

# 引擎源码及管线深度定制

《航海王热血航线》

朝夕光年 无双工作室 陈到



无双工作室  
—HEROES • STUDIO—



朝夕光年  
NUVERSE



无双工作室  
HEROES · STUDIO

# 目录

C O N T E N T S

01 简介

02 引擎源码定制

03 管线深度定制

03 优化及工具经验

# 简单介绍

## • 航海王 热血航线

- 3D ARPG
- 朝夕光年与中手游联合出品
- 登顶了iOS免费榜，畅销榜最高Top 3

## • 无双工作室

- 朝夕光年上海游戏工作室
- ACT、卡牌品类
- 欢迎大家加入，引擎、TA、客户端岗位丰富





# 源码自动构建

Unity源码编... 流水线 > Unity2018.4.13源码构建 > 执行预览流水线

参数配置: ^

引入参数模板

TYPE: \* release  
folder: \* D:\res\unity2018.4.13-build  
branch: \* master

component: \* windows,android,ios  
ClearEditorCache: \* false  
GameVersion: \* EP\_1

MajorVersion: 1  
MinorVersion: 0  
FixVersion: 0

主版本 (MajorVersion) 特性版本 (MinorVersion) 修正版本 (FixVersion)

选择可执行的插件: (若你的流水线已经调试成功, 可在手动触发插件中关闭该选项)  全选/全不选

```
graph LR; S1[1-1 构建触发] --> S2_1[2-1 构建环境-WINDOWS]; S1 --> S2_2[2-2 构建环境-MACOS]; S2_1 --> S3_1[3-1 构建apk进行测试]; S2_2 --> S3_1; S3_1 --> S4_1[4-1 windows提交]; S4_1 --> S5_1[5-1 Mac提交]; S5_1 --> S6_1[6-1 构建环境-WINDOWS];
```

1-1 构建触发

- 手动触发

2-1 构建环境-WINDOWS

- 拉取Gitlab仓库代码
- 构建
- Windows通知

2-2 构建环境-MACOS

- 拉取Gitlab仓库代码
- 构建
- Mac构建通知

3-1 构建apk进行测试

- 拉取demo源码
- 构建apk
- 归档构件
- 云测试插件
- 飞书通知插件

4-1 windows提交

- 拷贝上传

5-1 Mac提交

- 拷贝上传

6-1 构建环境-WINDOWS

- 飞书通知插件

# 自制启动器

WSUnityLauncher

配置 工具 关于

项目

[使用说明](#) [使用反馈](#)

项目名称	Unity版本	路径
hctest	EP2	D:\
client	官方版本	E:\
client	官方版本	E:\
OnePiece	EP3	D:\
RunTestUnityProj2018	官方版本	E:\
New Unity Project	官方版本	D:\
testCrunch2018	官方版本	E:\
New Unity Project	官方版本	D:\
OnePiece	官方版本	D:\
New Unity Project	官方版本	C:\
client	官方版本	D:\
client	官方版本	D:\
testab	官方版本	E:\
OnePiece	官方版本	E:\

```
C:\unity\launcher\unity\2018.4.13-master\AndroidPlayer\Variations\il2cpp\Development\Libs\x86\libunity.so
U
C:\unity\launcher\unity\2018.4.13-master\AndroidPlayer\Variations\il2cpp\Development\Symbols\arm64-v8a\libunity.sym.so
U
C:\unity\launcher\unity\2018.4.13-master\AndroidPlayer\Variations\il2cpp\Development\Symbols\armeabi-v7a\libunity.sym.so
U
C:\unity\launcher\unity\2018.4.13-master\AndroidPlayer\Variations\il2cpp\Development\Symbols\x86\libunity.sym.so
U
C:\unity\launcher\unity\2018.4.13-master\AndroidPlayer\Variations\mono\Development\Classes\classes.jar
U
C:\unity\launcher\unity\2018.4.13-master\AndroidPlayer\Variations\mono\Development\Libs\armeabi-v7a\libunity.so
U
C:\unity\launcher\unity\2018.4.13-master\AndroidPlayer\Variations\mono\Development\Symbols\armeabi-v7a\libunity.sym.so
Updated to revision 73724.
Updating 'C:\unity\launcher\unity\2018.4.13-master\iOSSupport':
U
C:\unity\launcher\unity\2018.4.13-master\iOSSupport\Trampoline\Libraries\libiPhone-lib-dev.a
U
C:\unity\launcher\unity\2018.4.13-master\iOSSupport\Trampoline\Libraries\libiPhone-lib-il2cpp-dev.a
U
```

