



HYBRID RENDERING PIPELINE

自研公司级渲染管线



周家骥

游族网络 - 引擎技术中心

Hybrid Rendering Pipeline

目标:

HRP 是一套基于Unity SRP的

满足移动、PC跨平台画面效果表现

满足多种不同类型产品设计需求

高性能, 易扩展的

渲染框架

UNITY · HRP

渲染效果

问题

如何解决以下两对矛盾？

跨平台

不同平台设备对画面品质的要求

高性能

尽可能下探到更多移动端设备，对兼容性和性能的苛刻要求。

跨产品

不同项目对功能复用的需求

定制化

项目中差异化的表现需求方便扩展

跨平台

移动端性能基线

年份	系统	机型分类	推荐手机	CPU排名	推荐设备 (内存, CPU型号, 核数, 频率, GPU型号)
2024 预测	Android	高端机	三星Galaxy S20 Huawei P40 pro	/	8GB, 高通骁龙865, KIRIN990
		中端机	Xiaomi 10 Lite 荣耀 X10	/	6GB, 高通骁龙765G, KIRIN820,
		低端机	vivo X23 荣耀8X	/	4GB, 高通骁龙665, KIRIN710
	IOS	高端机	iPhone12	/	4GB Apple A14
		中端机	iPhoneXR	/	3GB Apple A12
		低端机	iPhone 7 Plus	/	3GB Apple A10
2023	Android	高端机	三星Galaxy S9+ Huawei P30	42	6GB 高通骁龙845, KIRIN980 GPU-高通 Adreno630
		中端机	Redmi K30 4G 荣耀10	70	6GB 高通骁龙730G, KIRIN970 GPU-高通 Adreno 618
		低端机	VIVO X20A 荣耀8	114	4GB 高通骁龙660, KIRIN950 GPU-高通 Adreno 512
	IOS	高端机	iPhone11	16	4GB Apple A13
		中端机	iPhoneX	50	3GB Apple A11
		低端机	iPhone 7 Plus	77	3GB Apple A10
2022	Android	高端机	三星 S8+	56	6GB 高通骁龙835, 2.35GHz(大四核)+1.9GHz(小四核) GPU-高通 Adreno 540
		中端机	Redmi Note7	77	4GB 高通骁龙660 2.2GHz(八核) GPU-高通 Adreno 512
		低端机	VIVO Y85	123	4GB 高通骁龙450 1.5GHz(八核) GPU-高通 Adreno 506
	IOS	高端机	iPhone X	42	3GB Apple A11 Bionic APL1072
		中端机	iPhone 7	54	2GB Apple A10 Fusion APL1024
		低端机	iPhone 6s	66	2GB Apple A9 APL0898 (S8000)

- 性能基线划定的依据主要是市场设备占比不低于40% 60% 80%
- 性能基线收到采样数据发行区域不同影响会有调整

客户端内存消耗	2022年	2023年
Android平台	1档机型指标: 最高PSS<=1400MB	1档机型指标: 最高PSS<=1700MB
	2档机型指标: 最高PSS<=1200MB	2档机型指标: 最高PSS<=1400MB
	3档机型指标: 最高PSS<=1000MB	3档机型指标: 最高PSS<=1200MB
IOS平台	1档机型指标: 最高Rsize+Vsize增量<=1100MB	1档机型指标: 最高Rsize+Vsize增量<=1300MB
	2档机型指标: 最高Rsize+Vsize增量<=900MB	2档机型指标: 最高Rsize+Vsize增量<=1100MB
	3档机型指标: 最高Rsize+Vsize增量<=800MB	3档机型指标: 最高Rsize+Vsize增量<=1000MB

跨平台

PC端性能基线

公司设备标准参考Steam数据

机器\配置	CPU	显卡	内存
高端机	i7-9700	2060	16G
中端机	i7-7700	1050T	16G
低端机	i5-9500	G730	8G

同时我们也可以参考Unity标准

Desktop				
Operating system	Windows	Universal Windows Platform	macOS	Linux
Operating system version	Windows 7 (SP1+), Windows 10 and Windows 11	Windows 10+, Xbox One, Xbox Series X S, HoloLens	Mojave 10.14+	Ubuntu 20.04, Ubuntu 18.04, and CentOS 7
CPU	x86, x64 architecture with SSE2 instruction set support.	x86, x64 architecture with SSE2 instruction set support, ARM, ARM64.	Apple Silicon, x64 architecture with SSE2.	x64 architecture with SSE2 instruction set support.
Graphics API	DX10, DX11, DX12 capable.	DX10, DX11, DX12 capable GPUs.	Metal capable Intel and AMD GPUs	OpenGL 3.2+, Vulkan capable.
Additional requirements	Hardware vendor officially supported drivers. For development: IL2CPP scripting backend requires Visual Studio 2015 with C++ Tools component or later and Windows 10+ SDK.	Hardware vendor officially supported drivers. For development: Windows 10+ (64-bit), Visual Studio 2015 with C++ Tools component or later and Windows 10+ SDK.	Apple officially supported drivers. For development: IL2CPP scripting backend requires Xcode. Targeting Apple Silicon with IL2CPP scripting backend requires macOS Catalina 10.15.4 and Xcode 12.2 or newer.	Gnome desktop environment running on top of X11 windowing system Other configuration and user environment as provided stock with the supported distribution (such as Kernel or Compositor) Nvidia and AMD GPUs using Nvidia official proprietary graphics driver or AMD Mesa graphics driver.
For all Desktop operating systems, the Unity Player is supported on workstations, laptop or tablet form factors, running without emulation, container or compatibility layer.				

