

黎明之海手游技术分享

搜狐畅游-张华

- 基于Unity定制24小时昼夜交互系统
- 如何丰富DIY角色定制
- 手游优化经验浅谈

《黎明之海》是一款写实风格的航海冒险MMORPG手游，具备高自由度和社交属性。游戏以真实世界无缝大地图为舞台，玩家化身为新入船长，带领舰队环游世界，开启全球贸易之旅，探索上古文明和遗迹宝藏，在环游世界的历险中不断成长，最终成为伟大的冒险家。

基于Unity定制24小时昼夜交互系统

昼夜交互系统要素

- 日出日落
- 明月高悬
- 星海灿烂
- 晴空万里
- 夕阳西下
- 狂风暴雨
- 风霜落雪
- 电闪雷鸣
- 友好的编辑器







昼夜天气系统编辑器是针对黎明之海手游项目定制的一个24小时昼夜轮转系统，支持多个24小时TOD(Time Of Day)的轮转，并且支持在24小时TOD时间中进行气候变化。

已经完成的模块

- **Day Profiles**文件系统：用来存储多个24小时TOD变化文件；
- **气候控制模块**：控制在24小时TOD轮转时突发的气候变化，如雨、雪、浓云密布等等气候效果；
- **声音控制模块**：控制天气或气候变化的时候的声音同步效果；
- **高度雾控制模块**：集成了新开发的高度雾控制；
- **日历系统模块**：目前开放了可供美术编辑的多个24小时TOD；
- **天空系统模块**：包括日光夜光、太阳月亮的动态变化；
- **后处理控制模块**：可以控制Unity官方提供的PostProcessing，以便可以更好的让后处理效果跟随24小时TOD进行变化；

技术要点

- 程序化天空+真实大气散射+光照
 - 可调空气密度、Rayleigh散射、Mie散射等大气光照要素
- 外太空
 - 日月+银河+星空
- 高度雾、云
 - 基于纵向和深度可控的高度雾，不同海拔支持不同效果的调整
 - 多种类型云层叠加







马尔代夫
夜光沙滩

技术要点

- 多样化的天气

 - 多层雨水效果可调

 - 场景模型自动计算出雨雪材质

 - 闪电+雾天资源调节

 - 支持多套气候瞬切、渐变过度、多套融合

- 海洋系统

 - 真实海水效果，多套水体系统适配各种机器







Time Of Day



优势：

极大的提升了美术和策划的工作效率，通过简洁的编辑界面可以无限扩展美术场景的效果及方便策划快速成型游戏。

支持多昼夜切换，支持各种天气变化，可视化编辑，提供所见即所得的实时预览效果。

支持真实时TOD及烘焙模式的TOD编辑。



Inspector

TOD_dizhonghai_B30426_zzy_4_ye2

DAY PROFILE

散射

空气密度 2.52
Kr 4.63
Km 3
Wavelength R 850
Wavelength G 600
Wavelength B 479

The above properties will not take effect if the Wavelength is set to Precomputed.

Rayleigh
Mie
Night Intensity
Rayleigh Color
Mie Color
Sun Intensity

此值控制在天空盒上的太阳和太阳轮盘的强度,不要和太阳平行光中控制场景的那个强度混淆。

日轮

日轮
太阳大小
太阳光晕强度
曝光系数

外太空

月亮:
月亮颜色
月空亮度颜色
月空亮度范围
月亮自发光因子
银河
星星(新):
星星大小
星星闪烁数量
星星闪烁速度

高度雾 Show
云层 Show
光耀 Show
天气 Show
海水 Show
新海水 Show
质处理 Show
音频 Show
特殊效果 Show

DAY PROFILE

散射

Hide

空气密度 2.52 R

Kr 4.63 R

Km 3 R

Wavelength R 850 R

Wavelength G 600 R

Wavelength B 479 R

The above properties will not take effect if the Wavelength is set to Precomputed.

Rayleigh R

Mie R

Night Intensity R

Rayleigh Color

Mie Color

Sun Intensity R

此值控制在天空上的太阳和太阳轮盘的强度,不要和太阳平行光中控制场景的那个强度混淆.

日轮 R

太阳大小 R

太阳光晕强度 R

曝光系数 R

外太空

Hide

月亮:

月轮颜色

月空亮度颜色

月空亮度范围 R

月亮自发光因子 R

银河 R

星星(新):

星星大小 R

星星闪烁数量 R

星星闪烁速度 R

高度雾

Show

云层

Show

光耀

Show

天气

Show

海水

Show

新海水

Show

后处理

Show

音频

Show

特殊效果

Show

天气

Hide

使用雨&闪电

雨:

雨水强度 R

雨水速度 R

雨水第一层方向 R

雨水第二层方向 R

屏幕雨滴强度 R

闪电:

绕X轴旋转角度 -10 R

绕Z轴旋转角度 -90 R

绕Z轴角度随机范[0 R

闪电相机距离 100 R

闪电大小 50 R

闪电生命周期 1 R

闪电峰值时间 0.4 R

闪电间隔时间 3 R

闪电间隔时间在 3.0 到 5.0 秒内随机.

雨亮度增强系数 0.2 R

周期亮度曲线 R

周期闪白曲线 R

闪电控制(24小时) R

闪电音效(Event) R

雷声立即切换

使用雪

海水

Show

新海水

Hide

***细节法线贴图:

扰动强度 R

***散射:

直视水面基本色

斜视水面基本色

***次表面散射:

主光贡献基本色

主光贡献量倍乘器 R

主光方向上穿过波的光 R

主光方向上的散射衰减 R

***浅水效果:

浅水深度范围 R

浅水颜色

***环境反射:

高光反射强度 R

菲涅尔衰减参数 R

水折射率 R

***海水平面反射:

反射扭曲程度 R

反射强度 R

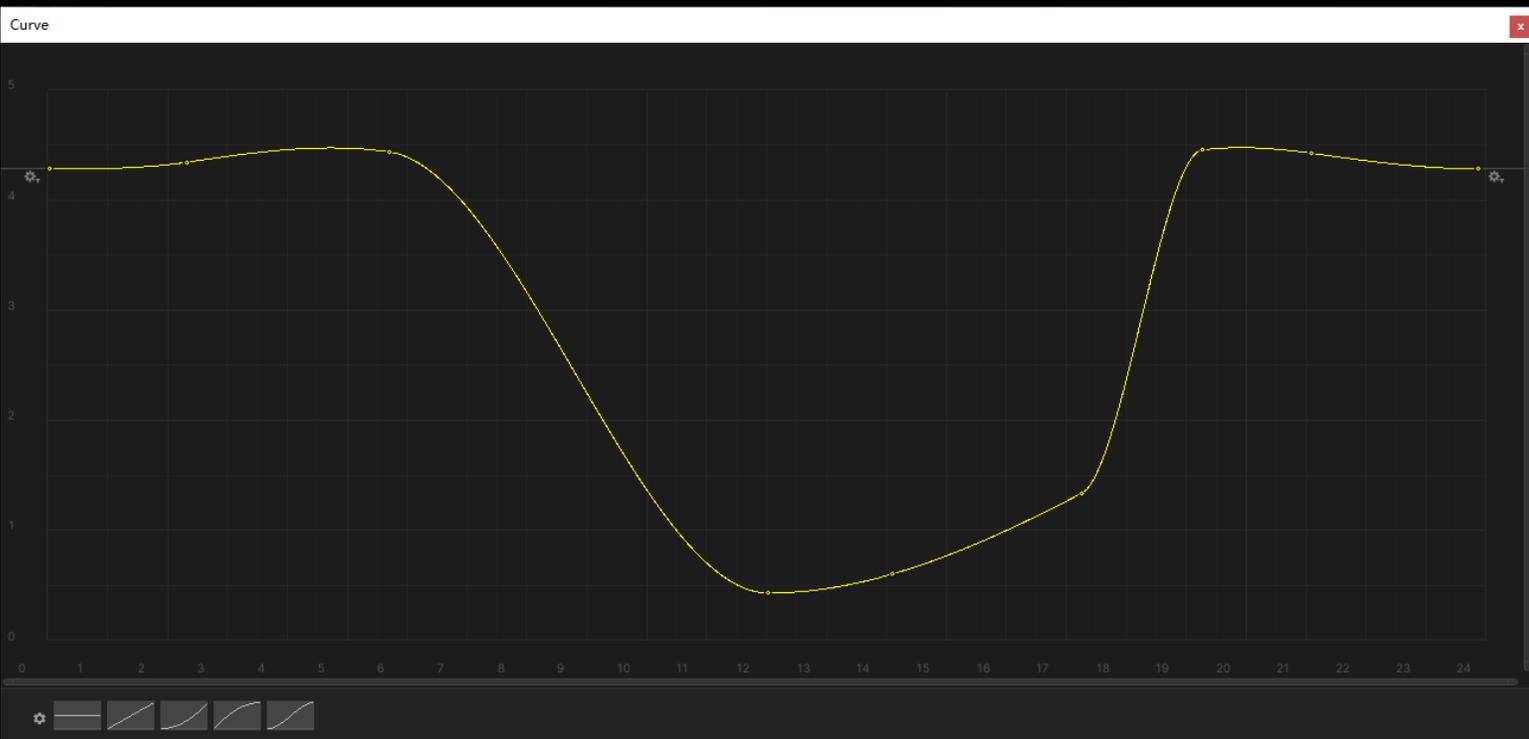
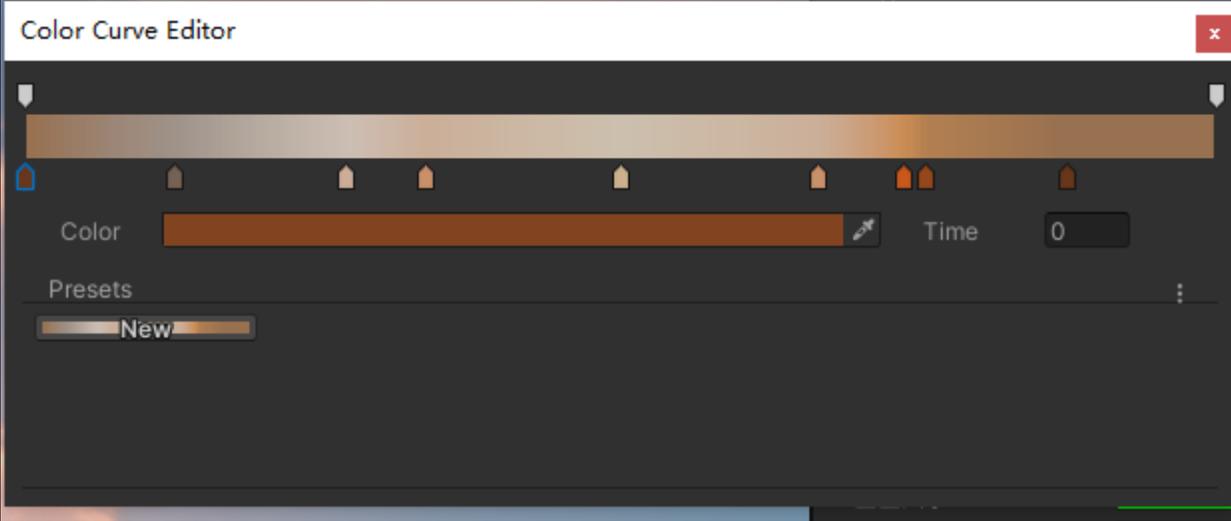
***方向光光带:

