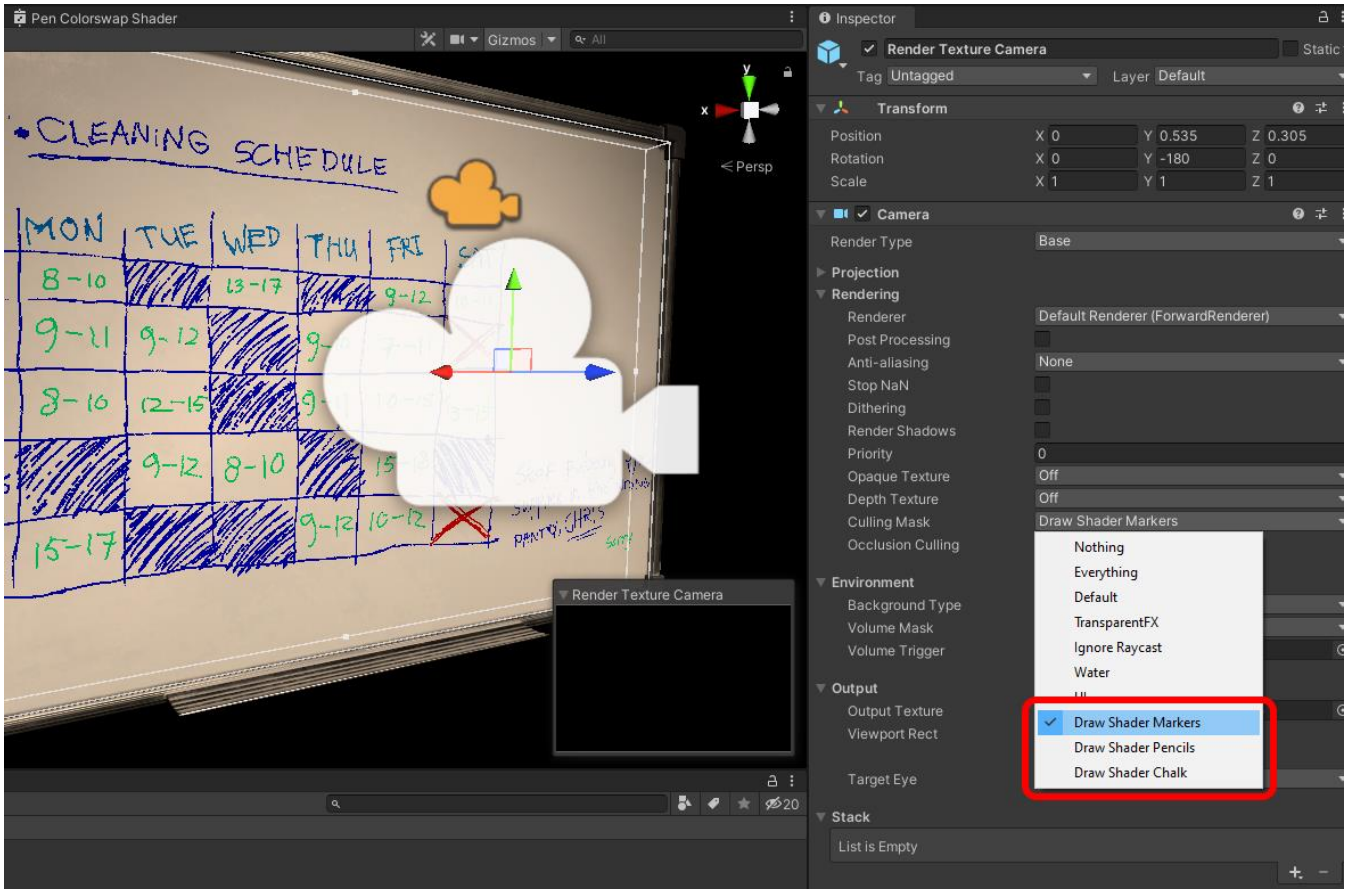
The Layer system is being used to **cull everything from the Camera except for the Particle Systems** that are attached to the pens. This way, it renders the Particle System – and the Particle System ONLY – to the Render Texture.



Creating new User Layers, one for each type of Draw Pen Set.

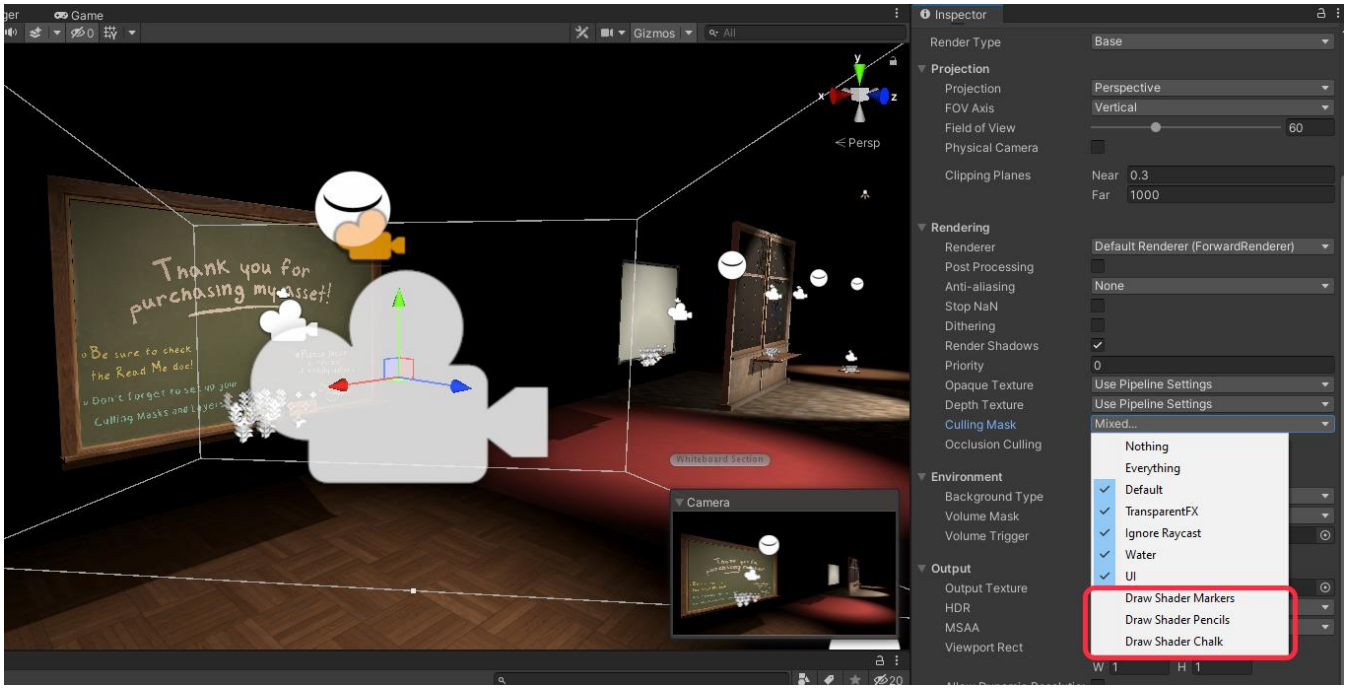
Once a new Layer has been created, we have to **assign the Particle System to that Layer**. After that, we'll select the **Render Texture Camera to only render that Layer**.