跨平台

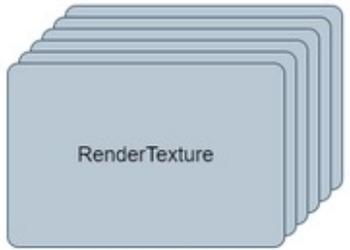
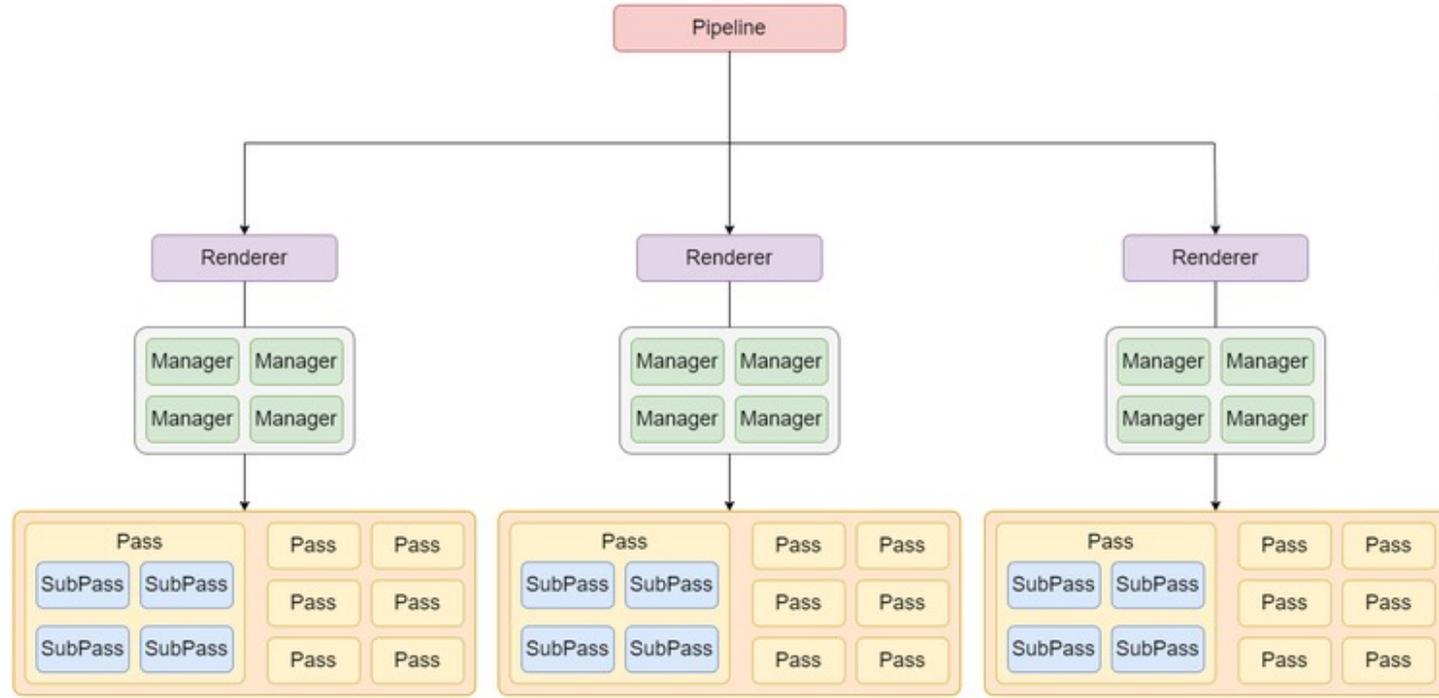
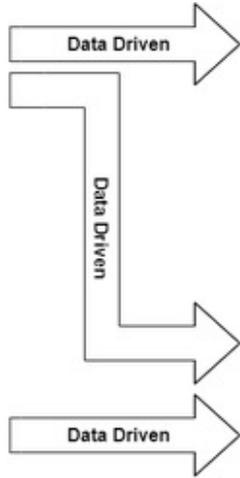
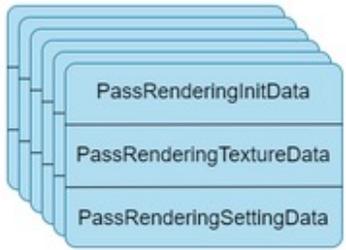
- 渲染管线的性能压力
 - *Cpu*
 - *Gpu*
 - *显存*
 - *带宽等*
- 内存的压力
 - *iOS端的内存限制*
 - *移动端的烘焙方案更费内存*

跨平台



HRP2.x

HRP2 FRAMEWORK



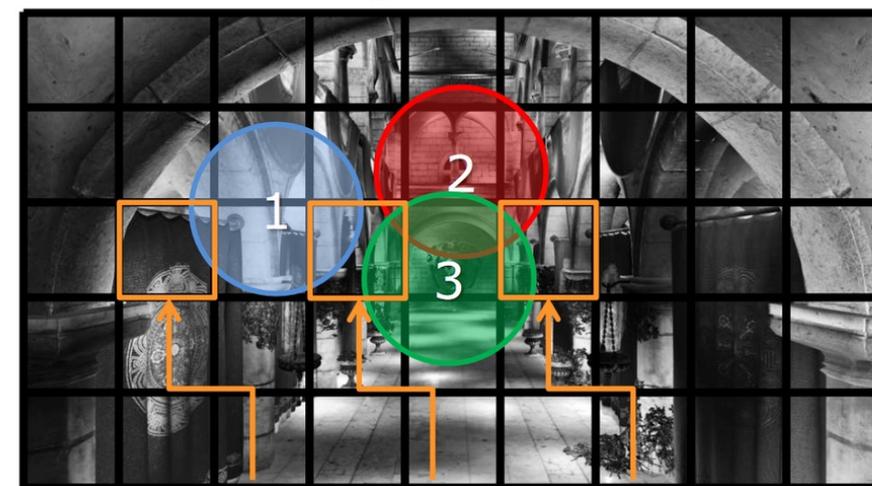
50.0 FPS (20.0 ms) | 1920x1080 | 12/12/16



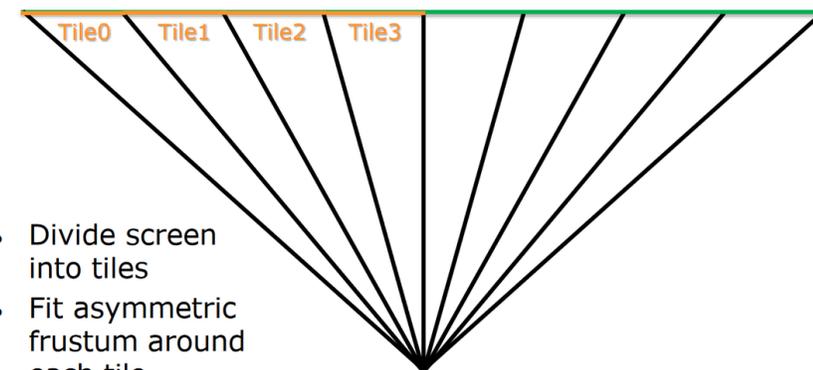
TBDR

针对延时渲染的性能优化，基于TBDR (Tile-Based Deferred Rendering) 的特性：

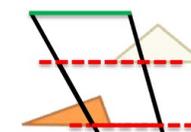
- 支持大量的实时光源 (每个Tile最大16个灯)
- Cluster-Based Deferred Rendering (可以将灯光数进一步提升到每个Tile32, 64个灯)
- Forward+ Rendering (优化半透明物体光照)



[1] [1,2,3] [2,3]



- Divide screen into tiles
- Fit asymmetric frustum around each tile



- Use z buffer from depth pre-pass as input
- Find min and max depth per tile
- Use this frustum for intersection testing

TBDR

TBDR在移动端的优化

GBuffer

Computer Shader

Vulkan

GBuffer & multi-Material

	Format	R	G	B	A
GBuffer0	ARGB32	Normal			Roughness
GBuffer1	ARGB32	Albedo			AO
GBuffer2	ARGB32	Motion Vector		Material ID	Metallic
ColorRT	RGB111110Float	Lightmap + Emission + IBL			
DepthRT	D24S8	Depth			
MetalDepth	RFloat	Custom Depth			
GBuffer0	ARGB32	Normal	Metallic		Roughness
GBuffer1	ARGB32	Albedo			AO
GBuffer2	ARGB32	Motion Vector X		Motion Vector Y	
ColorRT	RGB111110Float	Lightmap + Emission + IBL			
DepthRT	D24S8	Depth			
MetalDepth	RFloat	Custom Depth			
GBuffer0	ARGB32	Normal			Roughness
GBuffer1	ARGB32	Albedo			Metallic
GBuffer2	ARGB32	Motion Vector X		Motion Vector Y	

