《诛仙手游》性能优化与质量保证

完美世界 刘彦麟

CONTENTS

CONTENTS

01 Assets Compression

02 Runtime Optimization

03 Unity Performance Reporting

Assets Compression

PART ONE

Assets Compression

Mesh

Animation

Texture

Audio

PART ONE

Assets Compression

Mesh

Animation

Texture

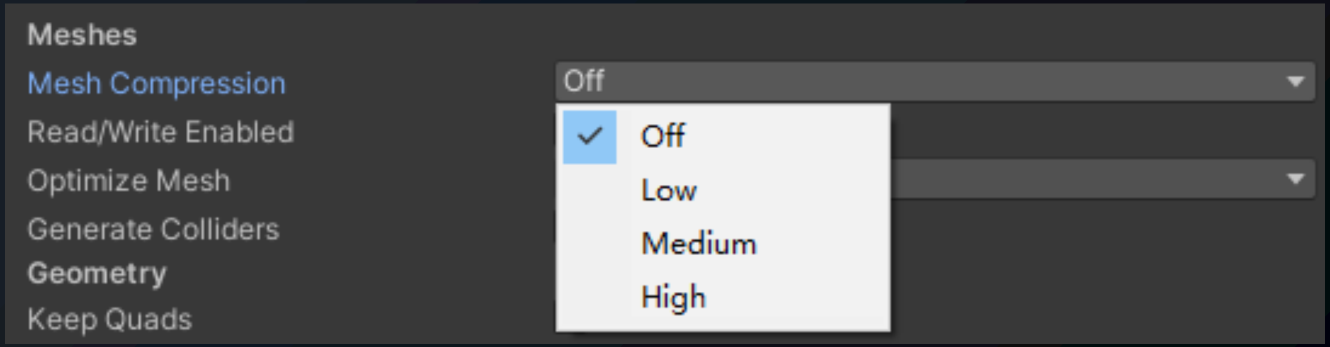
Audio

PART ONE

Mesh Compression

FBX Import - Model

- Read / Write Enabled：upload data to GPU，（No）keep data in CPU
- Mesh Compression
  - file size on disk：ipa、apk、assetbundle
  - uses additional CPU resources
  - higher the compression ratio，the higher the loss



Meshes

Mesh Compression: Off

Read/Write Enabled

Optimize Mesh

Generate Colliders

Geometry

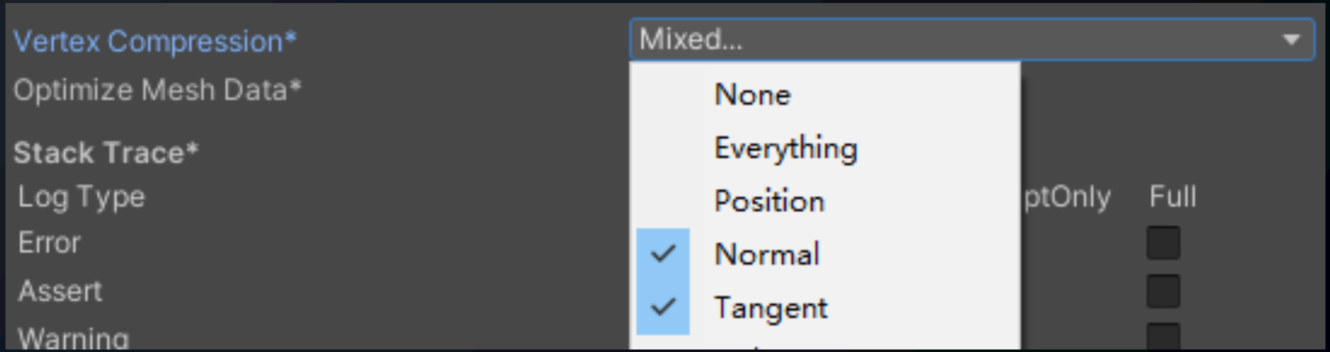
Keep Quads

PART ONE

Mesh Compression

Project Settings - Player

- Vertex Compression : from FP32 to FP16
  - Mesh Compression value set to “off”
  - file size in memory
  - not support skinned mesh
  - global setting
  - lossy



Vertex Compression* Mixed...

Optimize Mesh Data*

Stack Trace*

Log Type ptOnly Full

Error

Assert

Warning

None

Everything

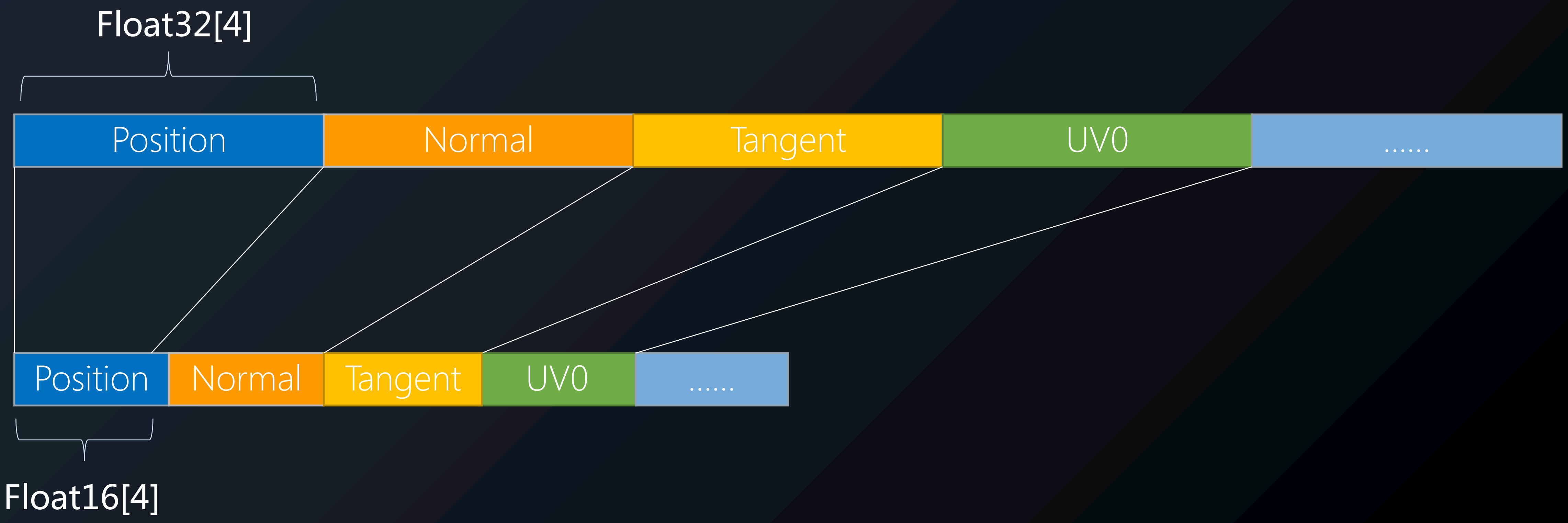
Position

✓ Normal

✓ Tangent

PART ONE

Mesh Compression



Float32[4]

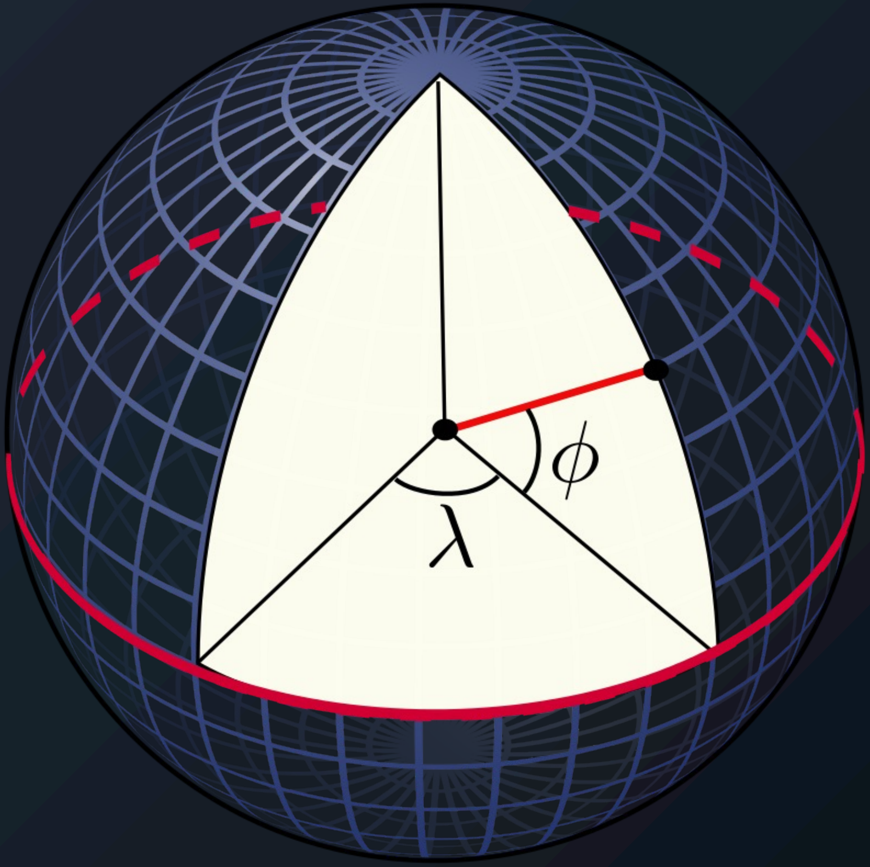
Position Normal Tangent UV0 ......

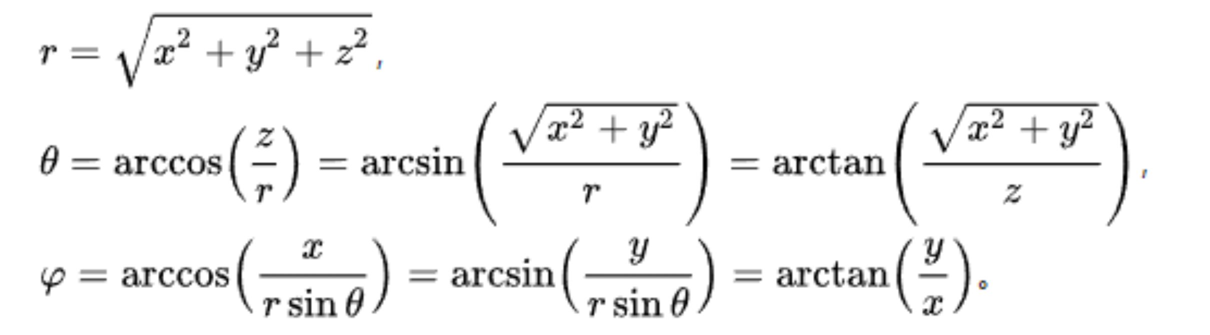
Position Normal Tangent UV0 ......

Float16[4]

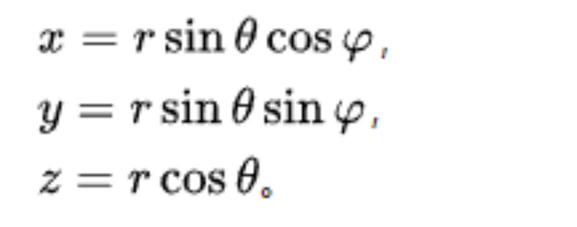
PART ONE

Mesh Compression





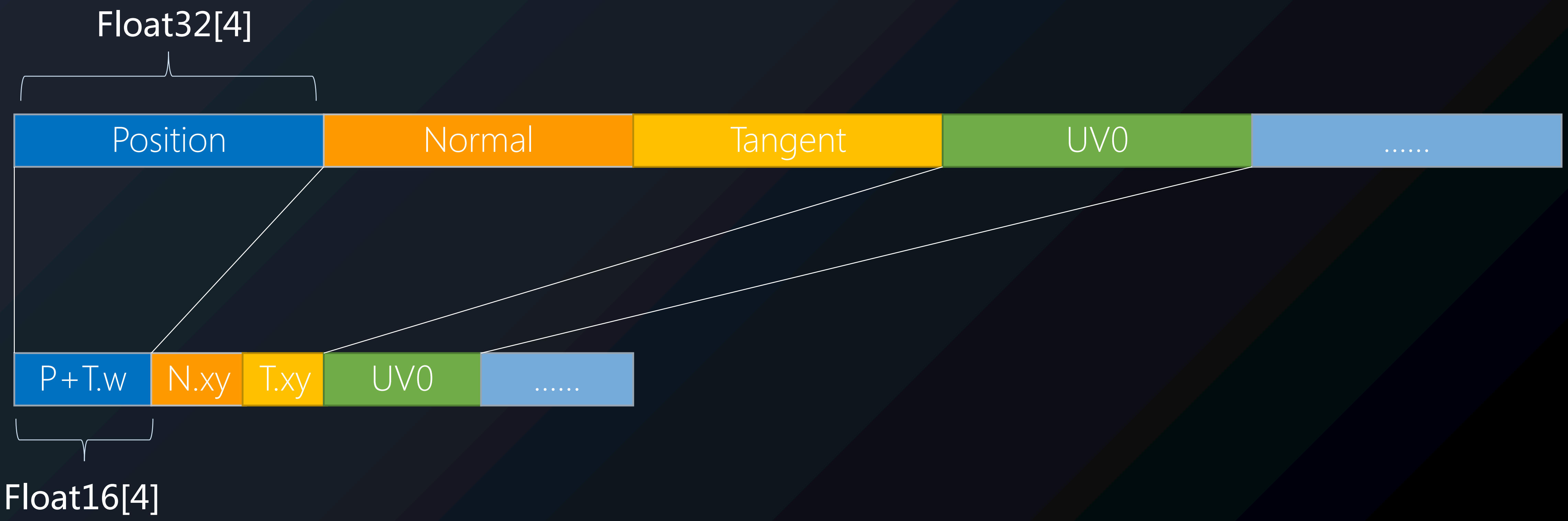
$$r=\sqrt{x^2+y^2+z^2},$$
$$\theta=\arccos\left(\frac{z}{r}\right)=\arcsin\left(\frac{\sqrt{x^2+y^2}}{r}\right)=\arctan\left(\frac{\sqrt{x^2+y^2}}{z}\right),$$
$$\varphi=\arccos\left(\frac{x}{r\sin\theta}\right)=\arcsin\left(\frac{y}{r\sin\theta}\right)=\arctan\left(\frac{y}{x}\right)。$$



$$x=r\sin\theta\cos\varphi,$$
$$y=r\sin\theta\sin\varphi,$$
$$z=r\cos\theta。$$

PART ONE

Mesh Compression



Float32[4]

Position

Normal

Tangent

UV0

......

P+T.w

N.xy

T.xy

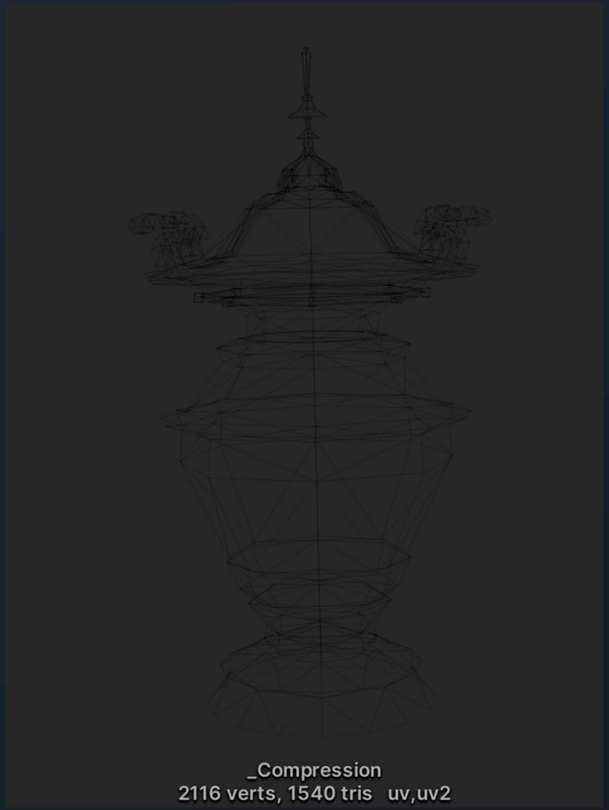
UV0

......

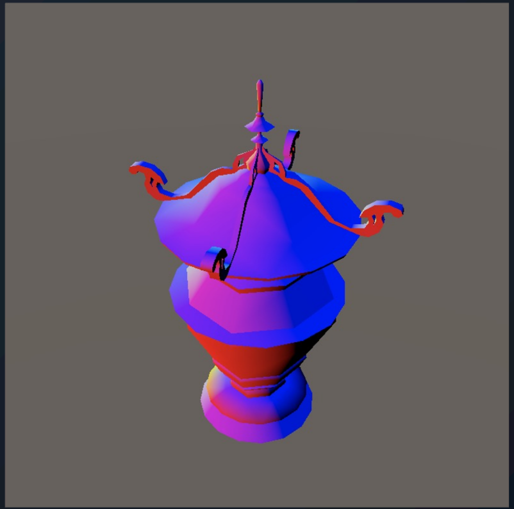
Float16[4]

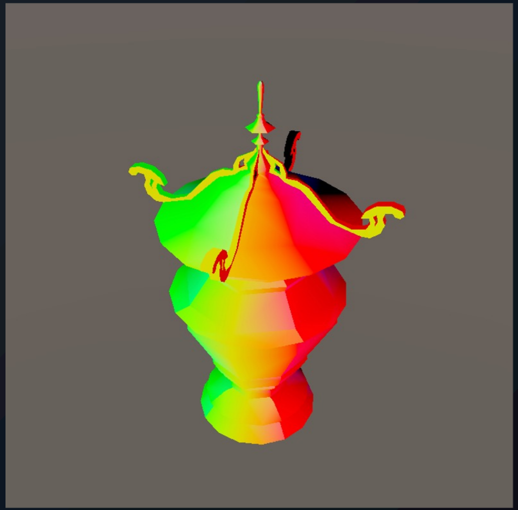
PART ONE

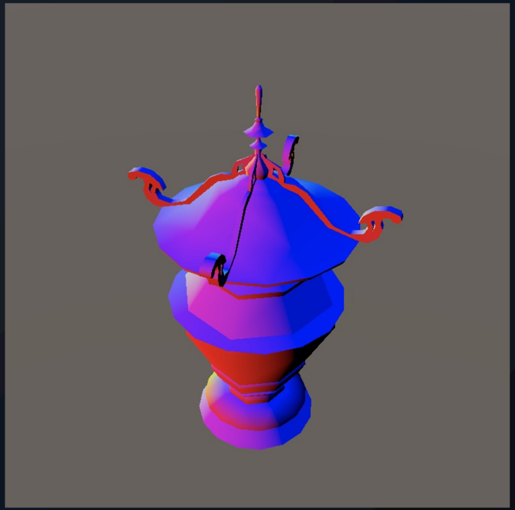
Mesh Compression

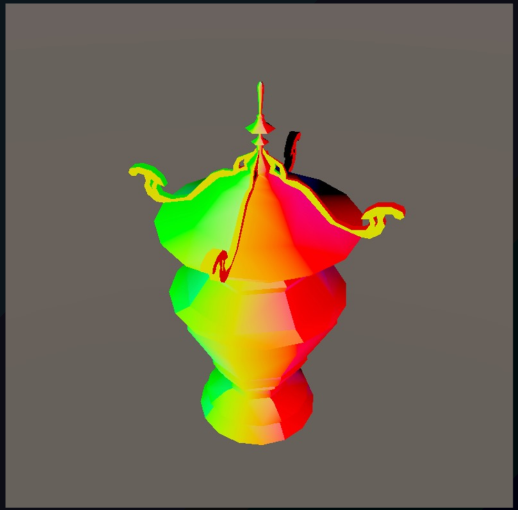


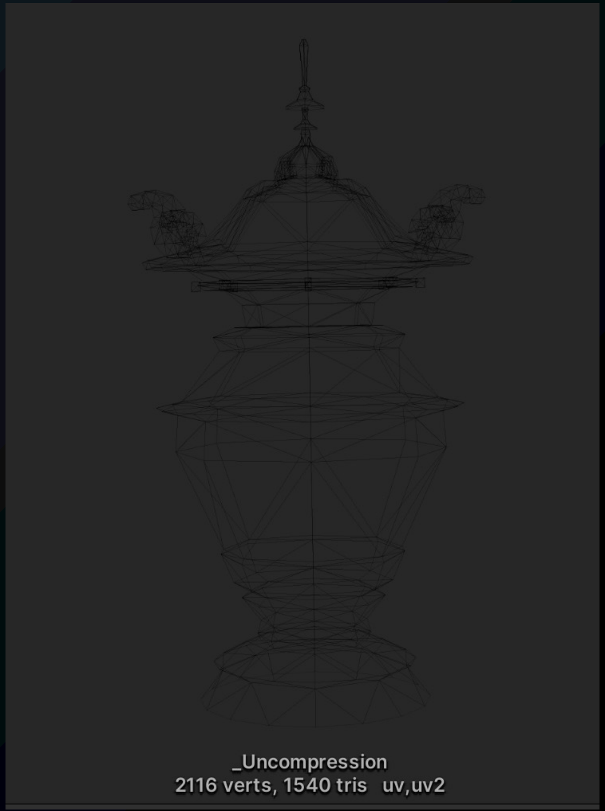
_Compression
2116 verts, 1540 tris uv,uv2







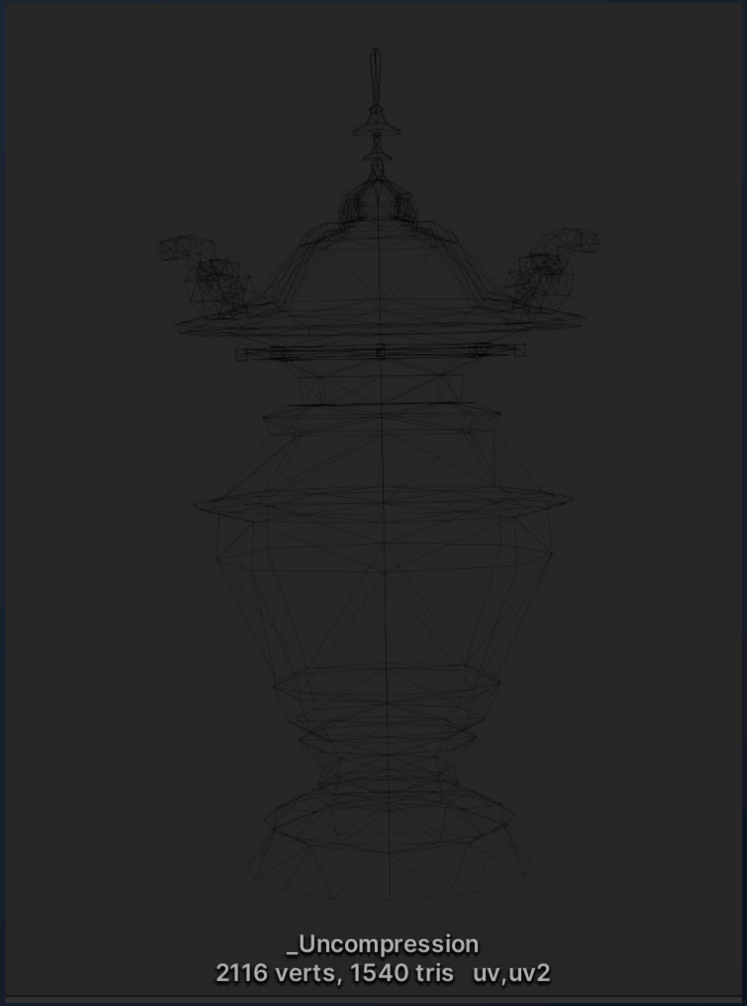




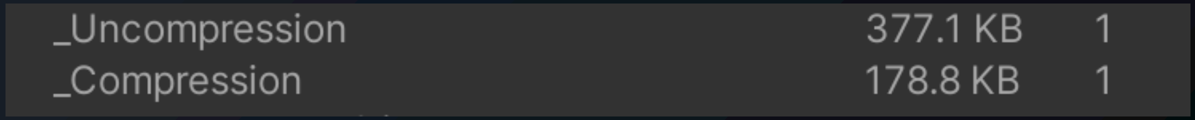
_Uncompression
2116 verts, 1540 tris uv,uv2