# 更新发布机制

结合飞书加强版本管理能力

Hgame

unity版本发布

unity分支

unity版本

分支版本描述

可选操作  发送通知  强制更新

登录器版本发布

登录器版本

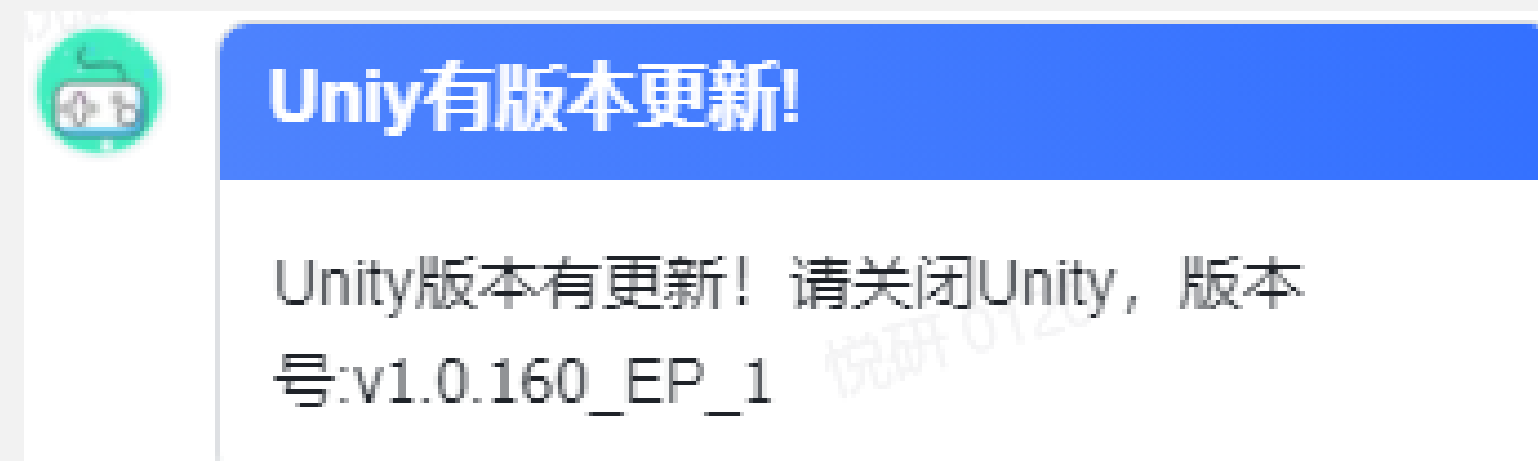
版本描述

可选操作  发送通知

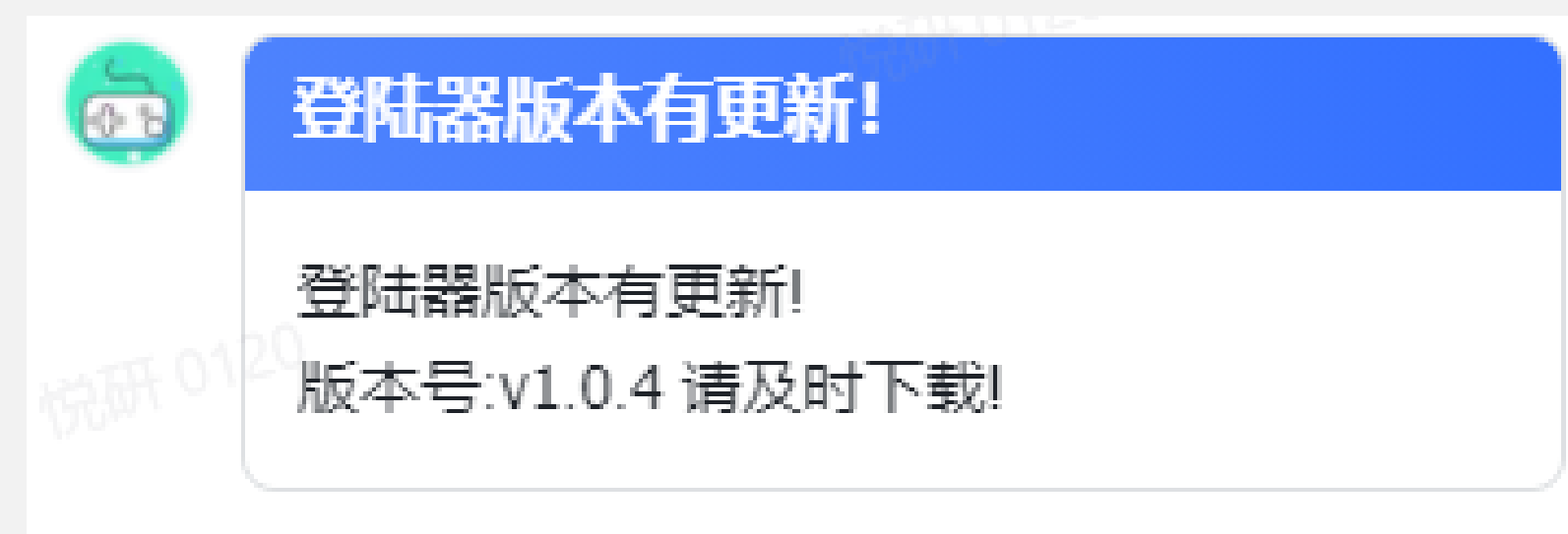
成员信息

用户	使用次数
yueyan.erik@bytedance.com	93
wangl.09@bytedance.com	1
zhangdong.04@bytedance.com	5
lijun.cs21@bytedance.com	12

## Unity发布



## Launcher发布



# 引擎源码定制流程

- 1 源码构建流程
- 2 引擎发布流程
- 3 崩溃上报流程
- 4 源码修改机制
- 5 源码阅读及调试小组

SetPosition 和 SetRotation 都会走一遍 SendMessage 流程，没有必要。 ←

里面提供了 SetPositionAndRotation 方法： ←

```
Transform.h X Collider.cpp Collider.h BoxCollider.h BoxCollider.cpp Camera.cpp MessageHandler.cpp GameObject.h Canvas.cpp Mess
Transform class EXPORT_COREMODULE Transform : public Unity::Component
134 void SetWorldRotationAndScale (const Matrix3x3f& worldRotationAndScale);
135
136 // Sets the world position and rotation
137 void SetPositionAndRotation (const Vector3f& position, const Quaternionf& rotation);
138 void SetLocalPositionAndRotation (const Vector3f& position, const Quaternionf& rotation);
139
140 // Return matrix that converts a point from World To Local space
141 Matrix4x4f GetWorldToLocalMatrix () const;
142 // Return matrix that converts a point from Local To World space
143 Matrix4x4f GetLocalToWorldMatrix () const;
144
145 Matrix4x4f GetWorldToLocalMatrixNoScale () const;
146 const Matrix4x4f& GetWorldToLocalMatrixNoScale (Matrix4x4f& m) const;
147 Matrix4x4f GetLocalToWorldMatrixNoScale () const;
```