With the Layers in place, next we can tell the Camera to only render this specific Layer. This will also make the Camera a lot less expensive to be active in your scene.



This can also be used if you have several different drawing boards in the game. To, for example, prevent being able to draw with chalk on a whiteboard.

Additionally, tell the camera you use for your Game view to ***not*** render these layers. That way the particle trail won't be rendered for the player.



Learn more about Layers by consulting the official Unity Documentation on Layers.

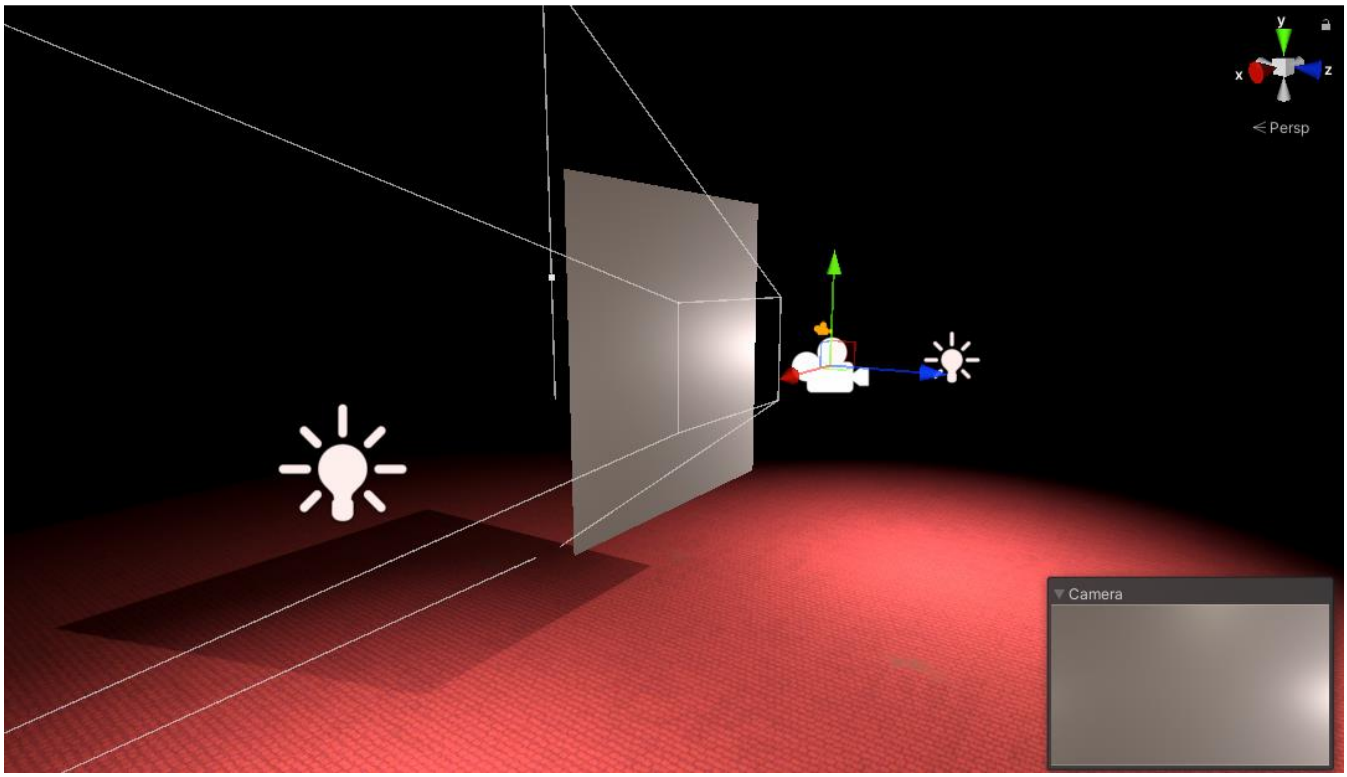
SETTING UP A CUSTOM MODEL

Introduction

This section will show you how to set up the shader yourself from scratch, using a **Render Texture**, a **Camera**, and the **Shader on a Material**. Following this guide, you could apply this to your own 3D Models!

