

RenderPipeline

1. LightData

类型	变量名	备注
int	mainLightIndex	主光源索引
int	additionalLightsCount	允许的最多光源数量
int	maxPerObjectAdditionalLightsCount	最大的Light数
List< visibleLights >	visibleLights	可见光队列
bool	shadeAdditionalLightsPerVertex	自己品
bool	supportsMixedLighting	是否支持混合灯光

2. CameraData

类型	变量名	备注
Matrix4x4	m_ViewMatrix	View矩阵
Matrix4x4	m_ProjectionMatrix	Pro矩阵
Camera	camera	附带的相机
CameraRenderType	renderType	相机的渲染类型是Base还是Overlay
RenderTexture	targetTexture	相机的渲染目标，如果不在编辑器中设置为Null
RenderTextureDescriptor	cameraTargetDescriptor	相机纹理图的格式描述符
Rect	pixelRect	一个方框，代表相机的渲染尺寸
int	pixelWidth	方框的宽
int	pixelHeight	方框的长
float	aspectRatio	宽高比
float	renderScale	渲染到与当前尺寸不一的时候怎么缩放
bool	clearDepth	是否清理深度
CameraType	cameraType	相机的类型是Game还是View
bool	isDefaultViewport	是默认的渲染尺寸吗
bool	isHdrEnabled	是否支持HDR
bool	requiresDepthTexture	是否需要深度图，需要会拷贝
bool	requiresOpaqueTexture	同理拷贝颜色图
bool	requireSrgbConversion	是否需要将渲染转换到Srgb空间
SortingCriteria	defaultOpaqueSortFlags	不知道干嘛的
bool	isStereoEnabled	不知道干嘛的

类型	变量名	备注
float	maxShadowDistance	最远阴影距离
bool	postProcessEnabled	该相机是否支持后处理
Action<RenderTargetIdentifier, CommandBuffer>	captureActions	委托方法，是否捕获相机
LayerMask	volumeLayerMask	控制后处理的Mask
Transform	volumeTrigger	后处理的控制组件
bool	isStopNaNEnabled	一个后处理类型
bool	isDitheringEnabled	一个后处理类型
AntialiasingMode	antialiasing	抗锯齿Mode
AntialiasingQuality	antialiasingQuality	抗锯齿质量
ScriptableRenderer	renderer	当前相机的Renderer，URP默认的ForwardRenderer
bool	resolveFinalTarget	是否是最后一个相机，如果只有一个相机则为True，如果有多个，则最后一个为True

3. ShadowData

类型	变量名	备注
bool	supportsMainLightShadows	主光源是否支持阴影
bool	requiresScreenSpaceShadowResolve	屏幕空间
int	mainLightShadowmapWidth	主光源的Map的宽
int	mainLightShadowmapHeight	高
int	mainLightShadowCascadesCount	级联的数量
Vector3	mainLightShadowCascadesSplit	级联分级比例
bool	supportsAdditionalLightShadows	是否支持附加光源阴影
int	additionalLightsShadowmapWidth	附加光的宽
int	additionalLightsShadowmapHeight	高
bool	supportsSoftShadows	是否支持软阴影
int	shadowmapDepthBufferBits	深度图的bits
List<< Vector4 >>	bias	深度采样或生成时的bias

ScriptableRenderer

类型	变量名	备注
RenderBufferStoreAction	m_ActiveDepthStoreAction	深度纹理纹理在完成渲染后，GPU执行什么操作，但只在移动端有效

类型	变量名	备注
RenderBufferStoreAction[]	m_ActiveColorStoreAction	激活的颜色纹理，完成渲染后的执行的操作
RenderTargetIdentifier	m_ActiveDepthAttachment	当前激活的深度纹理附件
RenderTargetIdentifier[]	m_ActiveColorAttachments	当前激活的颜色附件
bool	isCameraColorTargetValid.	当相机目标被移除时变为true
bool	m_IsPipelineExecuting	在调用Excute方法是变为True，最后一个pass时变为false
bool	m_IsPipelineExecuting	第一次CameraTarget被赋值时变为true。调用clear方法时
bool	m_FirstTimeCameraColorTargetIsBound	同上
RenderTargetIdentifier	m_CameraDepthTarget	相机的深度目标， 在RenderSingleCamera调用的clear方法被初始化为Builtin RenderTextureType.CameraTarget
	m_CameraColorTarget	相机的颜色缓冲目标
List<ScriptableRenderFeature>	m_RendererFeatures	Renderer的所有feature
List<ScriptableRenderPass>	m_ActiveRenderPassQueue	所有被激活的pass