但是没有对外暴露！ ←

对外暴露之后，开销有望下降接近 50%。 ←



# 定制开发内容

## 基于Cache Server，优化工程效率



- 工程庞大
- 资源导入慢，经常Hold On
- 杀毒软件拖慢速度
- 网关不稳定，cache server总断线

## Cache Server



- 主动弹出连接断开提示，用户控制
- 只处理贴图转换和动画转换
- 配置黑名单，屏蔽部分目录，提高转换速度

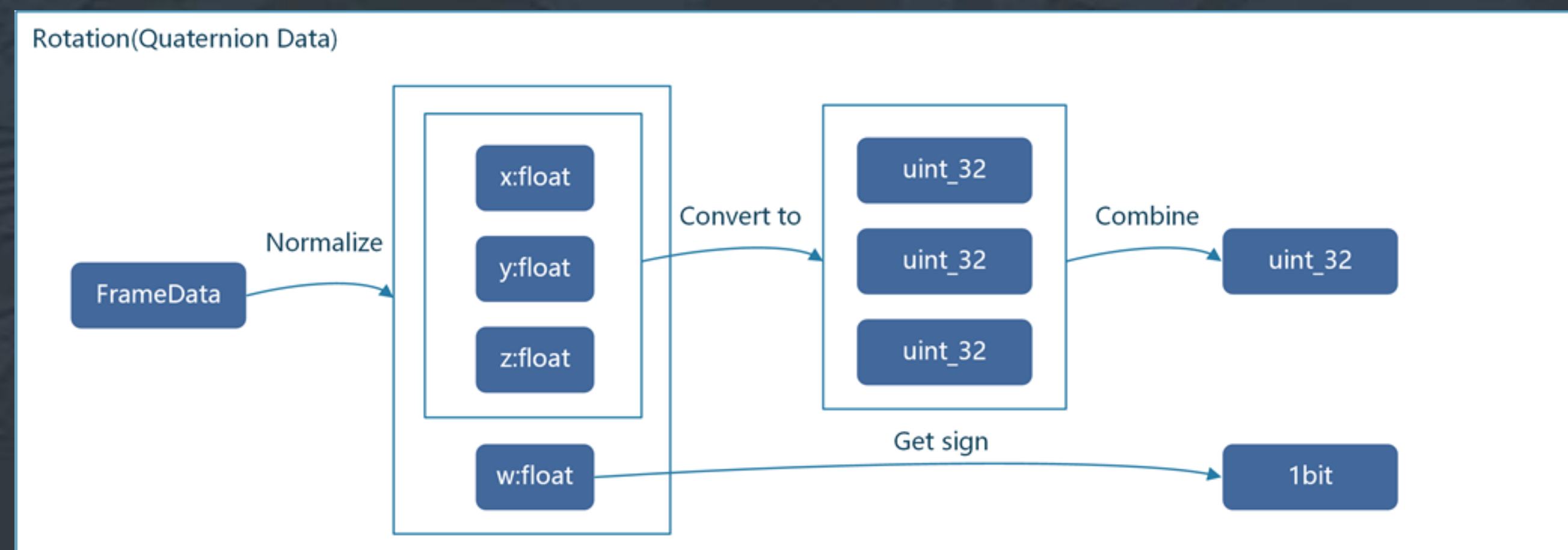
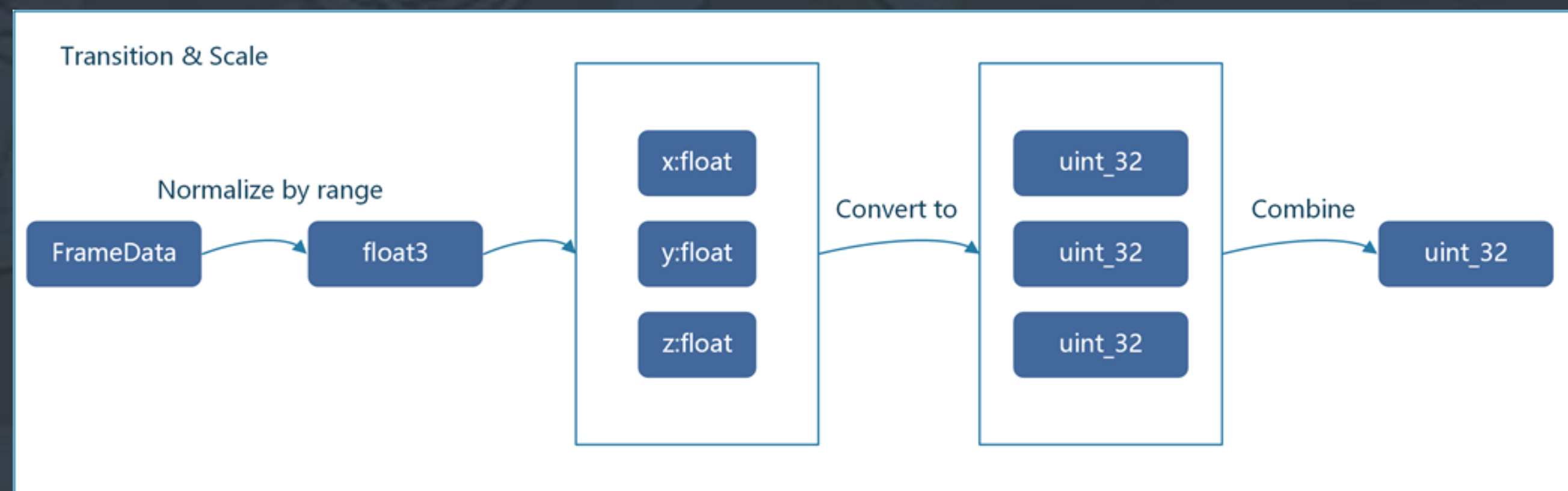


- 工程打开速度提高1倍。
- 工程崩溃等情况下，不会再进行无谓的全量导入
- 网络波动下，Unity不会在傻傻等待cache server连接

# 定制开发内容

## 针对ACT游戏的动画压缩

- 移植引入ACL (Animation Compression Library)  
GDC2017 <http://nfrechette.github.io/>
- 减少采样帧
- 减少数据精度
- 优化的同时还能保证动画质量尽量小的失真
- 误差预测
- 降低采样率, 尝试丢弃的采样帧采用插值或者曲线拟合算出。
- 基本的数据压缩



# 定制开发内容

## 其他

- 引擎中的IBO三角形排序解决AlphaBlend的问题
- 部分计算C++化（动态骨骼、寻路等）
- 新引擎版本的feature、Bugfix的merge
- ASTC升级：替换为新版本的ASTC
- 引入定制服务方案（实例化多线程方案、C++层AB加密）



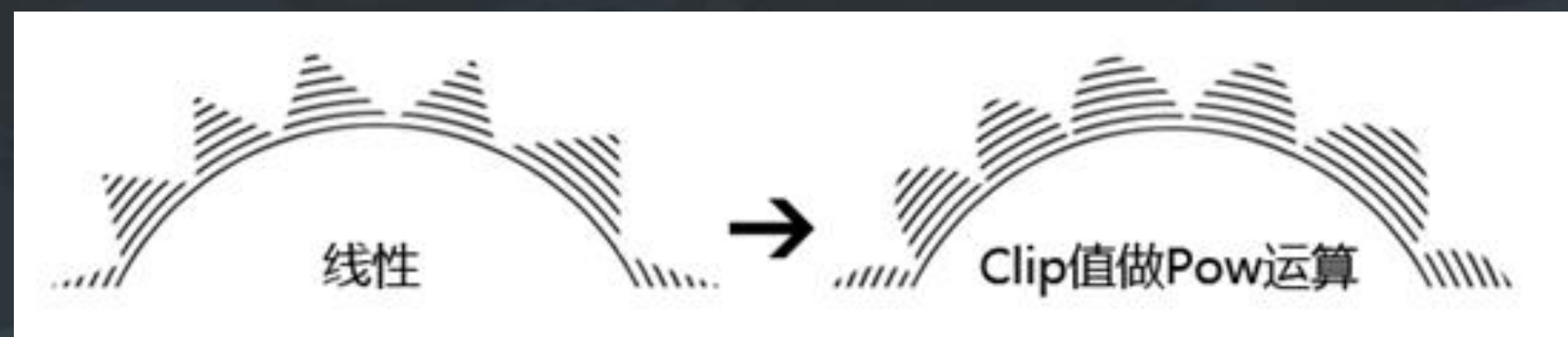
# 渲染定制

## 海浪效果

- 多层Gerstner Wave叠加
- 多种反射技术结合
  - Reflection Probe进行BoxProjection  
Blend差值
  - Planar Reflection的平面反射技术
  - 屏幕空间反射SSR