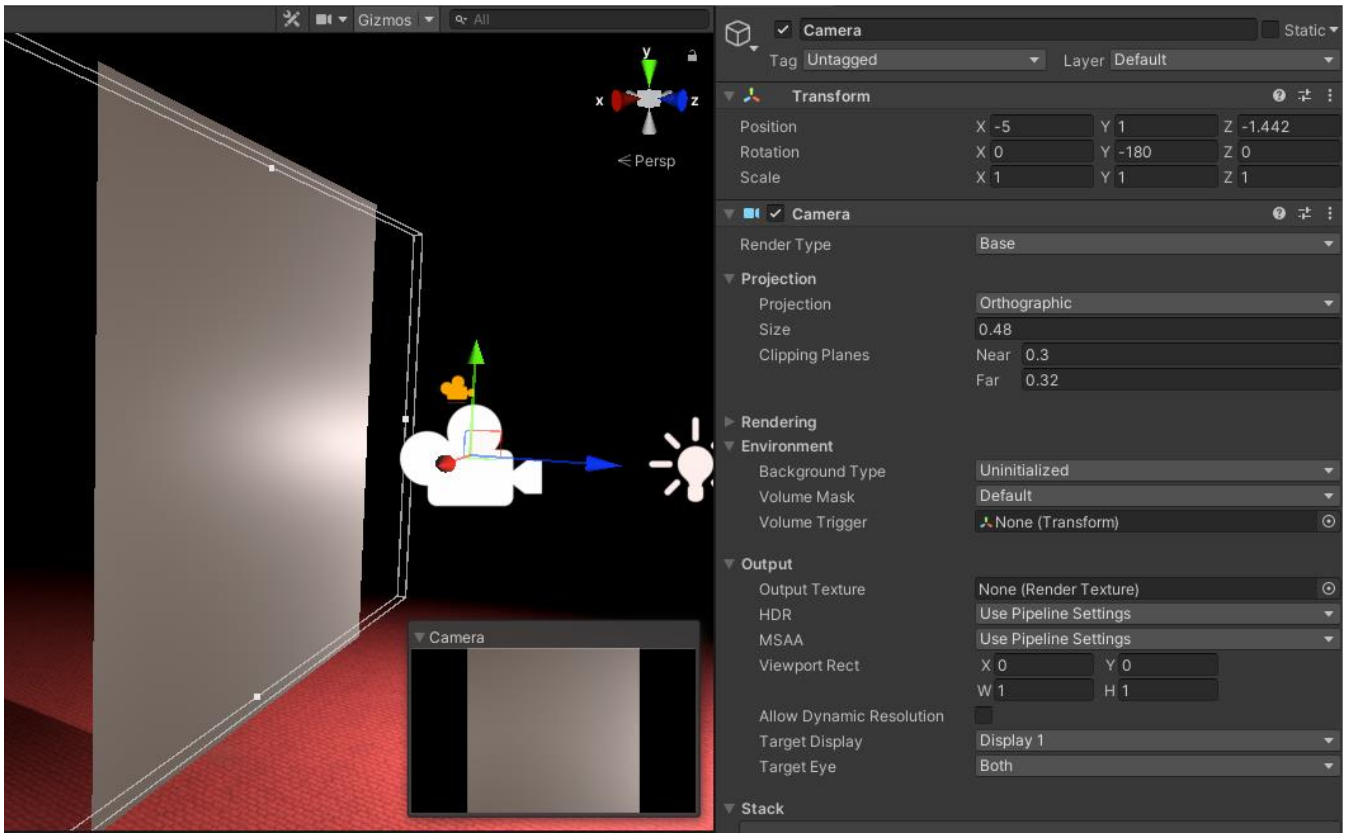
Step 1. A Surface and a Camera

Firstly, you need a Camera and a 3D surface to render the Material/Shader on. For the set-up guide I'll simply use a default Unity Quad, spawned from *GameObject > 3D Object > Quad*. Position your surface, and your camera in front of it, facing the surface.

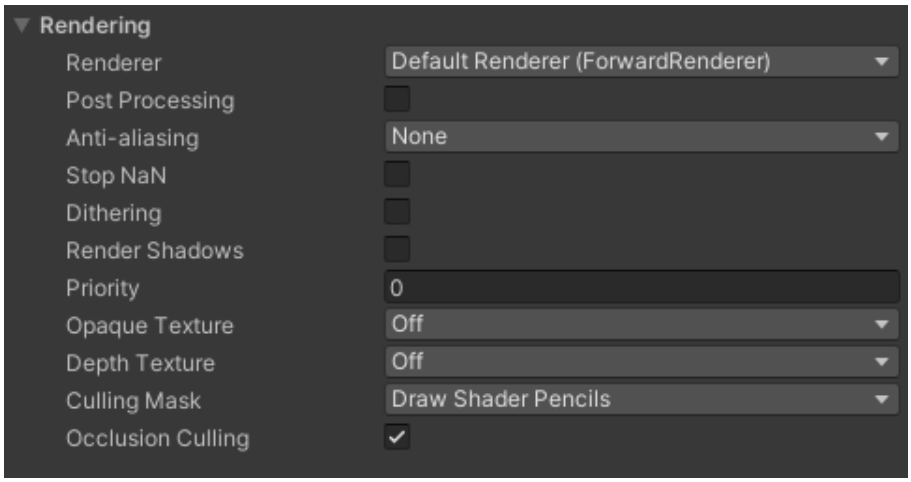


Step 2. Camera Settings

Change the Camera from Perspective to **Orthographic Mode**. Adjust the Size/Clipping Planes to fit your surface. Under the Environment tab, set the Background Type to Uninitialized.

