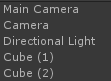
对于OnRenderImage的触发时机时机上是在当前的Camera渲染完成后，颜色缓冲区中的数据

例子：

场景中有两个Camera，其中Camera的depth为0，Main Camera的depth为1



Camera带的脚本为Script2，Main Camera带的脚本为Script1

void OnRenderImage(RenderTexture src, RenderTexture dest)

{

Debug.Log("OnRenderImage script2, src = " + src);

RenderTexture.active = src;

Texture2D texture = new Texture2D(Screen.width, Screen.height, TextureFormat.RGB24, false);

texture.ReadPixels(new Rect(0, 0, Screen.width, Screen.height), 0, 0);

texture.Apply();

RenderTexture.active = null;

byte[] bytes = texture.EncodeToPNG();

System.IO.File.WriteAllBytes(Application.dataPath + "/b.png", bytes);

}

void OnRenderImage(RenderTexture src, RenderTexture dest)

{

Debug.Log("OnRenderImage script1, src = " + src);

RenderTexture.active = src;

Texture2D texture = new Texture2D(Screen.width, Screen.height, TextureFormat.RGB24, false);

texture.ReadPixels(new Rect(0, 0, Screen.width, Screen.height), 0, 0);

texture.Apply();

RenderTexture.active = null;

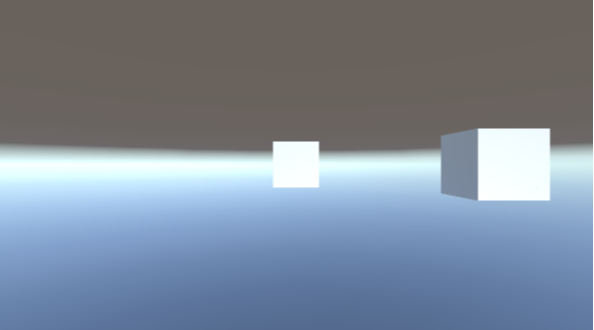
byte[] bytes = texture.EncodeToPNG();

System.IO.File.WriteAllBytes(Application.dataPath + "/a.png", bytes);

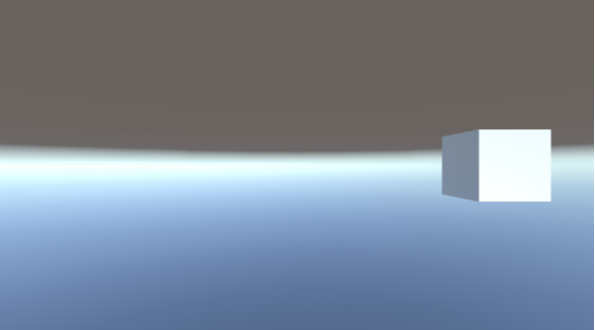
}

得到的结果：

1. png



1. png



也就是说先把Camera执行完了，得到了b.png，在执行Main Camera，得到了a.png，发现a.png实际上包含了b.png渲染的内容