# 渲染定制



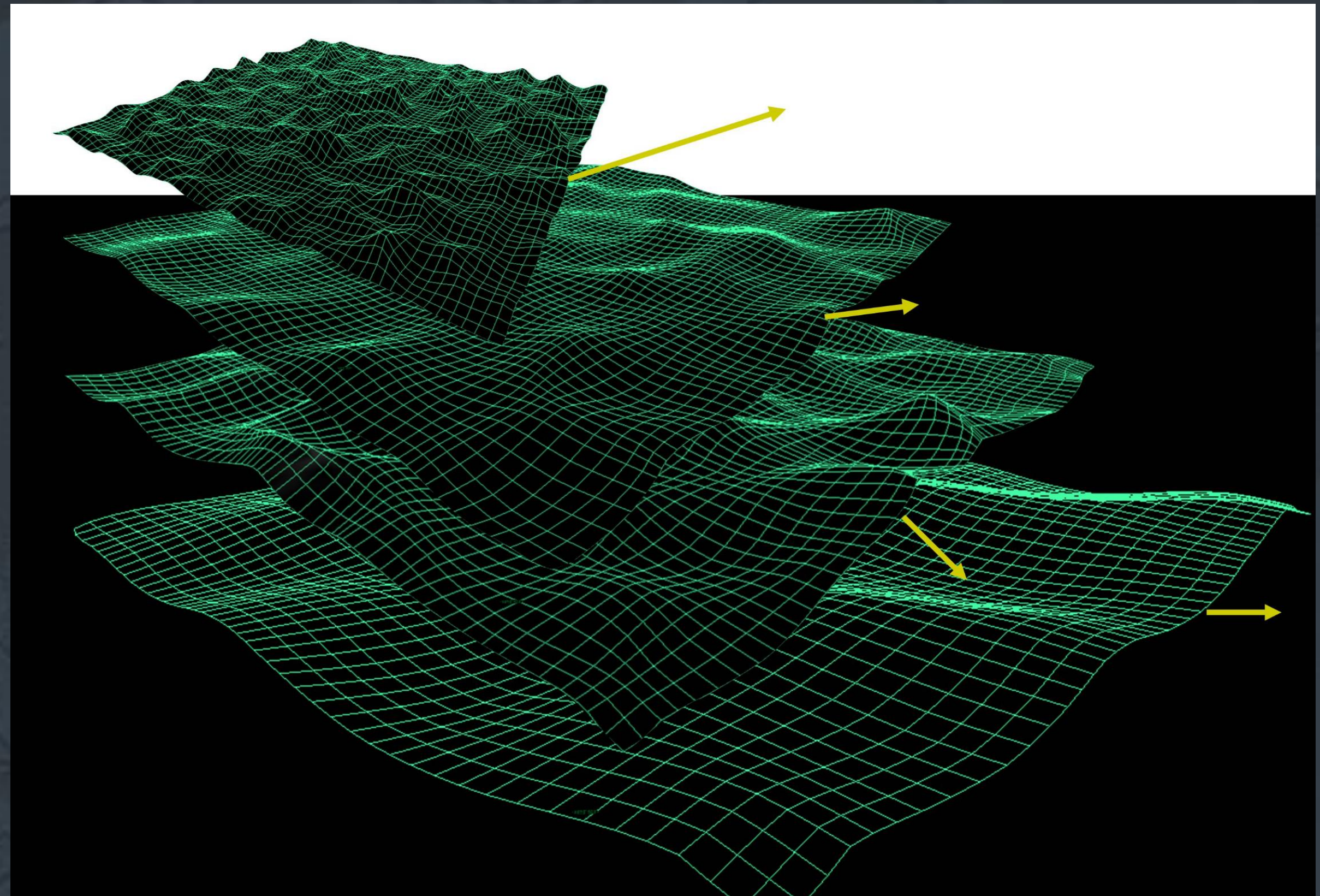
## 岛的云海效果

- Volume Cloud、
- Raymarching
- MultiPass Parallax Mapping
- 层渲染
- GPU instancing代替多pass
- 高端depth prepass



# 海水优化

- 动态超大范围、覆盖整个世界的海水
- 四叉树加CDLOD的动态网格



# 渲染优化

## 灯光数量\*物件数量的压力

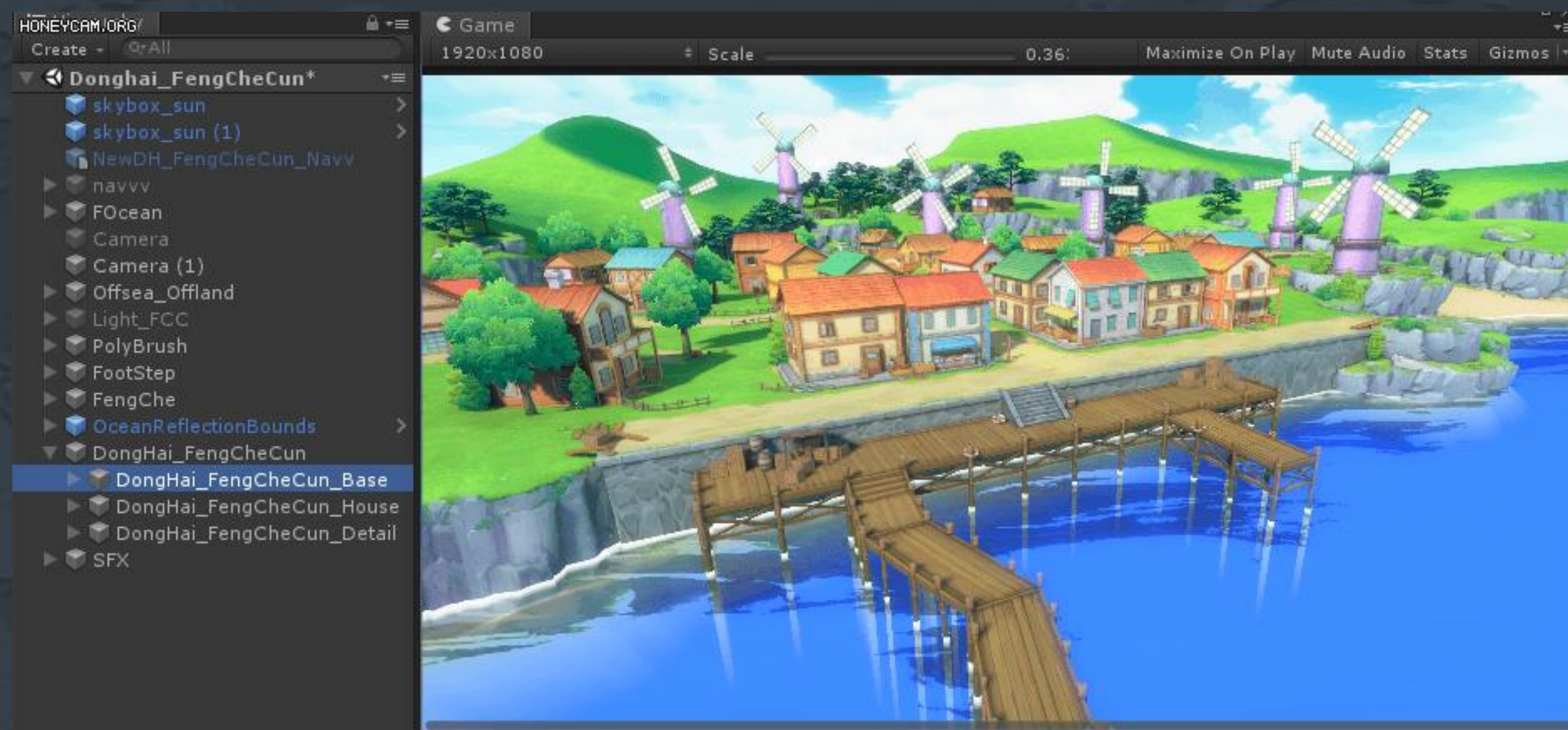
- 灯光信息组织到一个CBUFFER中，一个drawcallpass完成所有光源绘制
- 统一处理CPU设置工作，减少渲染设置开销
- 用专用的代码将引擎的属性（比如objects transform）直接放入GPU显存，避免了逐个材质去设置GPU的CBUFFER

```
45
46 half4 _AdditionalLightsCount;
47 #if USE_STRUCTURED_BUFFER_FOR_LIGHT_DATA
48 StructuredBuffer<LightData> _AdditionalLightsBuffer;
49 StructuredBuffer<int> _AdditionalLightsIndices;
50 #else
51 // GLES3 causes a performance regression in some devices when using CBUFFER.
52 #ifndef SHADER_API_GLES3
53 CBUFFER_START(AdditionalLights)
54 #endif
55 float4 _AdditionalLightsPosition[MAX_VISIBLE_LIGHTS];
56 half4 _AdditionalLightsColor[MAX_VISIBLE_LIGHTS];
57 half4 _AdditionalLightsAttenuation[MAX_VISIBLE_LIGHTS];
58 half4 _AdditionalLightsSpotDir[MAX_VISIBLE_LIGHTS];
59 half4 _AdditionalLightsOcclusionProbes[MAX_VISIBLE_LIGHTS];
60 #ifndef SHADER_API_GLES3
61 CBUFFER_END
62 #endif
63 #endif
```