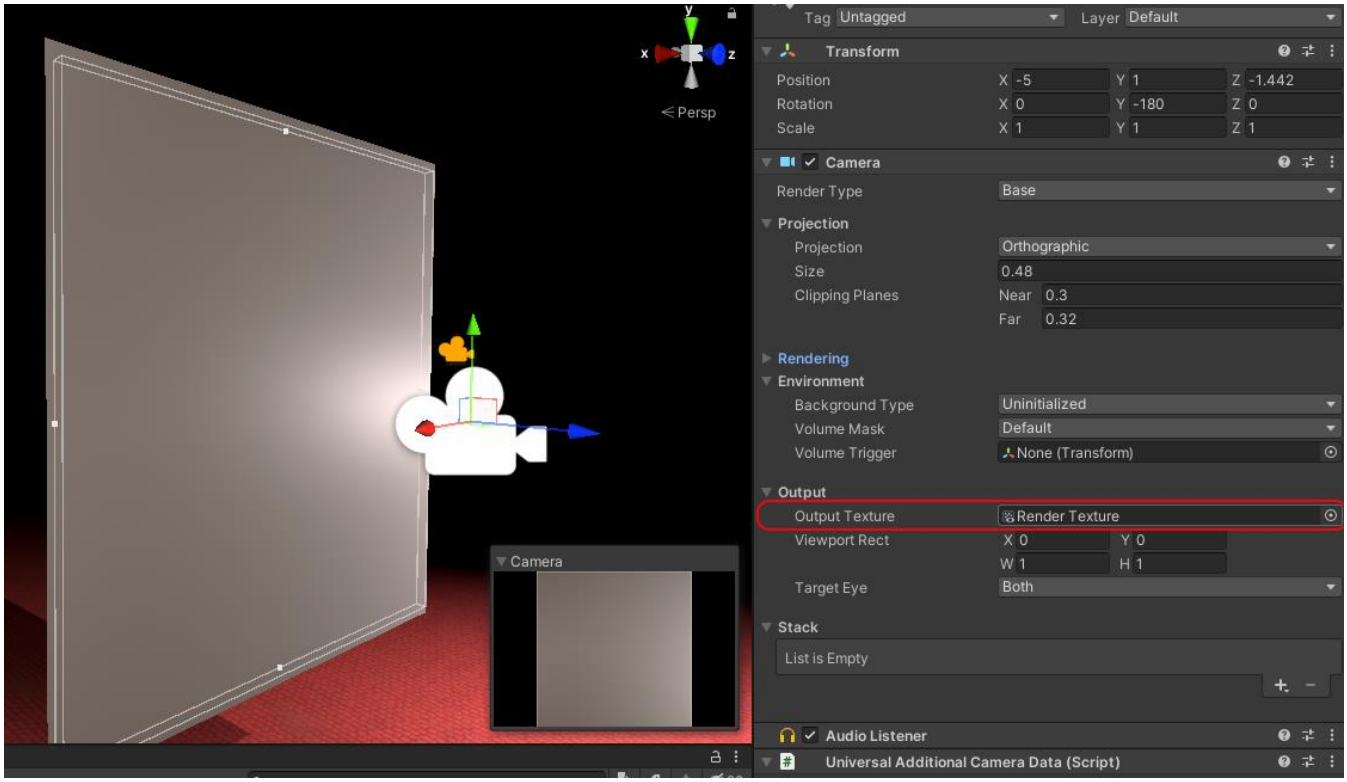


In the **Rendering** tab, we'll have most things turned off since they aren't necessary, and to make the camera as lightweight as possible. The Culling Masks will be covered later!



Step 3. Render Texture

Create a Render Texture in your Assets project folder by right-clicking and going to *Create > Render Texture*. Once created, put it as the Camera's **Output Texture**.

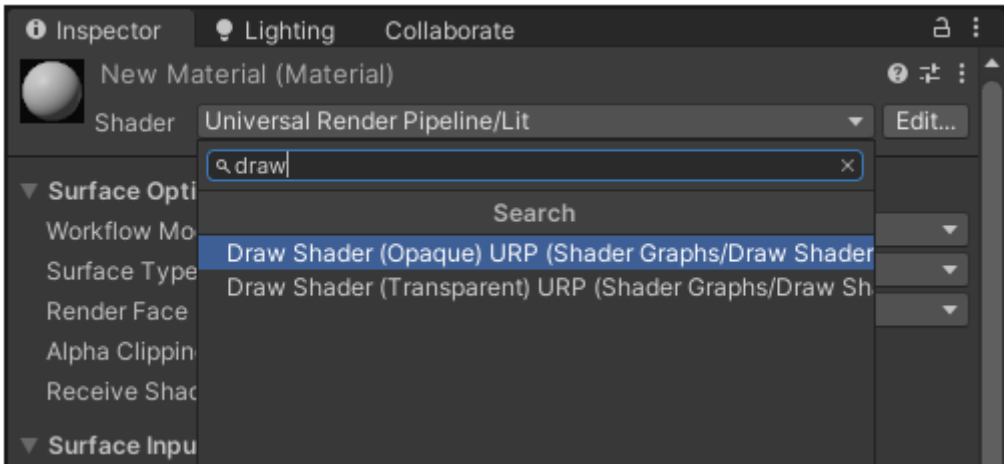


Now the Camera's size should also adjust to the aspect ratio of your Render Texture. For this example we'll just keep it at a perfect 1x1 square.

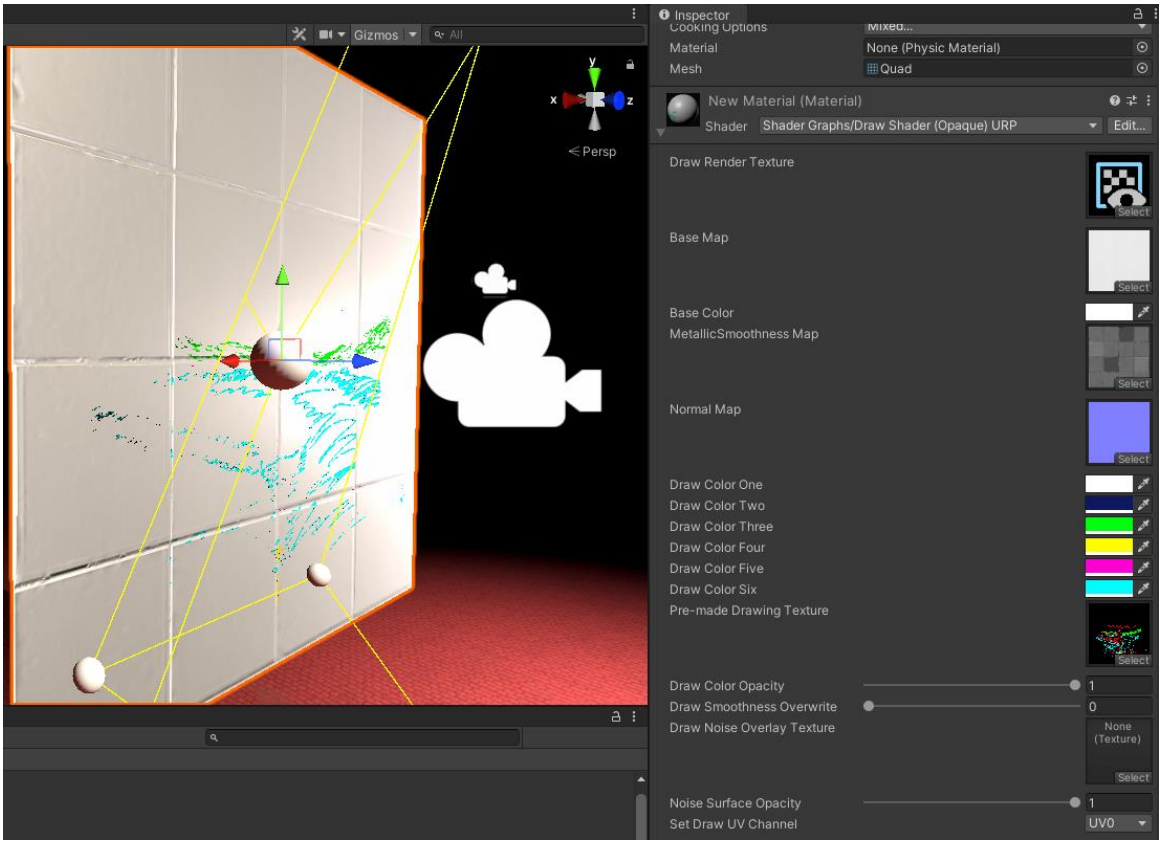
Note: this may be a little finnicky with the Camera's Size not updating immediately, I've found that resizing and then undoing (ctrl+Z) updates the Camera Size properly.