光带衰减参数 R

光带衰减最远距离 R

光带在远距离的衰减参数 R

效果增强参数 R



真实海洋交互系统

真实海洋交互系统是针对手游和端游项目定制的一套模拟移动端真实海水起伏动态真实感变化。

提供了多套海洋渲染解决方案，用于解决不同配置机器的性能差距。

功能特点

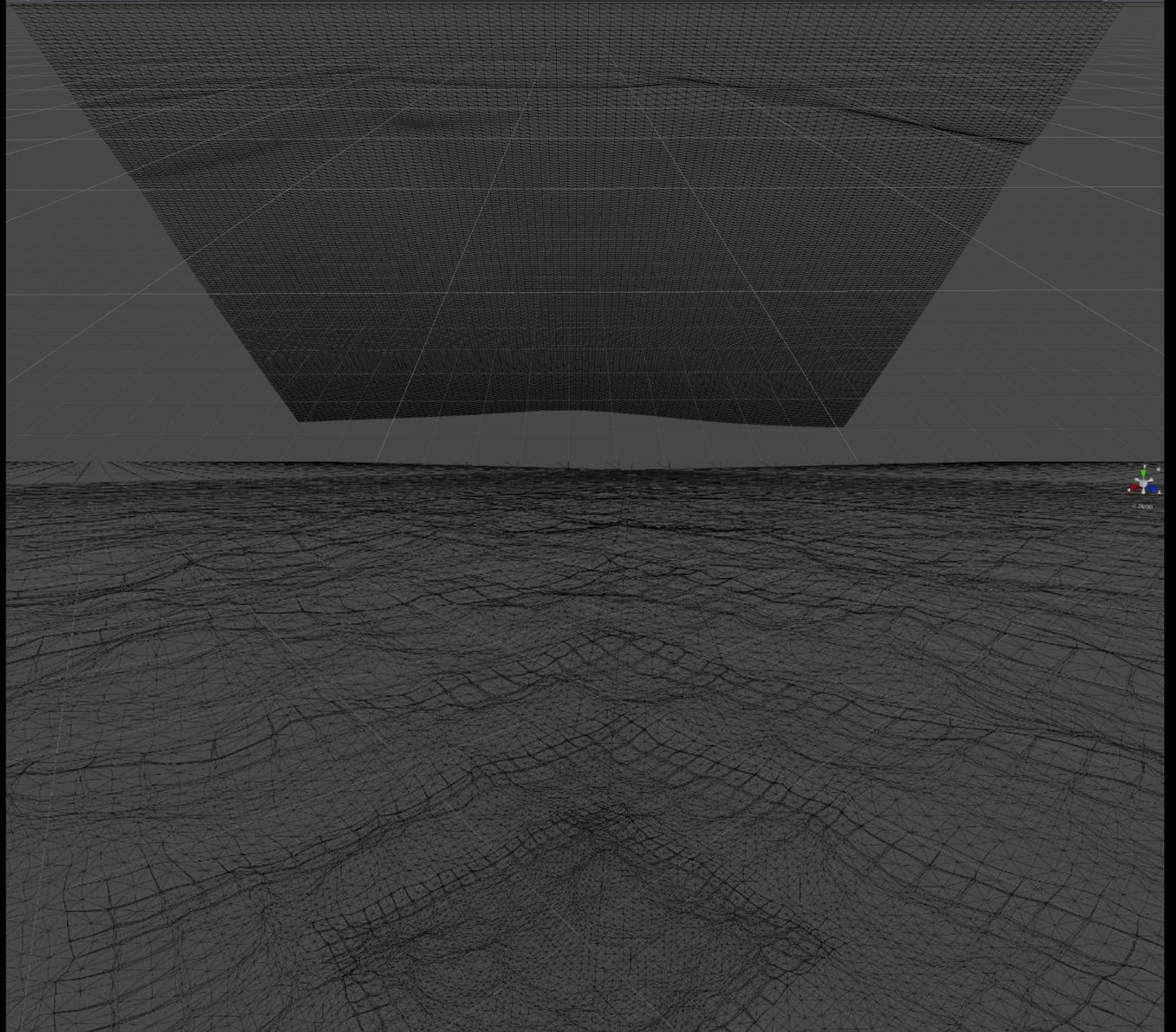
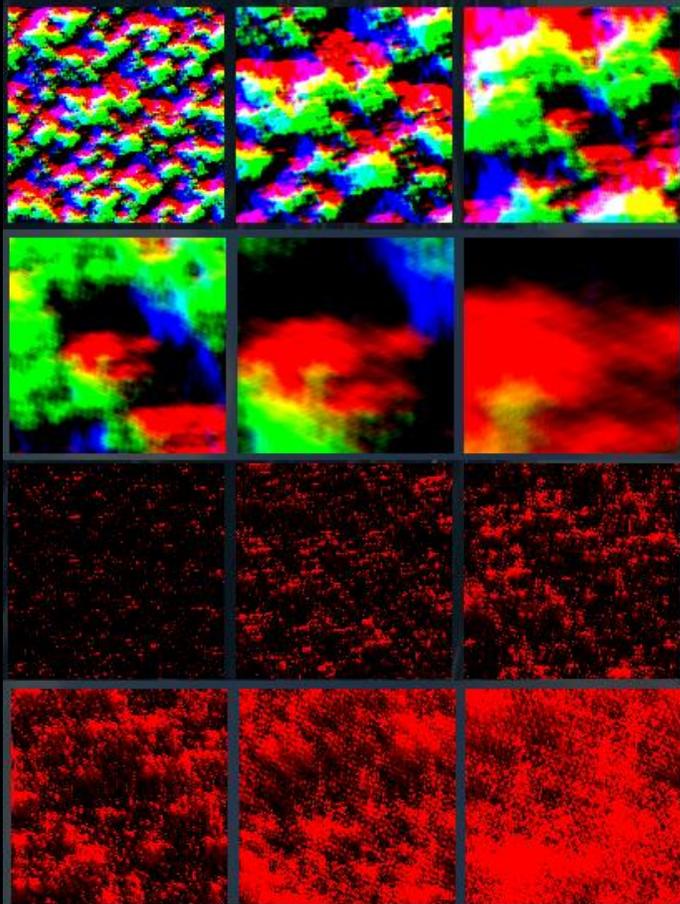
- 基于通用着色计算技术，在一些支持GPU异步回读的硬件平台实现了更自然的海水大起伏模拟查询系统，并且实现了多层次LOD海水变化，进一步扩展了更真实的基于物理光照的海洋材质效果。
- 支持多种波形叠加模拟更自然的真实海水。
- 可实时变化各种大小波形渐变。