# 渲染优化

## 自定义渲染

- 自定义的级联阴影Pass
- 海水的PreDepth Pass
- 其他





# 渲染优化

- 动态开关部分pass
  - Motion Blur
  - 海水反射区域
  - Distortion
  - Radial Blur等
- 关闭Stencil Test
  - UI方面的Mask功能失效
  - 针对性的实现一套UIMask功能
- 使用Memoryless模式、降内存带宽
- 不同Adreno, Mali, PowerVR等性能做MSAA适配
- 机型和效果的分级策略

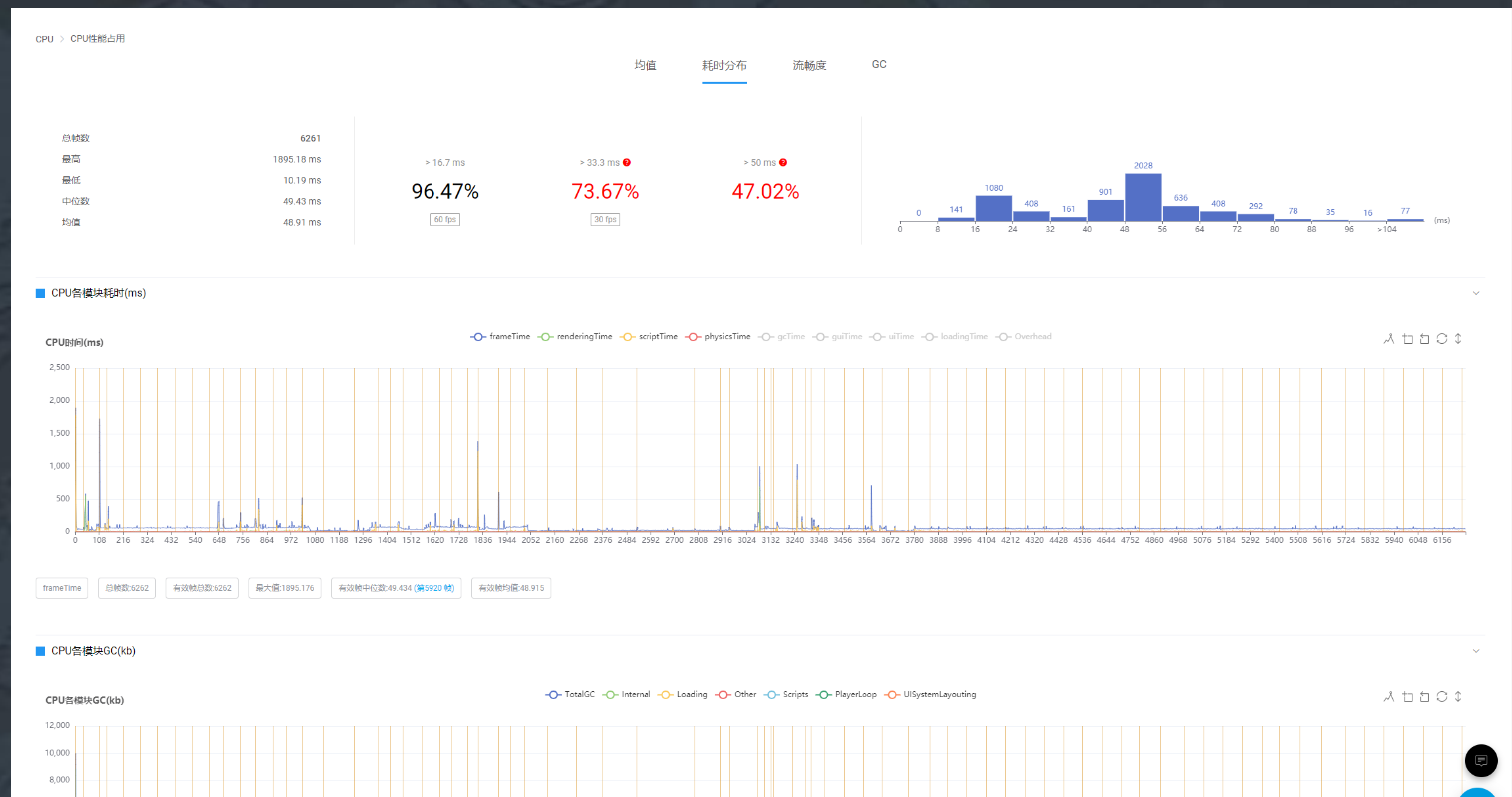
# 性能优化

## ● 开箱即用一站式工具套件UPR

- 运行时检测
- 静态检测
- 企业部署

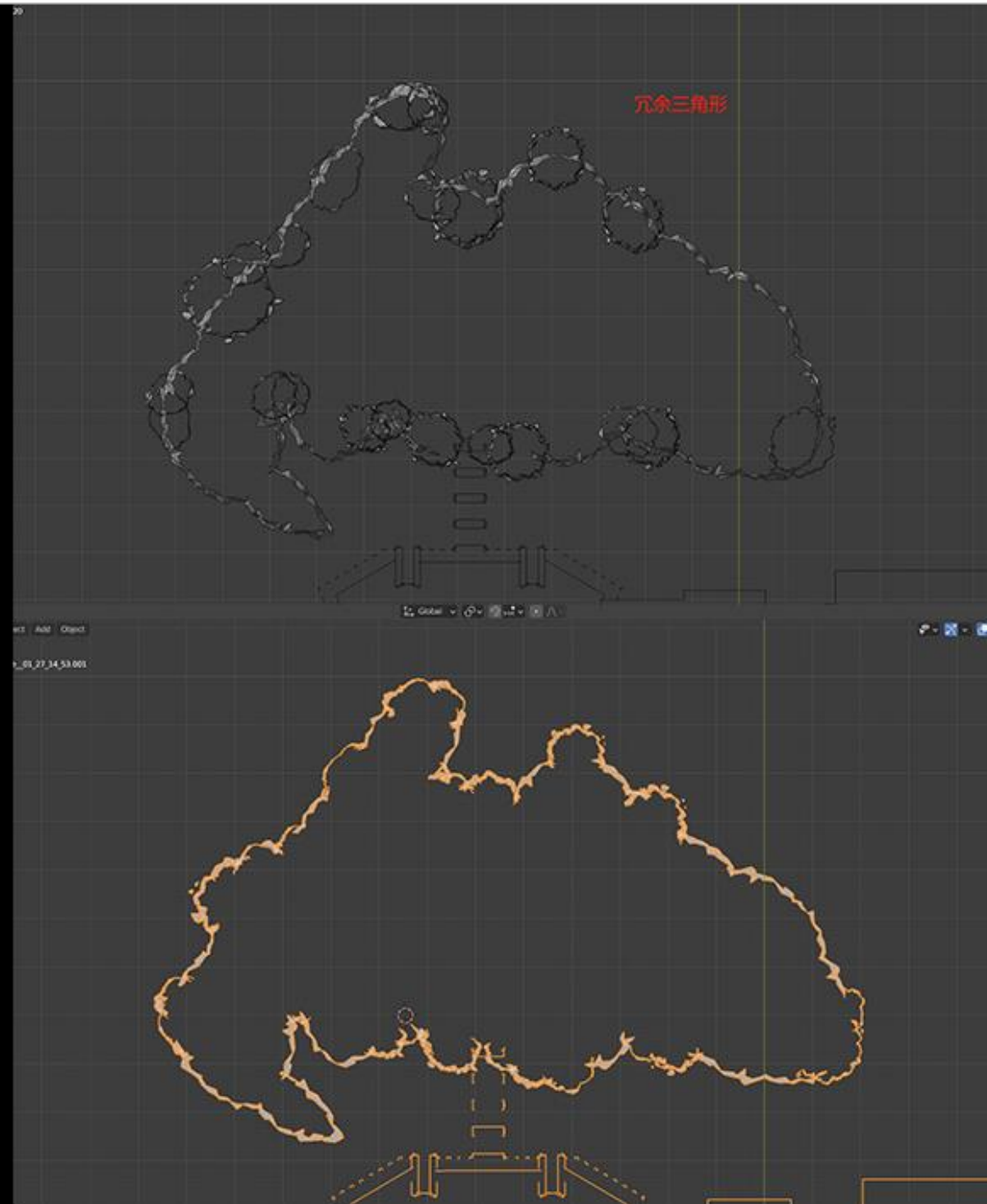
## ● Desktop CLI搭配Open API, UPR集成到了自动化测试流程