PART ONE

Mesh Compression



_Uncompression
2116 verts, 1540 tris uv,uv2



_Uncompression	377.1 KB	1
_Compression	178.8 KB	1

ratio：52.5%

avg ratio：≈50%

PART ONE

Assets Compression

Mesh

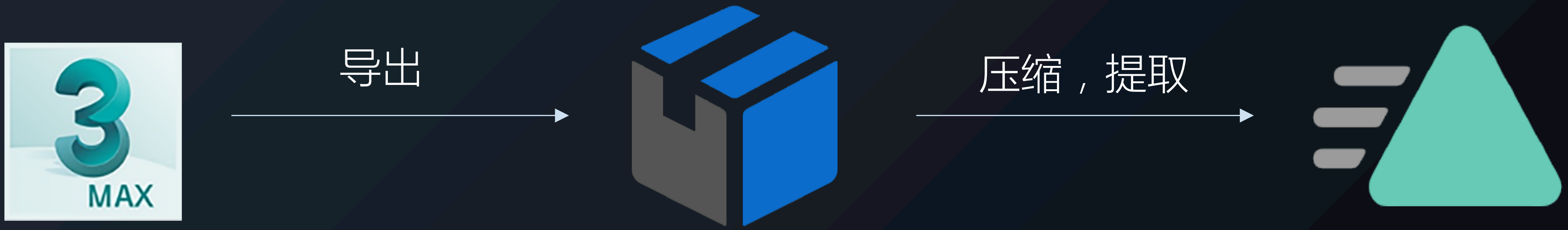
Animation

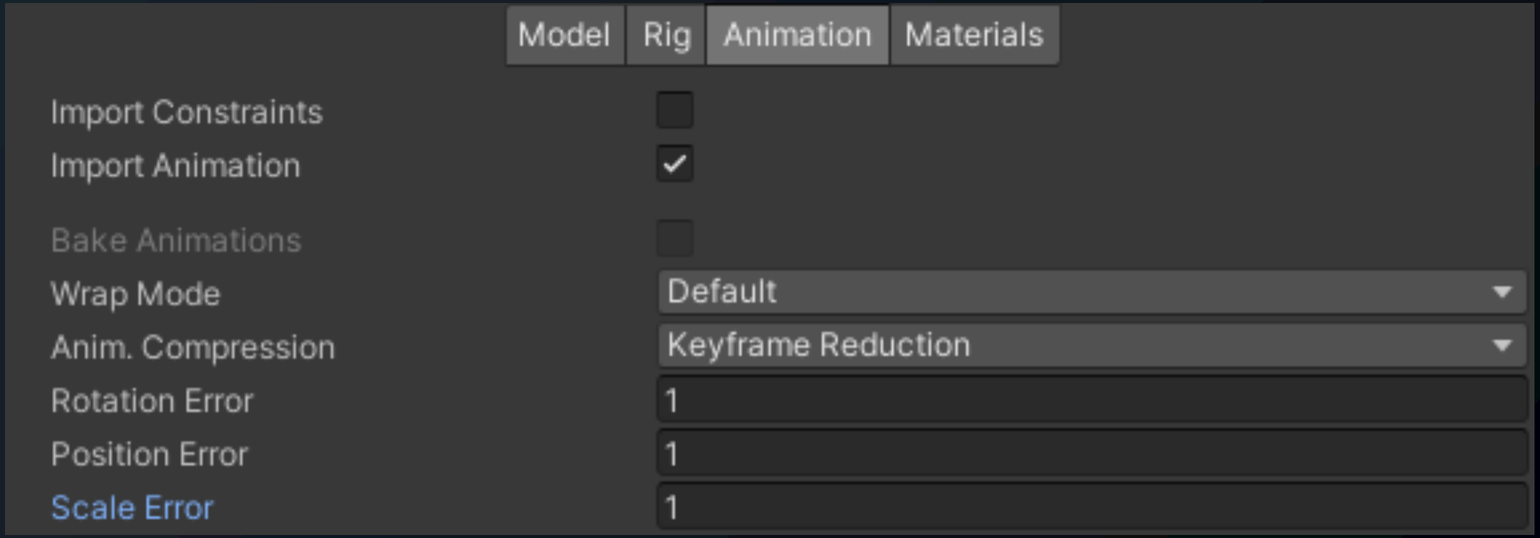
Texture

Audio

PART ONE

Animation Compression





Import Constraints □

Import Animation ☑

Bake Animations □

Wrap Mode: Default

Anim. Compression: Keyframe Reduction

Rotation Error: 1

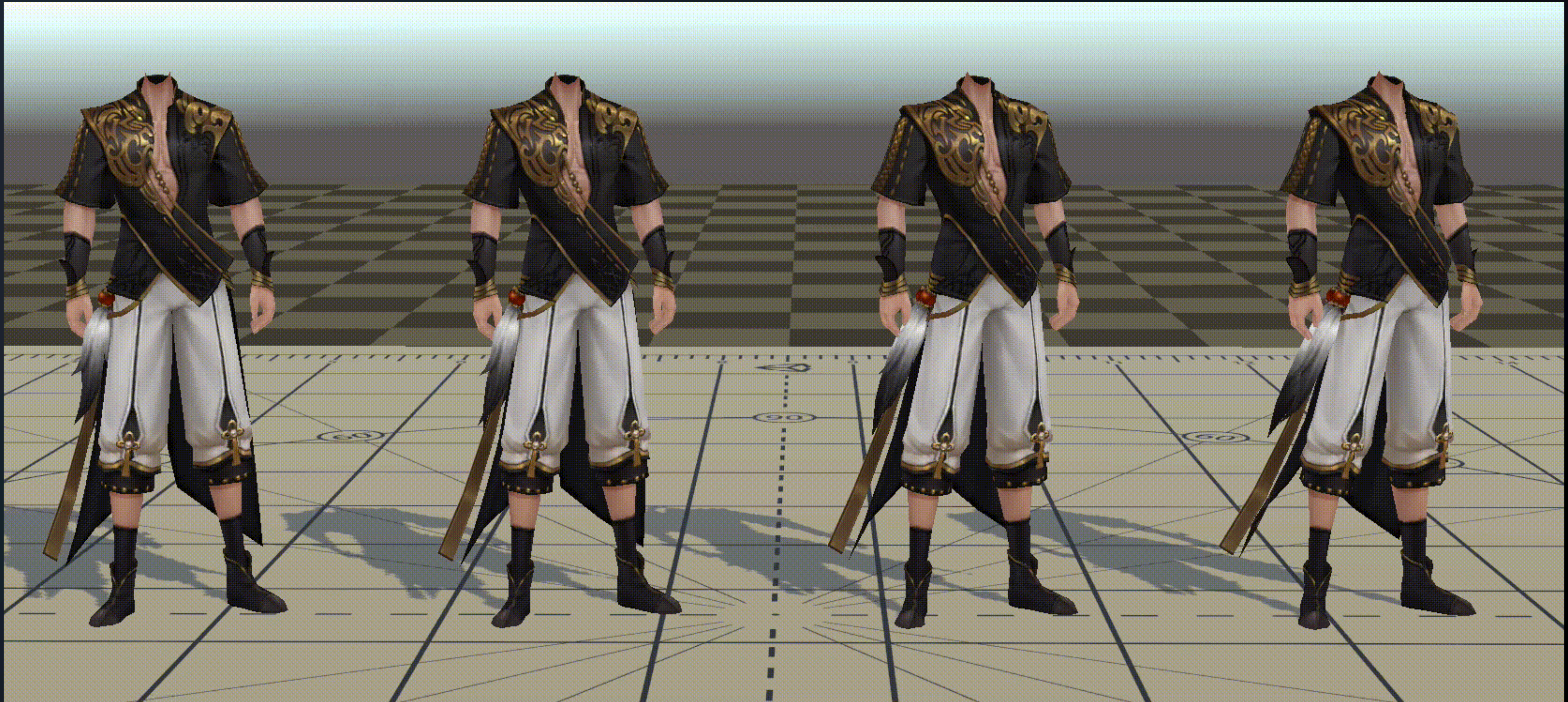
Position Error: 1

Scale Error: 1

PART ONE

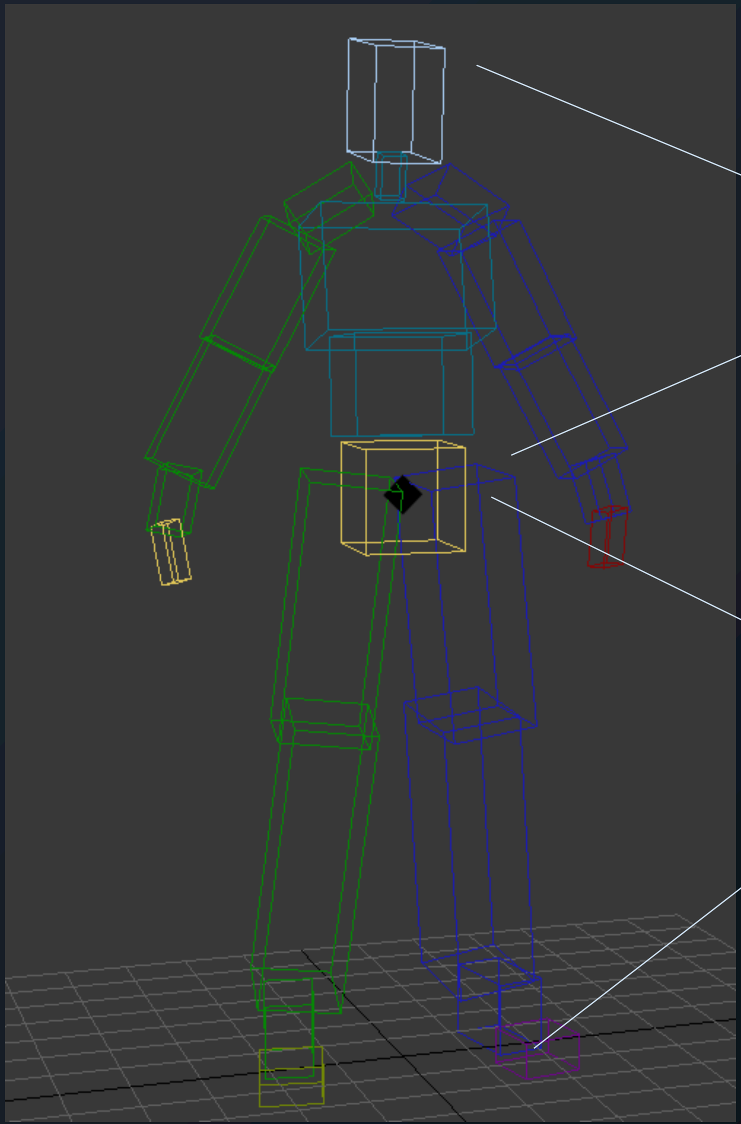
Animation Compression

Error1.0 Error0.5 不压缩 组合压缩



PART ONE

Animation Compression



高误差，低精度

低误差，高精度

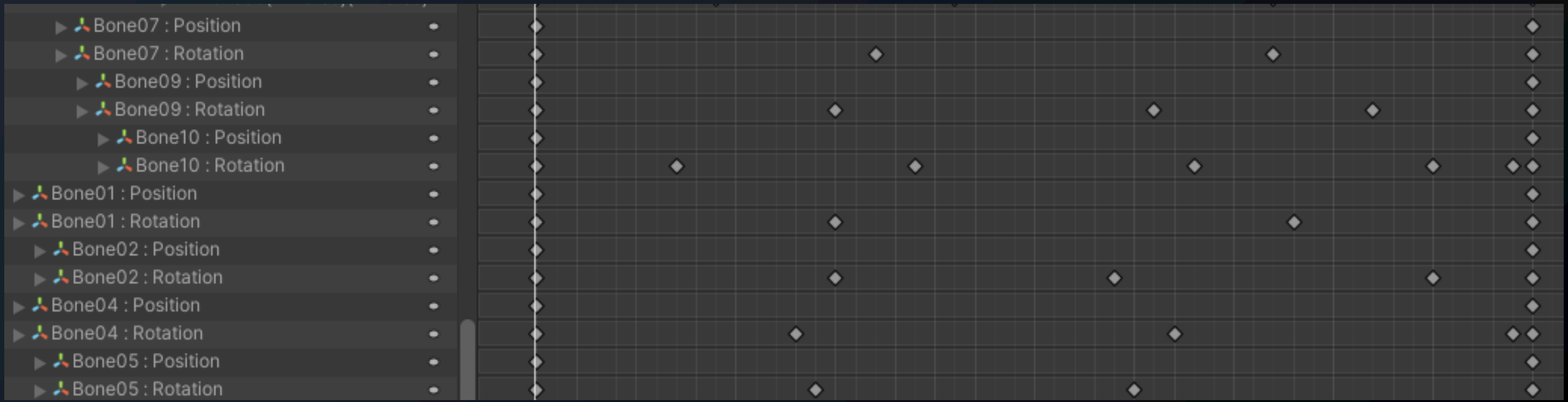
删除Scale曲线

PART ONE

Animation Compression

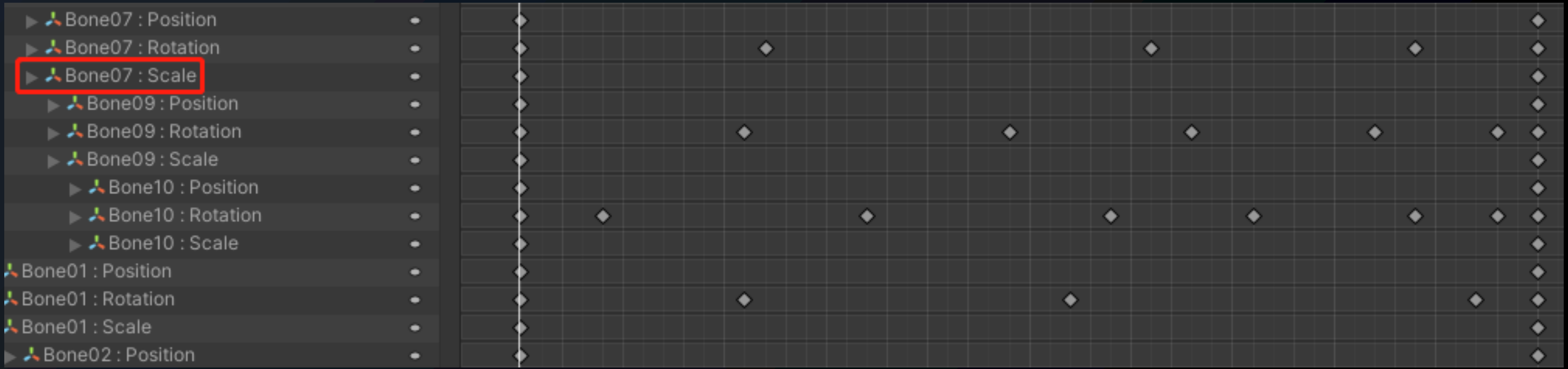
腿部

组合压缩



Bone07 : Position
Bone07 : Rotation
Bone09 : Position
Bone09 : Rotation
Bone10 : Position
Bone10 : Rotation
Bone01 : Position
Bone01 : Rotation
Bone02 : Position
Bone02 : Rotation
Bone04 : Position
Bone04 : Rotation
Bone05 : Position
Bone05 : Rotation

Error 0.5



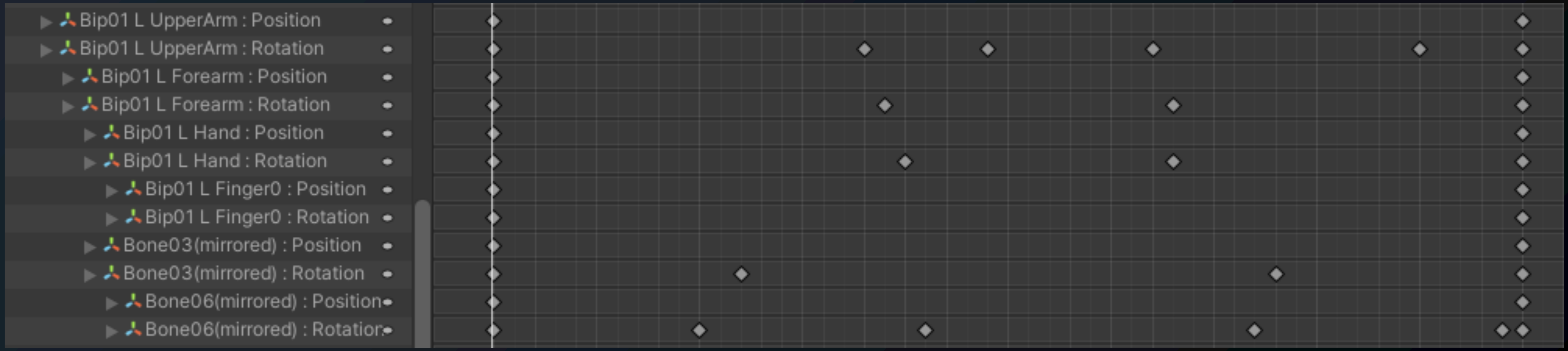
Bone07 : Position
Bone07 : Rotation
Bone07 : Scale
Bone09 : Position
Bone09 : Rotation
Bone09 : Scale
Bone10 : Position
Bone10 : Rotation
Bone10 : Scale
Bone01 : Position
Bone01 : Rotation
Bone01 : Scale
Bone02 : Position

PART ONE

Animation Compression

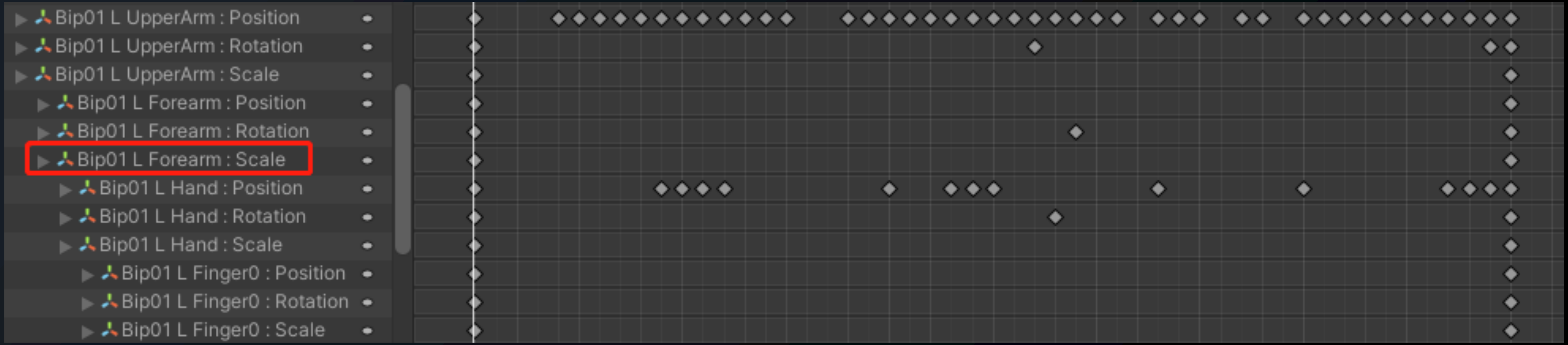
手臂

组合压缩



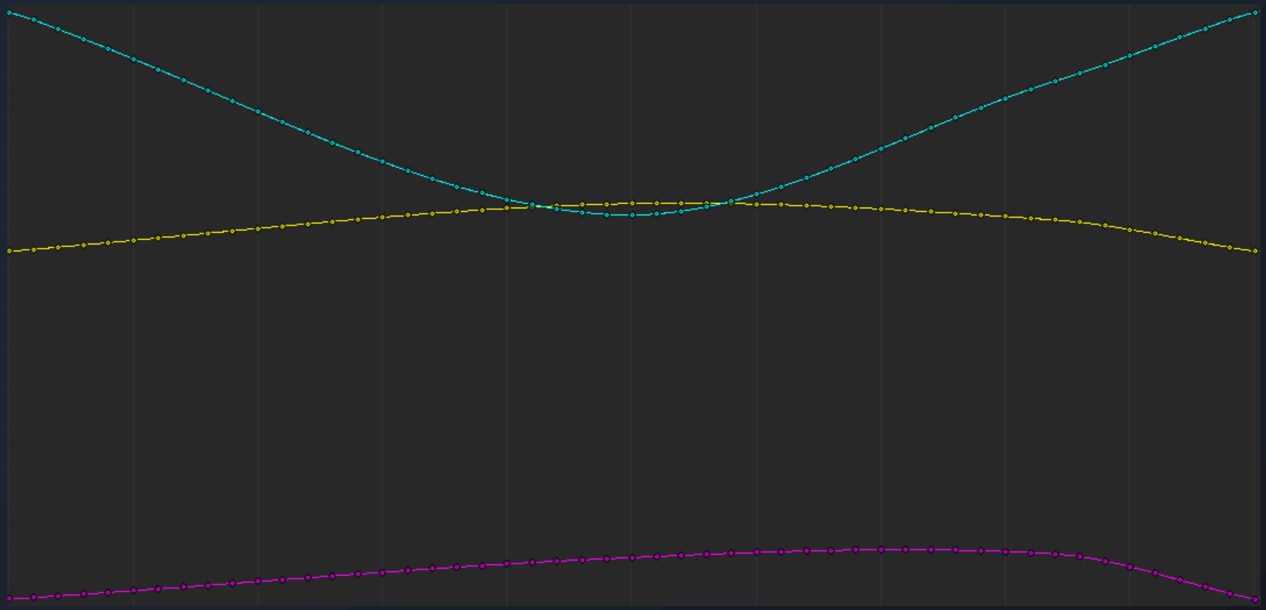
Bip01 L UpperArm : Position
Bip01 L UpperArm : Rotation
Bip01 L Forearm : Position
Bip01 L Forearm : Rotation
Bip01 L Hand : Position
Bip01 L Hand : Rotation
Bip01 L Finger0 : Position
Bip01 L Finger0 : Rotation
Bone03(mirrored) : Position
Bone03(mirrored) : Rotation
Bone06(mirrored) : Position
Bone06(mirrored) : Rotatior

Error 0.5

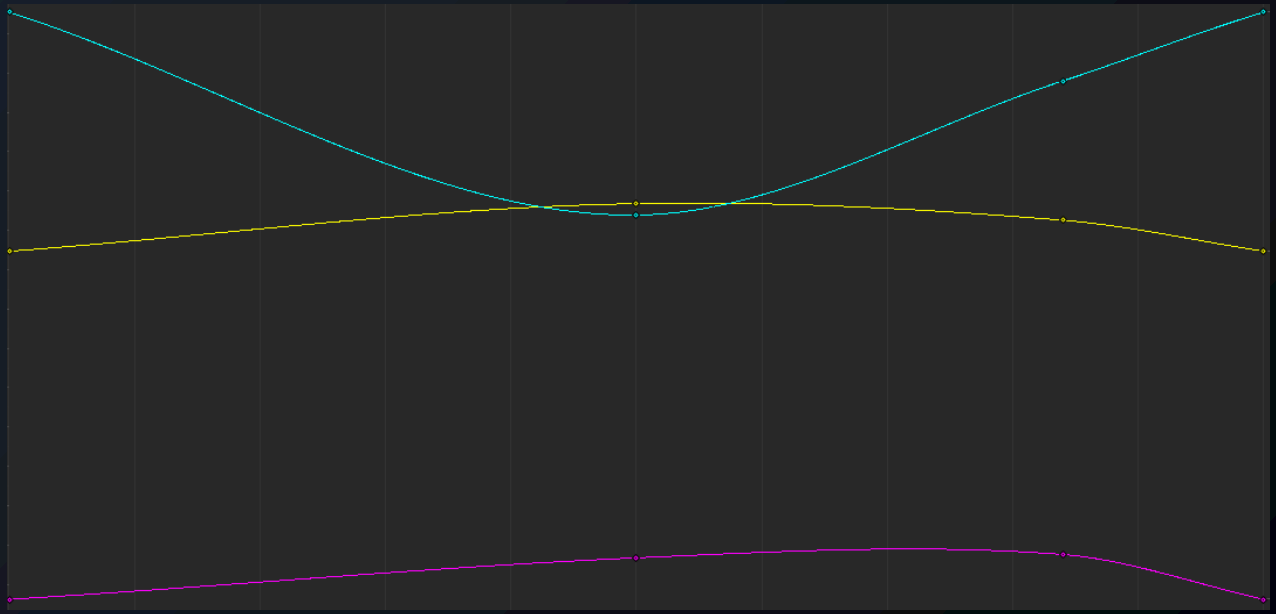


Bip01 L UpperArm : Position
Bip01 L UpperArm : Rotation
Bip01 L UpperArm : Scale
Bip01 L Forearm : Position
Bip01 L Forearm : Rotation
Bip01 L Forearm : Scale
Bip01 L Hand : Position
Bip01 L Hand : Rotation
Bip01 L Hand : Scale
Bip01 L Finger0 : Position
Bip01 L Finger0 : Rotation
Bip01 L Finger0 : Scale