多套海洋效果对比



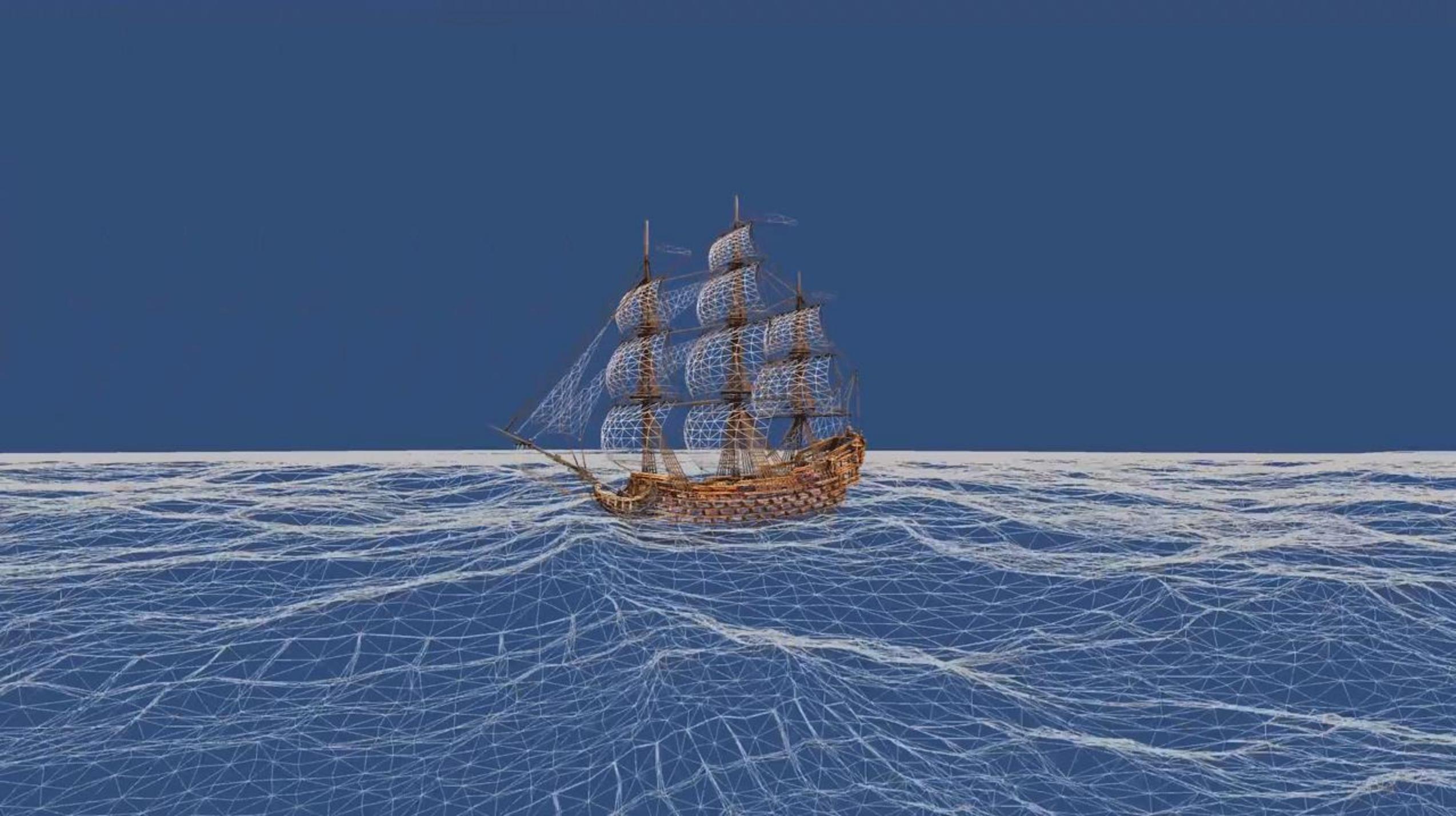
多种类型波形

- 采用投影网格方式
- 多层次LOD Mesh



回读高度数据

- CPU模拟GPU实时计算高度数据
- GPU异步回读
Copy resource data from the GPU to the CPU without any stall
- 支持获取多个位置的数据