- QA即可消化很多分析数据
- 甚至直接给出优化建议



# 海岸线优化工具

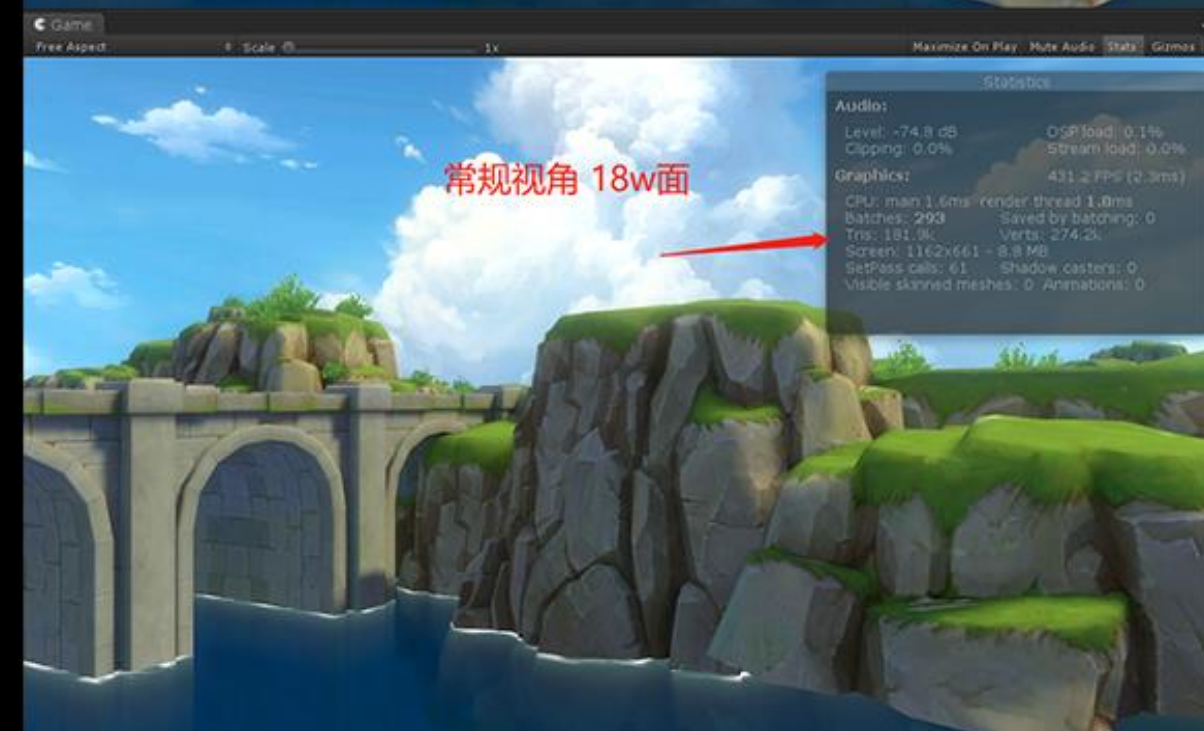
## 大型场景性能优化



海岸线优化工具



优化前: 1.4 M tris 优化后: 0.93 M tris



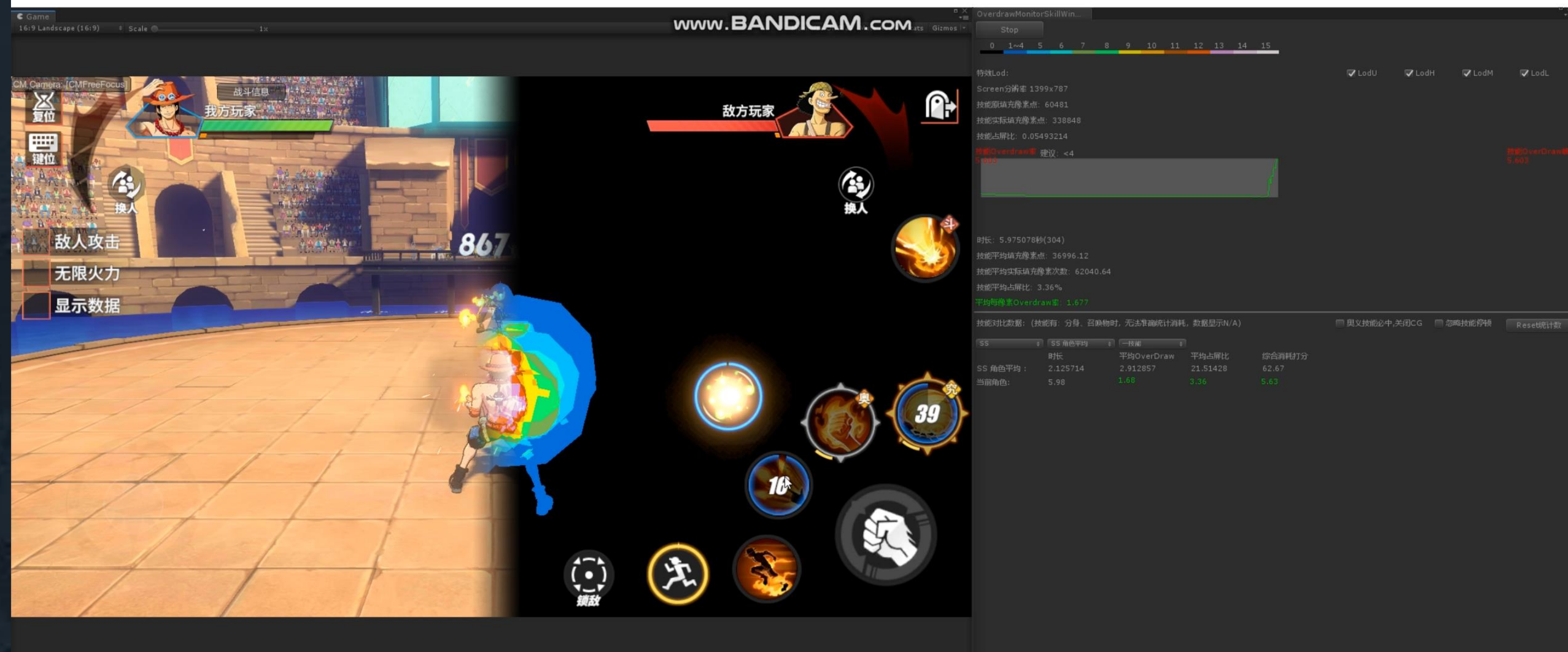
优化前: 22.5 W tris 优化后: 18 W tris