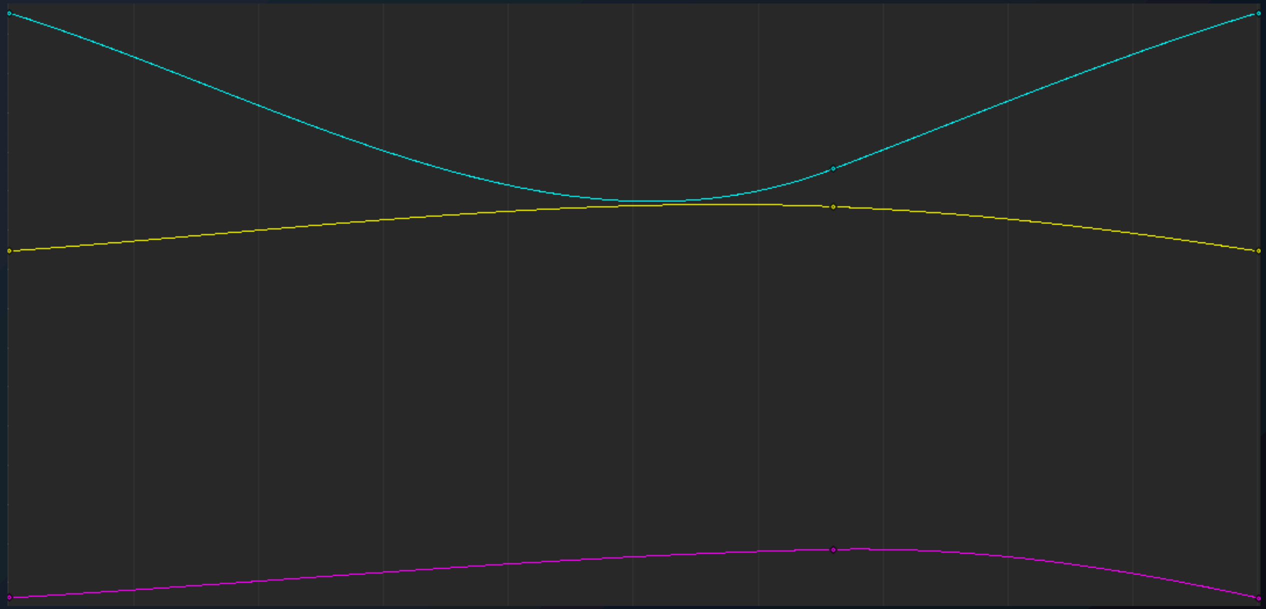
Origin



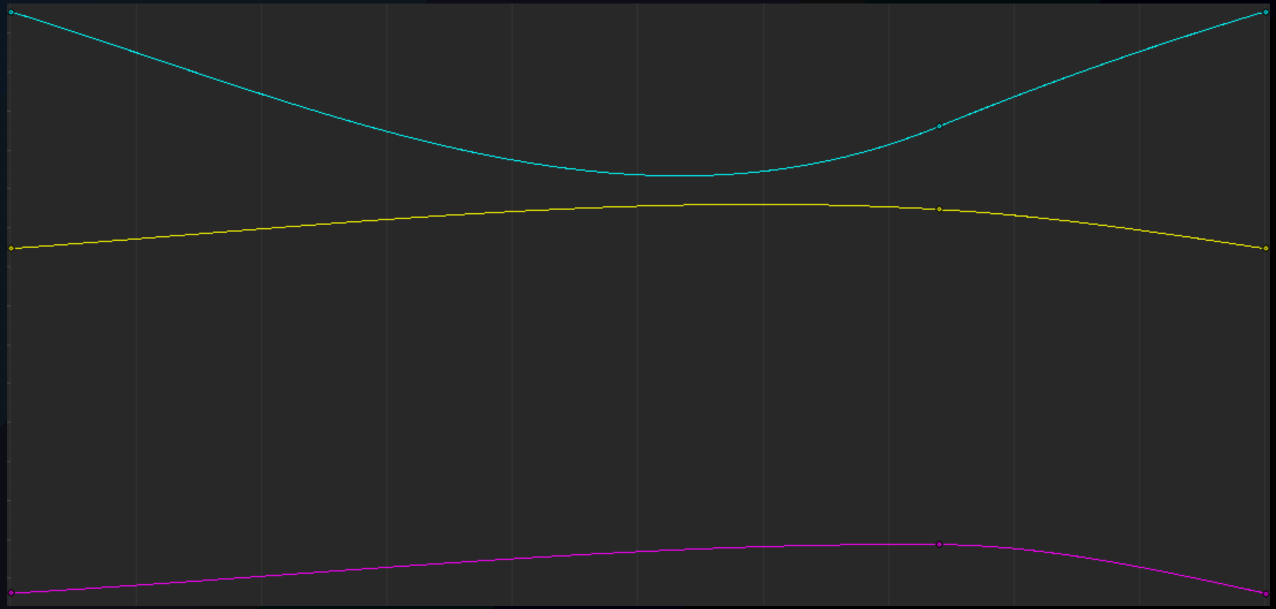
组合压缩



Error 0.5



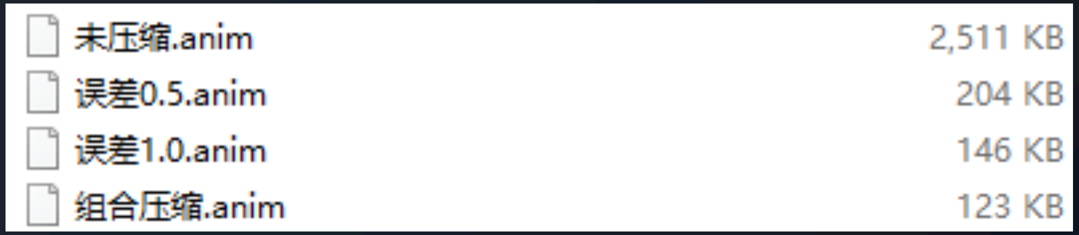
Error 1.0



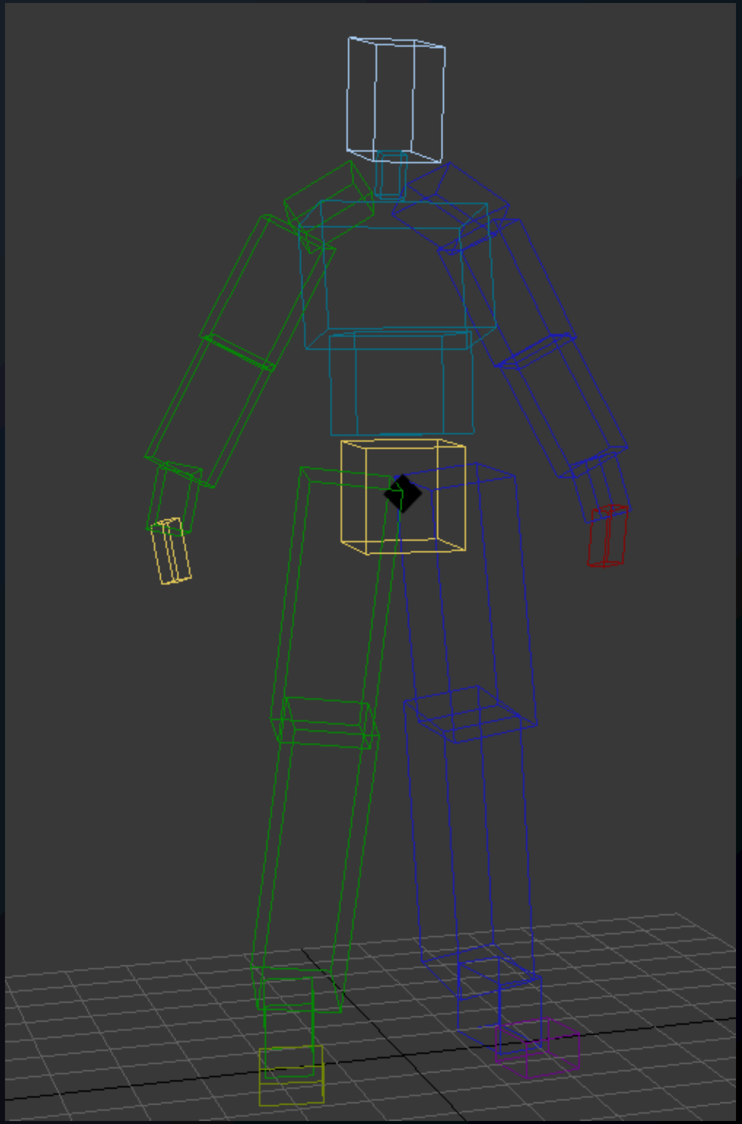
PART ONE

Animation Compression

站立动作压缩结果



未压缩.anim	2,511 KB
误差0.5.anim	204 KB
误差1.0.anim	146 KB
组合压缩.anim	123 KB



PART ONE

Assets Compression

Mesh

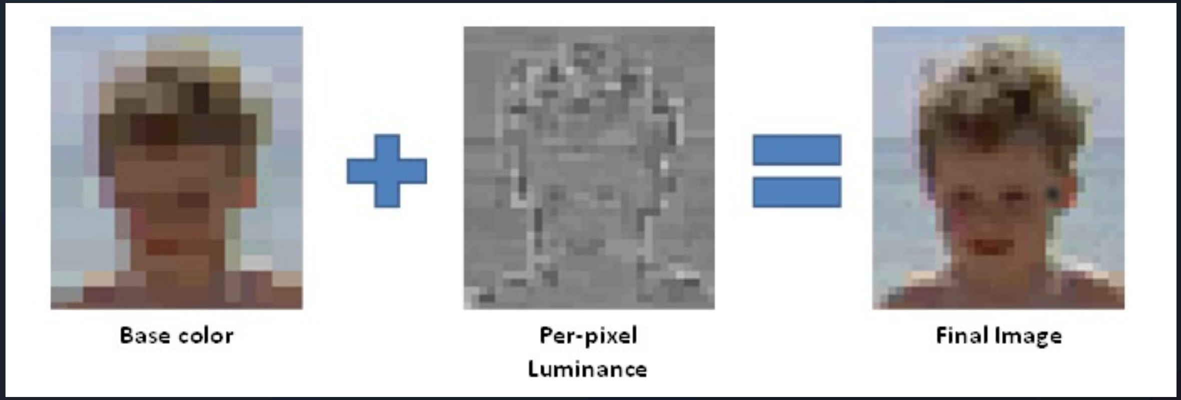
Animation

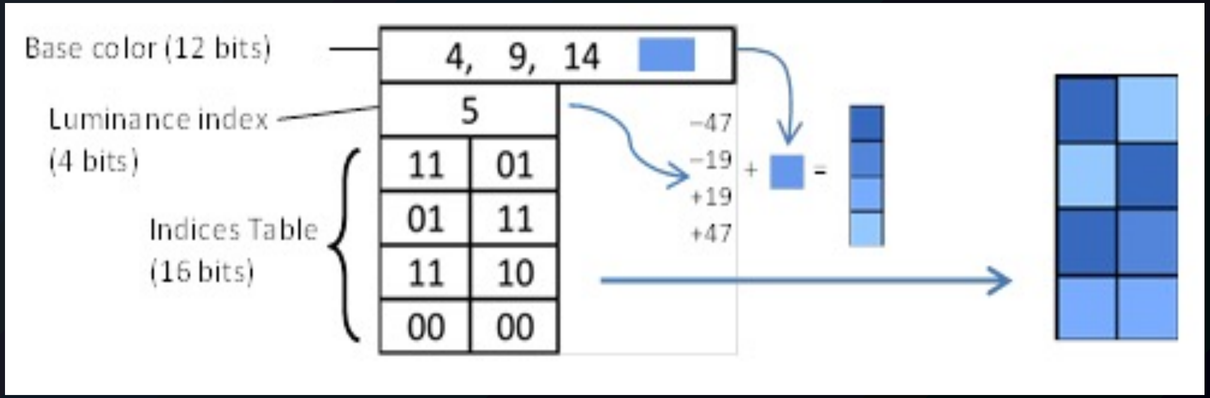
Texture

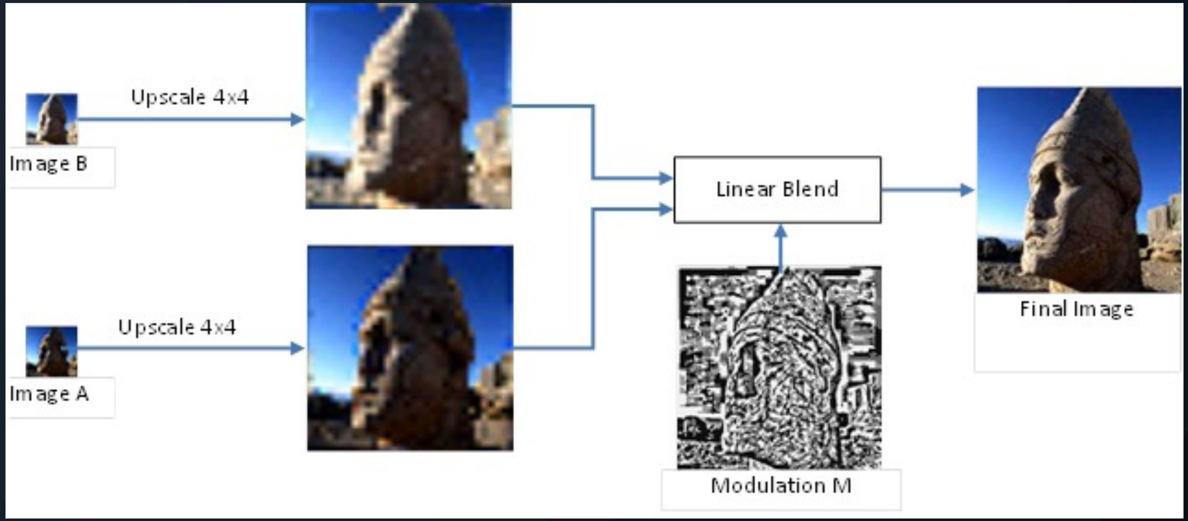
Audio

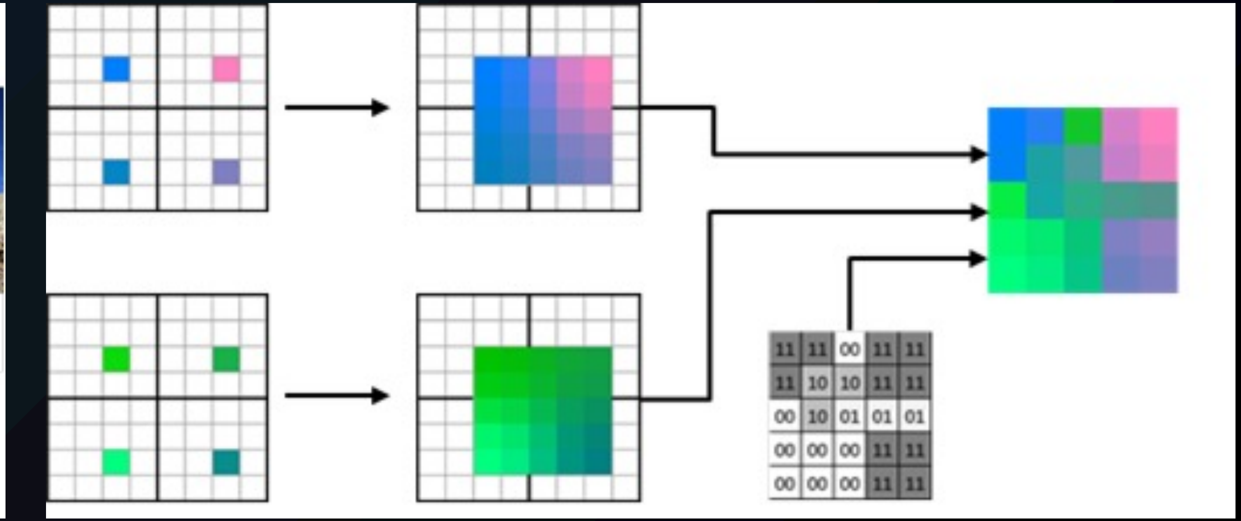
PART ONE

Texture Compression









PART ONE

Texture Compression

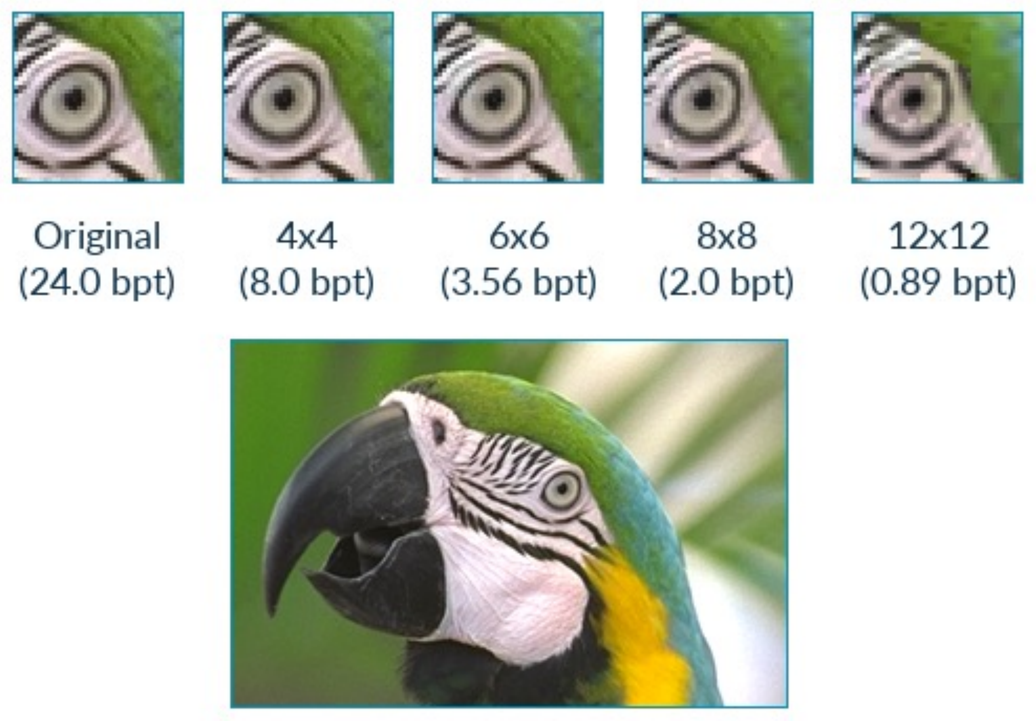
Adaptive Scalable Texture Compression 自适应可伸缩纹理压缩

跨平台

无尺寸限制

更多的压缩模式

更高的压缩率



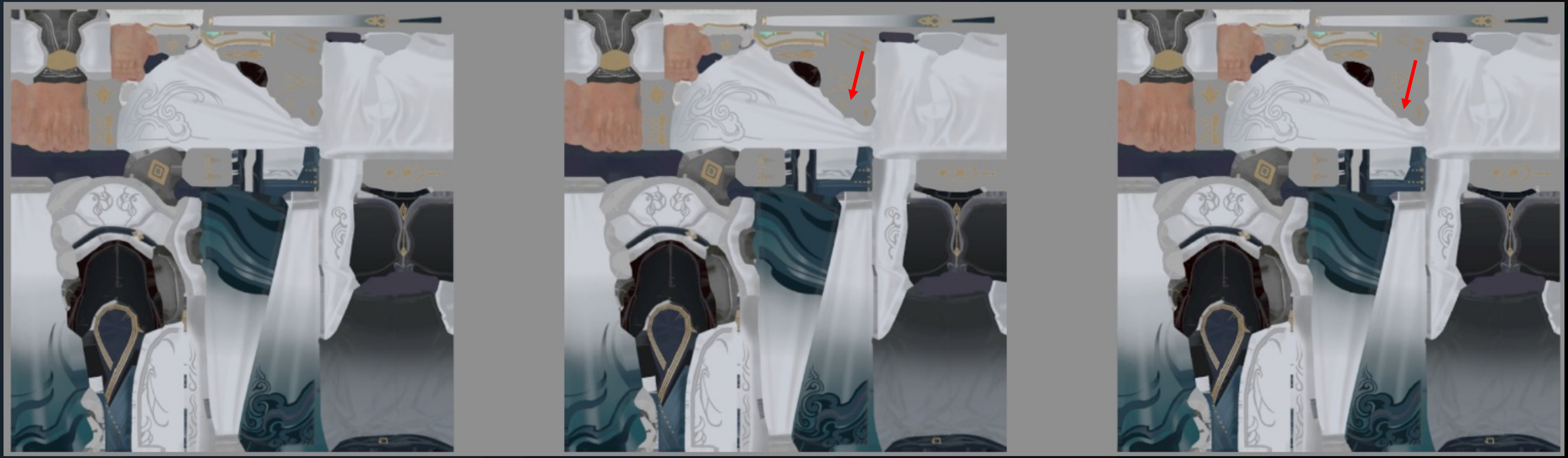
Block footprint	Bits/texel
4x4	8.00
5x5	5.12
6x6	3.56
8x8	2.00
10x10	1.28
12x12	0.89

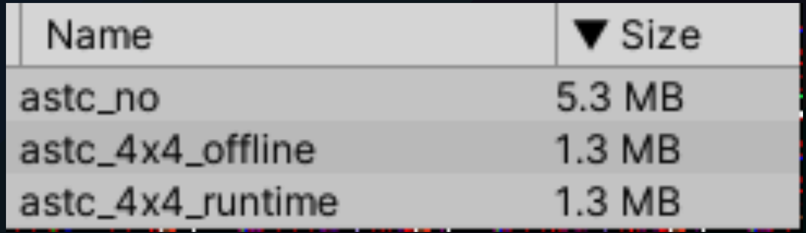
Block footprint	Bits/texel
5x4	6.40
6x5	4.27
8x5	3.20
8x6	2.67
10x5	2.56
10x6	2.13
10x8	1.60
12x10	1.07

PART ONE

Runtime ASTC

ASTC 4x4 : 1024x1024 RGBA





Name	▼ Size
astc_no	5.3 MB
astc_4x4_offline	1.3 MB
astc_4x4_runtime	1.3 MB

PART ONE

Assets Compression

Mesh

Animation

Texture

Audio

PART ONE

Audio Compression

.OGG ：技能、动作、UI、NPC、环境音效

.MP3 ：TV、场景等高质量音效

LoadType

- Decompress on load ：length < 3
- Compressed in Memory ：length < 10
- Streaming ：length >= 10

Preload Audio Data ：length < 3

单声道：

Force To Mono ：true

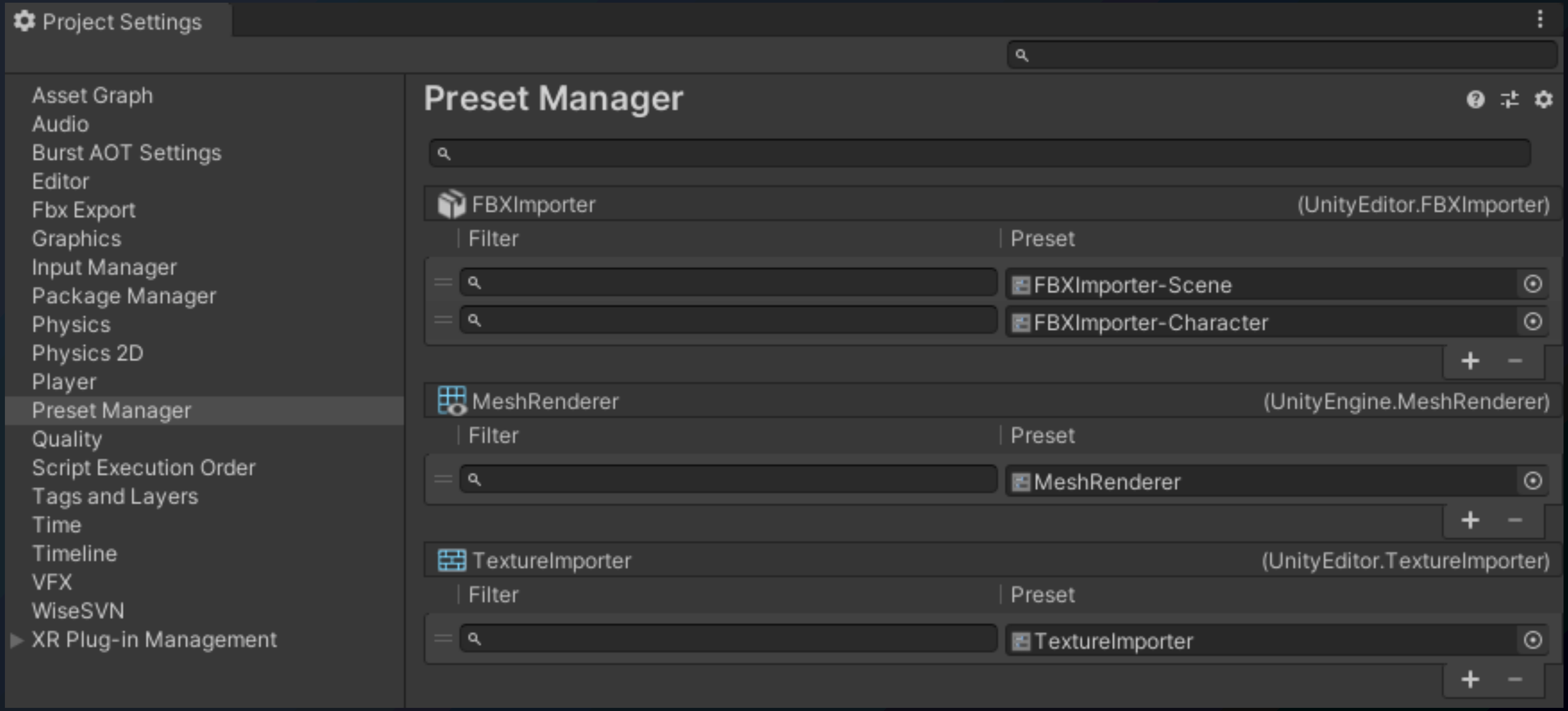
Sample Rate ：22050Hz

PART ONE

Assets Importer

Asset Postprocessor API

Preset Manager



Preset Manager

FBXImporter (UnityEditor.FBXImporter)

