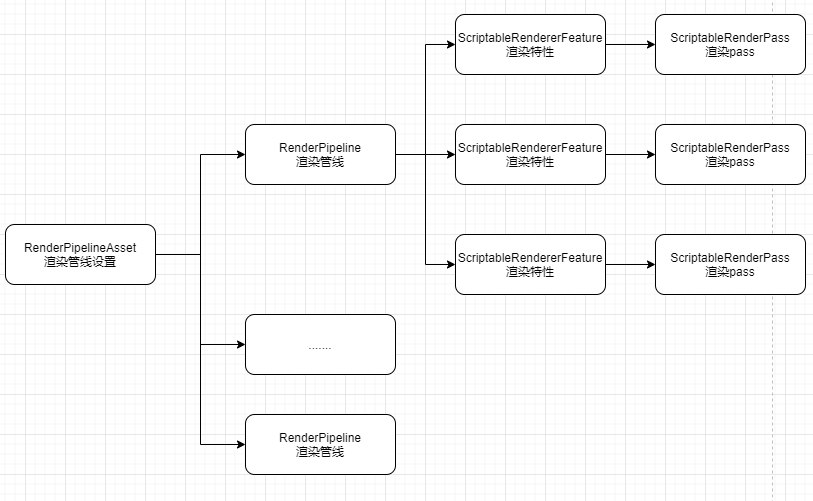
Urp源码分析



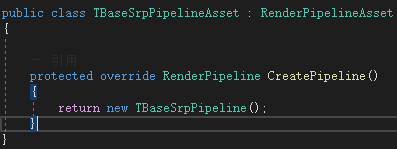
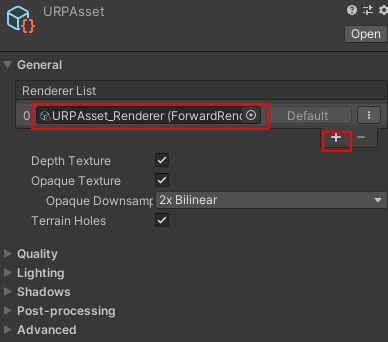
Asset/Create/Rendring/Universal Render pipeline/Pipeline Asset

创建出两个文件

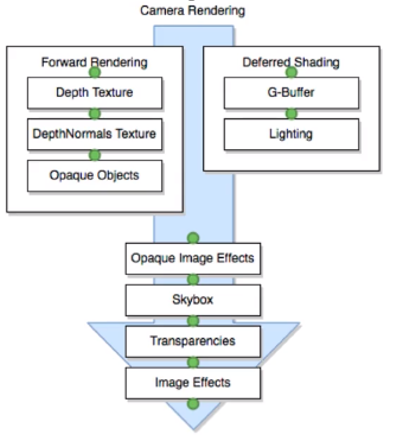
1 URPAsset

注意这里创建的URPAsset继承自RenderPipelineAsset

URPAsset中可以包含多个渲染管线，可以对比build in(包含前向，延迟两种)

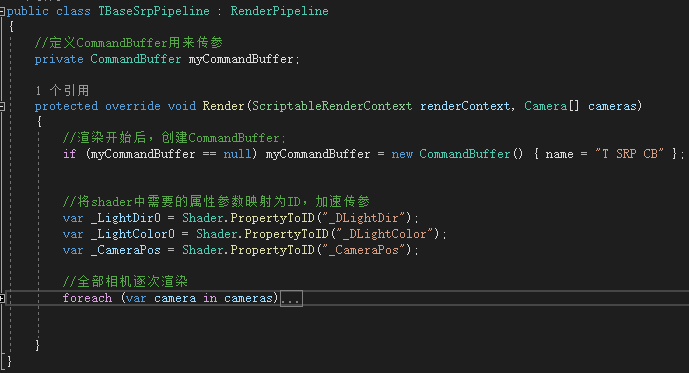
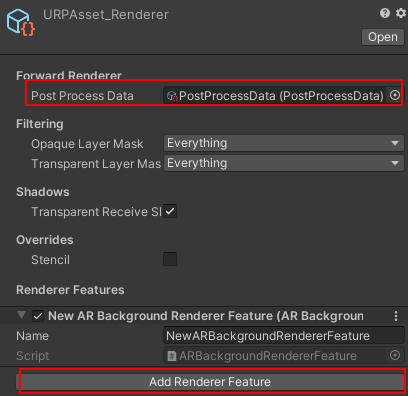


Srp之前的buildin渲染管线



2 URPAssetRender

Render就是一个渲染管线，里面可以包含多个Renderer Feature



URP对应的文件夹

1 asset Runtime/Data/UniversalRenderPipelineAsset.cs

2 render Runtime/UniversalRenderPipeline.cs

3 feature Runtime/RenderFeatures/

4 pass Runtime/Passes/

说明

Urp demo中动态切换GrphicsSetting的方法，就是不同场景不同piplineAsset的设置方法

是通过下面这个类来实现的

[ExecuteAlways]

public class AutoLoadPipelineAsset : MonoBehaviour

{

public UniversalRenderPipelineAsset pipelineAsset;

private void OnEnable()

{

UpdatePipeline();

}

void UpdatePipeline()

{

if (pipelineAsset)

{

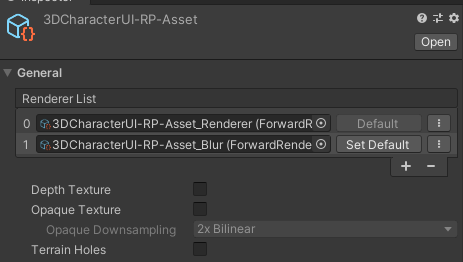
GraphicsSettings.renderPipelineAsset = pipelineAsset;

}

}

}

一个pipleAsset上可能有多个Renderer



切换可能的方法为

1. **var** x = GetComponent<UnityEngine.Rendering.Universal.UniversalAdditionalCameraData>();
2. **if** (Input.GetKey(KeyCode.LeftShift)) x.SetRenderer(1);
3. **else** x.SetRenderer(0);

另一种demo中实际用到的方法，就是协会Camera上的Renderer

1 重新理解feature时什么，就是pass和输入数据的载体

2 关于urp与传统管线理解，类似不过前向可以单pass多光源，

3 RenderPassEvent插入的事件是urp自己实现管道某些具体位置事件，与buildin无关

4 关于srp与传统相比，到底能定制些什么，观看下传统管线以及urp开发出的feateur基本

能得到结论，srp完全可以实现跟传统一模一样的管线，（传统）管线的每个部分(节点)

可以由不同shader渲染一次或多次