UnderWater

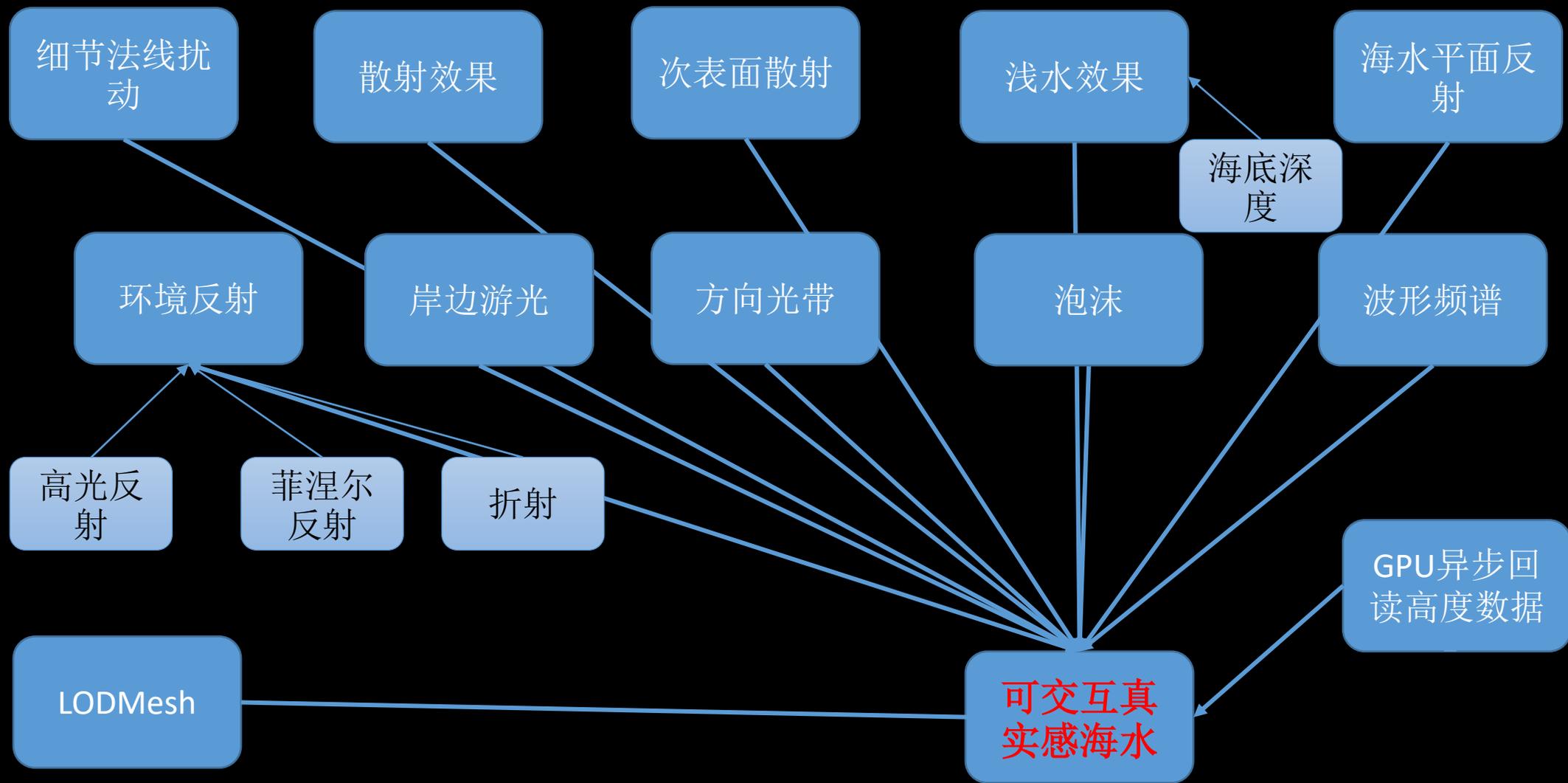


浅滩优化





架构图



<http://advances.realtimerendering.com/s2017/index.html>

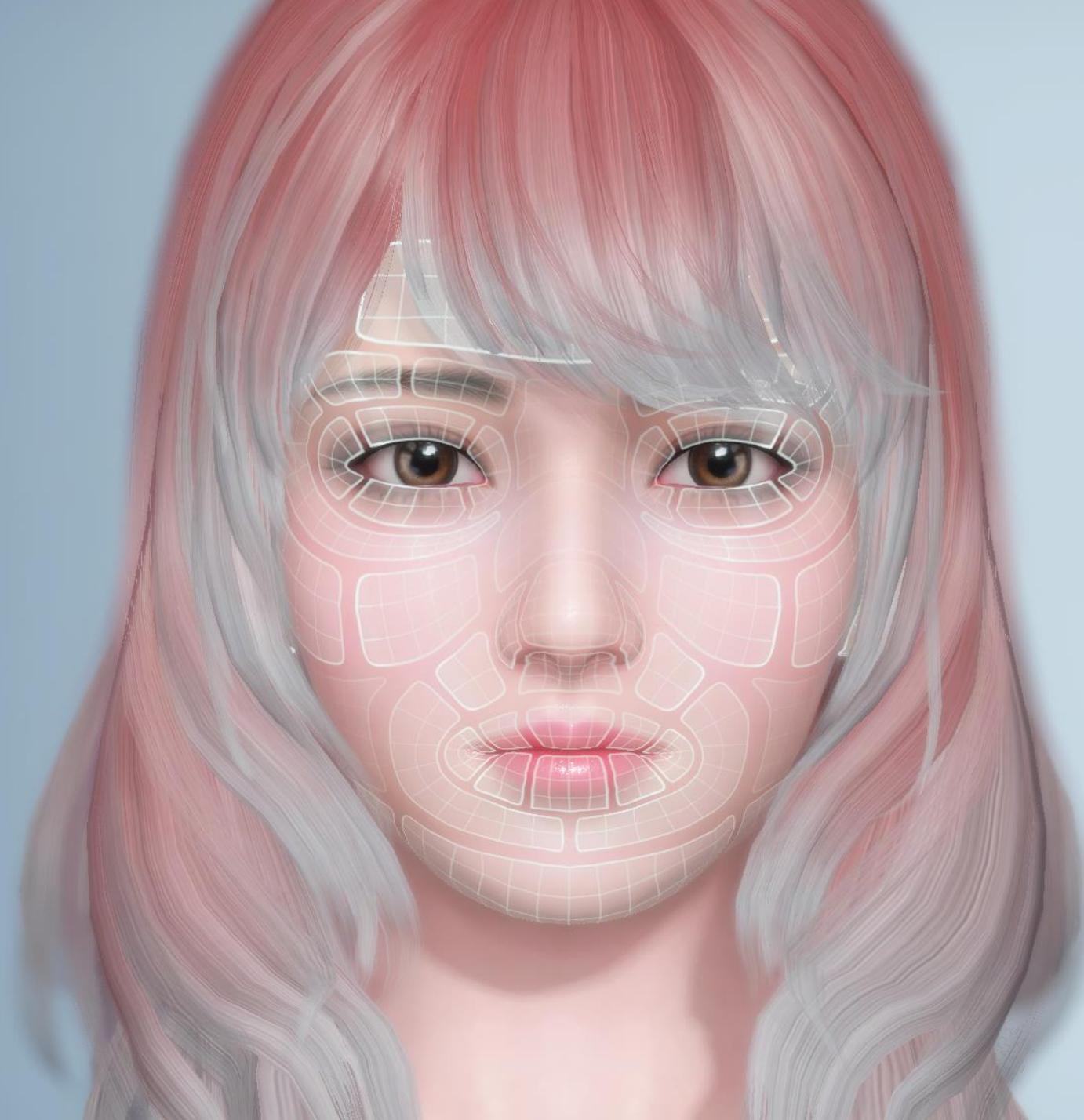
<http://advances.realtimerendering.com/s2019/index.htm>

如何丰富DIY角色定制



DIY元素

- 发型
- 面纹
- 肤色
- 眉型
- 美瞳
- 眼影
- 唇彩
- 睫毛
- 唇釉
- 脸型
- 眉毛
- 眼睛
- 鼻子
- 嘴巴
- 耳朵





技术要点

- 动捕表情
- 物理布料(Cloth+Dynamic Bone)
- 骨骼蒙皮+BlendShape

Morpher表情编辑器

ARKit Face Tracking

专业动捕工具

声音驱动口型插件

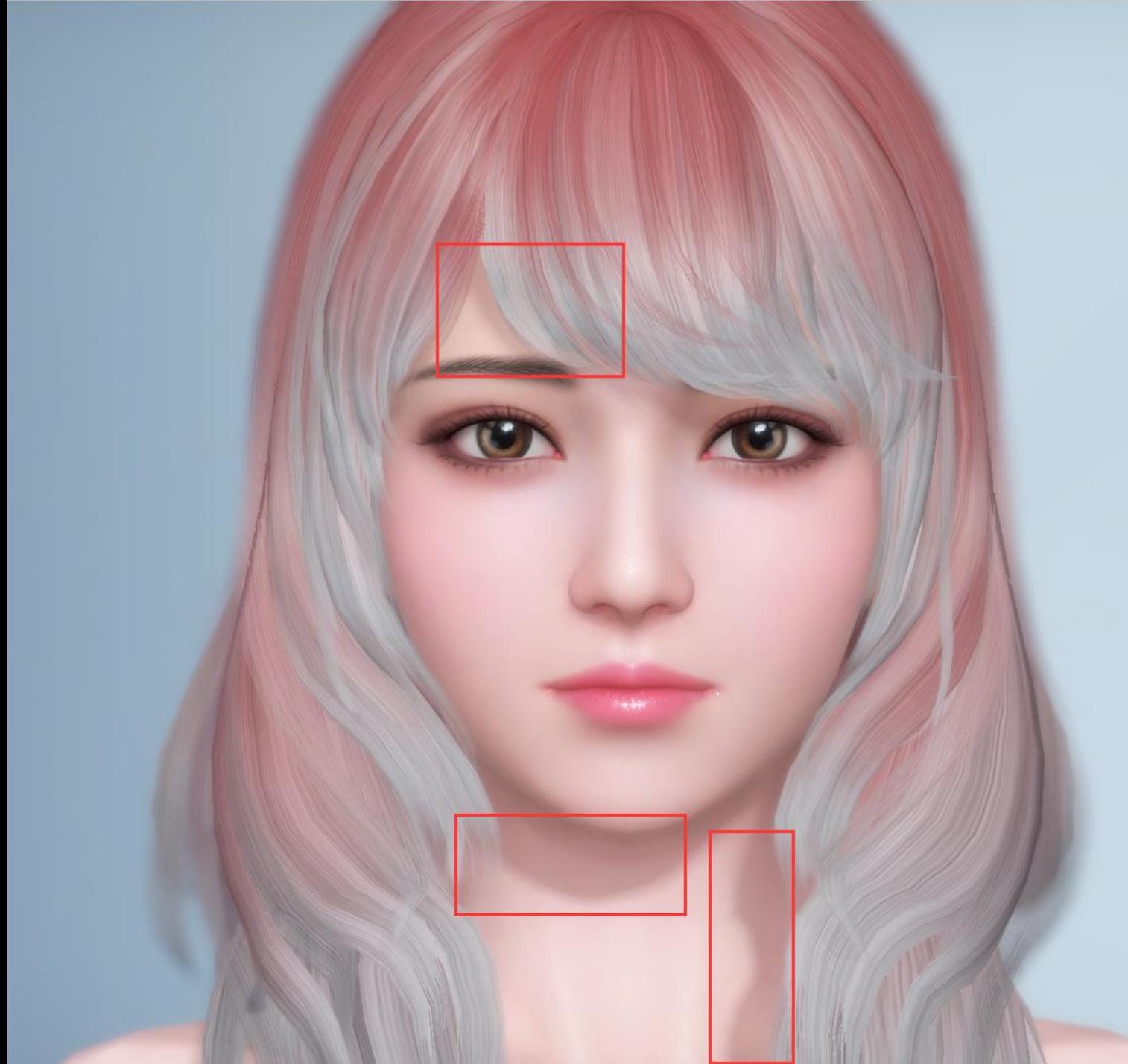
技术要点

Shadow Map分辨率不足，导致如果角色开启受阴影，即使开最大分辨率也会导致阴影的锯齿明显，所以一般角色的受影可能会被关闭。

调整Shadow Map分辨率或者动态调整阴影距离，可能会缓解，但是有些情况下无法满足场景阴影和自阴影共存。

解决方案：

对角色单独渲染ShadowMap+PCF
Soft Shadow



技术要点

特化 Shader

内置的Standard Shader
比较适用于金属、木质、
石头等大部分常见物体，
部分内部结构比较复杂的
材质比如布料、皮肤等不
太好表现。

Pre-Integrated Skin Shading

次表面散射

多反射眼球

特制唇釉

湿润效果

各向异性(Kajiya-Kay)

其它: HDRP



```

65     };
66 };
67
68 #define SAMPLE_COUNT 16
69 //rg: PCF采样宽度, b: 阴影强度
70 uniform half4 _SelfShadowParams;
71 uniform float4x4 _SelfShadowMatrix;
72 UNITY_DECLARE_SHADOWMAP(_SelfShadowTexture);
73
74 half SampleUnique(const half4 coords) {
75     half4 uv = coords;
76     half shadow = 0.f;
77     for(int i = 0; i < SAMPLE_COUNT; ++i) {
78         uv.xy = coords.xy + poisson[i] * _SelfShadowParams.rg;
79         shadow += UNITY_SAMPLE_SHADOW(_SelfShadowTexture, uv.xyz);
80     }
81
82     shadow /= SAMPLE_COUNT;
83     shadow = _SelfShadowParams.b + shadow * (1-_SelfShadowParams.b);
84     return shadow;
85 }
86

```

```

#ifdef _RAIN_ON
    float coefficient = _RainParam.x * (1 - metallic_roughness.r);
    o.Metallic = lerp(metallic_roughness.r, max(0.2, metallic_roughness.r), coefficient);
    o.Smoothness = lerp(o.Smoothness, max(0.5, o.Smoothness), coefficient);
    float2 uvRipper = IN.uv_MainTex * float2(25, 25)/* + _Time.x*_FlowRate * float2(-1, 0)*/;
    half3 rainNormal = UnpackNormal(tex2D(_CharRainNormal, uvRipper));
    baseNormal = lerp(baseNormal, BlendNormals(baseNormal, rainNormal), coefficient);
#endif

    o.Normal = baseNormal;

```

扩展

- 丝绸
- 棉布
- 绒布
- 毛料
- 皮质
- 丝袜
- 纱



目前业界对于通用布料类渲染光照模型的探索

- 边缘光, Rim Lighting 经验模型, 非物理, 效果一般。
- Ashikhmin分布相关: 《Order 1886》 2013-2014
Crafting a Next-Gen Material Pipeline for The Order: 1886
《Uncharted 4》 2016
The Process of Creating Volumetric-based Materials in Uncharted 4
- Charlie分布相关: 《Sony Pictures Imageworks》 2017年论文
Production Friendly Microfacet Sheen BRDF