Step 4. Material and Shader

Simply create a new **Material** in your Project Folder (*Create > Material*) and assign the **Draw Shader (Opaque)** to it.

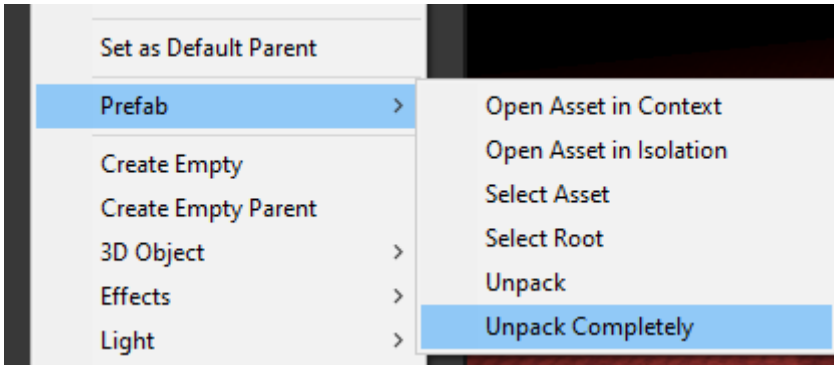


Now we have a Material with the Shader set up, we can start adding **Textures**. For my example I'll be applying some bathroom tiles, with a Pre-drawn Texture. In the Material we can also chance the Colors we'll be drawing with with Draw Color One through Six.

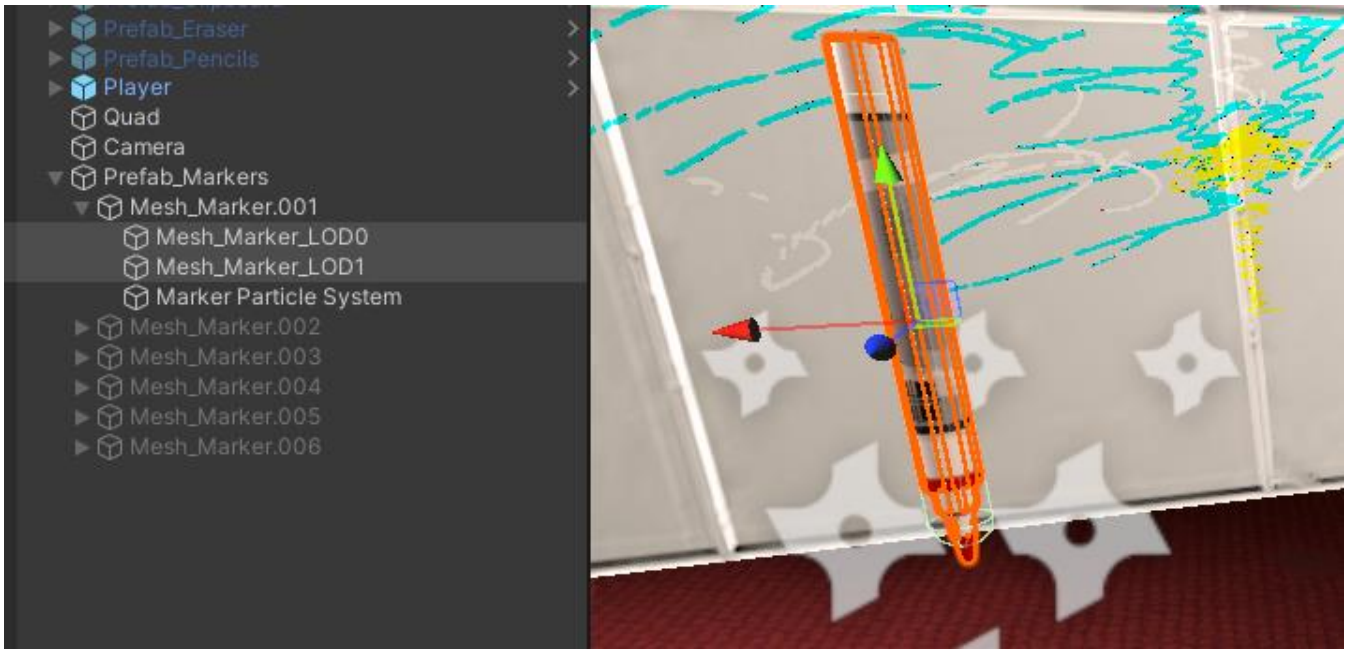


Step 5. Markers and Particles

For this part, I recommend using one of the **Prefabs**! This is the easiest and quickest way to make your own pens. Simply grab one of the pen prefabs, drag it into the scene, and Unpack the Prefab.