# 特效性能监测工具

## 以现有的合格资源建立初步参考标准

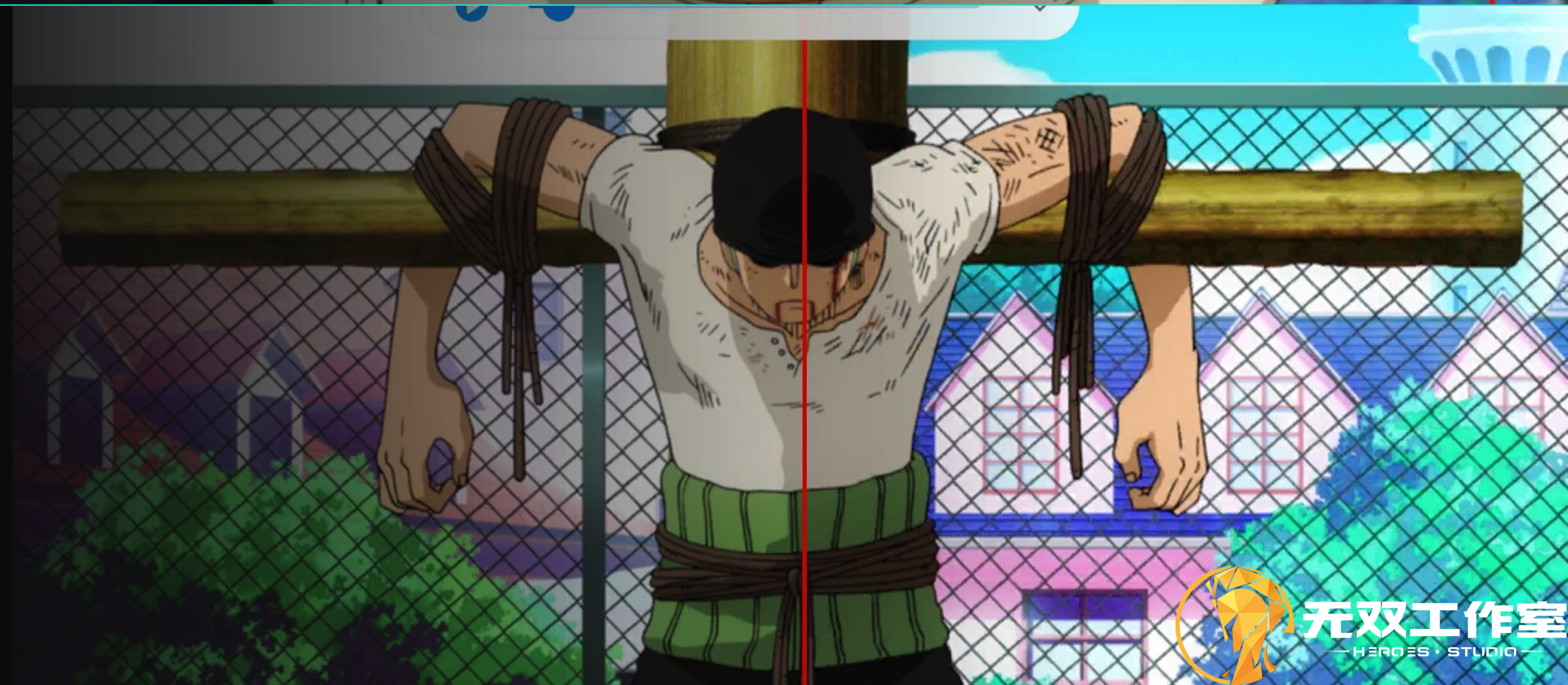
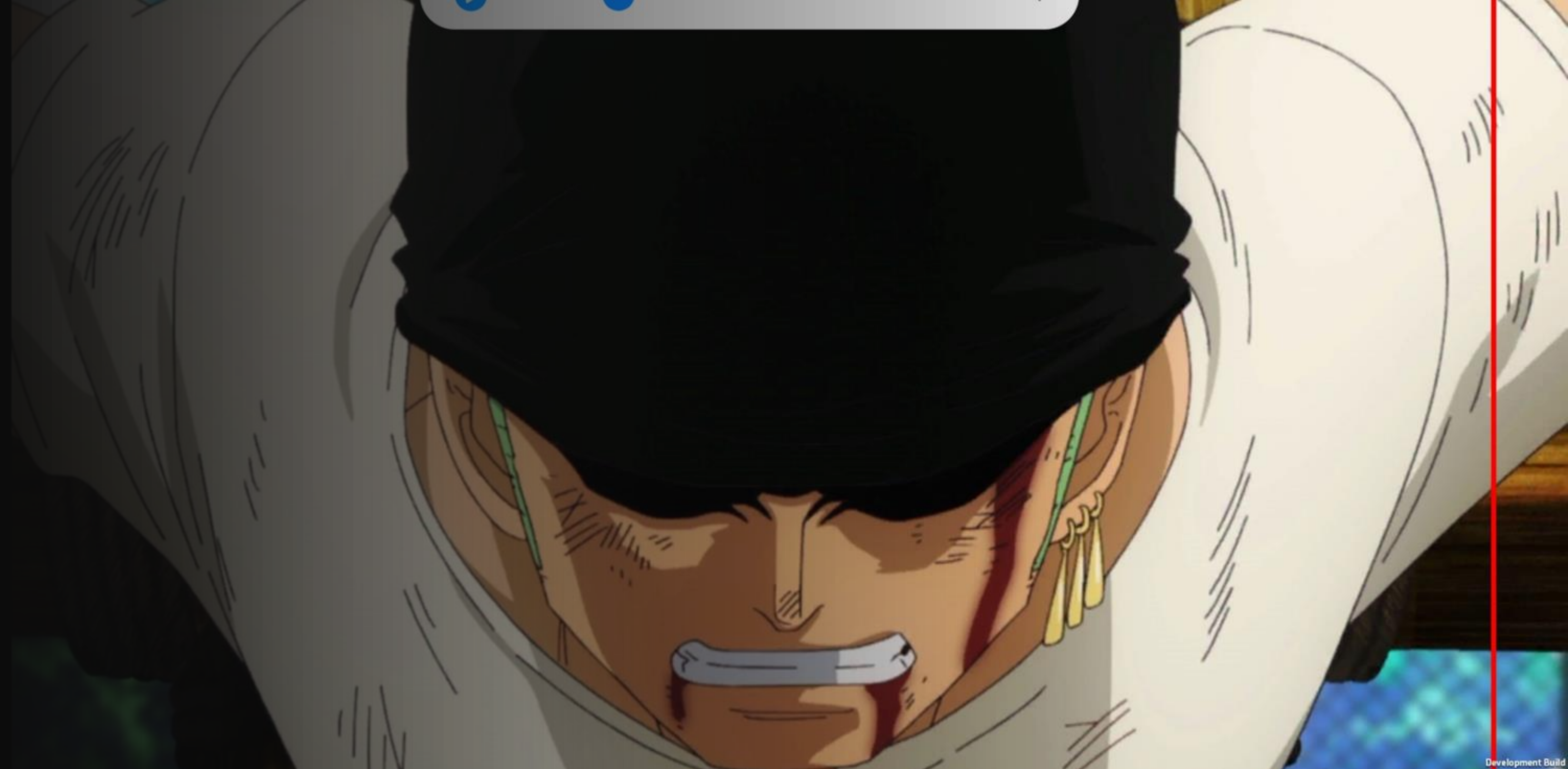
- 空像素数：当前帧被绘制0次的像素数量
- 像素填充次数：当前帧所有像素被绘制的次数之和
- OverDraw倍率：当前帧像素填充次数/当前帧实际渲染像素
- OverDraw倍率峰值：
- 其他参数指标——计算公式
- 实时预览、离线批量报告