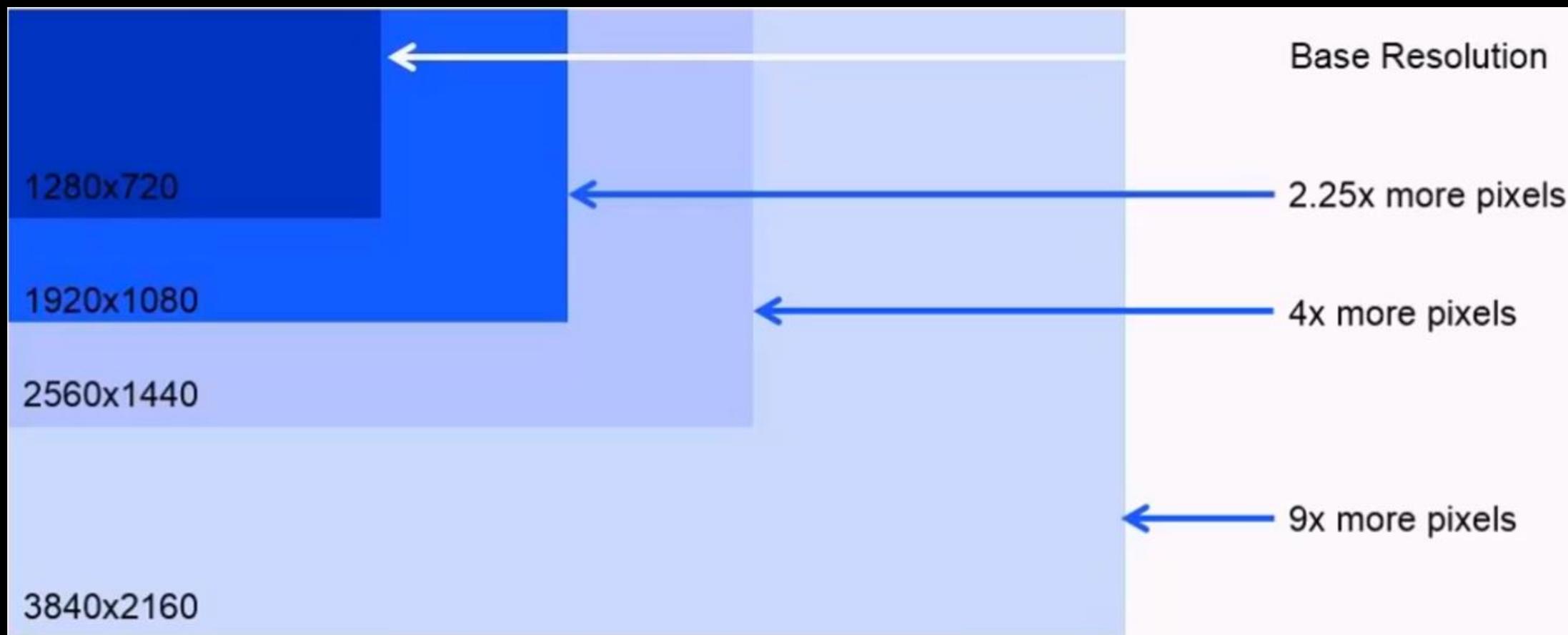
Game Optimization

Game Optimization

- FPS
- Less DrawCall
 - Batch&Instancing
- Less Fragment
 - Resolution
 - Overdraw
- Less Texture Fetch
- Less ALU
- Shader Optimization

Game Optimization

- **Resolution:** 使用低于本地设备分辨率的大小大大能减少Fragment处理次数



Game Optimization

- Shader优化

Prefer medium over highp

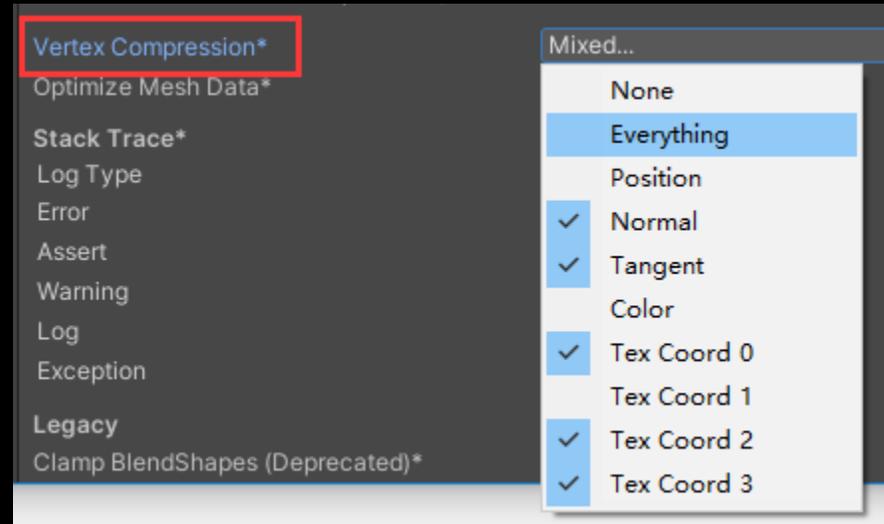
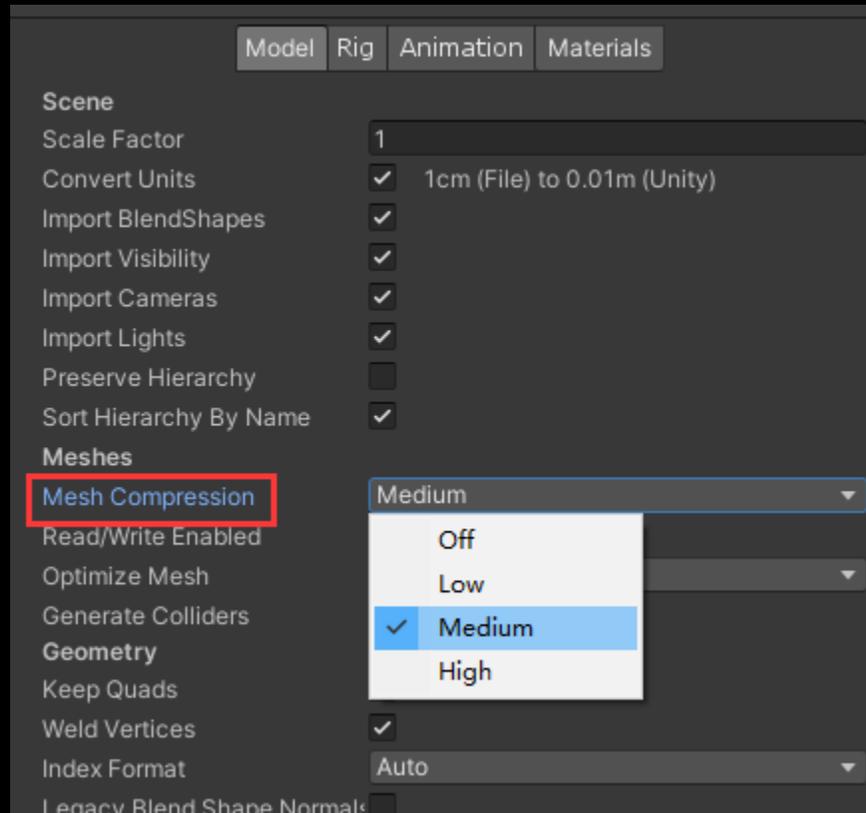
一些高端设备，性能影响小，但是对于一些中低端设备有较大影响

- 避免不必要的精度转换
- 最小化使用一些复杂的指令: sqrt, sin, cos, and recip
- 越大的场景纹理越建议开启Mipmap
- 使用Bilinear代替Trilinear过滤
- 减少Anisotropic 过滤采样数量
- 选择合适的纹理压缩格式 eg:ASTC Texture compression
- SIMD(单指令多数据)
- 选择合适的纹理压缩格式