Filter	Preset
	FBXImporter-Scene
	FBXImporter-Character

MeshRenderer (UnityEngine.MeshRenderer)

Filter	Preset
	MeshRenderer

TextureImporter (UnityEditor.TextureImporter)

Filter	Preset
	TextureImporter

Runtime Optimization

PART TWO

Runtime Optimization

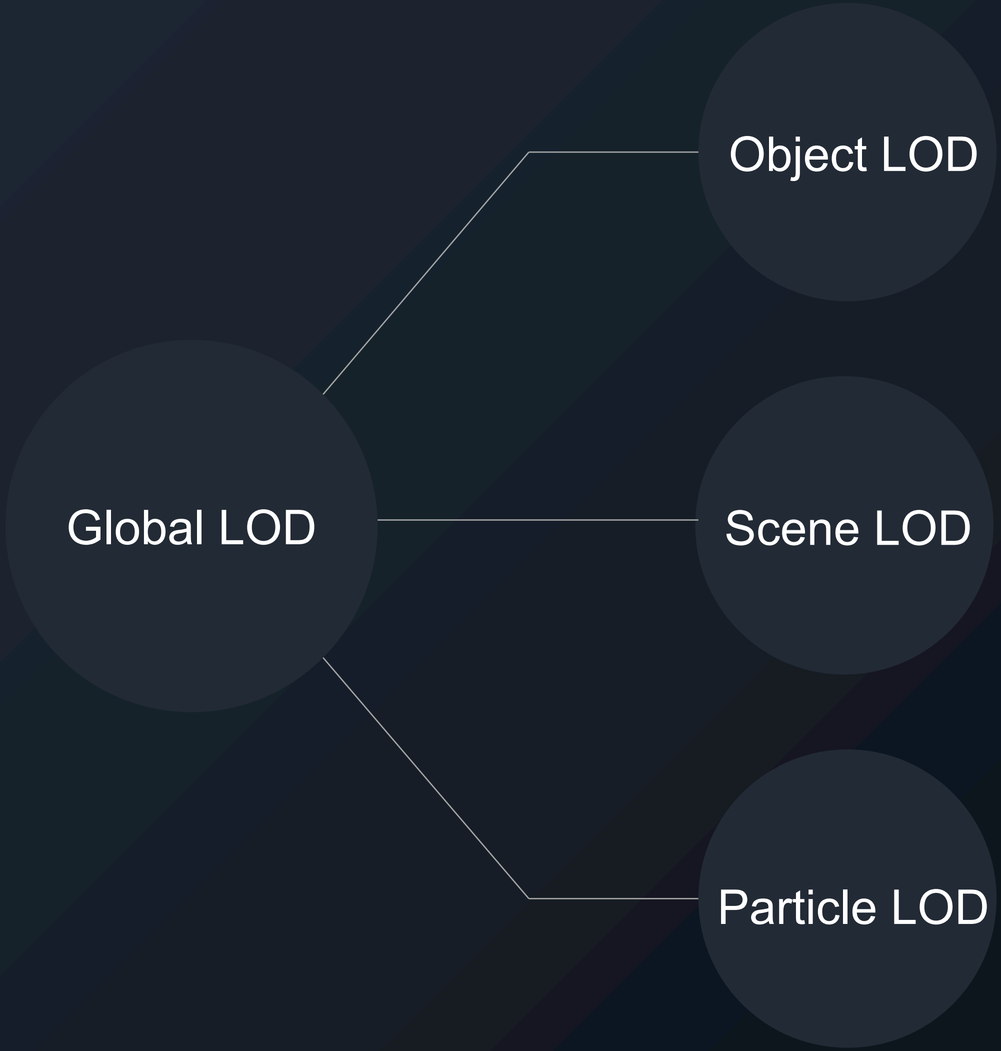
Global LOD

Scene Streaming

Universal RP

PART TWO

Runtime Optimization



Global LOD

Object LOD

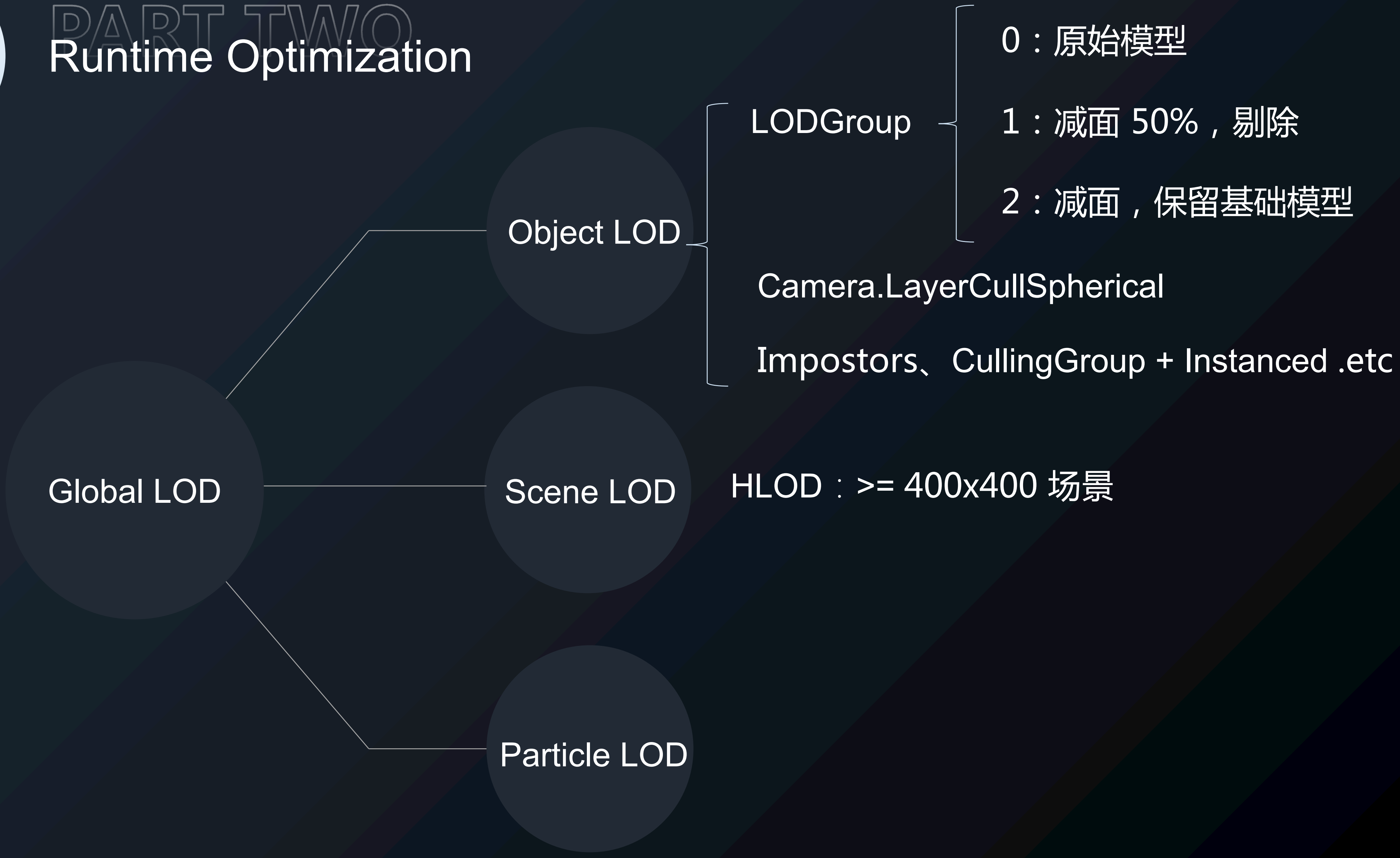
Scene LOD

Particle LOD

HLOD：>= 400x400 场景

PART TWO

Runtime Optimization



Global LOD

Object LOD

- LODGroup
  - 0：原始模型
  - 1：减面 50%，剔除
  - 2：减面，保留基础模型
- Camera.LayerCullSpherical
- Impostors、CullingGroup + Instanced .etc

