

Rendering 3d objects behind 2d Objects unity

Asked 3 years, 11 months ago Modified 2 years, 5 months ago Viewed 6k times



2



1



I am making a an isometric game with a combination of 2d and 3d elements. I wish to know how to make my player and enemy sprites (all made up of 2d elemnts) not render behind or inside my 3d elements.



So I have tried messing about with renderQueue and setting the renderqueue of the materials these rocks and such have to a high value to have them drawn behind the 2d elements.

However through my searching I have found out that I actually need to be setting the ztest of the objects to correct this. This confused me a bit and I am unsure how to do it as I have not really used shaders that much. I could not find any decent references that explain exactly how to go about this just stuff written that assumes prior knowledge like [this](#). I tried downloading the unity shaders and and opened the default sprite one but I was unsure where to change the ztest or tagging to solve this.

here is the standard sprite shader from unity:

```
unity_SpriteRendererColorArray)
#define _Flip          UNITY_ACCESS_INSTANCED_PROP(PerDrawSprite,
unity_SpriteFlipArray)

#endif // instancing

CBUFFER_START(UnityPerDrawSprite)

#ifndef UNITY_INSTANCING_ENABLED
fixed4 _RendererColor;
fixed2 _Flip;
#endif
float _EnableExternalAlpha;
CBUFFER_END

// Material Color.
fixed4 _Color;

struct appdata_t
{
    float4 vertex    : POSITION;
    float4 color     : COLOR;
    float2 texcoord  : TEXCOORD0;
```

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up



```
{
    float4 vertex    : SV_POSITION;
    fixed4 color      : COLOR;
    float2 texcoord   : TEXCOORD0;
    UNITY_VERTEX_OUTPUT_STEREO
};

inline float4 UnityFlipSprite(in float3 pos, in fixed2 flip)
{
    return float4(pos.xy * flip, pos.z, 1.0);
}
```

Update 1



unity3d rendering shader

Share Improve this question

edited Aug 7, 2018 at 10:44

asked Aug 7, 2018 at 9:33

Follow



SteenPetersen

148 ● 2 ● 12

Have you tried to position your 2D elements in 3D space just like your 3D elements, with their transform position in front/behind whatever they need to be in front of / behind?

– [Henrik L. Schou-Pedersen](#) Aug 7, 2018 at 10:30

In my case its not a good idea. because my camera is isometric so it looks down at the world at a 41 degree angle. Placing 2d elements "infront" of 3 elements means they get placed at weird angles which makes the aiming system and position of the 2d elements all skewed and really becomes messy to work with. Im quite sure the correct way to do this is to be able to tell certain 3D elements that they should simply be rendered behind the 2D elements. I just dont know how to write a shader that does this.

UPDATE 1 illustrates how it looks when in-game. Sprite renders infront of tile but not rock.

– [SteenPetersen](#) Aug 7, 2018 at 10:40

Could you use another camera? If you could you can set another camera culling mask and depth. That could make something in the layer will show front. – [Ron Tang](#) Aug 7, 2018 at 11:16

shader can do this, but i think use another camera would be easy to impl. – [Ron Tang](#) Aug 7, 2018 at 11:18

Sorted by:

Highest score (default)



1 Answer

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up



2 Correctly set the two cameras's **culling mask** and **depth** and **clear flags**.



The depth of the extra camera should be **higher** and the clearflag should be set to **depth only** and culling mask only set the layer you want render in front.

And your original camera culling mask should delete the layer render in extra camera.

You can make objects of a particular layer render in front.

Yesterday Svp tell me that he want fine-grained control drawing-order of 2d and 3d objects. My old answer can not does this well. So after research, there are three solutions.

UPDATE:

1. Treat 2d object as 3d. Use z-pos to sort all objects. Keep few 3d object z-pos always less than or greater than player z-pos (The offset of z-pos of 3d object and player need greater than fixed value that make the entire 3d object behind the player.). The fixed value could select 3d object bounds max.
2. Treat 3d object as 2d. **3d object mesh render use 2d sprite-default shader**. Use 2d sorting layer and order in layer control 3d object render. But we need **custom mesh render inspector** use follow code copy from [here by uvivagabond](#).