Game Optimization

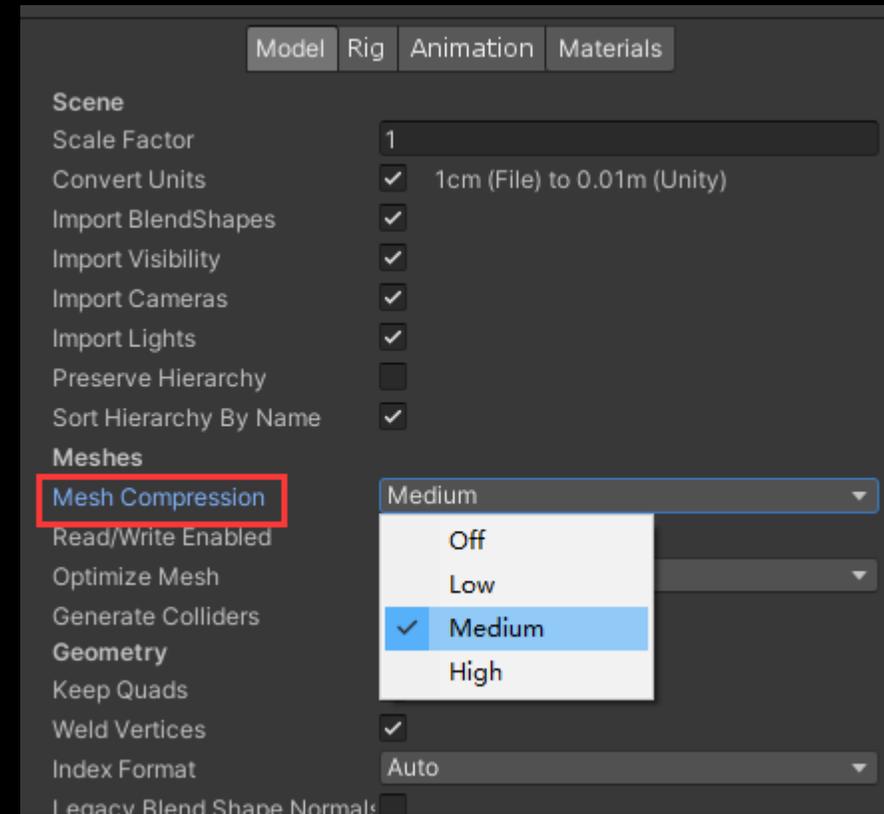
- 适当时候考虑压缩Vertex data: 如果可能请使用16位顶点格式
- 压缩或使用更少的Output attributes channels

Mesh压缩



Mesh Compression

- 使用压缩算法，将mesh数据进行压缩，减少占用硬盘空间
- 运行时会被解压为原始精度数据
- 有损压缩



Optimize Mesh Data

- 用来剔除不需要的Channels
(StripUnusedMeshComponents)
- 注意打开时如果模型和材质拆开打包
可能导致模型数据丢失

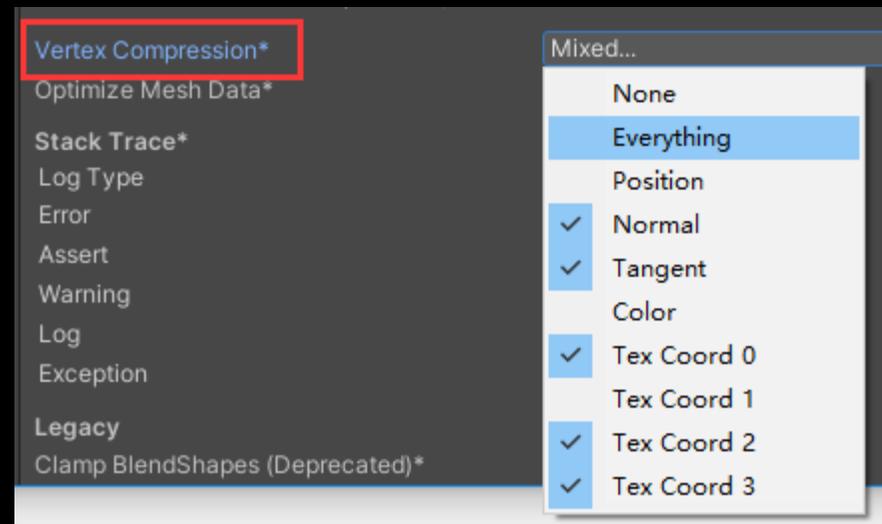


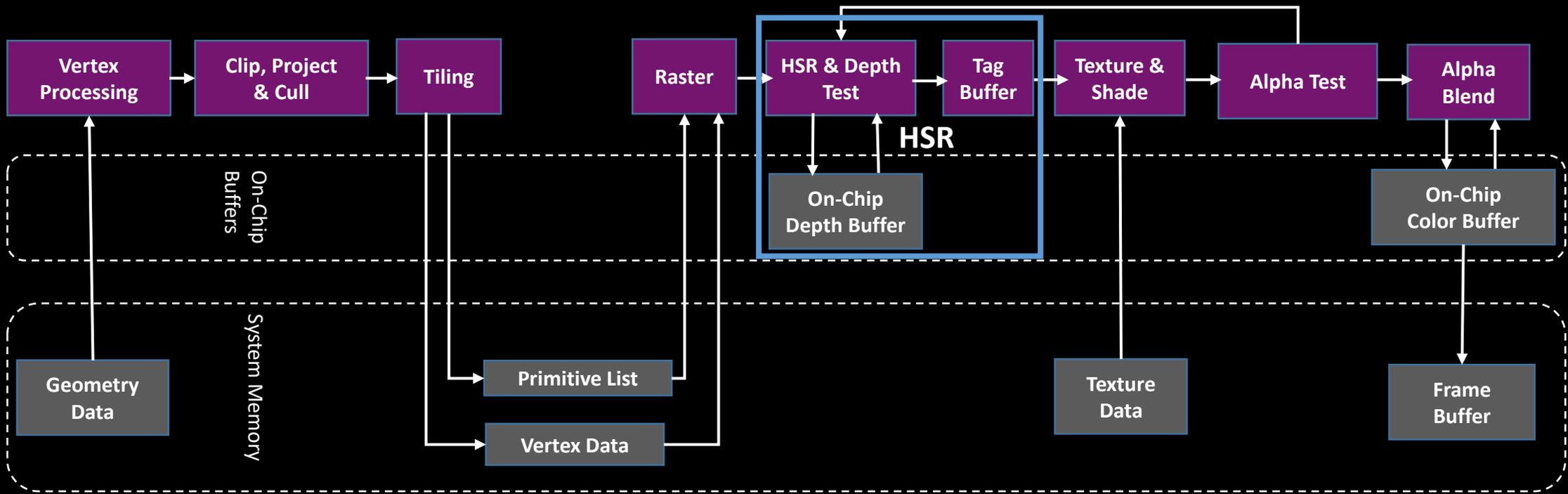
Vertex Compression

将选择的顶点channel的数据格式format设置为16bit(float->half), 减少运行时内存。

Conditions:

- 是否可以进行Dynamic Batching
- Mesh不能开读写
- 不能是SkinMesh
- 不能使用Mesh Compression(需要为Off)





https://docs.imgtec.com/Architecture_Guides/PowerVR_Architecture/topics/powervr_architecture_tile_based_deferred_rendering_tldr.html

TBDR管线新增了一个阶段，叫做HSR(隐藏面消除)

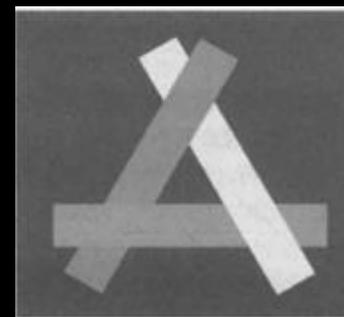
- TBR依然没有解决Overdraw的问题，TBR的设计主要是减少IMR的带宽开销；
- 通常用来解决Overdraw的是EarlyZ技术：对不透明的图元逐像素进行处理时，先进行Depth Read & Test，通过后直接写入深度，后续再执行该像素上的着色，否则就等待下一个要处理的像素，减少不必要的绘制；
- TBR在渲染不透明物体时要按从近到远的顺序去绘制；
- EarlyZ是无法完全避免Overdraw的，因为我们在真正对一个复杂场景去渲染的时候是不可能进行严格的由近到远的绘制的。



TBDR管线新增了一个阶段，叫做HSR(隐藏面消除)

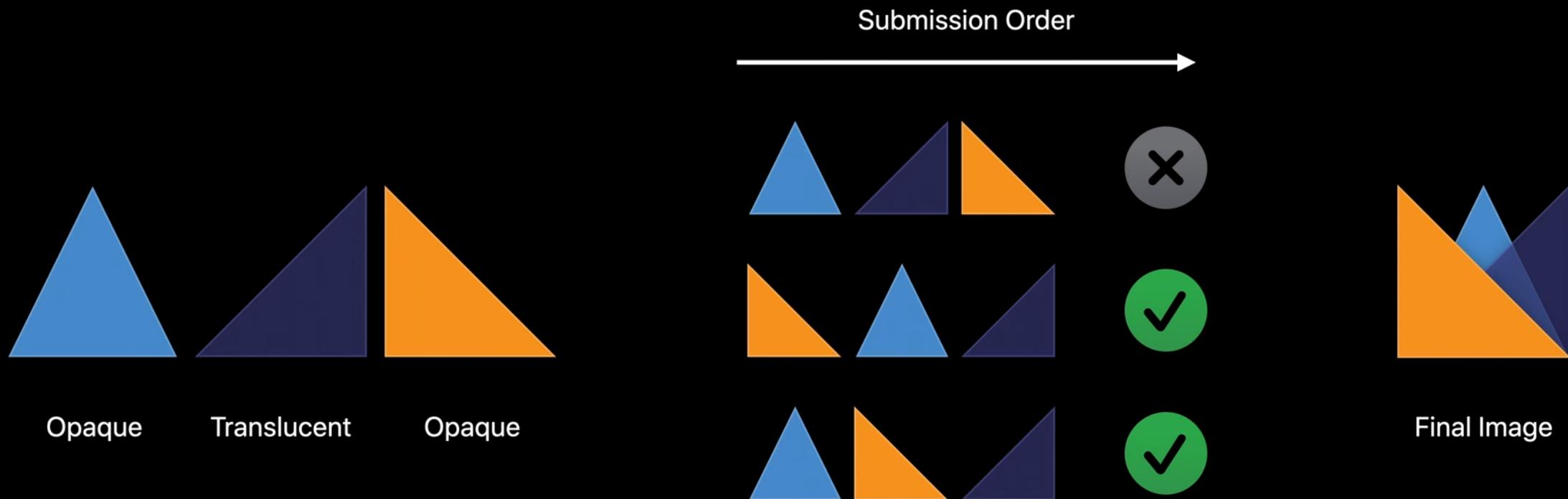
HSR (PowerVR提出): 不需要在软件层面对物体进行排序，直接提供硬件级别的支持来解决OverDraw。