Scene LOD

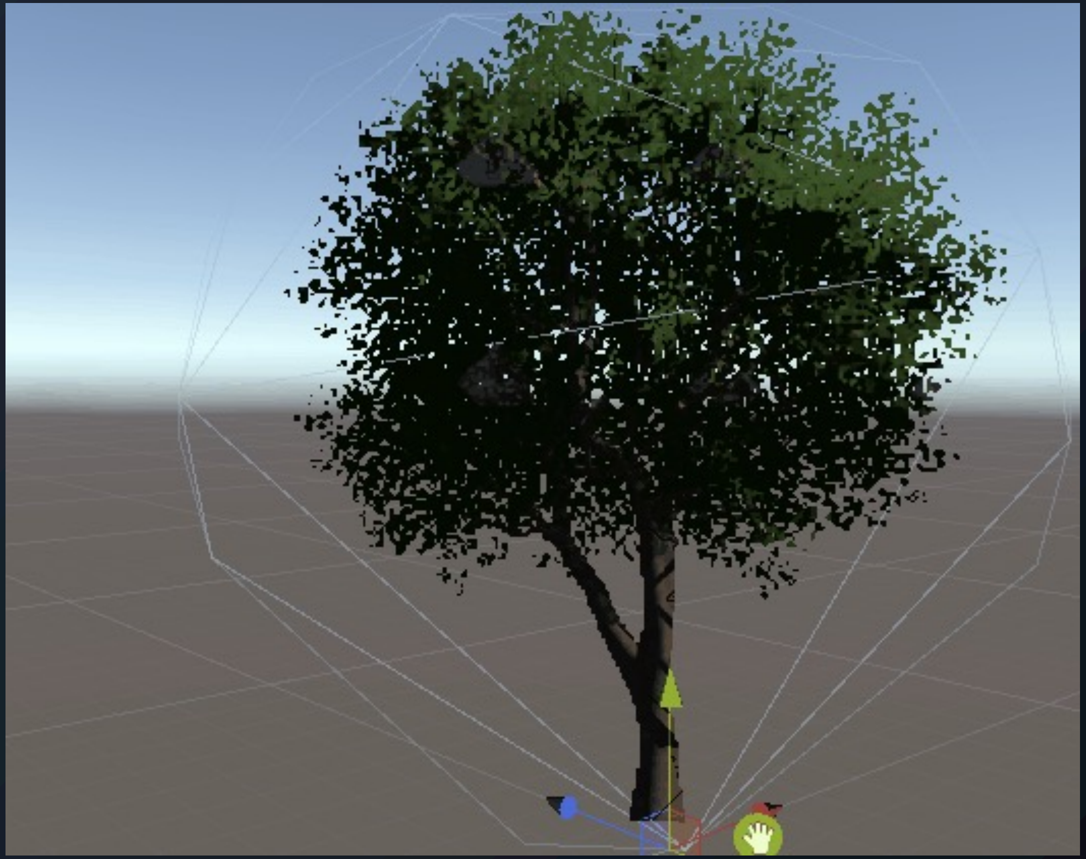
- HLOD：>= 400x400 场景

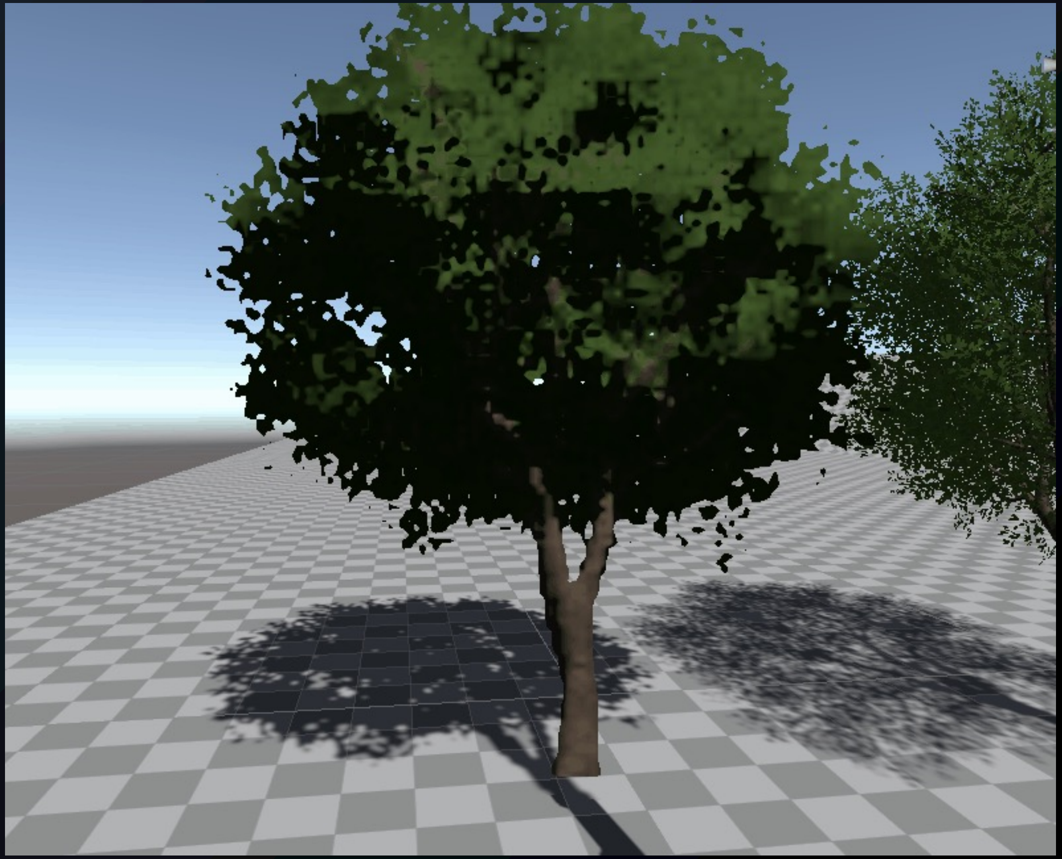
Particle LOD

PART TWO

Runtime Optimization

Amplify Impostors

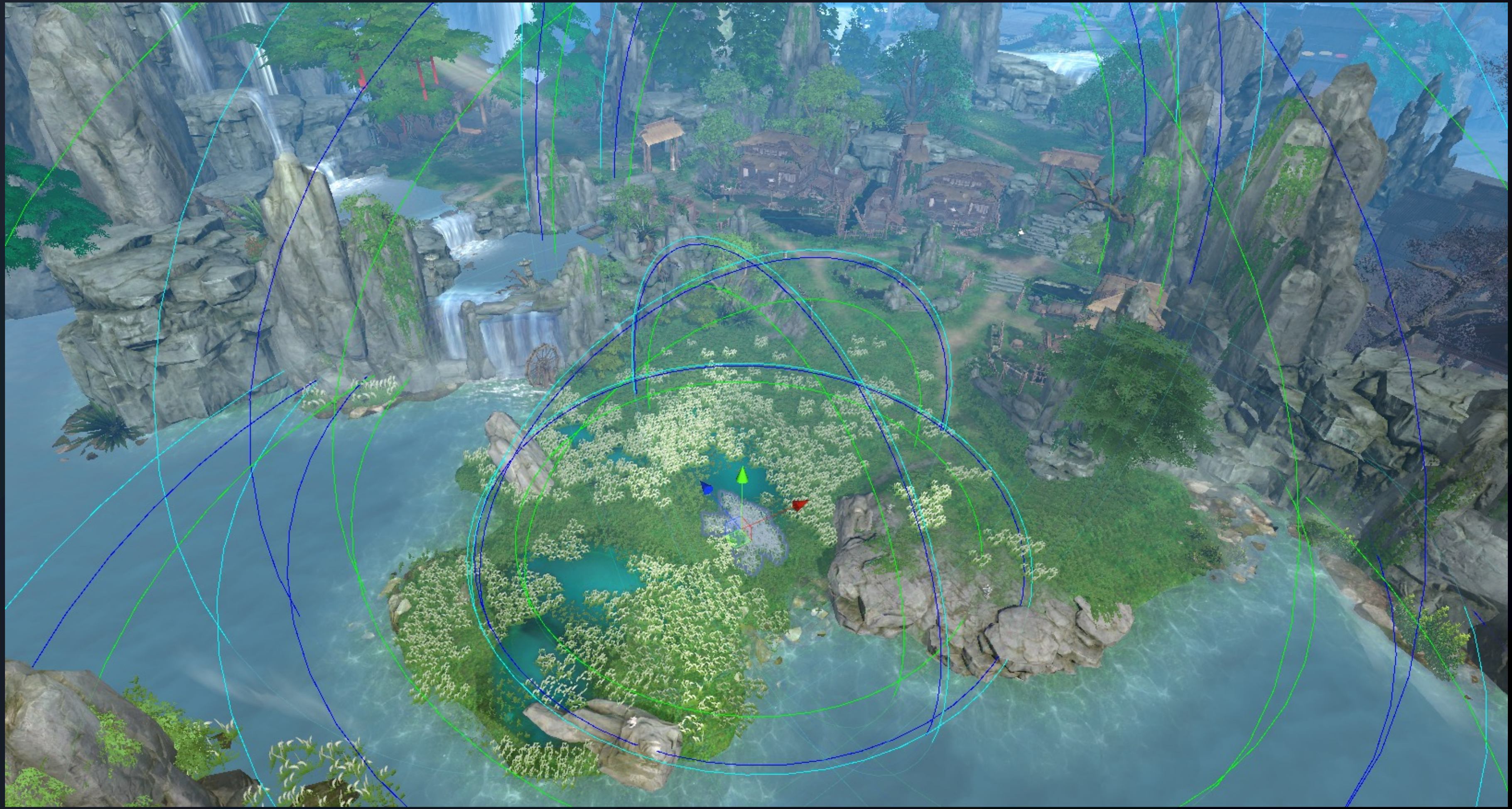




PART TWO

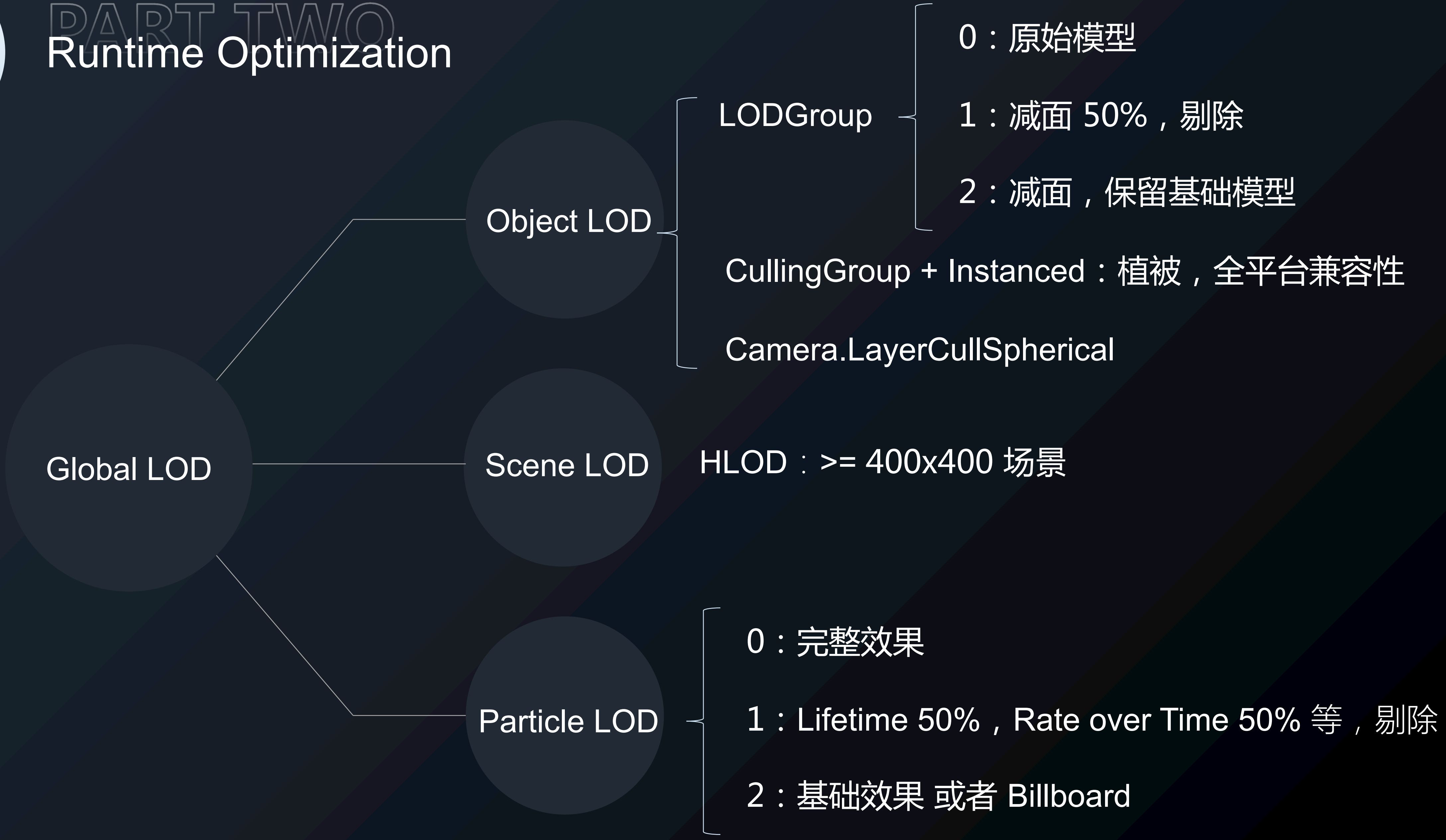
Runtime Optimization

CullingGroup + Instanced



PART TWO

Runtime Optimization



Global LOD

Object LOD

LODGroup

0：原始模型

1：减面 50%，剔除

2：减面，保留基础模型

CullingGroup + Instanced：植被，全平台兼容性

Camera.LayerCullSpherical

Scene LOD

HLOD：>= 400x400 场景

Particle LOD

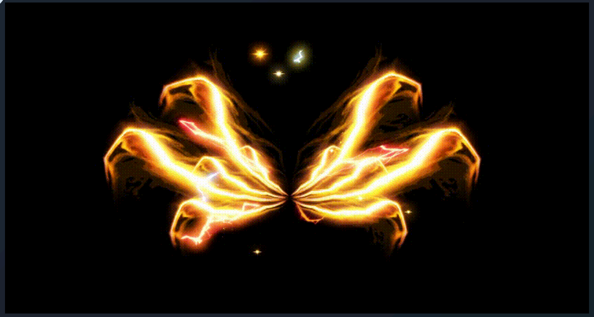
0：完整效果

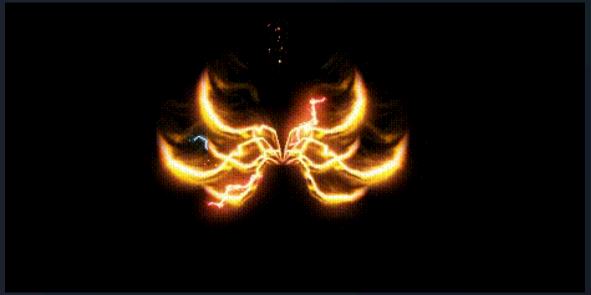
1：Lifetime 50%，Rate over Time 50% 等，剔除

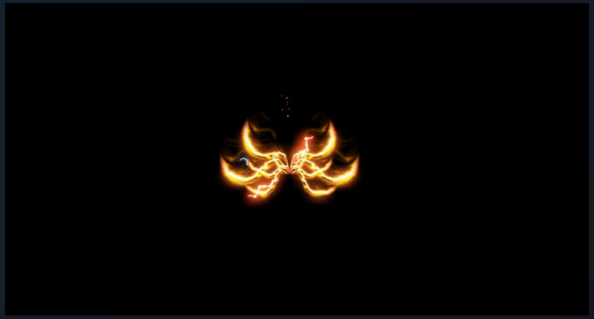
2：基础效果 或者 Billboard

PART TWO

Runtime Optimization

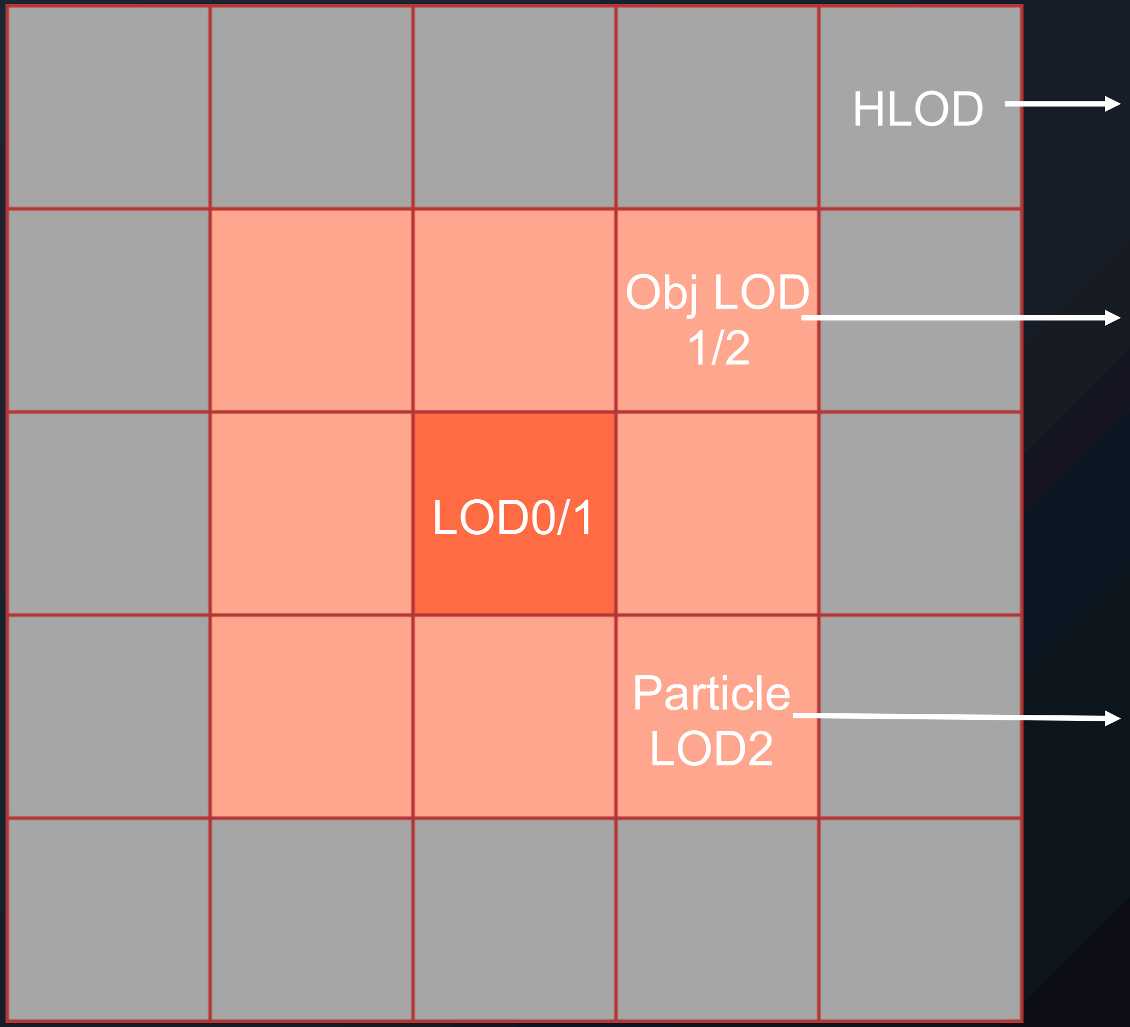






PART TWO

Runtime Optimization



Added fewer mesh and texture，20Mb

Added more mesh

Added fewer billboard

PART TWO

Runtime Optimization

Global LOD

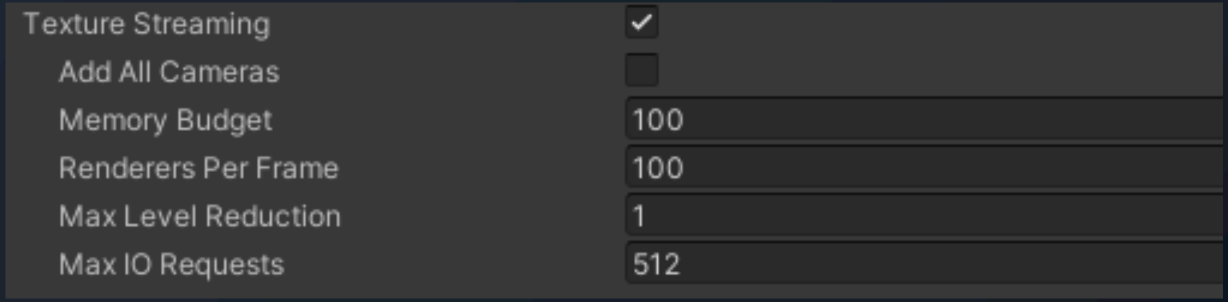
Scene Streaming

Universal RP

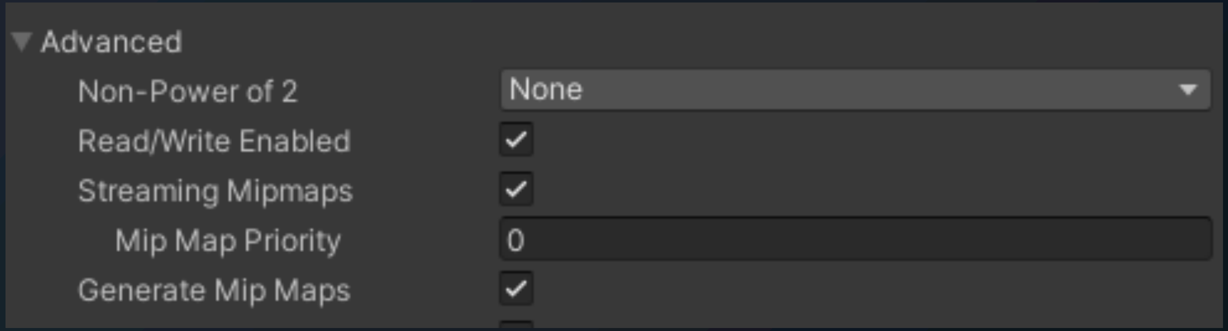
PART TWO

Runtime Optimization

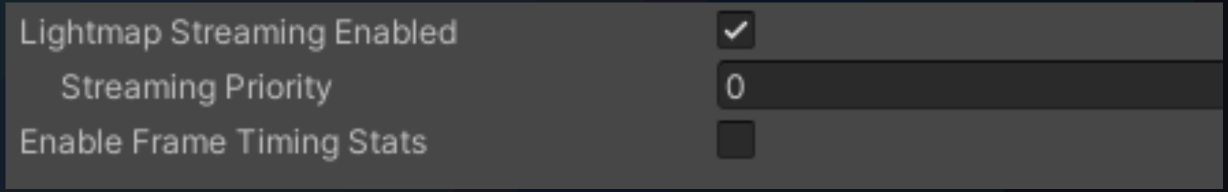
开启 TextureStreaming



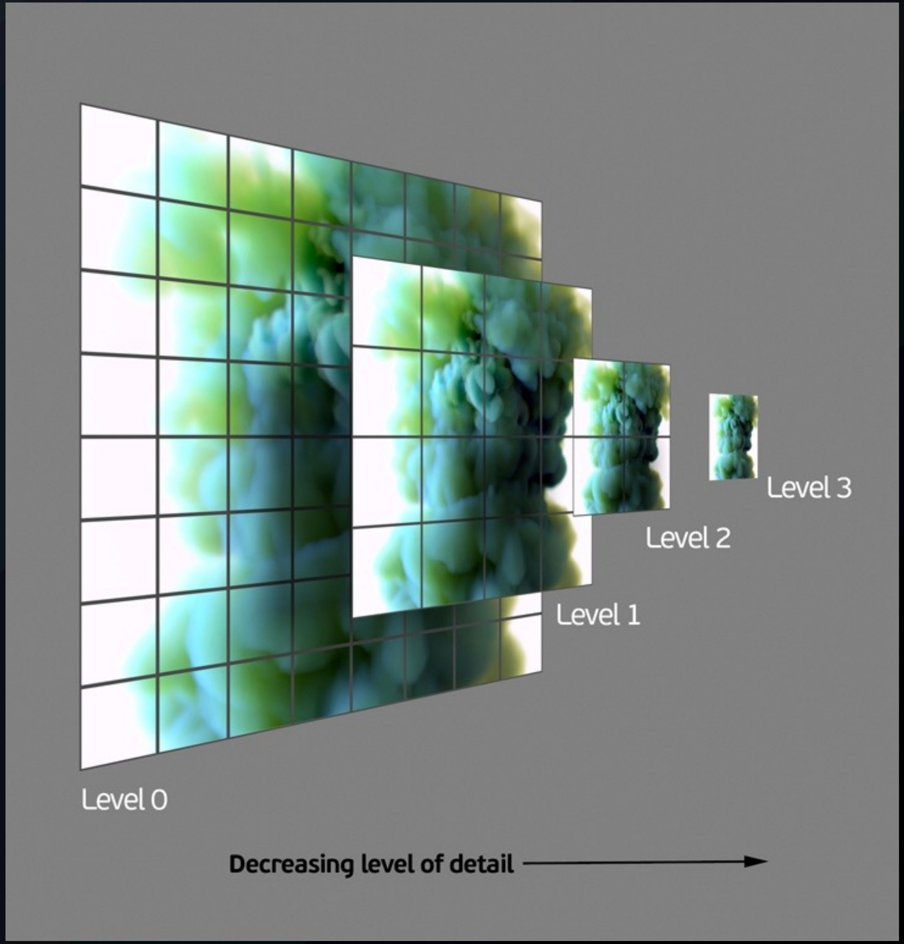
Texture Streaming ☑
Add All Cameras □
Memory Budget 100
Renderers Per Frame 100
Max Level Reduction 1
Max IO Requests 512



▼ Advanced
Non-Power of 2 None
Read/Write Enabled ☑
Streaming Mipmaps ☑
Mip Map Priority 0
Generate Mip Maps ☑



Lightmap Streaming Enabled ☑
Streaming Priority 0
Enable Frame Timing Stats □

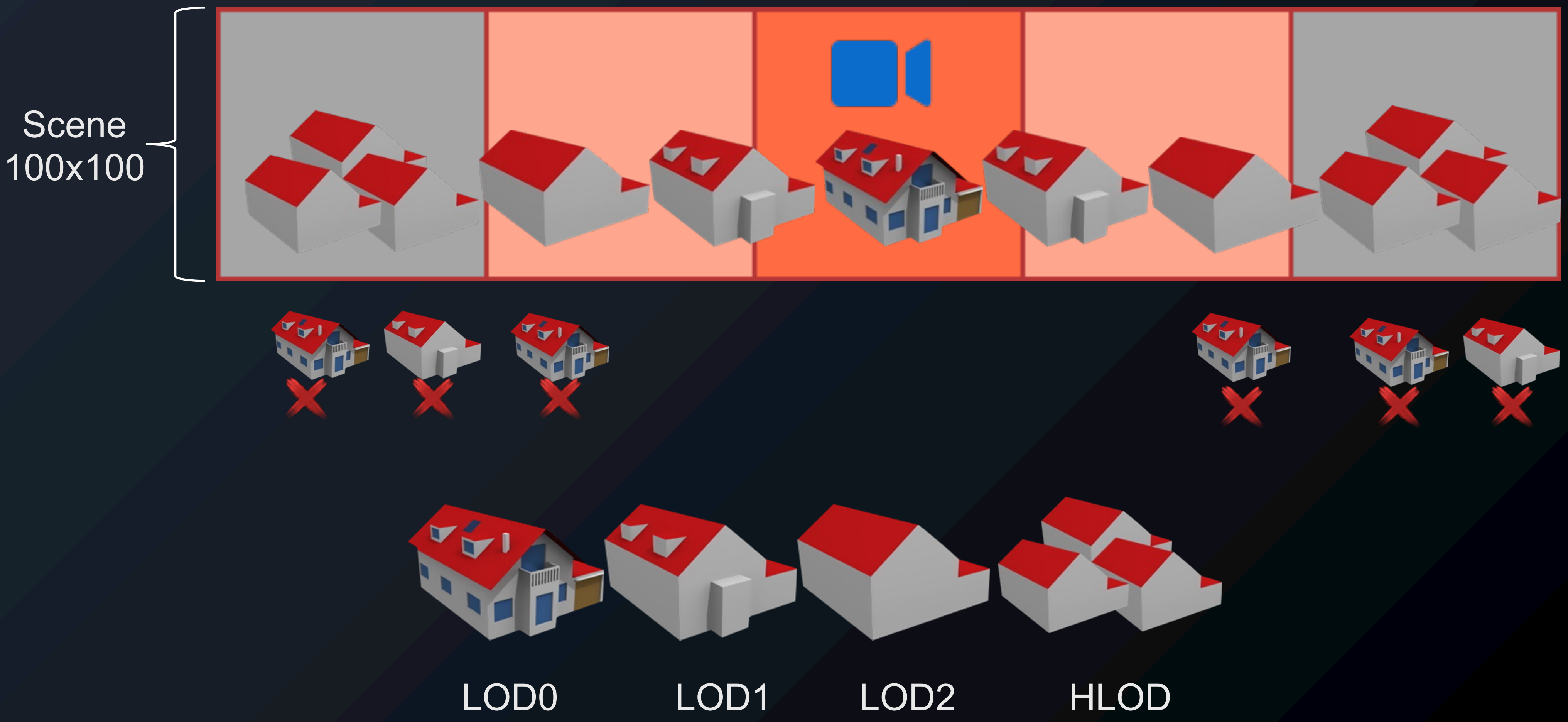


PART TWO

Runtime Optimization

Runtime Optimization

Mesh Streaming



PART TWO

Runtime Optimization

PART TWO

Runtime Optimization

~~ComputeShader + DrawMeshInstancedIndirect~~

DOTS + DrawMeshInstanced + SRPBatch 兼容性 √

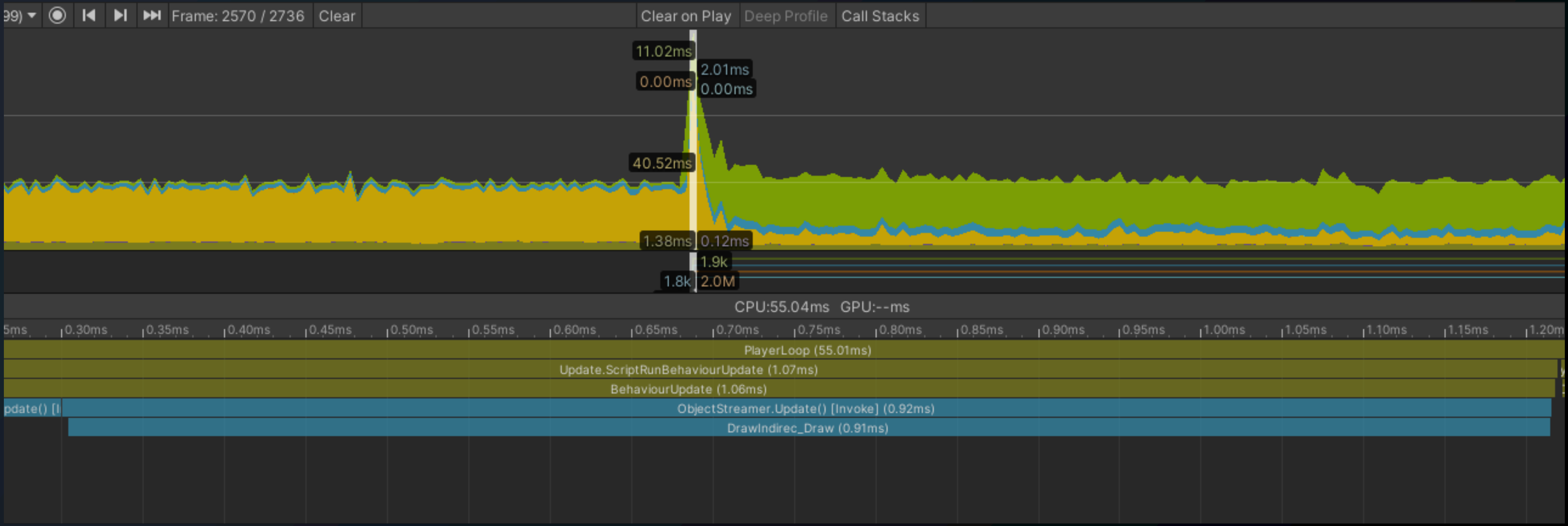
PART TWO

Runtime Optimization

~~LoadScene Additive~~

~~GameObject.Instantiate~~

AsyncLoad + Graphics.DrawMesh，1000个GameObject



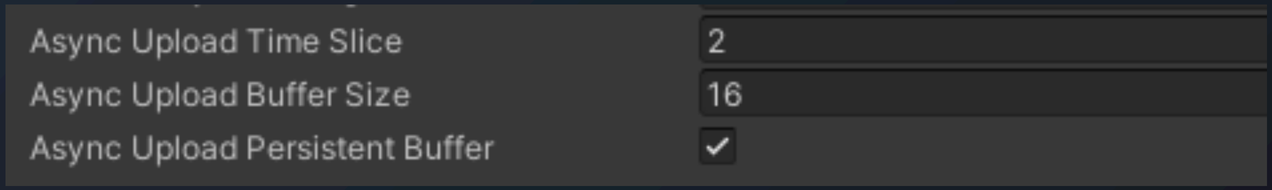
CPU:55.04ms GPU:--ms

PlayerLoop (55.01ms)
Update.ScriptRunBehaviourUpdate (1.07ms)
BehaviourUpdate (1.06ms)
pdate() [I
ObjectStreamer.Update() [Invoke] (0.92ms)
DrawIndirec_Draw (0.91ms)

PART TWO

Runtime Optimization

开启 AsyncUpload



Async Upload Time Slice 2

Async Upload Buffer Size 16

Async Upload Persistent Buffer ☑

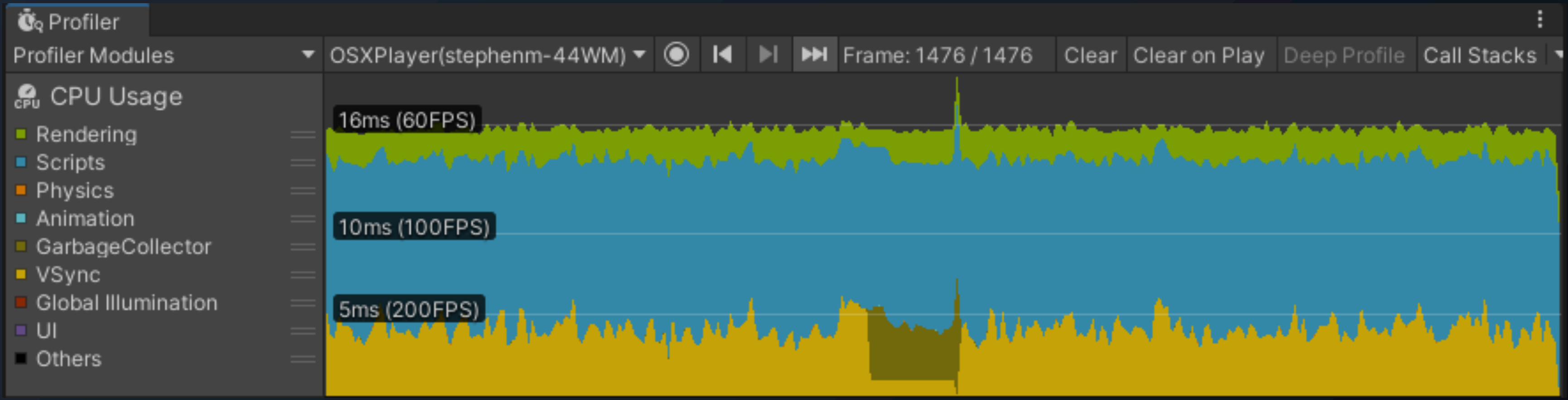
Recommendations

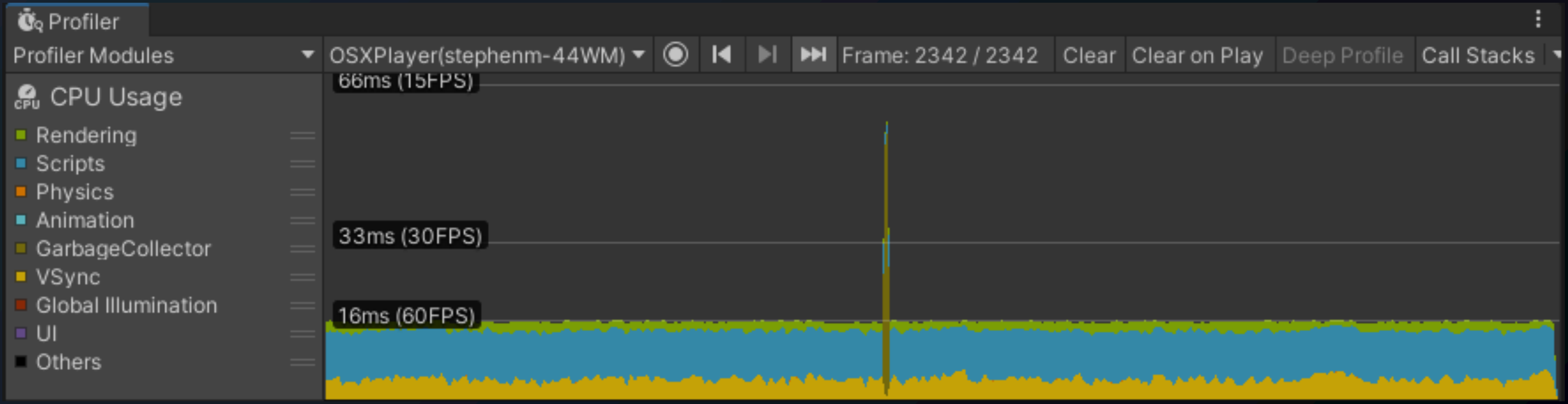
- Choose the largest QualitySettings.asyncUploadTimeSlice that doesn’t result in dropping frames.
- During loading screens, temporarily increase QualitySettings.asyncUploadTimeSlice.
- Use the profiler to examine the time slice utilization. The time slice will show up as AsyncUploadManager.AsyncResourceUpload in the profiler. Increase QualitySettings.asyncUploadBufferSize if your time slice is not being fully utilized.
- Things will generally load faster with a larger QualitySettings.asyncUploadBufferSize, so if you can afford the memory, increase it to 16MB or 32MB.
- Leave QualitySettings.asyncUploadPersistentBuffer set to true unless you have a compelling reason to reduce your runtime memory usage while not loading.