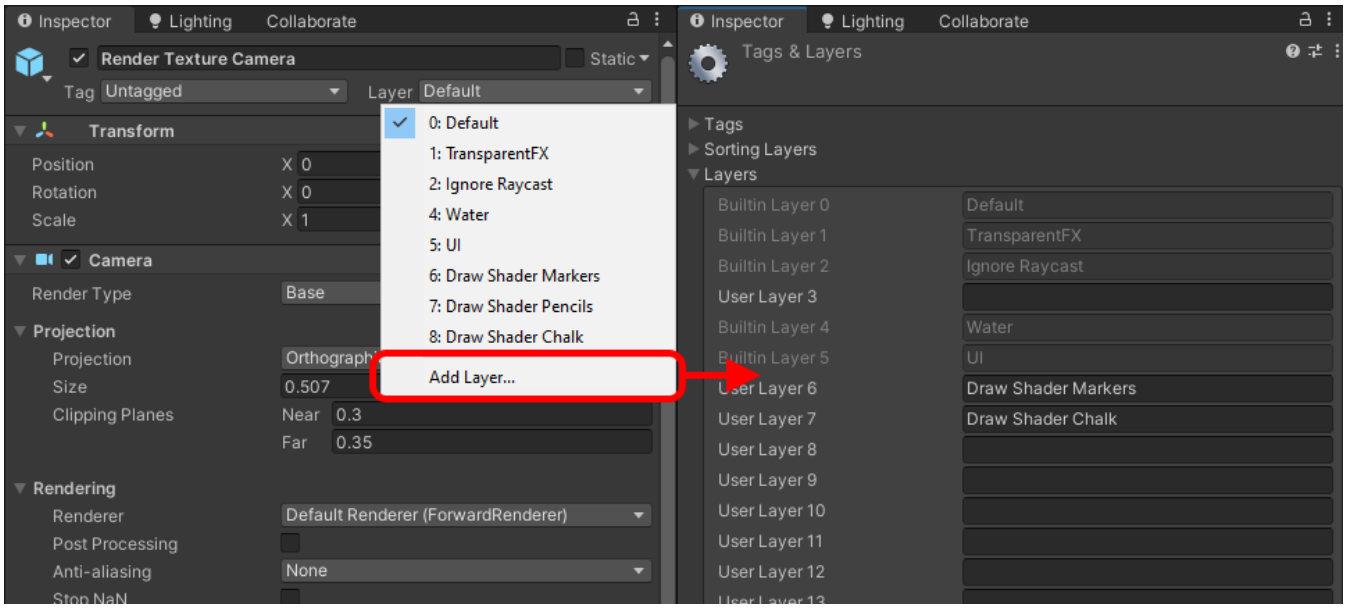


Now you can edit the Game Objects without changing the original Prefab. You can safely delete the current Marker models and swap it out for one of your own, with their own Materials.



Step 6. Culling Masks

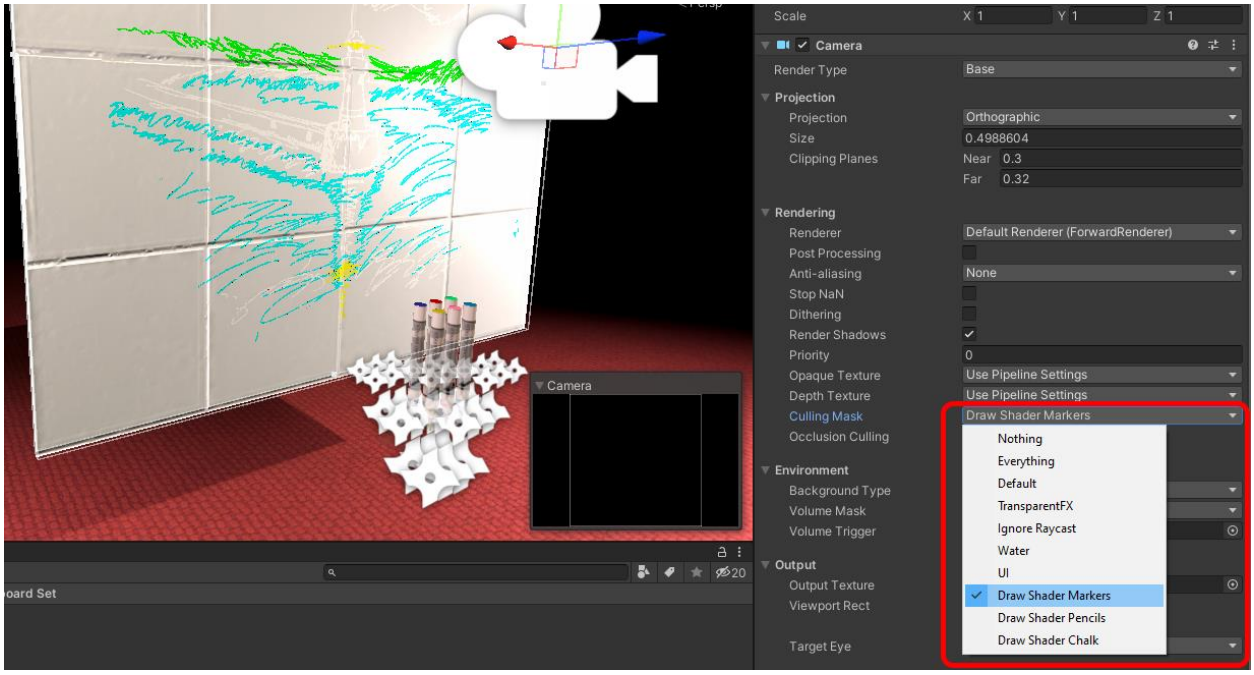
You might have noticed the Shader displaying weirdly, not yet working with the pens, or not fully displaying an image. This is because Culling Masks are the final crucial step to making the Shader work. We need to **cull everything from the Camera except for the Particle Systems** that are attached to the pens. This way, it renders the Particle System – and the Particle System ONLY – to the Render Texture.