当一个像素通过了EarlyZ准备执行PS进行绘制前，先不画，只记录标记这个像素归哪个图元来画。等到这个Tile上所有的图元都处理完了，最后再真正的开始绘制每个图元中被标记上能绘制的像素点，可以实现完全消除不可见三角形的片元渲染。



HSR最佳实践

避免不透明物体和透明物体交叉渲染



Game Optimization

Rendering order

For opaque objects

- Sort by distance

Adreno 3.x, 4.x, sort opaque object from near to far is preferred

- Sort by state

Adreno 5.x, 6.x, sort by state is preferred

Adreno 5.x, 6.x, have Low Resolution Z(LRZ) which could help Z rejection

Rendering order suggestion on Adreno 5.x, 6.x is

- Render Opaque queue by state
- Render AlphaTest queue
- Render transparent queue

<https://developer.qualcomm.com/docs/adreno-gpu/developer-guide/gpu/overview.html>

性能优化工具

Xcode

- Frame Capture
- Shader Debugger
- GPU Memory Viewer



Instruments

- Metal System Trace
- Game Performance Template



MetalTest01 Captured G...

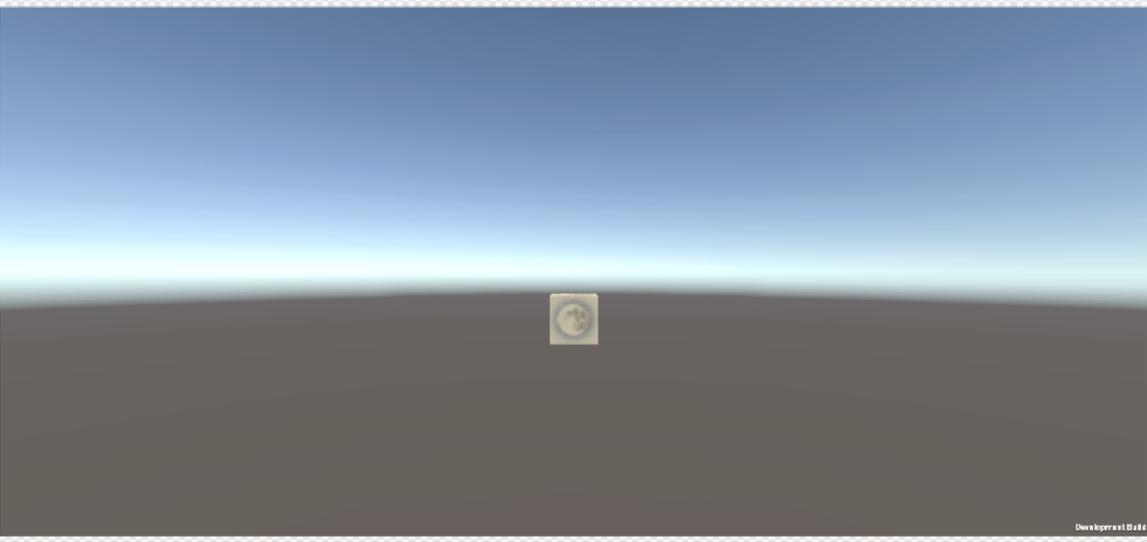
Summary

Counters

Memory **66.2 MB**

- 0 0x2827b0540 = [MTLLayer...
- 1 CAMetalLayer Display Drawa...
- Command Buffer 0 0x108b6...
 - 2 0x108b6cd10 = [0x10384...
 - Camera.Render 1.34 ms
 - 3 [pushDebugGroup:"Ca...
 - Drawing 1.01 ms
 - Camera.Image... 330.48 μs
 - 84 [pushDebugGroup:"...
 - [R] Camera.... 330.48 μs
 - 85 [R] Camera.Image...
 - 86 [setLabel:"[R] Ca...
 - 87 [setFragmentText...
 - 88 [setFragmentSam...
 - 89 [setViewport:{0, 0...
 - 90 [setVertexBuffer:S...
 - 91 [setFragmentBuffe...
 - 92 [setRenderPipelin...
 - 93 [setDepthStencilS...
 - 94 [setStencilReferen...
 - 95 [setCullMode:None]
 - 96 [setFrontFacingWi...
 - 97 [setDepthBias:-0...
 - 98 [setDepthClipMod...
 - 99 [setTriangleFillMo...
 - 100 [setVertexBuffer:...
 - 101 [drawIn... 306.70 μs
 - 102 [setViewport:{0,...

Summary Export



Resolution 1792x828
Pixel Format BGRA8Unorm_sRGB

Overview Show Dependencies

Command Buffers	1
Render Encoders	3
Blit Encoders	0
Compute Encoders	0
Draw Calls	5
Dispatch Calls	0

Performance Show Counters

GPU Time	1.34 ms
Vertices	5,124

Memory Show Memory

Textures	52.5 MB
Buffers	24.6 MB
Other	Zero KB

Insights

Category	Value	Insights
Memory	7.6 MB	7.6 MB
Bandwidth	7.4 MB	7.4 MB
Performance	No Insights	No Insights
API	5 Insights	5 Insights

Savings Insight

TempBuffer 1 1792x828	7.6 MB	Storage Mode
-----------------------	--------	--------------

TempBuffer 1 1792x828

Type	2D
Pixel Format	Depth32Float_Stencil8
Mipmap Level	0
Width	1792
Height	828

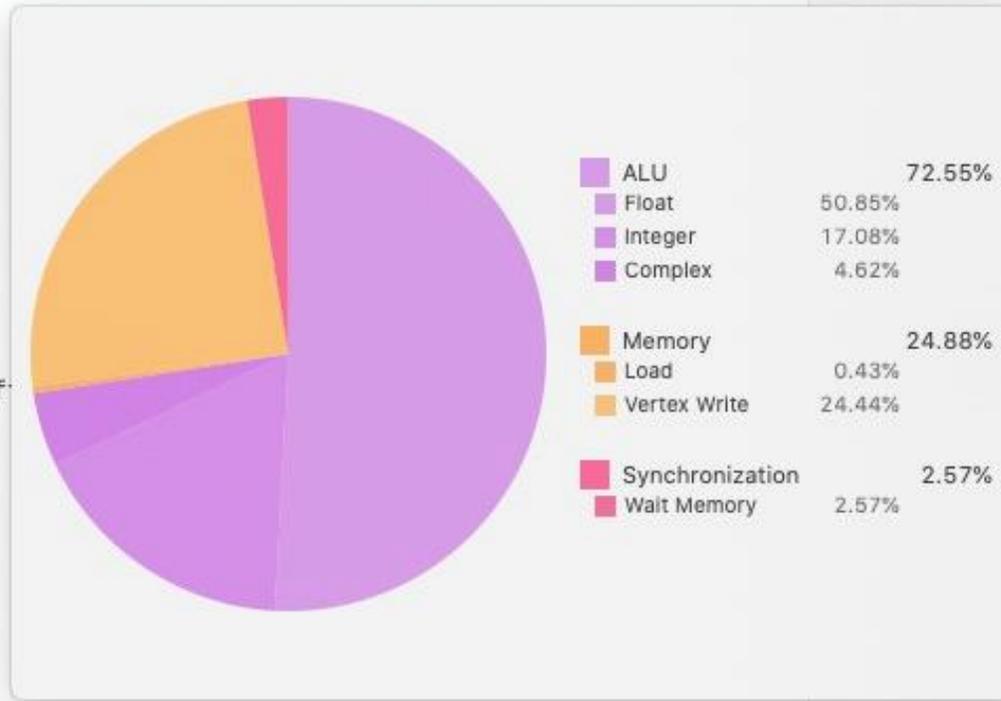
Resource "Texture TempBuffer 1 1792x828 0x1038737d0" has the wrong storage mode set.