	Enable MaterialID (Clocks)	Disable MaterialID (Clocks)
GBuffer	3200977	3173519
-----	-----	-----
Tile-Based MaterialID	1398910	-
MaterialID 1 Deferred Lighting	11742800	11094700
MaterialID 2 Deferred Lighting	403781	994927
MaterialID 3 Deferred Lighting	785379	1347090
MaterialID 4 Deferred Lighting	958810	1406310
MaterialID 5 Deferred Lighting	745945	1379430
MaterialID 6 Deferred Lighting	649796	1220220
MaterialID 7 Deferred Lighting	882192	1284070
MaterialID 8 Deferred Lighting	975190	1442980
MaterialID 9 Deferred Lighting	809479	1248510
MaterialID 10 Deferred Lighting	981527	1463680
Total Clocks	19352282	22881917

Compute Shader

- Opengl ES 3.1 + / Vulkan
- Light Sort
 - Parallel Reduction
 - Parallel Scan
 - Parallel Radix Sort
 - Multi-Radix Sort
- CPU&GPU Culling混合算法

Vulkan

- RenderPass和SubPass。SubPass无法独立存在，SubPass必须创建在RenderPass里面。SubPass才是实现渲染内容的真正主体。
- RenderPass里Attachment数量是 $8 + \text{Depth}$ ，但是实际并不是这样。从引擎源码分析可知，最大Attachment数量是7，而且是包含Depth在内的，换言之Color Attachment的数量最多只能有6个

TBDR

TBDR在移动端的优化结果

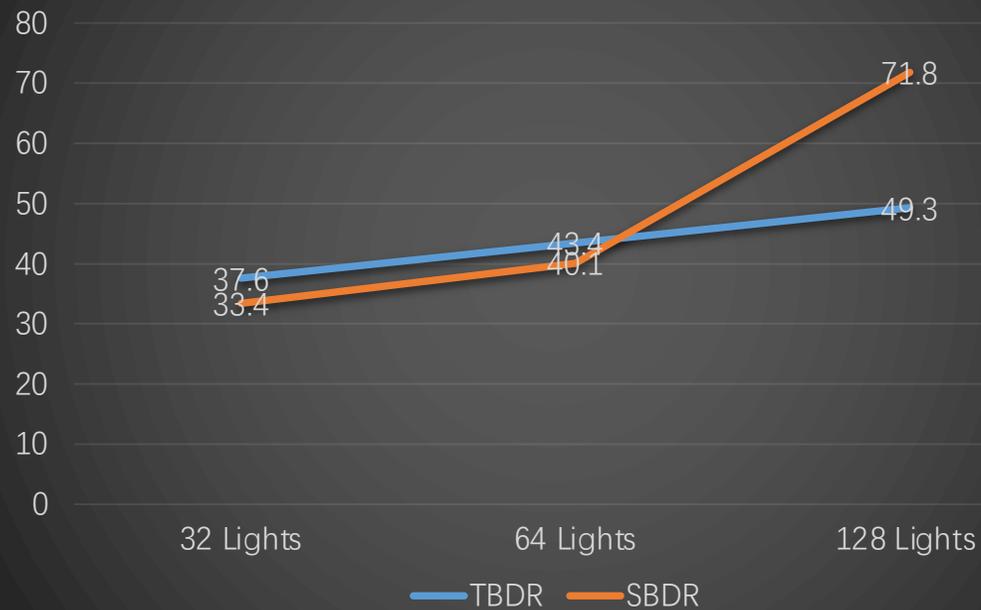
TBDR vs SBDR

测试场景1:

255 Balls

Xiaomi MIX2S Snapdragon845

Frame time



TBDR

TBDR在移动端的优化结果

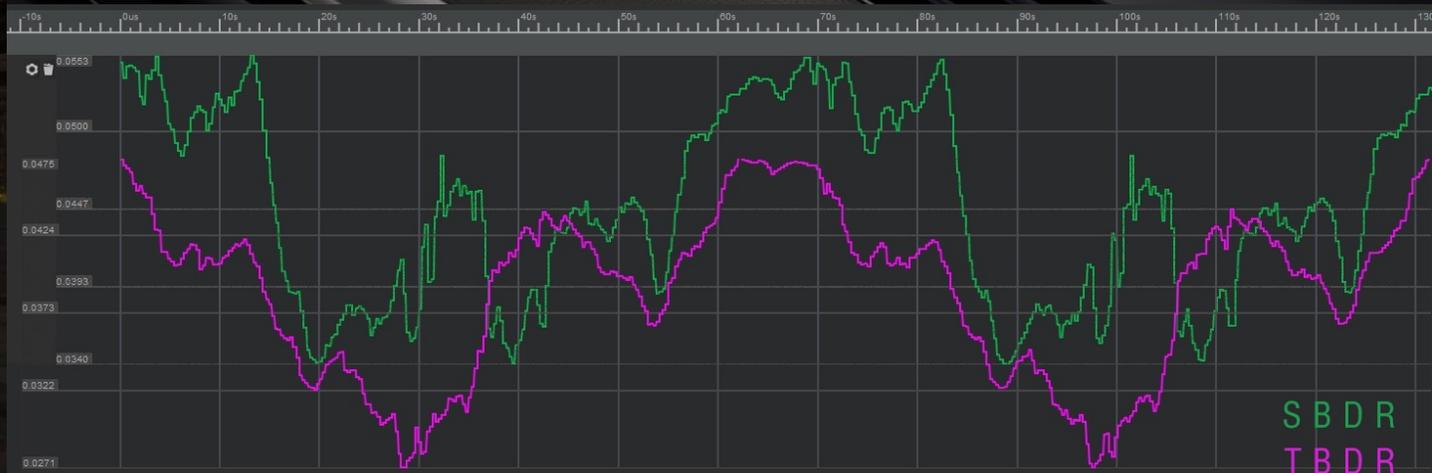
TBDR vs SBDR

测试场景2:

