太空科幻SLG手游的渲染与优化

邝圣凯 陈石

Agenda

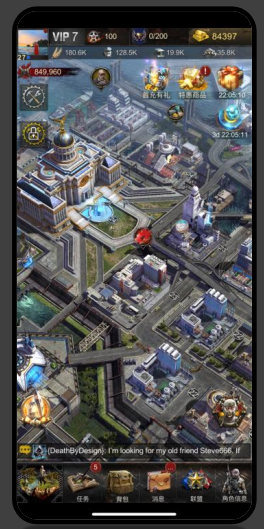
- 联网SLG游戏介绍及一般优化策略
- Infinite Galaxy项目介绍
- 项目整体优化策略分析
- 专题：Volume Lights
- 专题：Face Sync Solution

SLG游戏介绍

- 关键词：模拟、策略、群体协作/对抗
- 前期个人养成
- 中后期基于联盟的多人策略对抗：探索、扩张、采集、竞争

个人养成：基地

- 包装形式：城堡、城市、基地、控制室……
- 前期个人养成
- 单机向，建造+城内资源生产
- 单机向PvE玩法



群体交互：世界地图

- 基于坐标的世界地图
- 上帝视角：
  - 玩家基地
  - 联盟领地
  - 资源点、特殊建筑
  - 活动、事件点
  - 行军队列

高层世界地图（缩略视图）

- 宏观层面地域呈现
- 快速浏览、跳转
- 宏观结构逻辑

优化分析：基地

- 建造养成及各系统的功能入口
- （广角）中景或远景为主，宏观视角
- 静态对象为主：建筑、地标、功能对象、操作台……
- 功能对象数量可能较多
- 配合动态修饰性对象：行走的角色、动物、载具……
- 无复杂的计算逻辑或动态渲染需求

优化策略：基地

- 宏观视角，视觉以突出全局氛围为主
- 结合相对固定镜头角度优化
- 动静结合，注意配比及平衡
- 建造类对象数量堆积较多时考虑低端设备优化
- 注意建筑附属UI（名称、进度条、收获图标等）的堆叠效率

优化分析：世界地图

- 多玩家呈现、交互的核心场景
- 远景为主
  - 静态对象：基地、资源点……
  - 动态对象：野怪、行军队列、地图战斗……
- 瓶颈：集中地域内
  - 大量聚集的玩家基地
  - 大规模的行军、战斗

优化策略：世界地图

- 镜头推近/拉远：多层LOD
- 地表：基于GPU的快速地基构建 + 装饰对象
- 行军队列：
  - 控制对象种类数量，小对象单位合批
  - 小对象动画方案
    - 2D frame animation
    - 3D mesh frame animation
  - 基于同屏队列数量的动态LOD
- 多核并行优化：Job System, …

Infinite Galaxy项目介绍

- Infinite Galaxy (下简称IG）是一款3D SLG手游
- 太空科幻题材，写实风格渲染
- 2018立项初期基于传统Built-in管线，2019年下半年转为URP管线

主要场景包装

- 基地：空港（控制室）
- 世界地图：
  - 第1层：恒星系
  - 第2层：银河系

优化要点（1）

- 恒星系层为核心负载场景
- 光影渲染：光源（恒星）居中，补光方向指向光源
- 场景舰船不需要带动画，整体运动与特效、光影表现为主
- 群体战斗时的实时光影反馈

优化要点（2）

- 各场景低延迟切换
  - 空间换时间
  - 分帧加载
- 多队列、多战斗情形
- 功耗 vs 体验流畅度
  - 按用例区分的更新FPS
  - 按操作区分的更新FPS

Why URP? —— Year of 2019 (LWRP 6.9.x)