"Texture:0x1038737d0 "TempBuffer 1 1792x828"" has storage mode 'Shared' but was a render target populated exclusively by the GPU.

```

16 };
17
18 struct Mtl_VertexIn
19 {
20     float4 POSITION0 [[ attribute(0) ]] ;
21     float3 NORMAL0 [[ attribute(1) ]] ;
22 };
23
24 struct Mtl_VertexOut
25 {
26     float4 mtl_Position [[ position ]];
27 };
28
29 vertex Mtl_VertexOut xlatMtlMain(
30     constant VGlobals_Type& VGlobals [[ buf
31     Mtl_VertexIn input [[ stage_in ]]
32 {
33     Mtl_VertexOut output;
34     float4 u_xlat0;
35     float4 u_xlat1;
36     float4 u_xlat2;
37     float u_xlat6;
38     float u_xlat9;
39     bool u_xlatb9;
40     u_xlat0.x = dot(input.NORMAL0.xyz, VGlobals.hslscc_mtx4x4unity_WorldToObject[0].xyz);
41     u_xlat0.y = dot(input.NORMAL0.xyz, VGlobals.hslscc_mtx4x4unity_WorldToObject[1].xyz);
42     u_xlat0.z = dot(input.NORMAL0.xyz, VGlobals.hslscc_mtx4x4unity_WorldToObject[2].xyz);
43     u_xlat9 = dot(u_xlat0.xyz, u_xlat0.xyz);
44     u_xlat9 = rsqrt(u_xlat9);
45     u_xlat0.xyz = float3(u_xlat9) * u_xlat0.xyz;
46     u_xlat1 = input.POSITION0.yyyy * VGlobals.hslscc_mtx4x4unity_ObjectToWorld[1];
47     u_xlat1 = fma(VGlobals.hslscc_mtx4x4unity_ObjectToWorld[0], input.POSITION0.xxxx,
48                 u_xlat1);
49     u_xlat1 = fma(VGlobals.hslscc_mtx4x4unity_ObjectToWorld[2], input.POSITION0.zzzz,
50                 u_xlat1);

```



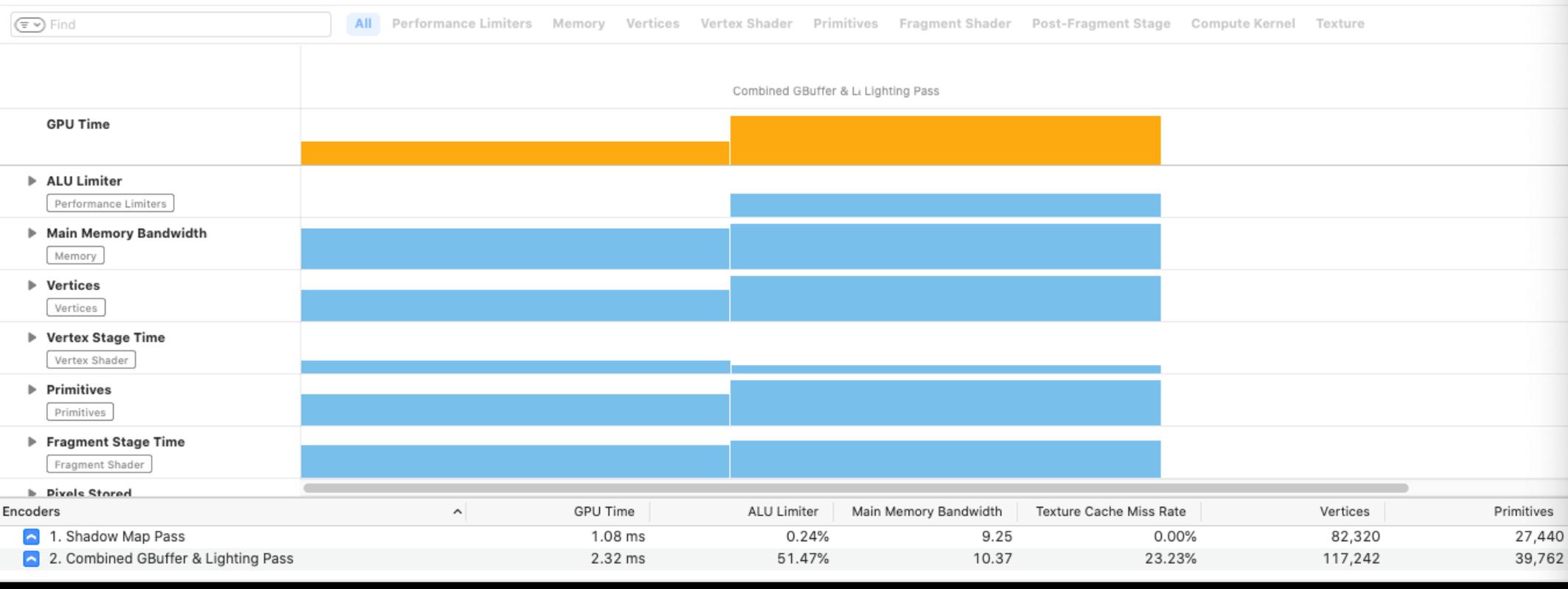
2.60% ●
 0.03% ●
 0.03% ●
 0.12% ●
 0.14% ●
 0.14% ●
 0.14% ●

Counters

Memory 101.4 MB

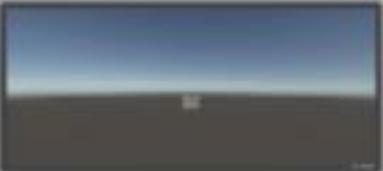
Shadow commands

- 0 Shadow commands = [0x1...
- 1 [addCompletedHandler:0x...
- 2 [setLabel:"Shadow comma...
- Shadow Map Pass** 1.07 ms
- 3 Shadow Map Pass = [re...
- 4 [setLabel:"Shadow Map...
- 5 [setRenderPipelineState...
- 6 [setDepthStencilState:S...
- 7 [setCullMode:Back]
- 8 [setDepthBias:0.015 slo...
- 9 [setVertexBuffer:Unifor...
- 10 [setVertexBuffer:Unifor...
- 11 [setVertexBuffer:0x103...
- 12 [setVertexBuffer:0x103...
- 13 [setFragmentTexture:T...
- 14 [setFragmentTexture:T...



Label	Issues	Type	Allocated Size	Storage Mode	Purgeable State	CPU Access	Time Since Last Bound
 _TargetPool0		Texture 2D	5.7 MB	Shared	NonVolatile	None	Now
 AreaTex		Texture 2D	448 KB	Shared	NonVolatile	Write	4.87 s
 Buffer:0x103253480		Buffer	144 KB	Shared	NonVolatile	ReadWrite	Now
 Buffer:0x10328c560		Buffer	128 KB	Shared	NonVolatile	ReadWrite	2.90 s
 Buffer:0x10328c860		Buffer	128 KB	Shared	NonVolatile	ReadWrite	Never
 CAMetalLayer Display Draw...		Texture 2D	5.7 MB	Shared	NonVolatile	None	Now
 Color Grading Log Lut		Texture 3D	1.0 MB	Shared	NonVolatile	None	Now
 Cube 0x10316c650		Buffer	128 KB	Shared	NonVolatile	ReadWrite	Now
 Cube 0x103263140		Buffer	128 KB	Shared	NonVolatile	ReadWrite	Now
 DrawableProxy		Texture 2D	5.7 MB	Shared	Empty	None	Never
 DrawableProxy		Texture 2D	5.7 MB	Shared	Empty	None	Never
 Fullscreen Triangle		Buffer	128 KB	Shared	NonVolatile	ReadWrite	Now
 Fullscreen Triangle		Buffer	128 KB	Shared	NonVolatile	ReadWrite	Now
 Internal Curves Texture		Texture 2D	128 KB	Shared	NonVolatile	Write	13.33 ms
 Internal Curves Texture		Texture 2D	128 KB	Shared	NonVolatile	None	Now
 LDR_LLL1_0 0x10314ea50		Texture 2D	128 KB	Shared	NonVolatile	Write	766.91 ms
 LDR_LLL1_0 0x10314f5d0		Texture 2D	80 KB	Shared	NonVolatile	Write	4.87 s
 LDR_LLL1_1 0x10314d350		Texture 2D	128 KB	Shared	NonVolatile	Write	733.45 ms
 LDR_LLL1_1 0x115128ab0		Texture 2D	80 KB	Shared	NonVolatile	Write	4.87 s
 LDR_LLL1_10		Texture 2D	128 KB	Shared	NonVolatile	Write	433.52 ms
 LDR_LLL1_11		Texture 2D	128 KB	Shared	NonVolatile	Write	400.21 ms

Color0: TempB...2x828	Depth: TempB...2x828	Stencil: TempB...2x828
		
Pixel Format RG11...0Float	Pixel Format Depth...encil8	Pixel Format Depth...encil8
Dimensions 1792x828	Dimensions 1792x828	Dimensions 1792x828
Allocated Size 5.70 MiB	Allocated Size 7.25 MiB	Allocated Size 7.25 MiB
Load Action Clear	Load Action Clear	Load Action Clear
Store Action Store	Store Action Store	Store Action Store

Color0: CAMet...awable	Depth: 0x100e53ae0	Stencil: 0x100e53ae0
		
Pixel Format BGRA...sRGB	Pixel Format Depth...encil8	Pixel Format Depth...encil8
Dimensions 1792x828	Dimensions 1792x828	Dimensions 1792x828
Allocated Size 5.67 MiB	Allocated Size 7.25 MiB	Allocated Size 7.25 MiB
Load Action Clear	Load Action Clear	Load Action Clear
Store Action Store	Store Action DontCare	Store Action DontCare

```
public void SetRenderTarget(Rendering.RenderTargetIdentifier rt, Rendering.RenderBufferLoadAction colorLoadAction, Rendering.RenderBufferStoreAction colorStoreAction, Rendering.RenderBufferLoadAction depthLoadAction, Rendering.RenderBufferStoreAction depthStoreAction);
```

<https://docs.unity3d.com/ScriptReference/Rendering.CommandBuffer.SetRenderTarget.html>

https://developer.apple.com/documentation/metal/rendering_a_scene_with_deferred_lighting_in_objective-c?preferredLanguage=occ



扬帆起航



谢谢