Xiaomi 10 Ultra Snapdragon865

Viking village

1 Directional Light 41 Point Lights



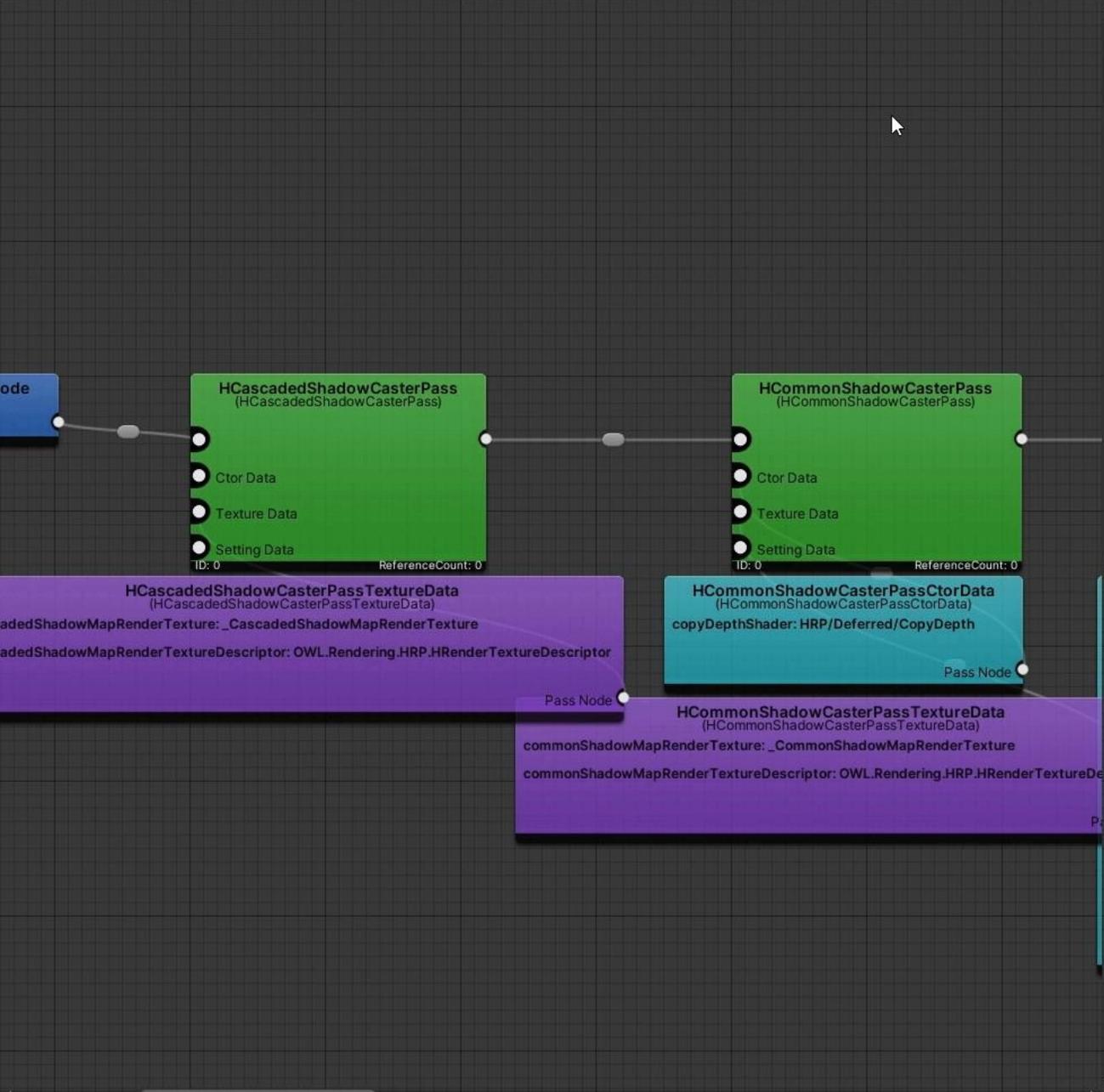
153.0 FPS (6.5 ms) | Direct3D 11.0 [level 11.1]



DELL

跨产品

- URP或者HDRP，大而全的复杂管线
- 项目组定制渲染管线常常与游戏逻辑耦合
- 功能重复开发
- RenderTexture生命周期管理
- 如何将创造性的工作交还给TA或美术



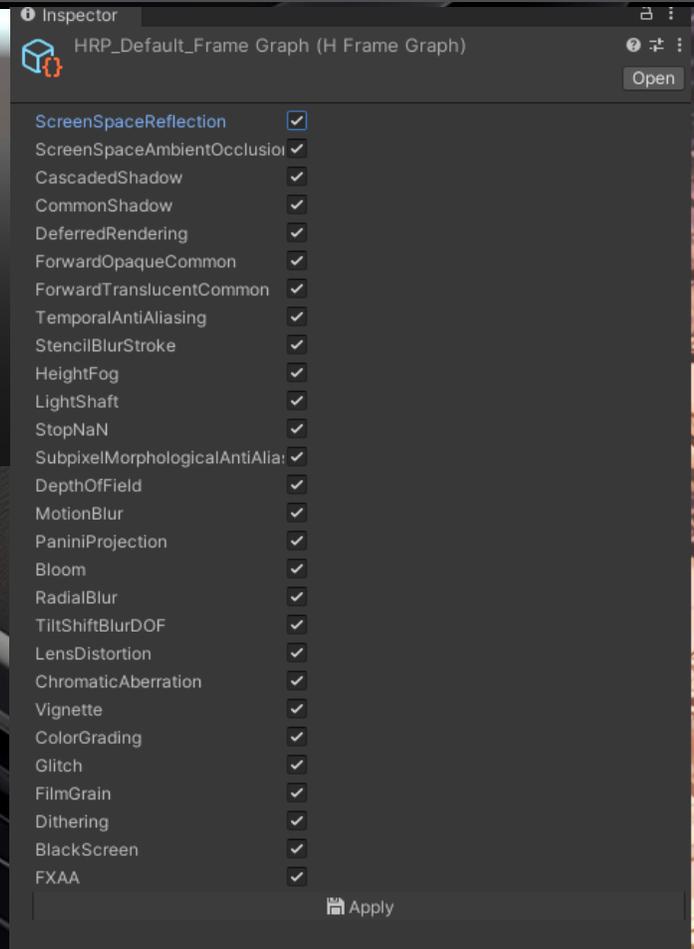
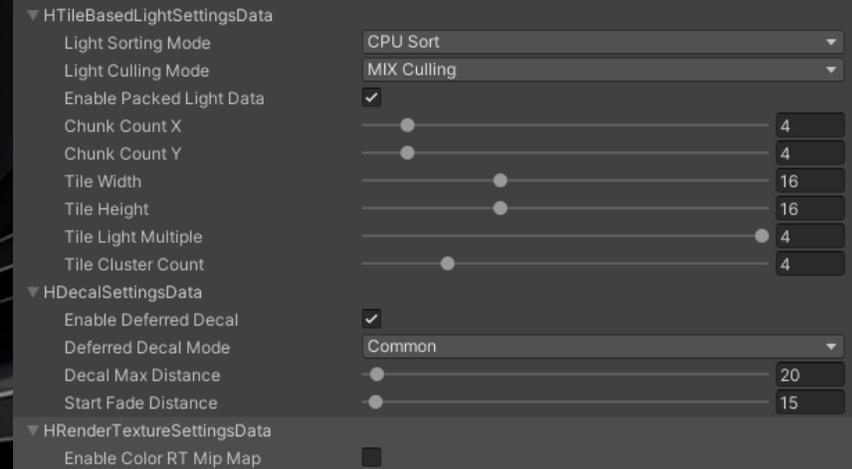
FrameGraph

- FrameGraph Feature管理

- 结点连出的特性统一管理开关
- 支持运行时开关调试
- 支持真机调试

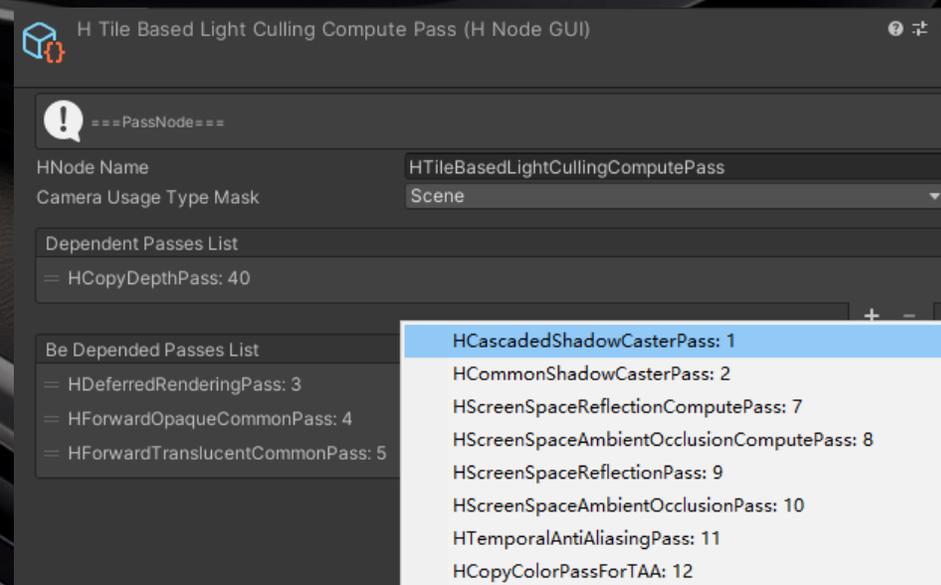
- 全局渲染设置、Pass间公用的数据：HRendererSettingsData