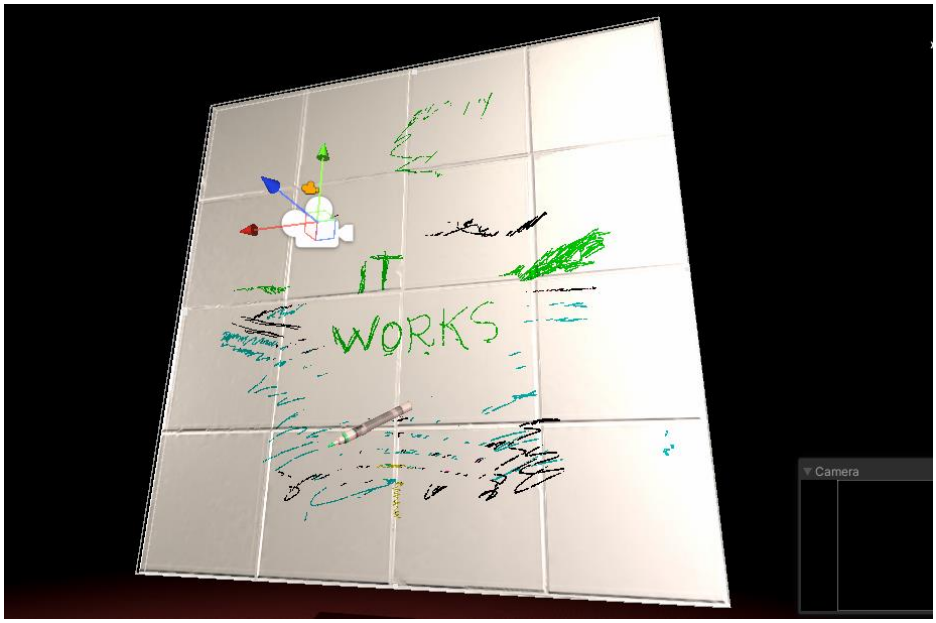
Creating new User Layers, one for each type of Draw Pen Set.

Once a new Layer has been created, we have to **assign the Particle System to that Layer**. After that, we'll select the **Render Texture Camera to only render that Layer**.

With the Layers in place, next we can tell the Camera to only render this specific Layer. This will also make the Camera a lot less expensive to be active in your scene.



This can also be used if you have several different drawing boards in the game. To, for example, prevent being able to draw with chalk on a whiteboard.



It should now be all in place and working!

TROUBLESHOOTING

Nothing appears when drawing on a board

1. Make sure you're in Play Mode.
2. Make sure you're in the Universal Render Pipeline (URP) and not the Built-in Render Pipeline.
3. Go to Window > Package Manager and check if you have Shader Graph loaded in. If you're in Universal Render Pipeline, Shader Graph should be in your project by default.
4. Check if the Particle System is playing.
5. Check if the Camera is able to see the Particle System (that it's not pushed too far into the 3D Model, for example).
6. Make sure your Layers are set up correctly. Do this by checking if the Particle System has the correct Layer assigned, and if the Render Texture Camera also renders this Layer.

My shaders are pink/give an error

1. Check if you are in Unity version 2020.3 or higher.
2. Make sure you're in the Universal Render Pipeline and not the Built-in Render Pipeline.
3. Go to Window > Package Manager and check if you have Shader Graph loaded in. If you're in Universal Render Pipeline, Shader Graph should be in your project by default.
4. Reload the Draw Shaders Package in from the Package Manager.

My shaders are grey

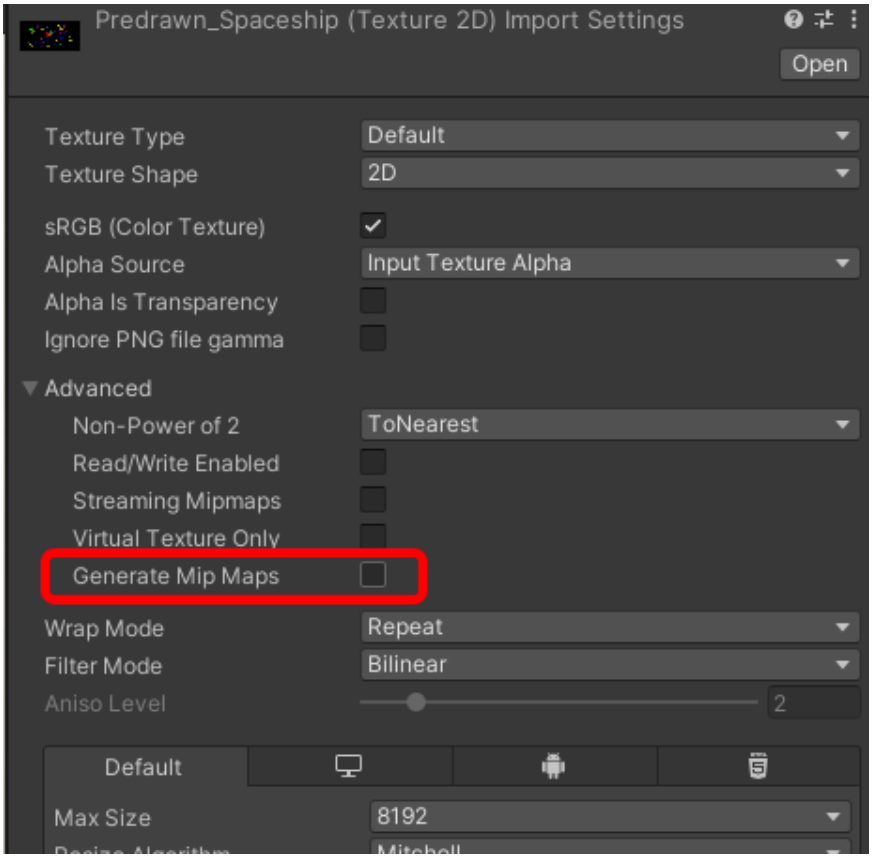
1. Check if your Shader Graph nodes are connected to the Master Node.
2. Make sure you're in the Universal Render Pipeline and not the Built-in Render Pipeline.
3. Go to Window > Package Manager and check if you have Shader Graph loaded in. If you're in Universal Render Pipeline, Shader Graph should be in your project by default.
4. Reload Shader Graph file by reloading in the Draw Shader Package in from the Package Manager.