PART TWO

Runtime Optimization

开启 Incremental GC





https://docs.unity3d.com/Manual/performance-incremental-garbage-collection.html

PART TWO

Runtime Optimization

Global LOD

Scene Streaming

Universal RP

PART TWO

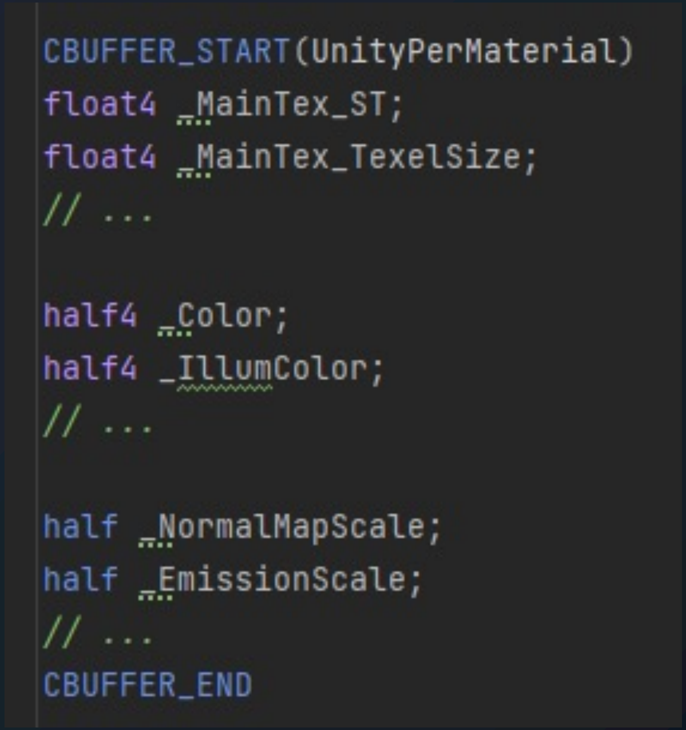
Runtime Optimization

- Static Batch：up to 64000 vertices
    - StaticBatchingUtility.Combine
- Dynamic Batch：900 vertex attributes
- GPU Instancing
    - DrawMeshInstanced：maximum of 1023 instances at once
    - DrawMeshInstancedIndirect：Compatibility On Android !!!
    - Limitation：GL_ARB_draw_instanced
        - OpenGLES Extension Viewer
- SRP Batch：Scriptable Render Pipeline
- DOTS Instancing：com.unity.rendering.hybrid(Unity 2019 preview)，Entities Graphics(Unity 2022)
        - Android：Vulkan

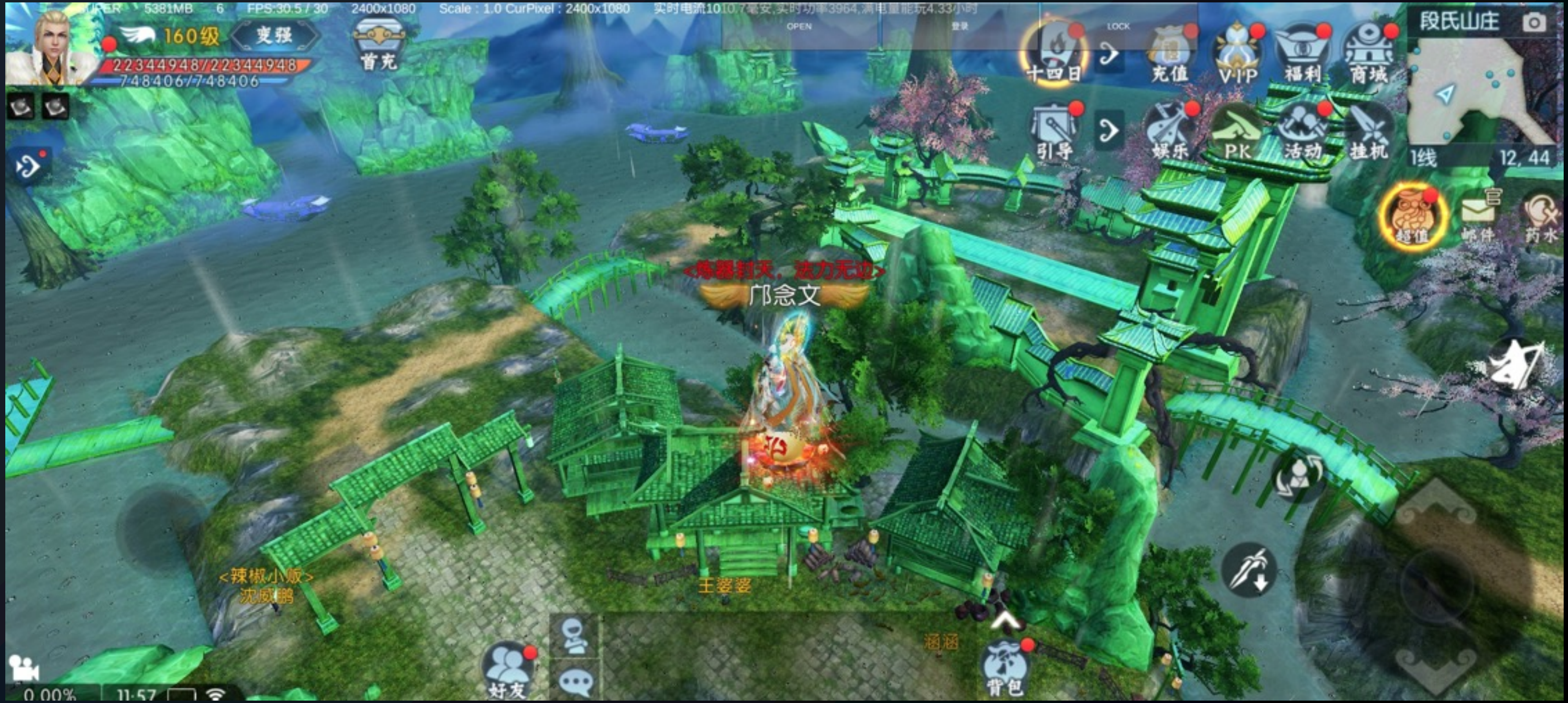
PART TWO

Runtime Optimization

CBuffer Byte Alignment



```
CBUFFER_START(UnityPerMaterial)
float4 _MainTex_ST;
float4 _MainTex_TexelSize;
// ...

half4 _Color;
half4 _IllumColor;
// ...

half _NormalMapScale;
half _EmissionScale;
// ...
CBUFFER_END
```

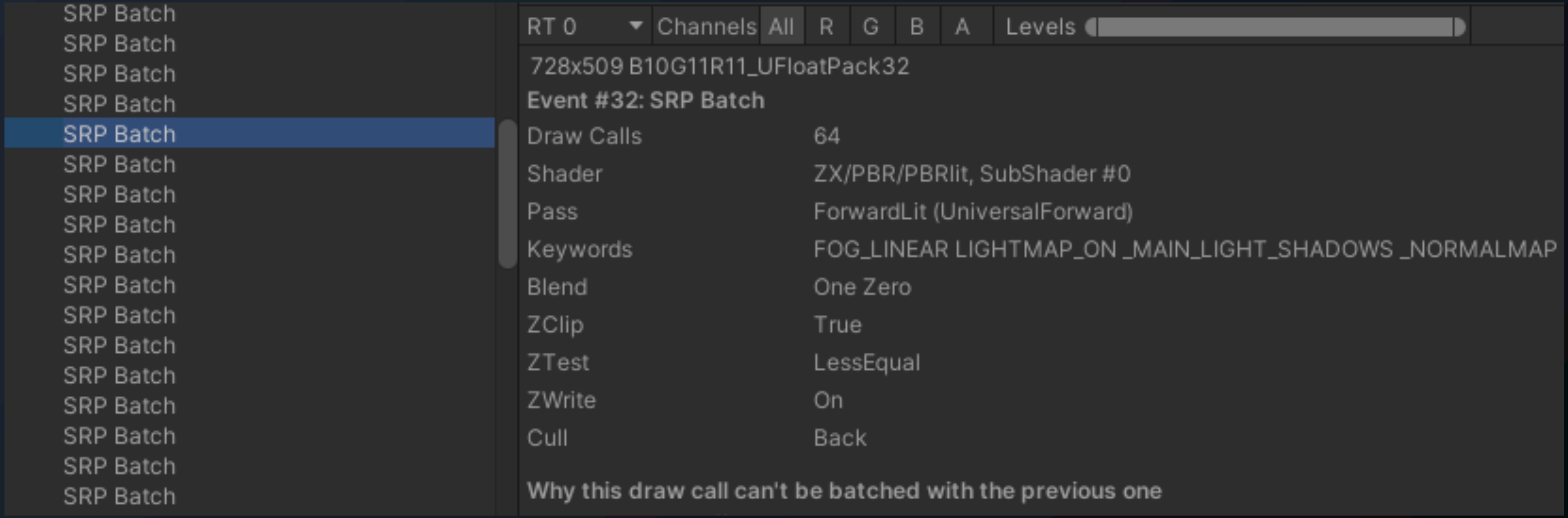


bytes unaligned caused error，unity 2019.4.33f1

PART TWO

Runtime Optimization

Shader Keywords



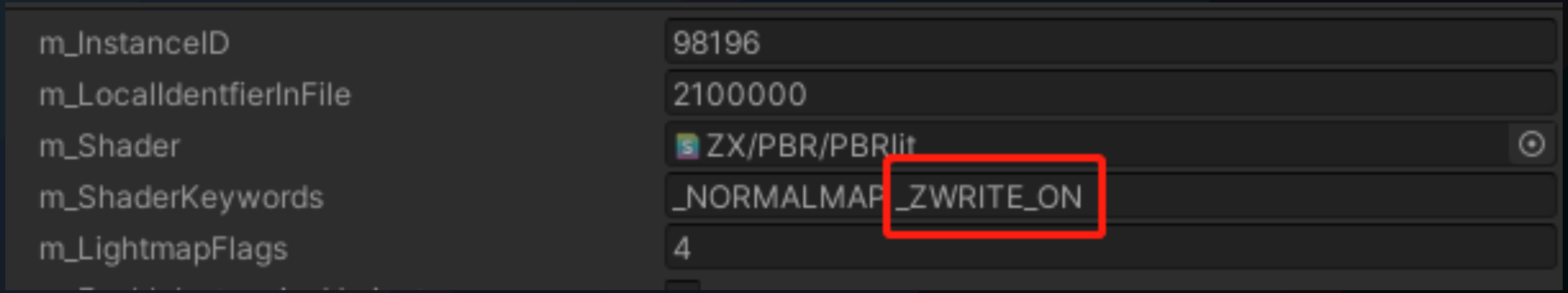
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch
SRP Batch

728x509 B10G11R11_UFloatPack32

Event #32: SRP Batch

Draw Calls	64
Shader	ZX/PBR/PBRlit, SubShader #0
Pass	ForwardLit (UniversalForward)
Keywords	FOG_LINEAR LIGHTMAP_ON _MAIN_LIGHT_SHADOWS _NORMALMAP
Blend	One Zero
ZClip	True
ZTest	LessEqual
ZWrite	On
Cull	Back

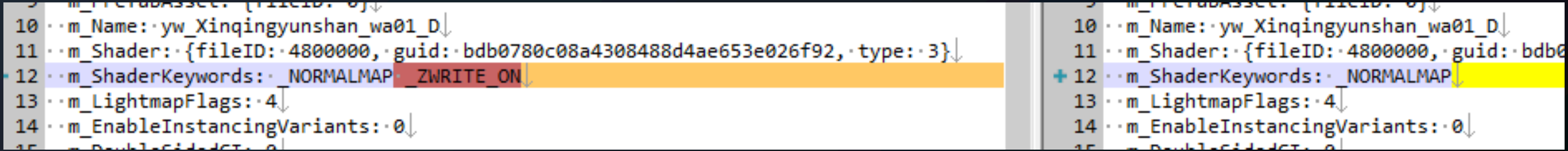
Why this draw call can't be batched with the previous one



m_InstanceID	98196
m_LocalIdentfierInFile	2100000
m_Shader	ZX/PBR/PBRlit
m_ShaderKeywords	_NORMALMAP _ZWRITE_ON
m_LightmapFlags	4

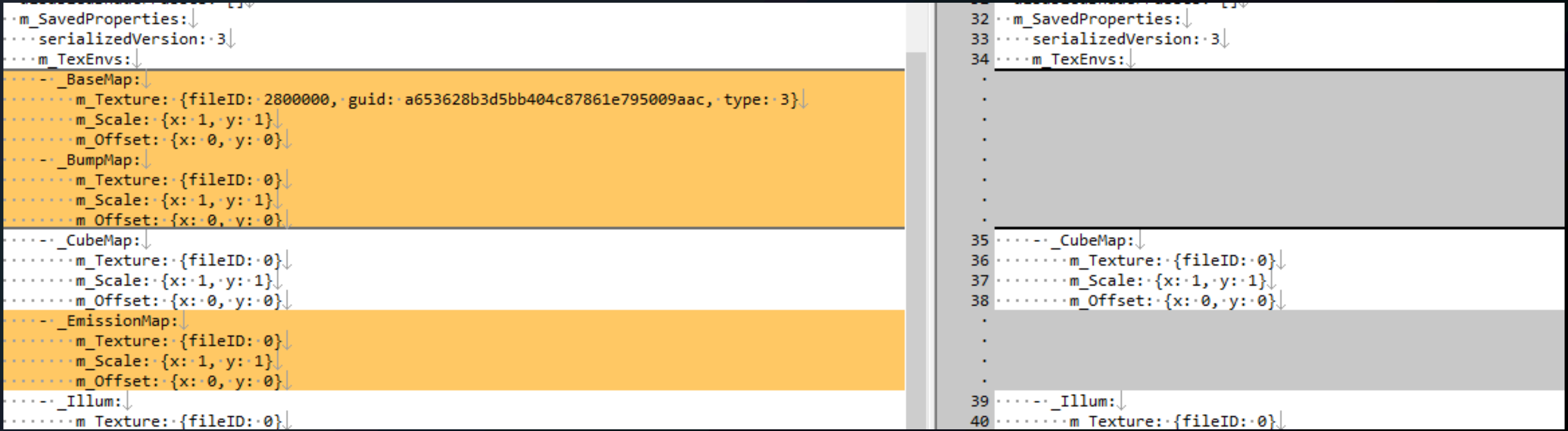
PART TWO

Runtime Optimization



```
  m_PrefabAsset: {fileID: 0}
  m_Name: yw_Xinqingyunshan_wa01_D
  m_Shader: {fileID: 4800000, guid: bdb0780c08a4308488d4ae653e026f92, type: 3}
  m_ShaderKeywords: _NORMALMAP _ZWRITE_ON
  m_LightmapFlags: 4
  m_EnableInstancingVariants: 0
  m_DoubleSidedGI: 0
```

```
  m_PrefabAsset: {fileID: 0}
  m_Name: yw_Xinqingyunshan_wa01_D
  m_Shader: {fileID: 4800000, guid: bdb0
  m_ShaderKeywords: _NORMALMAP
  m_LightmapFlags: 4
  m_EnableInstancingVariants: 0
  m_DoubleSidedGI: 0
```



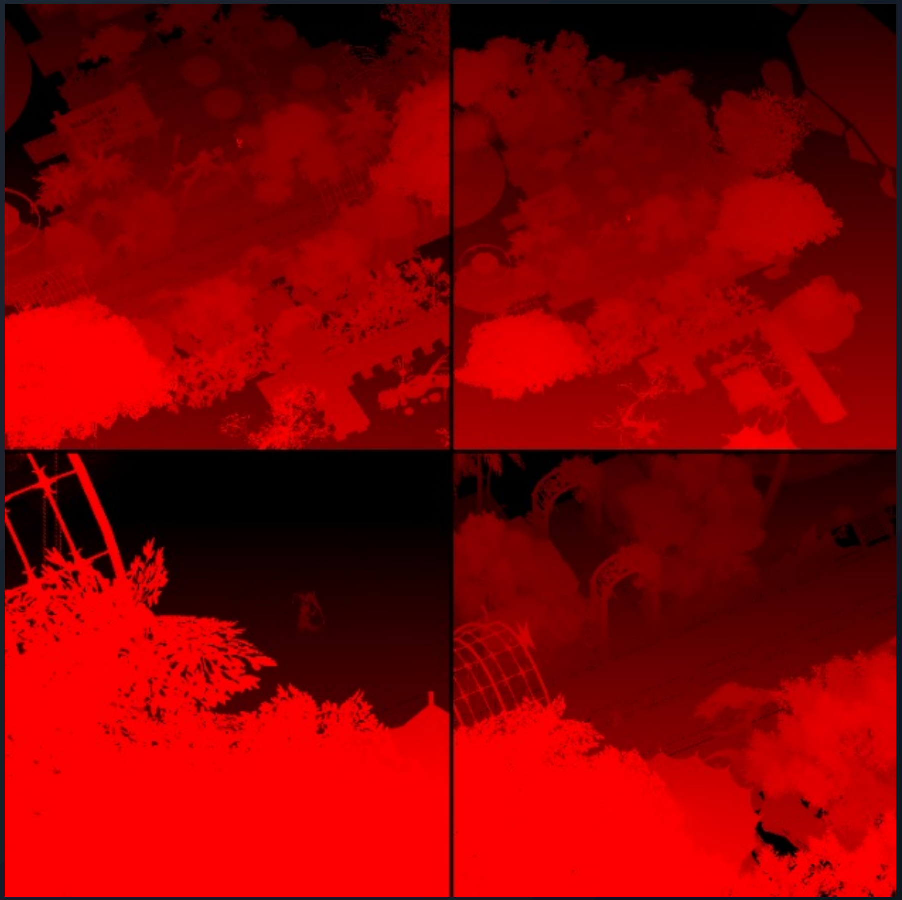
```
  m_SavedProperties:
    serializedVersion: 3
    m_TexEnvs:
    - _BaseMap:
        m_Texture: {fileID: 2800000, guid: a653628b3d5bb404c87861e795009aac, type: 3}
        m_Scale: {x: 1, y: 1}
        m_Offset: {x: 0, y: 0}
    - _BumpMap:
        m_Texture: {fileID: 0}
        m_Scale: {x: 1, y: 1}
        m_Offset: {x: 0, y: 0}
    - _CubeMap:
        m_Texture: {fileID: 0}
        m_Scale: {x: 1, y: 1}
        m_Offset: {x: 0, y: 0}
    - _EmissionMap:
        m_Texture: {fileID: 0}
        m_Scale: {x: 1, y: 1}
        m_Offset: {x: 0, y: 0}
    - _Illum:
        m_Texture: {fileID: 0}
```

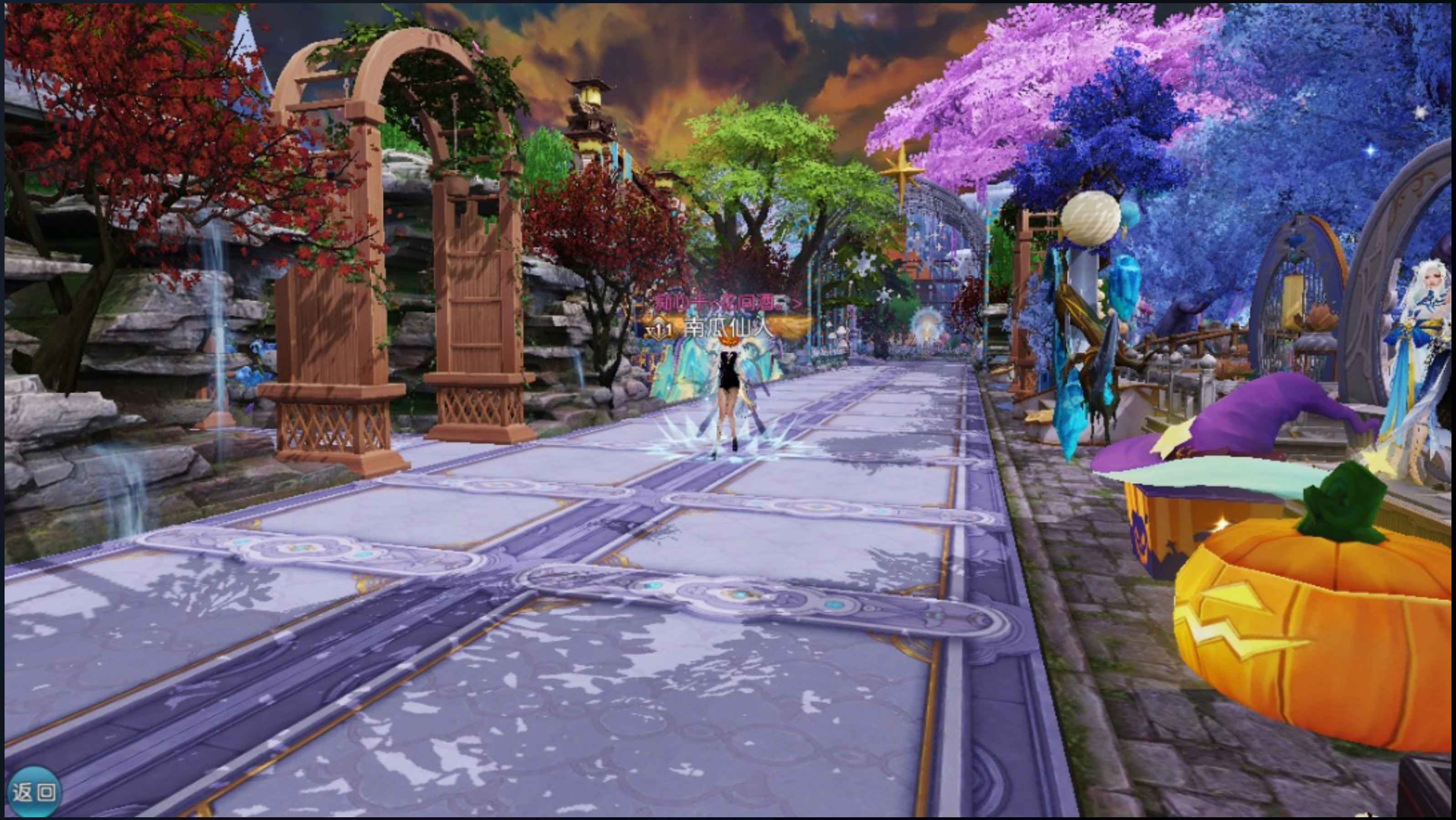
```
  m_SavedProperties:
    serializedVersion: 3
    m_TexEnvs:
    - _CubeMap:
        m_Texture: {fileID: 0}
        m_Scale: {x: 1, y: 1}
        m_Offset: {x: 0, y: 0}
    - _Illum:
        m_Texture: {fileID: 0}
```