- HSettingManager统一管理
- 支持同时Require多个RendererSettingsData，允许使用者自由组合



FrameGraph 运行时推断

- 运行时简化当前管线的渲染执行逻辑
- 对Pass执行的流程图进行剪枝
- 根据配置来推断
 - PassNode UID
- PassSetup自检
- TODO:
 - 实时推断



EngineDemo_TombRuins - Forest_EnvironmentSample - Android - Unity 2022.3.2f1* <DX11>

File Edit Assets GameObject Component Services Bakery HRPTerrain Tools Jobs HRP TATools Window Help

Game Z [Play/Pause/Stop] Layers Horizontal Layout

Game Display 1 2160x1080 Scale 0.85x Play Focused Stats Gizmos



78.5 FPS | Direct3D 11.0 [level 11.1]

Windows taskbar: 搜索, 任务栏图标, 14:50, 系统托盘

DELL

特性列表

Category	Features	Category	Features
Rendering	Deferred/forward++/forward	Post-Process	LensDistortion
Lighting/shadows	Base-physical Lighting		ChromaticAberration
	Multi-realtime lighting		RadialBlur
	Multi-realtime Shadows/CSM		God-Ray
	Shadow-Cache		Stroke
	Adaptive Probe Volume		Glitch
	Decal/Post Decal		Screen Space Reflection
Anti-Alias	TAA/FXAA/SMAA		Grain
Post-Process	SSAO		Vignette
	SSGI		ToneMapping(ACES,Neutral...)
	Auto Exposure		HeightFog
	MotionBlur		LightShaft
	DepthOfField		
	ColorGrading		
	Bloom		

PostProcess - URP

- 所有的效果都在postprocesspass中实现,

Pass内容复杂

- 改动成本大, 不够灵活

```
// Optional NaN killer before post-processing kicks in
// stopNaN may be null on Adreno 3xx. It doesn't support full shading
if (useStopNaN)...
```

```
// Anti-aliasing
if (useSubPixelMorpAA)...
```

```
// Depth of Field
// Adreno 3xx SystemInfo.graphicsShaderLevel is 35, but instancing
// DOF shader uses #pragma target 3.5 which adds requirement for instancing
if (useDepthOfField)...
```

```
// Motion blur
if (useMotionBlur)...
```

```
// Panini projection is done as a fullscreen pass after all depth-based passes
// and before bloom kicks in
if (usePaniniProjection)...
```

```
// Lens Flare
if (useLensFlare)...
```

```
// Combined post-processing stack
using (new ProfilingScope(cmd, ProfilingSampler.Get(URPProfileId.UpperHalf)))
{
    // Reset uber keywords
    m_Materials.uber.shaderKeywords = null;

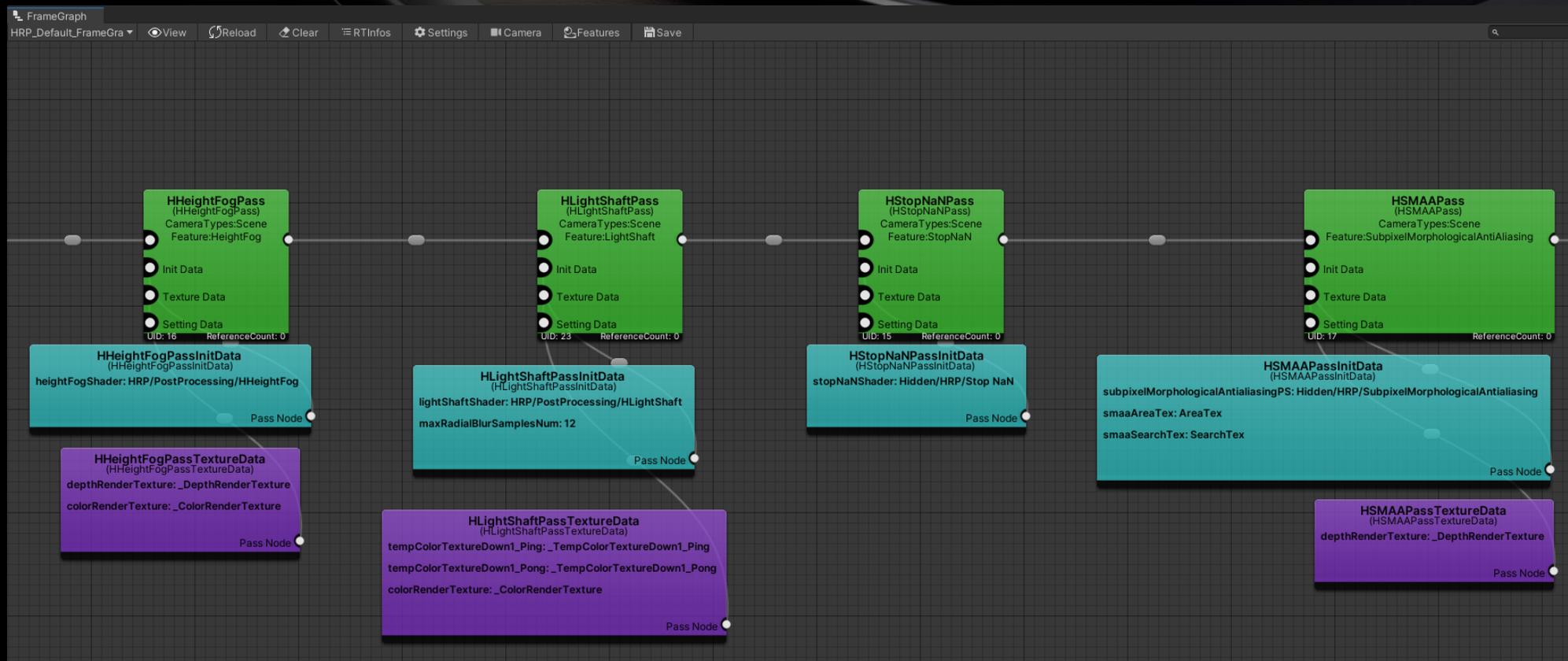
    // Bloom goes first
    bool bloomActive = m_Bloom.IsActive();
    if (bloomActive)...
```

```
// Setup other effects constants
SetupLensDistortion(m_Materials.uber, isSceneViewCamera);
SetupChromaticAberration(m_Materials.uber);
SetupVignette(m_Materials.uber);
SetColorGrading(cmd, ref renderingData, m_Materials.uber);

// Only apply dithering & grain if there isn't a final pass.
SetupGrain(cameraData, m_Materials.uber);
SetupDithering(cameraData, m_Materials.uber);
```