## 特效性能监测工具

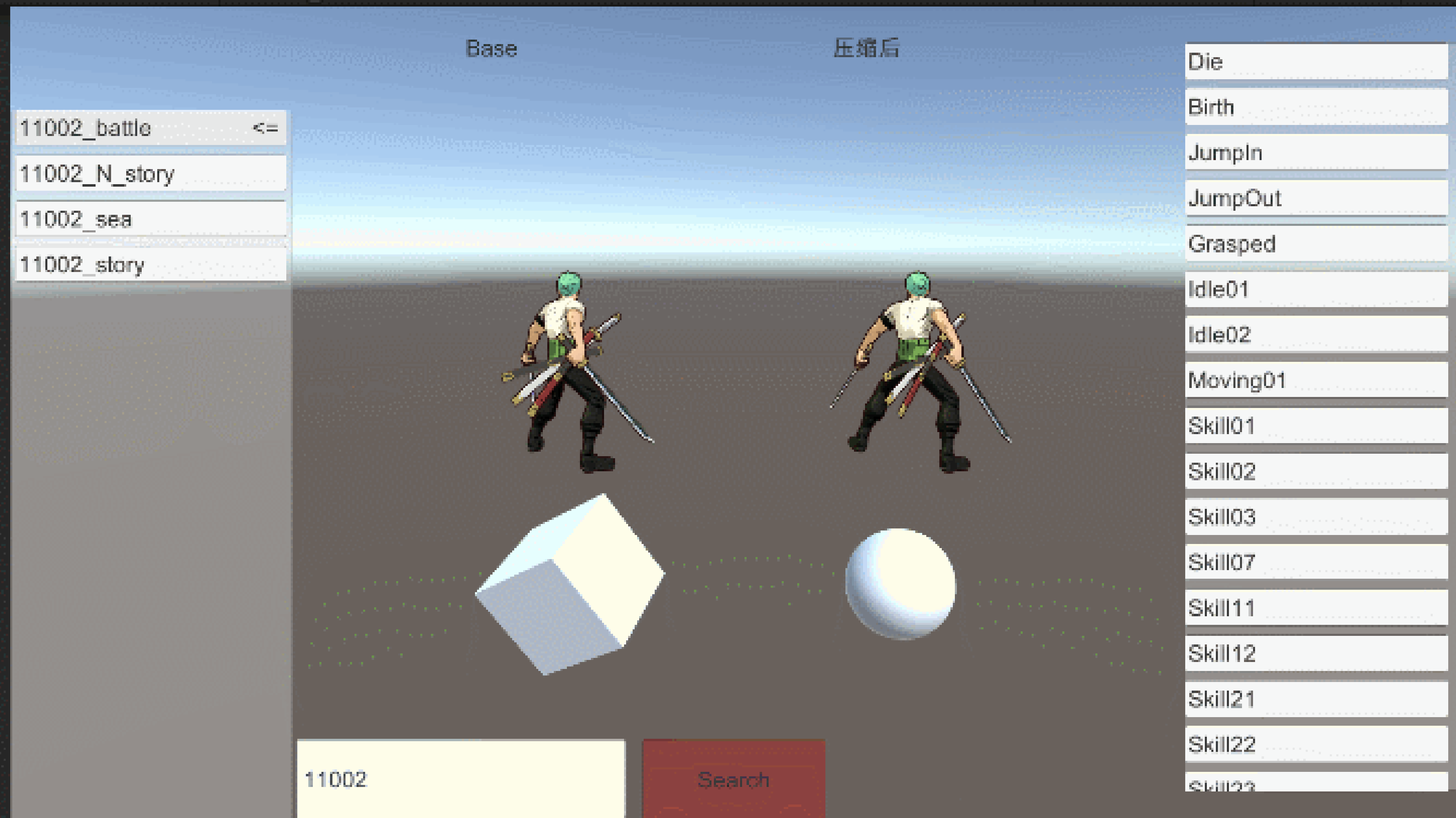


动态性能监测工具，特效制作过程中实时反馈性能状态，性能与设计同时兼顾，优化前置

# 贴图压缩对比工具



# 动画压缩对比工具





技术交流



该二维码7天内(8月5日前)有效, 重新进入将更新

欢迎做客交流

# 谢谢



无双工作室  
—HEROES · STUDIO—



朝夕光年  
NUVERSE