ForwardRenderer

继承自ScriptableRenderer，主要的数据为各种pass和贴图。

Pass

```
ColorGradingLutPass m_ColorGradingLutPass;
DepthOnlyPass m_DepthPrepass;
DepthNormalOnlyPass m_DepthNormalPrepass;
MainLightShadowCasterPass m_MainLightShadowCasterPass;
AdditionalLightsShadowCasterPass m_AdditionalLightsShadowCasterPass;
GBufferPass m_GBufferPass;
CopyDepthPass m_GBufferCopyDepthPass;
TileDepthRangePass m_TileDepthRangePass;
TileDepthRangePass m_TileDepthRangeExtraPass; // TODO use subpass API to hide this pass
DeferredPass m_DeferredPass;
DrawObjectsPass m_RenderOpaqueForwardOnlyPass;
DrawObjectsPass m_RenderOpaqueForwardPass;
DrawSkyboxPass m_DrawSkyboxPass;
CopyDepthPass m_CopyDepthPass;
CopyColorPass m_CopyColorPass;
TransparentSettingsPass m_TransparentSettingsPass;
DrawObjectsPass m_RenderTransparentForwardPass;
InvokeOnRenderObjectCallbackPass m_OnRenderObjectCallbackPass;
PostProcessPass m_PostProcessPass;
PostProcessPass m_FinalPostProcessPass;
FinalBlitPass m_FinalBlitPass;
CapturePass m_CapturePass;
```

Texture

贴图名称	贴图名字	作用和使用Pass
RenderTargetHandle m_ActiveCameraColorAttachment	无名字，当前激活的相机颜色	
RenderTargetHandle m_ActiveCameraDepthAttachment	同上	
RenderTargetHandle m_CameraColorAttachment	_CameraColorTexture	
RenderTargetHandle m_CameraDepthAttachment	_CameraDepthAttachment	
RenderTargetHandle m_DepthTexture	_CameraDepthTexture	
RenderTargetHandle m_NormalsTexture	_CameraNormalsTexture	
RenderTargetHandle[] m_GBufferHandles	延迟渲染部分，略	
RenderTargetHandle m_OpaqueColor	_CameraOpaqueTexture	
RenderTargetHandle m_AfterPostProcessColor	_AfterPostProcessTexture	
RenderTargetHandle m_ColorGradingLut	_InternalGradingLut	
RenderTargetHandle m_TileDepthInfoTexture	_TileDepthInfoTexture	

RenderTargetHandle m_DepthInfoTexture

ScriptableRenderPass

数据类型	数据名称	备注	初始化值
RenderPassEvent	renderPassEvent	该pass的渲染时机	无所谓
RenderBufferStoreAction[]	m_ColorStoreActions	该pass的颜色附件的移动端GPU解析方式	store
RenderBufferStoreAction	m_DepthStoreAction	该pass的depth的移动端GPU解析方式	store
bool	overrideCameraTarget	是否覆盖了相机的target只要配置Pass自己的贴图就会覆盖	False
RenderTargetIdentifier[]	m_ColorAttachments	该pass的颜色贴图	CameraTarget, 既是默认帧缓冲
m_DepthAttachment			同上
ClearFlag	m_ClearFlag	贴图的clear标识	none
Color	m_ClearColor	颜色缓冲的celar颜色	黑色 Color.black

BaseCamera和OverLayCamera的数据区别

不同类型的区别

数据类型	BaseCamera	OverlayCamera
Camera	自己的	自己的

数据类型	BaseCamera	OverlayCamera
maxShadowDistance	$\text{Min}(\text{settings.shadowDistance}, \text{camera.farClipPlane})$	$\text{Min}(\text{settings.shadowDistance}, \text{camera.farClipPlane})$
renderType	base	overlay
clearDepth	True	自己设置
postProcessEnabled	自己设置	自己设置
requiresDepthTexture	自己设置	默认关闭
requiresOpaqueTexture	自己设置	默认关闭
renderer	自己设置	自己设置
resolveFinalTarget	没有CameraStack时为True	是CameraStack最后一个OverlayCamera时为True
矩阵	自己设置	自己设置

相同的

数据类型	值
targetTexture	BaseCamera设置的
cameraType	
volumeLayerMask	
volumeTrigger	
isStopNaNEnabled	
isDitheringEnabled	
antialiasing	
antialiasingQuality	
isHdrEnabled	
cameraRect相关数据	
renderScale	
defaultOpaqueSortFlags	
captureActions	