PostProcess

● PostProcessPass拆分 + Frame Graph

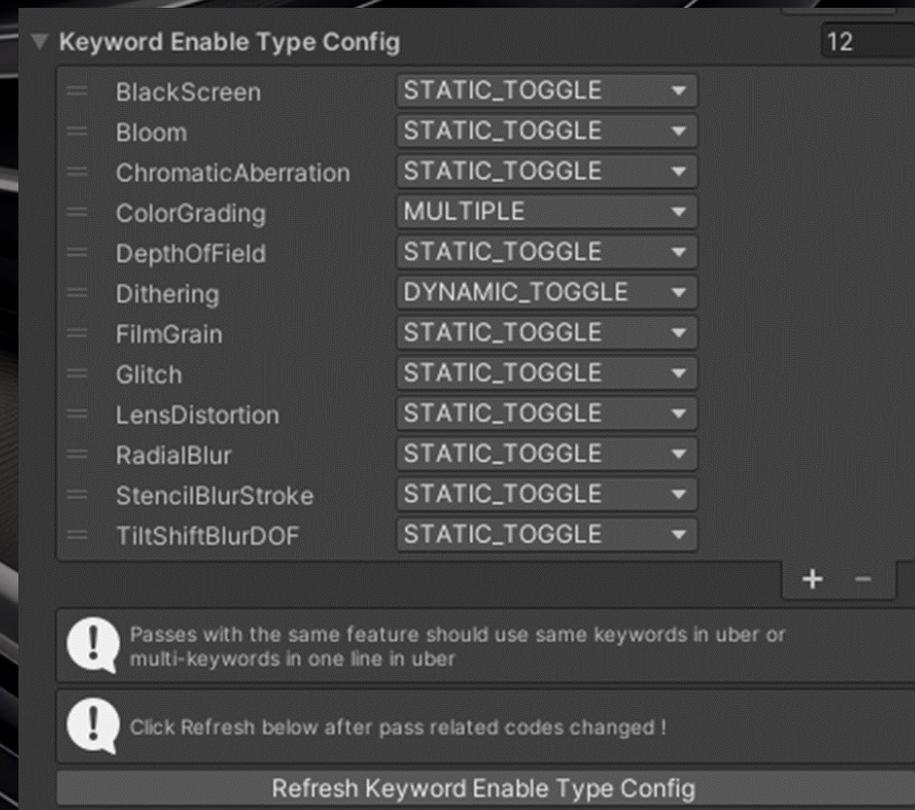


PostProcess - HRP

- 分类一：直接绘制到_ColorRenderTexture上
 - HHeightFogPass 雾效
 - HLightShaftPass 光束
 -
- 分类二：处理Color texture
 - HCameraMotionBlurPass 相机运动模糊
 - HMotionBlurPass 运动模糊
 - HDepthofFieldBokehPass Bokeh景深
 - HDepthofFieldGaussianPass 高斯景深
 - HFXAAPass
 - HSMAAPass
 -
- 分类三：UberPost.shader [HUberBlitPass]
 - HColorGradingPass
 - HBlackScreenPass
 - HBloomPass
 - HChromaticAberrationPass
 - HDepthofFieldLowPass
 - HGlitchPass
 - HLensDistortionPass
 - HRadialBlurPass
 - HStencilBlurStrokePass
 - HTiltShiftBlurDofPass
 - HVignettePass
 -

HRP/Uber shader 变体剔除

- 遍历指定路径下的 post process profile 文件来生成变体
- 开启某个关键字的条件：
 - profile中的参数设置
 - GlobalUberStrippingSettings中配置的feature关键字的启用类型
 - Frame graph中相应的pass是否生效