My draw pens aren't working

1. Check if your Camera is set up and positioned correctly.
2. Make sure your Particle Systems are working.
3. Make sure the Culling Masks are set up correctly, both for the Particle System AND the Camera.
4. Go to Window > Package Manager and check if you have Shader Graph loaded in. If you're in Universal Render Pipeline, Shader Graph should be in your project by default.

My custom Pre-Drawn texture looks faded

1. Compare your Pre-Drawn texture to the the ones included in the pack.
2. Check if your Pre-Drawn texture is using the correctly colors.
3. Make sure your Pre-Drawn texture has lines that are thick enough for the shader to recognize.
4. Try turning off Generate Mip Maps for the texture file.



TIPS AND RECOMMENDATIONS

Try to limit the amount of drawing boards in a single view

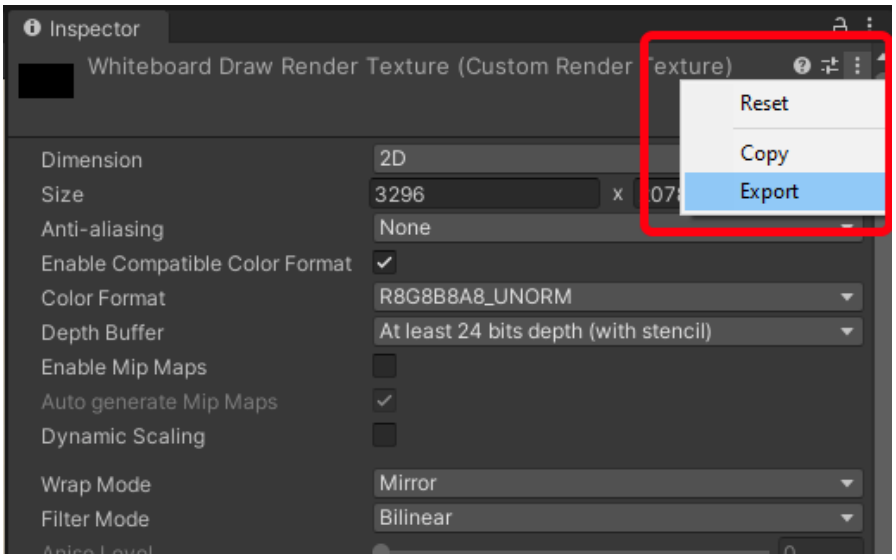
While the system is lightweight enough to have multiple times in a scene (like in the Demo Scene), I still have to recommend being smart with your Draw Shader placement/frequency. Either limit them per scene or make sure you cull the Drawing Boards after they're out of the Player's view.

Using SteamVR

Having used the **SteamVR plug-in** for recording the trailer footage and I can recommend it! It is easy to use to quicky start a project off with, or to quickly test the asset right after purchasing.

Exporting Drawings you made

With the Render Textures, you can easily export the drawing you've made! You could for example turn it into a custom Pre-drawn texture. Pause your project in playmode, navigate to top the Render Texture file you've been drawing on, and find the Export option in the dropdown menu.



ASSET PACK CONTENTS

Contents per version 1.1.1

Shaders

- Draw Shadergraph (Opaque)
- Draw Shadergraph (Transparent)
- Pen Colorswap Shadergraph

3D Meshes

- **Chalkboard Set**
  - Mesh_Chalkboard
  - Mesh_Chalk
  - Mesh_ChalkEraser

- **Whiteboard Set**
  - Mesh_Whiteboard
  - Mesh_Marker
  - Mesh_DryEraser

- **Greenhouse Set**
  - Mesh_GardenhouseFrame
  - Mesh_GardenhousePillar
  - Mesh_PlankShelf
  - Mesh_IvyFoliage-01
  - Mesh_IvyFoliage-02

- **Clipboard Set**
    - Mesh_Clipboard
    - Mesh_Paper
    - Mesh_Pencil
    - Mesh_Eraser

Materials

- **Chalkboard Set Materials** including:
    - M_Chalkboard (using the Draw Shader (Opaque))
    - M_Chalk (in assorted colors, using the Pen Colorswap shader)
    - M_ChalkEraser
    - M_WoodSquareBasket

- **Whiteboard Set Materials** including:
    - M_Whiteboard (using the Draw Shader (Opaque))
    - M_Marker (in assorted colors, using the Pen Colorswap shader)
    - M_DryEraser
    - M_Carpet

- **Greenhouse Set Materials** including:
    - M_GardenhouseGlass 1 (using the Draw Shader (Transparent))
    - M_GardenhouseGlass 2 (using the Draw Shader (Transparent))
    - M_GardenhouseFrame
    - M_PlankShelf
    - M_IvyFoliage
    - M_GreenhouseTiles

- **Clipboard Set Materials** including:
    - M_Clipboard (using the Draw Shader (Opaque))
    - M_Pencil (in assorted colors, using the Pen Colorswap shader)
    - M_Eraser

Textures

- **BaseColor Maps** (*AlbedoTransparency*), **Normal Maps** and **Metallic Smoothness Maps** for every Material, totalling up to:
    - 16 BaseColor Maps
    - 16 Normal Maps
    - 16 MetallicSmoothness Maps

- And a total of **8 Pre-Drawn textures**.

Prefabs

- **Demo Scene Props**
    - Prefab_IvyFoliage-01
    - Prefab_IvyFoliage-02
    - Prefab_GreenhouseFrame
    - Prefab_GreenhousePillar
    - Prefab_PlankShelf

- **Draw Shader Prefabs**
    - **Chalkboard Set**
        - Prefab_Chalkboard
        - Prefab_Chalk
        - Prefab_ChalkEraser
    - **Whiteboard Set**
        - Prefab_Whiteboard
        - Prefab_Markers
        - Prefab_DryEraser
    - **Clipboard Set**
        - Prefab_Clipboard
        - Prefab_Pencils
        - Prefab_Eraser

Scenes

- DemoScene.unity

THANK YOU NOTE

I hope you are able to get lots of use out of my Draw Shader! If you have any inquiries or need technical support, e-mail support@suggocreations.com. As I'm only a single person, it may take a few work days before I get back to you. But I'll try my best to help!

If you enjoyed this asset pack, please take the time to **leave a review**. I highly value any comments, feedback or suggestions you may have.

Thanks!