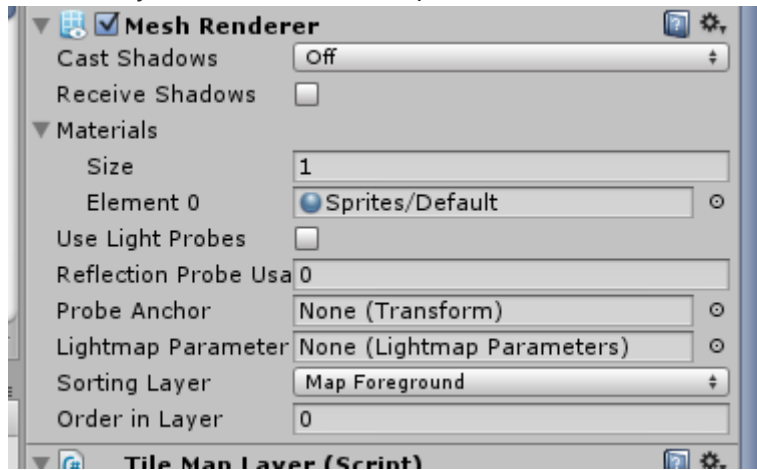
```
( m_SortingOrder );

        SerializedProperty castShadows = serializedObject.FindProperty
("m_CastShadows");
        SerializedProperty receiveShadows = serializedObject.FindProperty
("m_ReceiveShadows");
        SerializedProperty motionVectors = serializedObject.FindProperty
("m_MotionVectors");
        SerializedProperty materials = serializedObject.FindProperty ("m_Materials");
        SerializedProperty lightProbes = serializedObject.FindProperty
("m_LightProbeUsage");
        SerializedProperty reflectionProbes = serializedObject.FindProperty
("m_ReflectionProbeUsage");
        SerializedProperty anchorProbes = serializedObject.FindProperty
("m_ProbeAnchor");
        #endregion

        #region Draw Properties
        AddPropertyField (castShadows);
        AddPropertyField (receiveShadows);
        AddPropertyField (motionVectors);
        AddPropertyField (materials);
        AddPopup (ref lightProbes, "Light Probes", typeof(LightProbeUsage));
        AddPopup (ref reflectionProbes, "Reflection Probes",
typeof(ReflectionProbeUsage));
        AddPropertyField (anchorProbes, "Anchor Override");
        #endregion

        GUIStyle style = new GUIStyle (GUI.skin.label);
        style.richText = true;
        EditorGUILayout.Space ();
        EditorGUILayout.LabelField ("<b><color=#EE4035FF>SortingLayers Options:
</color></b>", style);
```

After this your render mesh inspector view as this:



3. Combine 2d render system and 3d render system. Handle Z-buffer Z-write and many many other ... I believe this great challenge.Unity should do this work not us.

I hope this can help you.

Share Improve this answer Follow

edited Aug 8, 2018 at 2:06

answered Aug 7, 2018 at 11:34



Ron Tang

1,472 • 12 • 19

1 Ahh I see I can of course just add all the layers I wish to be drawn infront of the player, will need some restructuring of the elements but this should work. – [SteenPetersen](#) Aug 7, 2018 at 12:37

@Svp extra camera should only set culling mask to one specify layer that you want in front.And other camera should delete this layer in culling mask – [Ron Tang](#) Aug 7, 2018 at 13:00

yes but once I decide to show the player infront of the stones, that now means player is shown infront of everything because the depth of the camera is higher. which now means I need to put all other objects in that camera as well. after a while I have tons of layers in the "front" camera and it gets mega confusing. And yes for every layer I add to the front one I remove it from the back. things just start getting wonky. – [SteenPetersen](#) Aug 7, 2018 at 13:10

Could uou do it Reverse. Put less thing to back render.@Svp – [Ron Tang](#) Aug 7, 2018 at 13:20

It seems your design that sometime A render front sometime A render back.So you change layer at runtime. – [Ron Tang](#) Aug 7, 2018 at 13:22

Let us [continue this discussion in chat](#). – [SteenPetersen](#) Aug 7, 2018 at 13:23