Decal

修改金属度



修改粗糙度



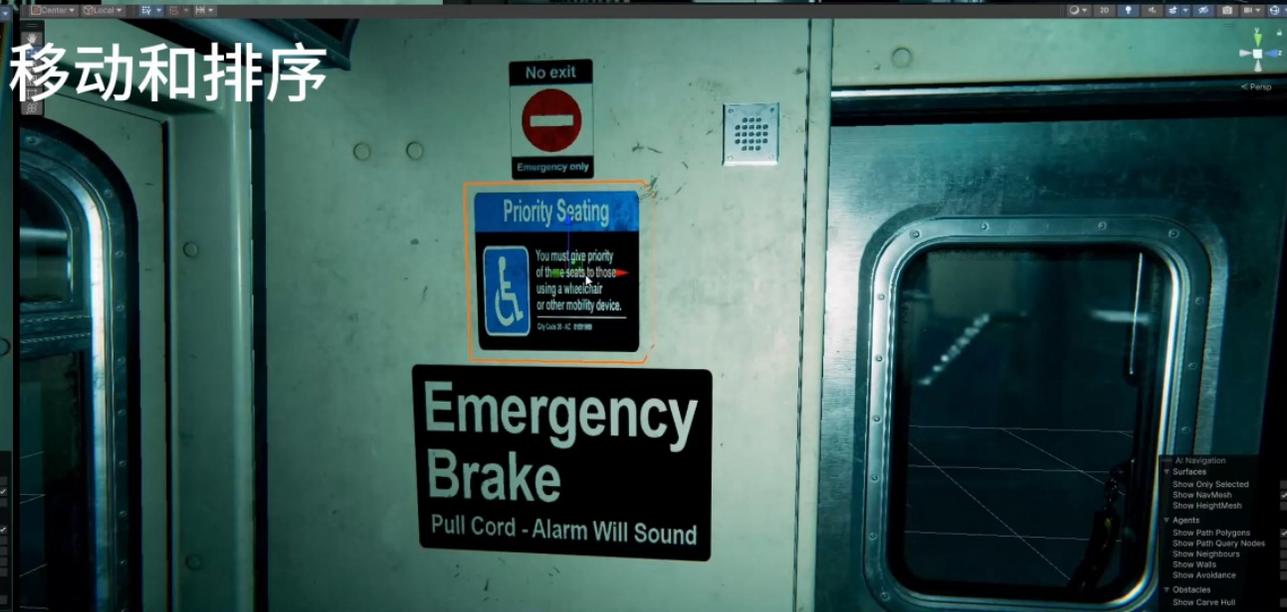
边缘处理



淡入淡出



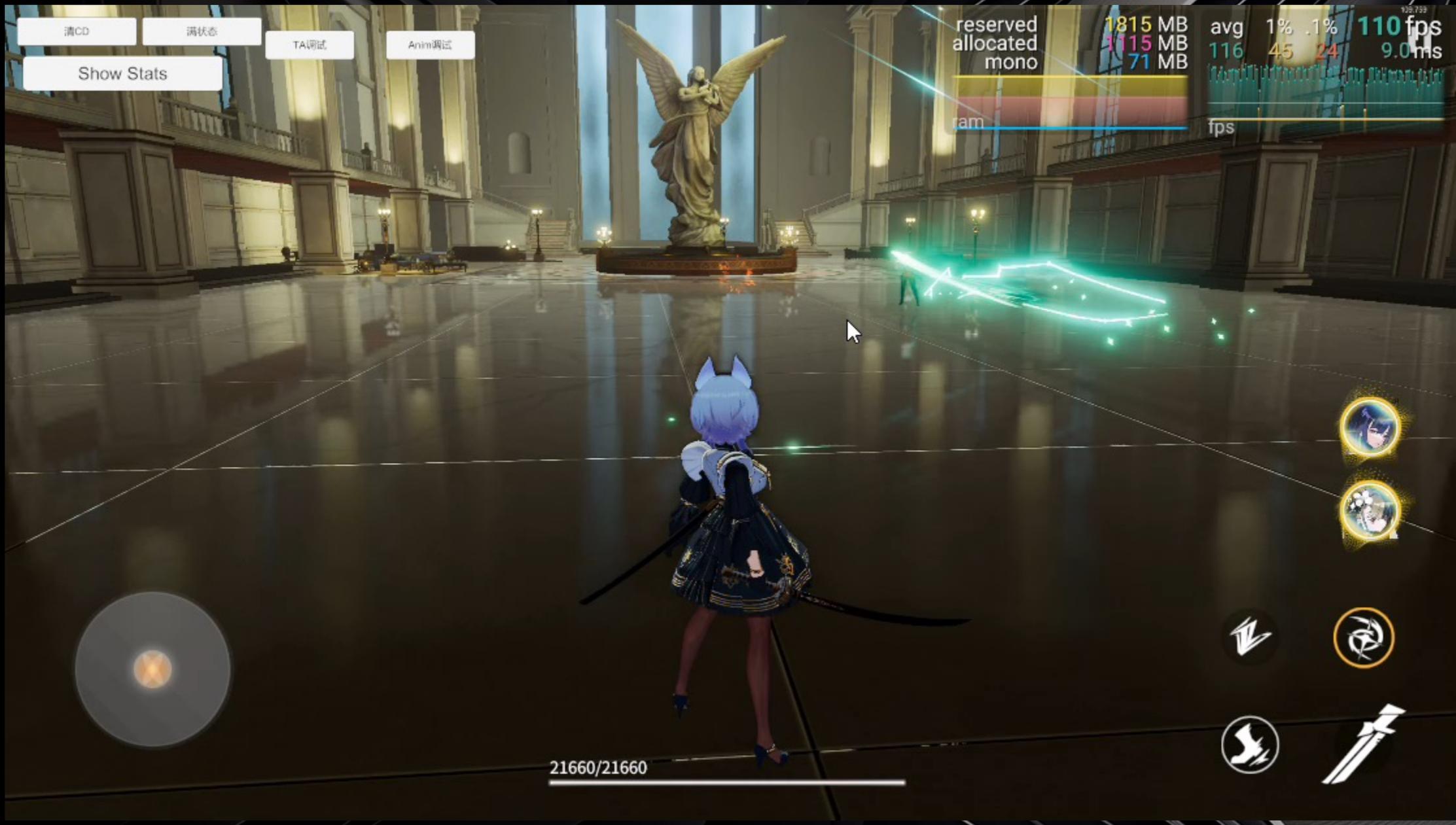
移动和排序



SSR



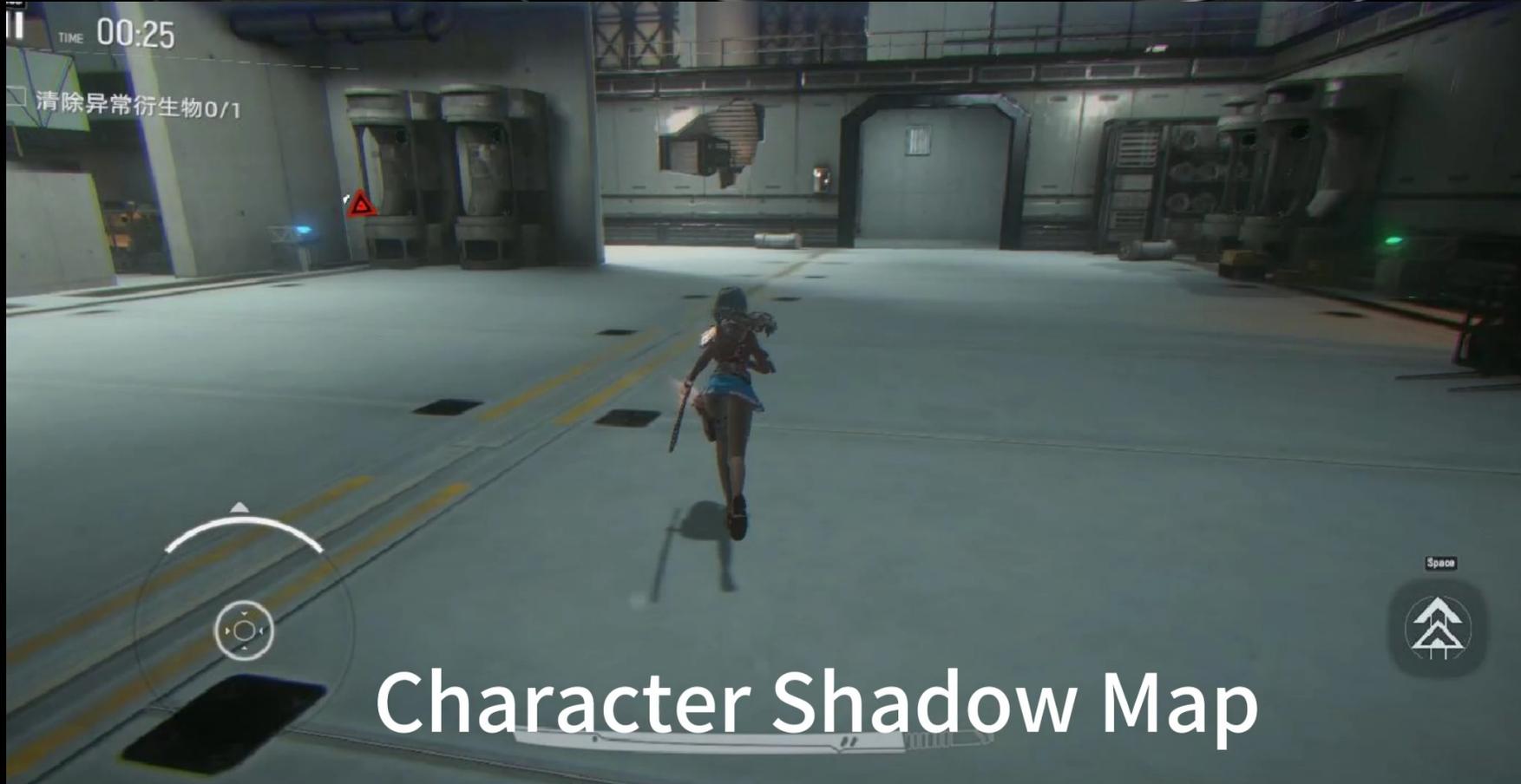
SSPR



Shadow Cache



SH Character ShadowMap



工具列表

Tool	Description
Bake2UTo1U	将物件的2U烘焙到1U
BakeSHCodedDAO	烘焙单个物件的基于SH（球谐）的DirectionalAO
BuildLightmapDataWithLOD	对包含LOD的场景Static物件进行Lightmap烘焙
DebugColor	调试着色计算过程中不同类型数据的着色结果
Foliage	草地编辑系统
SceneAssetsUsageStatistics	检查当前场景使用到所有资源的情况
SceneMipmapChecker	检查场景的Mipmap情况
SceneProbe	依据摄像机获取设定区域内的渲染数据，包括batch、draw call、三角形面数、顶点数，以及区域内渲染物体按顶点数、三角形面数、使用贴图大小排序的详细信息
Heatmap	扩展Unity的DrawMode（即Unreal的ViewMode），将场景中的信息（Shader复杂度，顶点密度，LOD等）以热图方式显示
ShadowMaskTool	生成并替换当前LightingData中的Shadowmask
ExposureDebugger	自动曝光的debug工具
Effect Sorting	批量修改特效的sorting信息
HRPTerrain	将unityTerrain切块，Splatmap生成，LOD,HLOD生成