PART TWO

Runtime Optimization

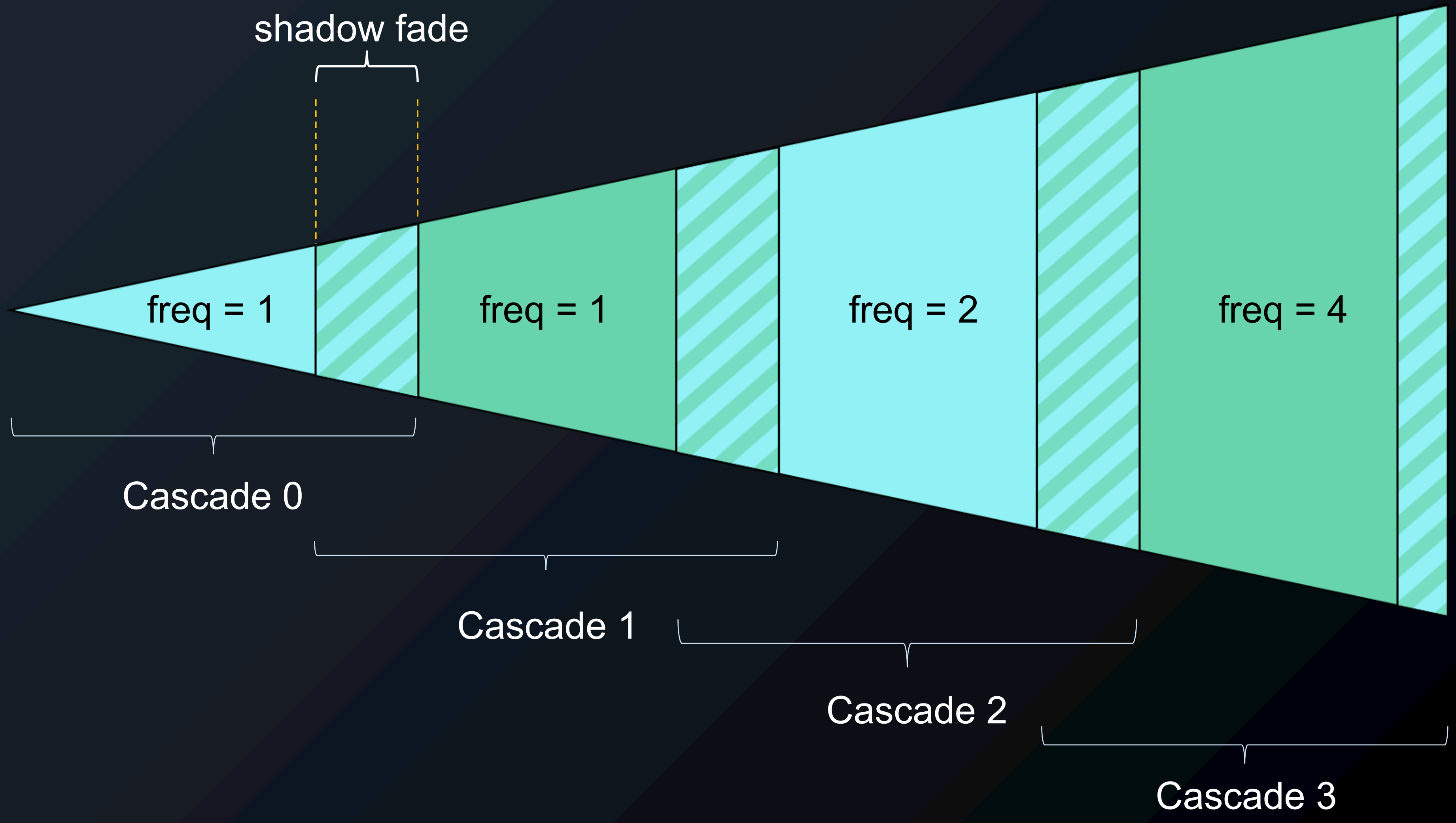
MultiFrame CSM





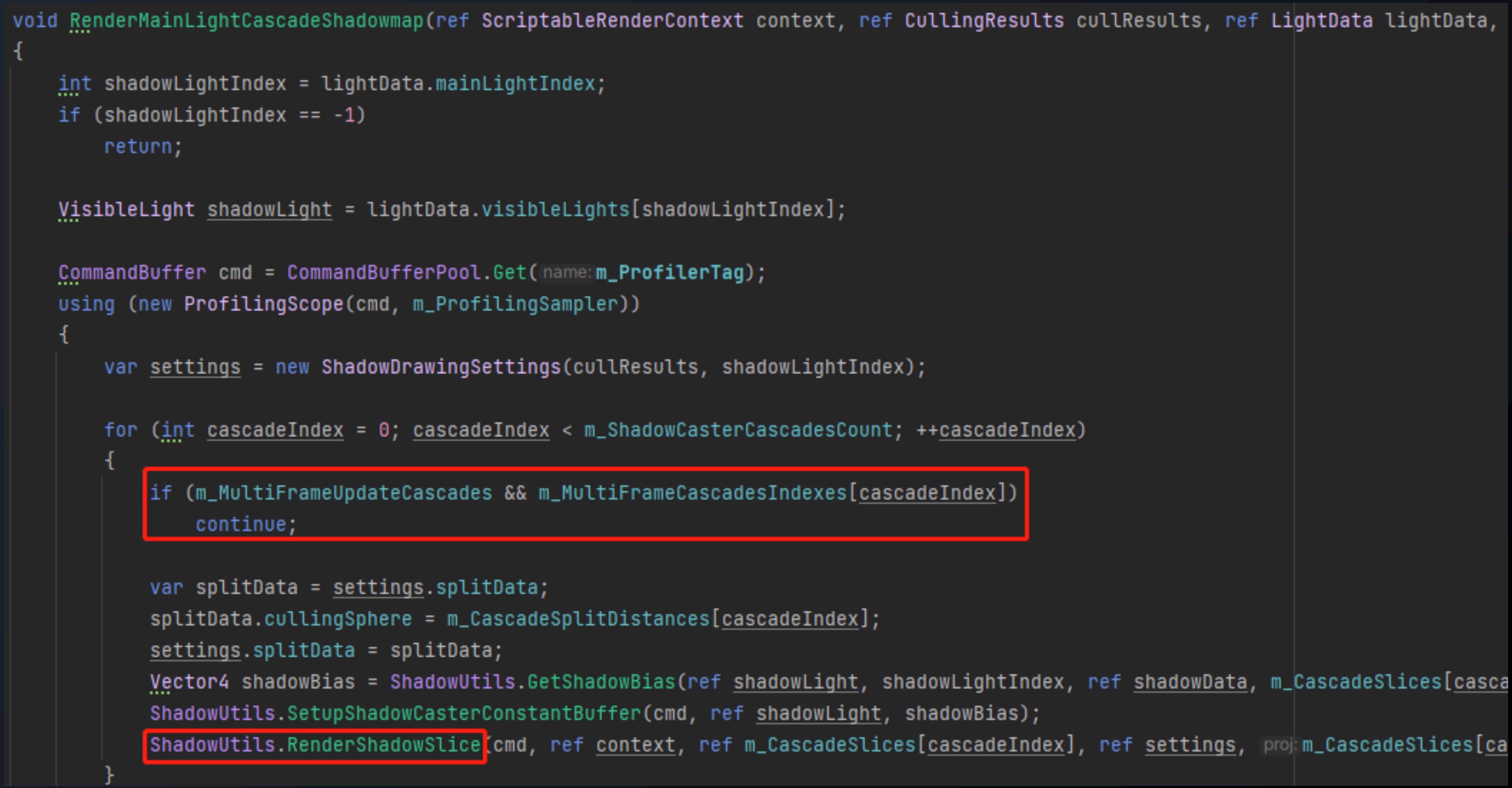
PART TWO

Runtime Optimization



PART TWO

Runtime Optimization

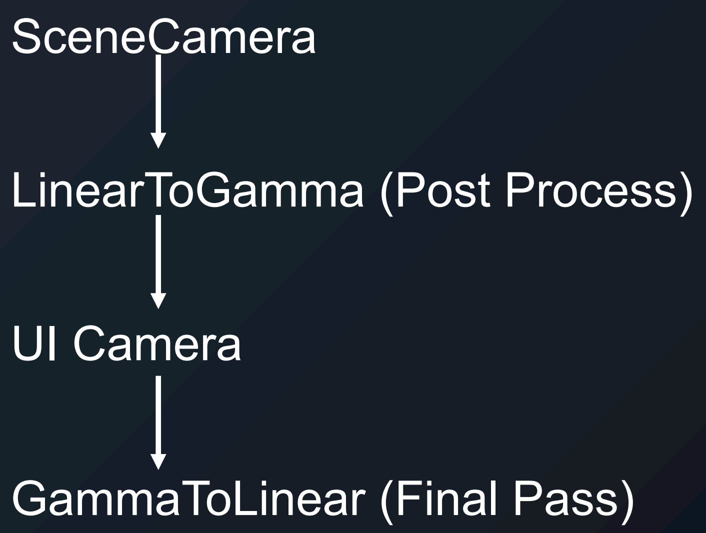


```
void RenderMainLightCascadeShadowmap(ref ScriptableRenderContext context, ref CullingResults cullResults, ref LightData lightData,
{
    int shadowLightIndex = lightData.mainLightIndex;
    if (shadowLightIndex == -1)
        return;

    VisibleLight shadowLight = lightData.visibleLights[shadowLightIndex];

    CommandBuffer cmd = CommandBufferPool.Get( name: m_ProfilerTag);
    using (new ProfilingScope(cmd, m_ProfilingSampler))
    {
        var settings = new ShadowDrawingSettings(cullResults, shadowLightIndex);

        for (int cascadeIndex = 0; cascadeIndex < m_ShadowCasterCascadesCount; ++cascadeIndex)
        {
            if (m_MultiFrameUpdateCascades && m_MultiFrameCascadesIndexes[cascadeIndex])
                continue;

            var splitData = settings.splitData;
            splitData.cullingSphere = m_CascadeSplitDistances[cascadeIndex];
            settings.splitData = splitData;
            Vector4 shadowBias = ShadowUtils.GetShadowBias(ref shadowLight, shadowLightIndex, ref shadowData, m_CascadeSlices[casca
            ShadowUtils.SetupShadowCasterConstantBuffer(cmd, ref shadowLight, shadowBias);
            ShadowUtils.RenderShadowSlice(cmd, ref context, ref m_CascadeSlices[cascadeIndex], ref settings, proj: m_CascadeSlices[ca
        }
```

PART TWO

Runtime Optimization

UPR Pipeline



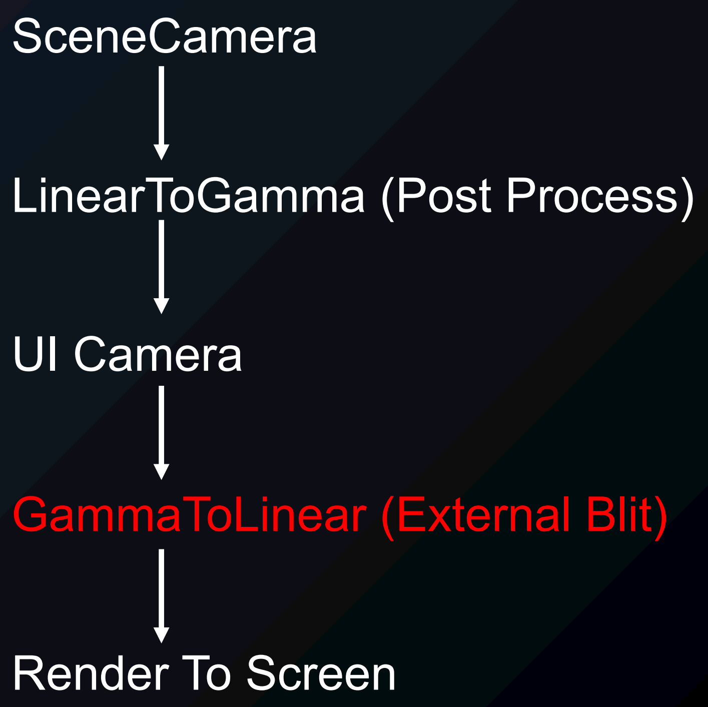
SceneCamera

LinearToGamma (Post Process)

UI Camera

GammaToLinear (Final Pass)

Builtin Pipeline



SceneCamera

LinearToGamma (Post Process)

UI Camera

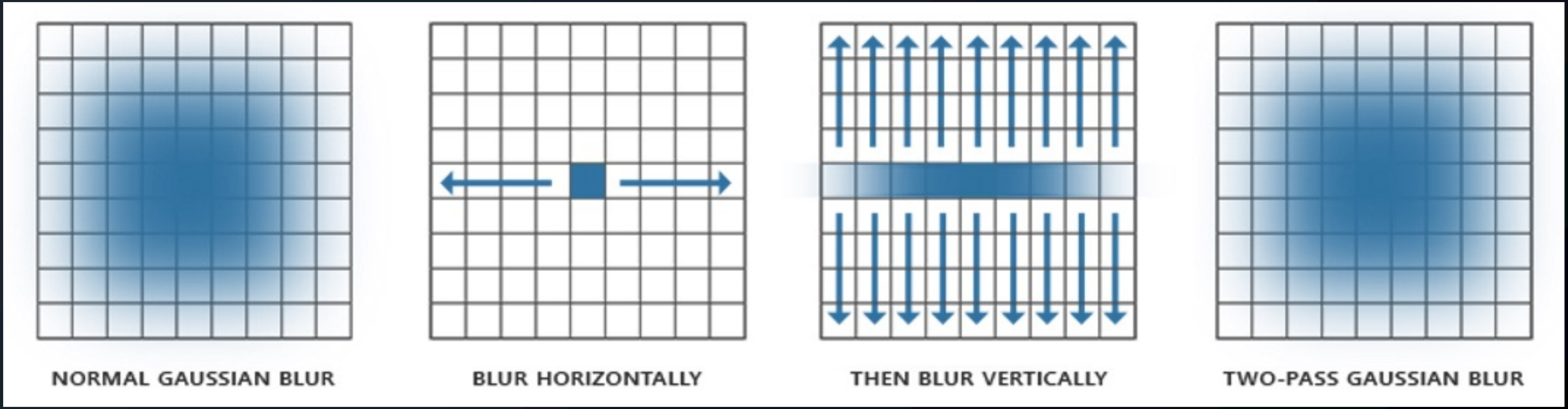
GammaToLinear (External Blit)

Render To Screen

PART TWO

Runtime Optimization

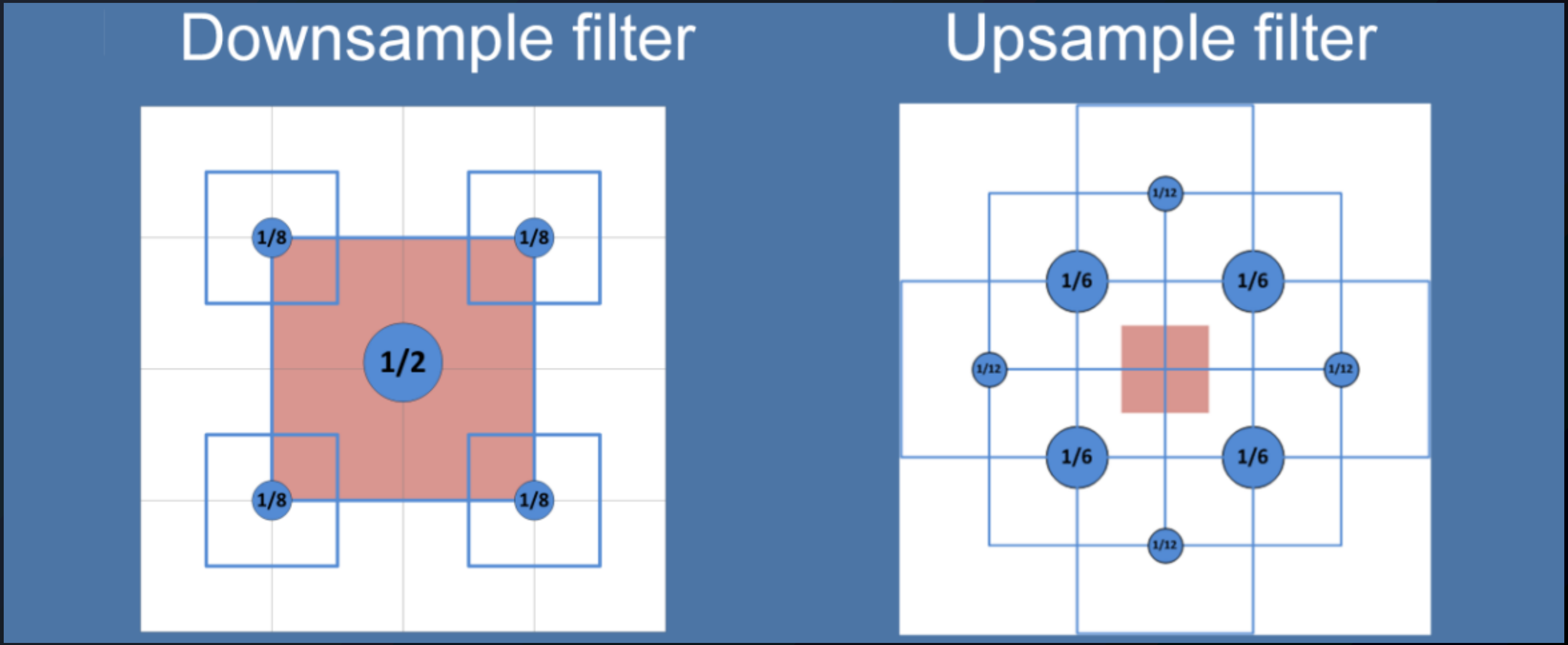
Gaussian Blur



PART TWO

Runtime Optimization

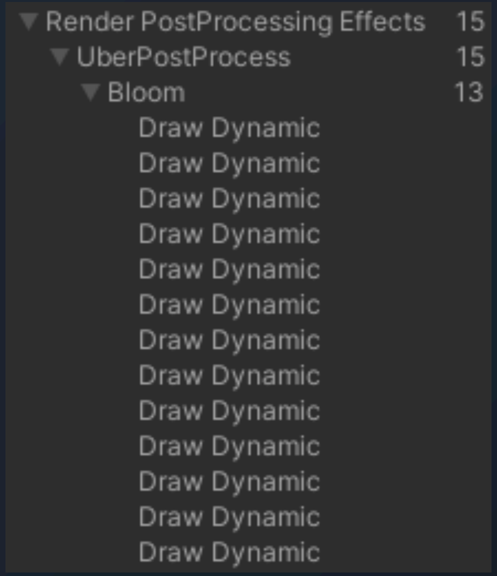
Dual Kawase Blur



https://www.intel.com/content/www/us/en/developer/articles/technical/an-investigation-of-fast-real-time-gpu-based-image-blur-algorithms.html

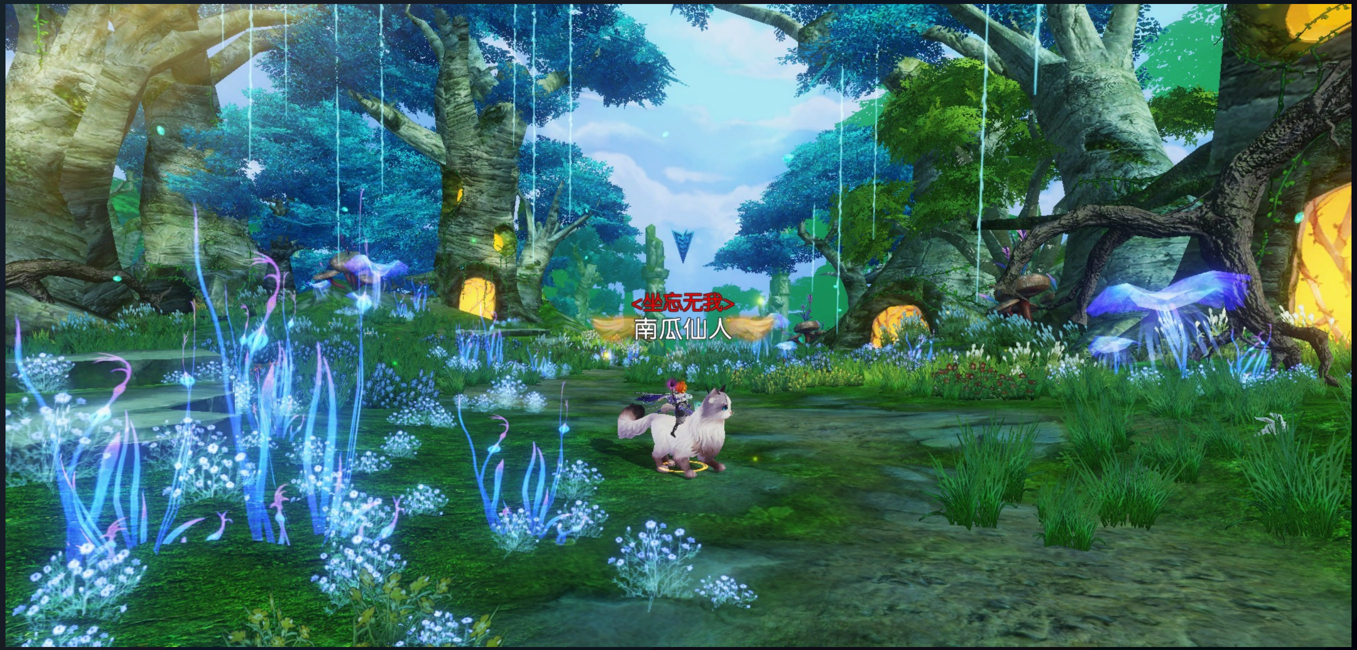
PART TWO

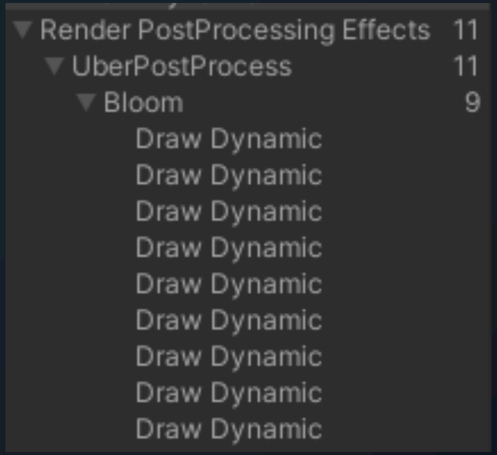
Runtime Optimization



```
▼ Render PostProcessing Effects   15
  ▼ UberPostProcess               15
    ▼ Bloom                       13
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
```

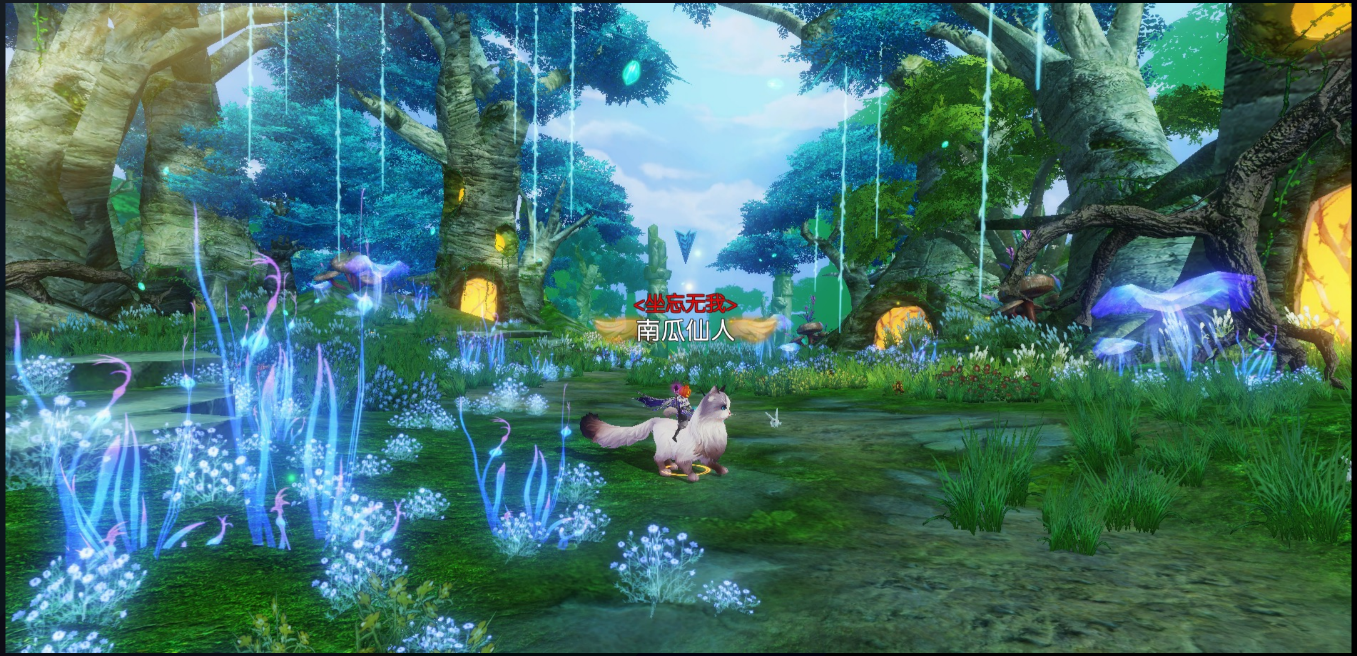
Gaussian





```
▼ Render PostProcessing Effects   11
  ▼ UberPostProcess               11
    ▼ Bloom                        9
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
        Draw Dynamic
```