兼容性测试



HRP1 0(0.1) 其他

2021/11/16 13:36:35



结果概况

终端数: 50 未通过机型数: 1 [查看测试步骤](#)



问题终端数 : 1
成功终端数 : 49
测试终端总数 : 50

问题定位

问题总数: 1



■ 安装失败 : 0
■ 启动失败 : 0
■ 闪退 : 0
■ 黑屏&白屏 : 0
■ 卡死 : 0

机型分布

终端数: 50 未通过机型数: 1



性能数据

总项: 6 优胜: 3



HRP 1(0.1) 其他

2021/10/29 10:26:28



结果概况

终端数: 300 未通过机型数: 18 [查看测试步骤](#)



问题终端数 : 18
成功终端数 : 282
测试终端总数 : 300

问题定位

问题总数: 18



■ 安装失败 : 0 ■ 卡顿 : 0
■ 启动失败 : 0 ■ 连接异常 : 0
■ 闪退 : 2 ■ UI异常 : 14
■ 黑屏&白屏 : 2 ■ 其他异常 : 0
■ 卡死 : 0 ■ 功能兼容性异常 : 0

机型分布

终端数: 300 未通过机型数: 18



性能数据

总项: 6 优胜: 4



最佳实践

渲染框架：HRP

新手必读

基本介绍

最佳实践

Color Space

G-Game接入HRP操作文档

竞品对比：BuildIn、HRP&URP的功能差异

更新公告

关于“重要说明”

1.0.47重要说明

1.0.60重要说明

1.0.68重要说明

1.1.0重要说明

功能介绍

HRP1

HRP2

框架介绍

延迟渲染

FrameGraph基础介绍

FrameGraph交流页面

FrameGraph更新日志

FrameGraph使用文档

FrameGraph F&Q

代码规范

ToolSet

Bake2UTo1U

BakeSHCodedDAO

BuildLightmapDataWithLOD

DebugCharacterGrade

DebugColor

DebugSceneGrade

Foliage

SceneAssetsUsageStatistics

SceneMipmapChecker

SceneProbe

热图

RenderPipelineDebugger

检查UNITY VERTEX INPUT INSTANCE ID

局部后处理

半透明物体写入深度图

如何烘焙自发光材质

RenderFeature使用DepthNormal

版本规划

HRP Roadmap 2022

延时测试版本

常见问题

PowerVR GE8320机型出现了花屏与卡顿（记录一次...）

镜像网格

Scene窗口中Gizmos显示层级异常

后效配置说明

HRP Lightmap压缩过程中稳定崩溃

如何控制Scene Camera的CopyColor

HRP样板间

Ggame样板间说明

Dawn样板间说明

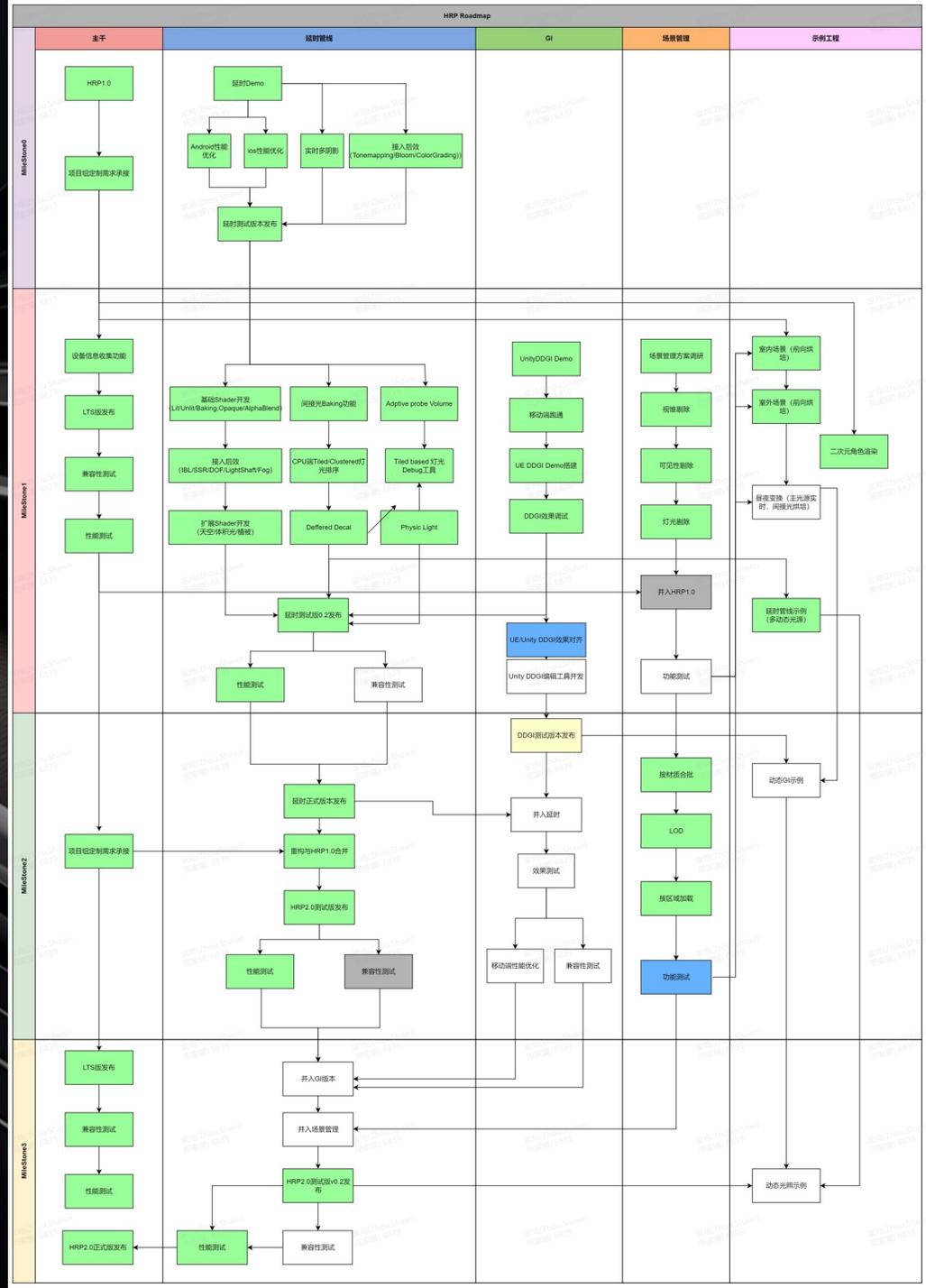
美漫样板间介绍

Dawn角色样板间

RoadMap

ToDo:

- GI
- 场景管理/开放世界
- GPU Driven



谢谢观看

HRP交流群



招聘