Dual Kawase



Unity Performance Reporting

PART THREE

Performance

Unity引擎升级：

2018.4 Builtin -> 2019.4 URP

2021.7 ~ 2021.12，共5个月

33个资料片，场景约200个，模型+特效约1w个

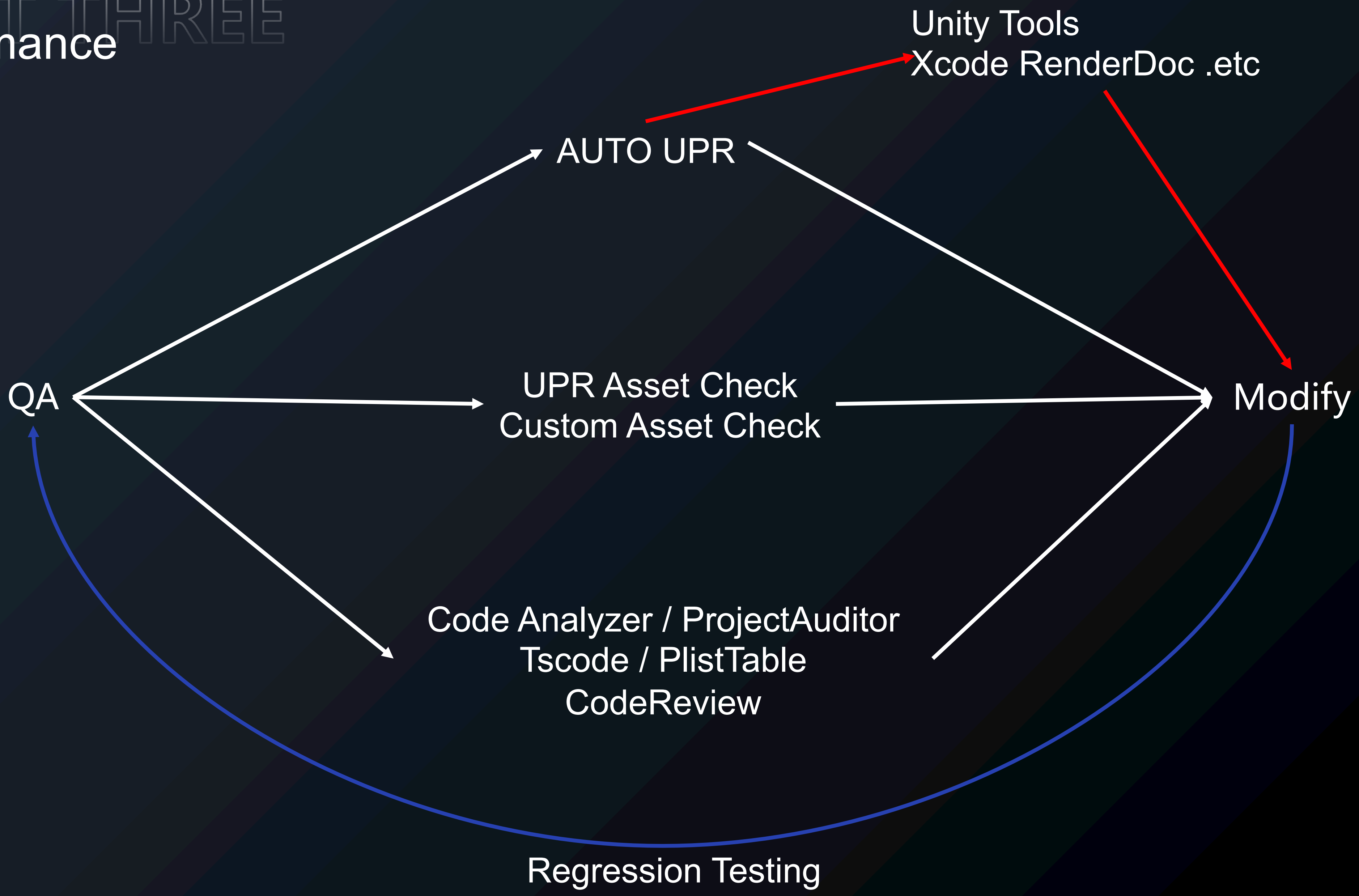
Unity Perfomance Report + 自动化测试

PART THREE

Performance

PART THREE

Performance



Unity Tools
Xcode RenderDoc .etc

AUTO UPR

QA

UPR Asset Check
Custom Asset Check

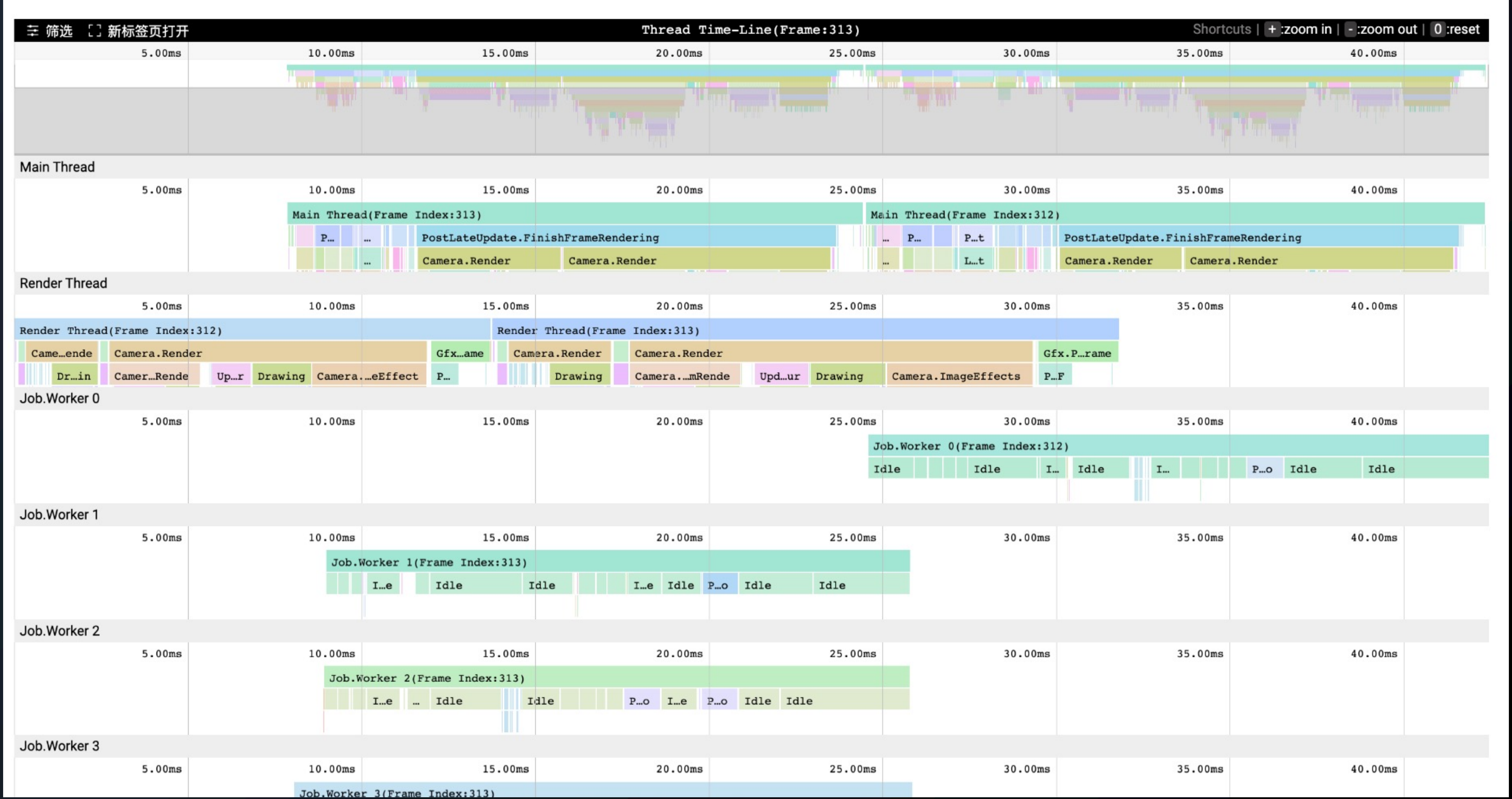
Modify

Code Analyzer / ProjectAuditor
Tscode / PlistTable
CodeReview

Regression Testing

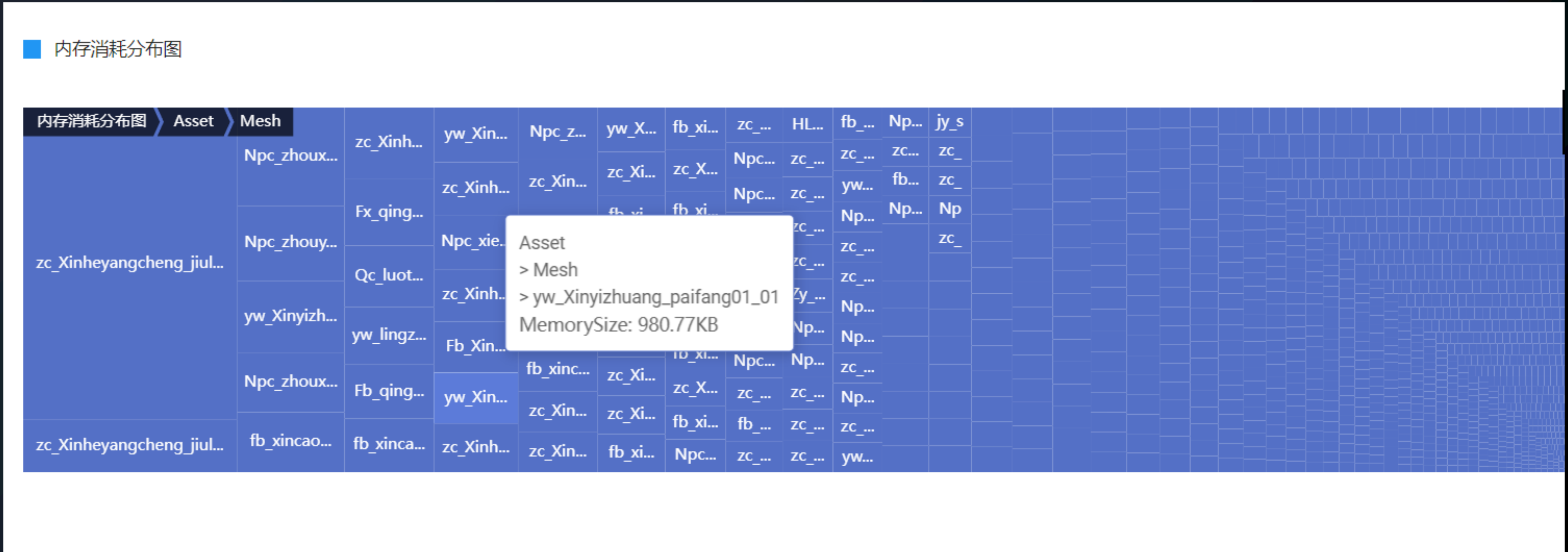
PART THREE

Performance


筛选
新标签页打开
Thread Time-Line(Frame:313)
Shortcuts | + :zoom in | - :zoom out | 0 :reset
5.00ms
10.00ms
15.00ms
20.00ms
25.00ms
30.00ms
35.00ms
40.00ms
Main Thread
Main Thread(Frame Index:313)
Main Thread(Frame Index:312)
PostLateUpdate.FinishFrameRendering
Camera.Render
Render Thread
Render Thread(Frame Index:312)
Render Thread(Frame Index:313)
Came…ende
Camera.Render
Gfx…ame
Gfx.P…rame
Dr…in
Camer…Rende
Up…r
Drawing
Camera.…eEffect
Camera.…mRende
Upd…ur
Camera.ImageEffects
P…F
Job.Worker 0
Job.Worker 0(Frame Index:312)
Idle
P…o
Job.Worker 1
Job.Worker 1(Frame Index:313)
I…e
Job.Worker 2
Job.Worker 2(Frame Index:313)
Job.Worker 3
Job.Worker 3(Frame Index:313)

PART THREE

Performance

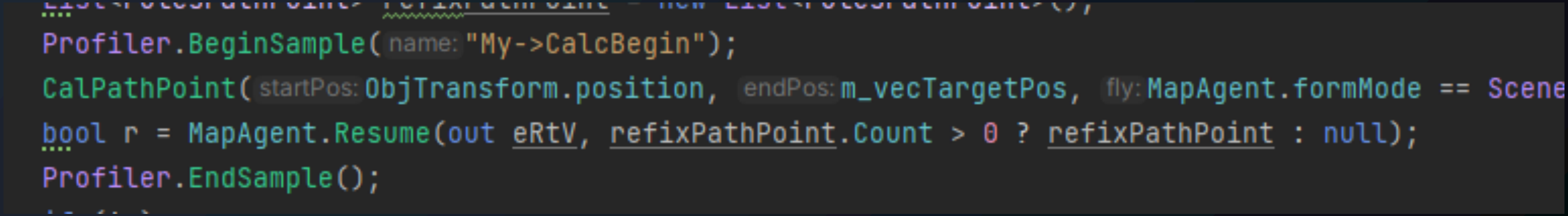


内存消耗分布图

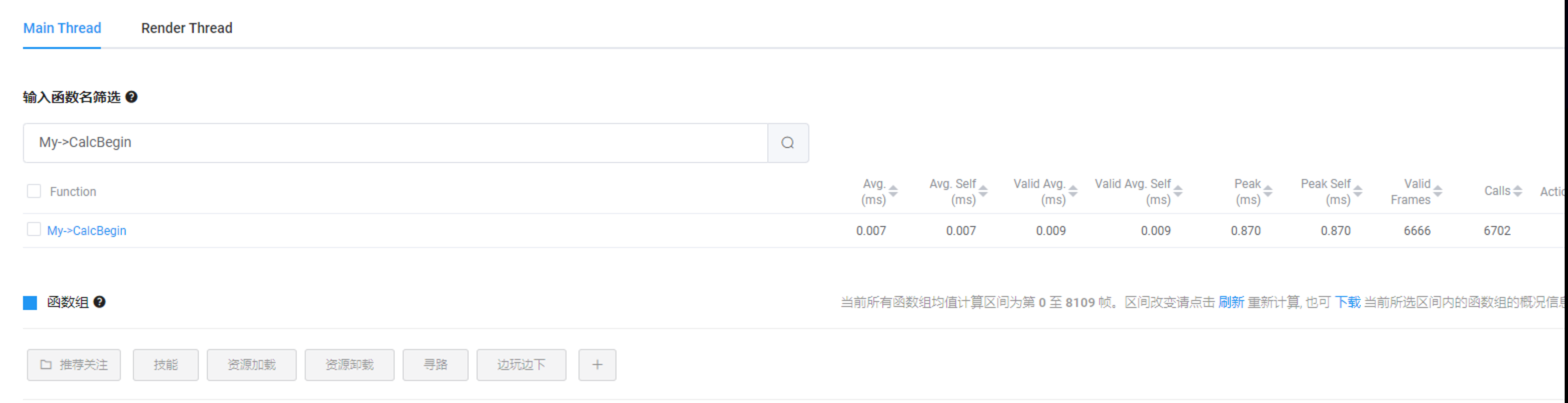
Asset
> Mesh
> yw_Xinyizhuang_paifang01_01
MemorySize: 980.77KB

PART THREE

Performance



```
Profiler.BeginSample(name: "My->CalcBegin");
CalPathPoint(startPos: ObjTransform.position, endPos: m_vecTargetPos, fly: MapAgent.formMode == Scene
bool r = MapAgent.Resume(out eRtV, refixPathPoint.Count > 0 ? refixPathPoint : null);
Profiler.EndSample();
```



输入函数名筛选 ❓

My->CalcBegin

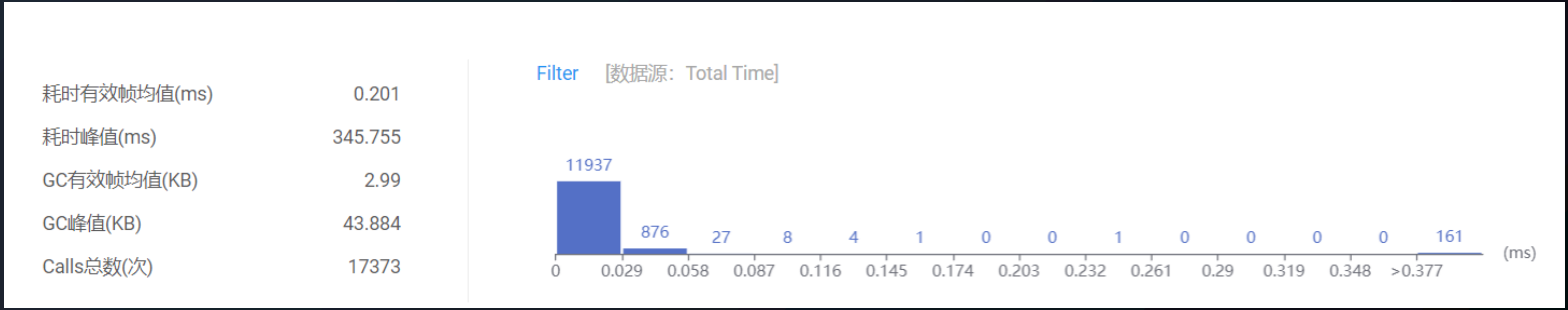
□ Function	Avg. (ms)	Avg. Self (ms)	Valid Avg. (ms)	Valid Avg. Self (ms)	Peak (ms)	Peak Self (ms)	Valid Frames	Calls	Acti
□ My->CalcBegin	0.007	0.007	0.009	0.009	0.870	0.870	6666	6702	

🟦 函数组 ❓

当前所有函数组均值计算区间为第 0 至 8109 帧。区间改变请点击 刷新 重新计算, 也可 下载 当前所选区间内的函数组的概况信

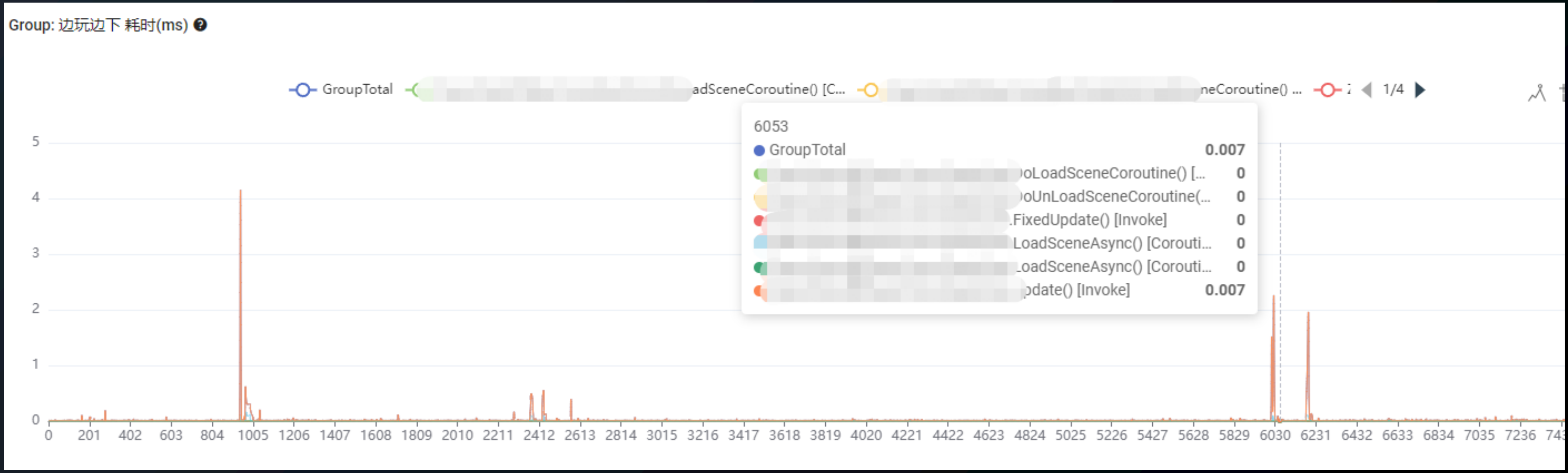
PART THREE

Performance



耗时有效帧均值(ms)	0.201
耗时峰值(ms)	345.755
GC有效帧均值(KB)	2.99
GC峰值(KB)	43.884
Calls总数(次)	17373

Filter [数据源：Total Time]



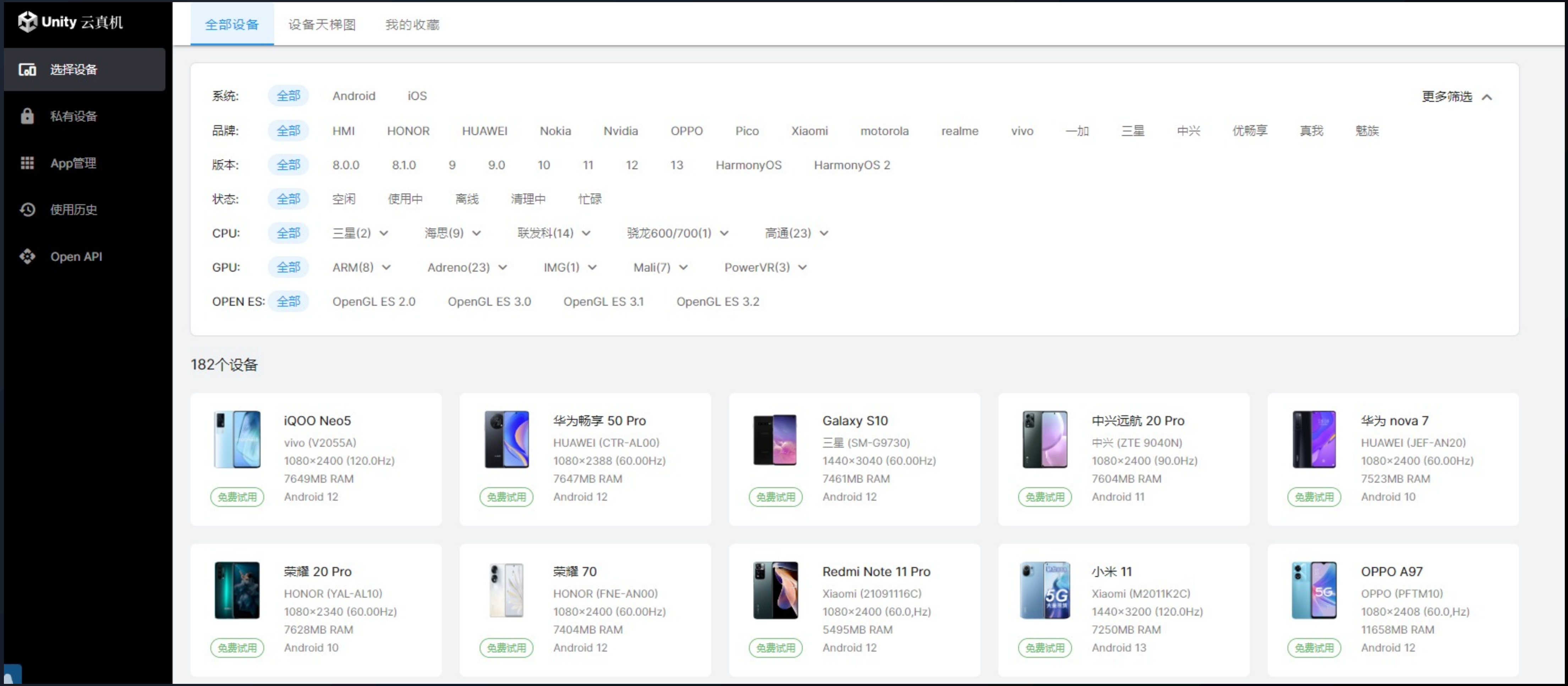
Group: 边玩边下 耗时(ms)

6053

● GroupTotal	0.007
[redacted]oLoadSceneCoroutine() [...	0
[redacted]oUnLoadSceneCoroutine(...	0
[redacted].FixedUpdate() [Invoke]	0
[redacted]LoadSceneAsync() [Corouti...	0
[redacted]LoadSceneAsync() [Corouti...	0
[redacted]pdate() [Invoke]	0.007

PART THREE

Performance



Unity 云真机

- 选择设备
- 私有设备
- App管理
- 使用历史
- Open API

全部设备　设备天梯图　我的收藏

系统: 全部　Android　iOS　　更多筛选

品牌: 全部　HMI　HONOR　HUAWEI　Nokia　Nvidia　OPPO　Pico　Xiaomi　motorola　realme　vivo　一加　三星　中兴　优畅享　真我　魅族

版本: 全部　8.0.0　8.1.0　9　9.0　10　11　12　13　HarmonyOS　HarmonyOS 2

状态: 全部　空闲　使用中　离线　清理中　忙碌

CPU: 全部　三星(2)　海思(9)　联发科(14)　骁龙600/700(1)　高通(23)

GPU: 全部　ARM(8)　Adreno(23)　IMG(1)　Mali(7)　PowerVR(3)

OPEN ES: 全部　OpenGL ES 2.0　OpenGL ES 3.0　OpenGL ES 3.1　OpenGL ES 3.2

182个设备

iQOO Neo5
vivo (V2055A)
1080×2400 (120.0Hz)
7649MB RAM
Android 12
免费试用

华为畅享 50 Pro
HUAWEI (CTR-AL00)
1080×2388 (60.00Hz)
7647MB RAM
Android 12
免费试用

Galaxy S10
三星 (SM-G9730)
1440×3040 (60.00Hz)
7461MB RAM
Android 12
免费试用

中兴远航 20 Pro
中兴 (ZTE 9040N)
1080×2400 (90.0Hz)
7604MB RAM
Android 11
免费试用

华为 nova 7
HUAWEI (JEF-AN20)
1080×2400 (60.00Hz)
7523MB RAM
Android 10
免费试用

荣耀 20 Pro
HONOR (YAL-AL10)
1080×2340 (60.00Hz)
7628MB RAM
Android 10
免费试用

荣耀 70
HONOR (FNE-AN00)
1080×2400 (60.00Hz)
7404MB RAM
Android 12
免费试用

Redmi Note 11 Pro
Xiaomi (21091116C)
1080×2400 (60.0,Hz)
5495MB RAM
Android 12
免费试用

小米 11
Xiaomi (M2011K2C)
1440×3200 (120.0Hz)
7250MB RAM
Android 13
免费试用

OPPO A97
OPPO (PFTM10)
1080×2408 (60.0,Hz)
11658MB RAM
Android 12
免费试用

THANK YOU

THANK YOU