- 官方推荐
- 高性能多光照
- 功耗可负载的后处理效果
- SRP Batching

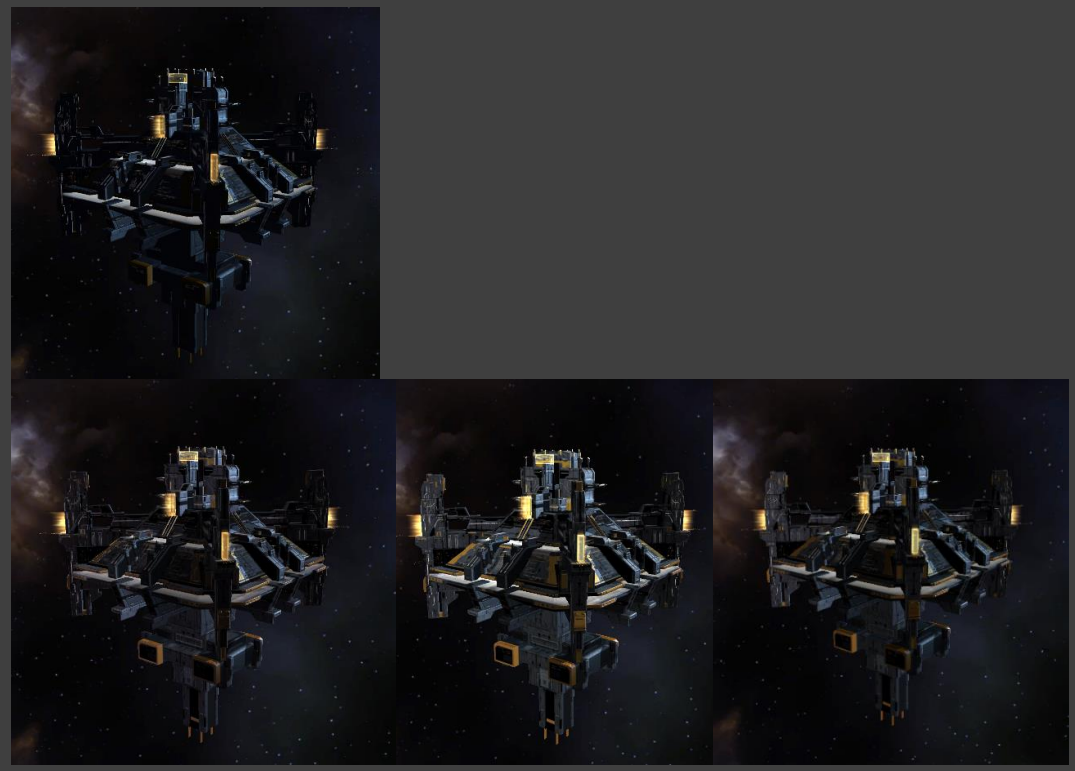
整体优化

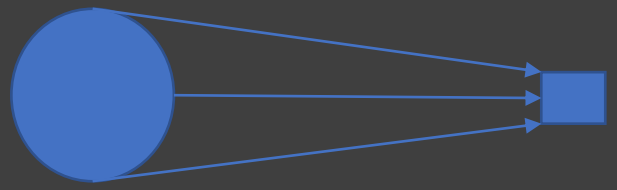
IG 整体优化-恒星系

- 模拟光照
- 陨石带Instancing
- 建筑LOD
- 建筑特效优化

恒星系模拟光照

- 基于Matcap与恒星颜色的环境光照
- 模拟球形光照
- 环境光也能有部分高光反射细节



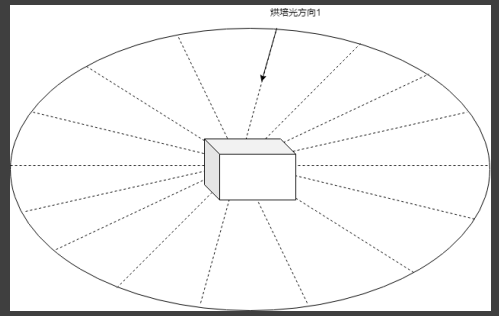


阴影烘培

- 面临的问题
  - 模拟太空环境，主光源为点光
  - 静态烘培（方向不变）的建筑无法移动
  - 高端机型使用点光源实时阴影开销也会很大
  - 太空中没有地表，不需要投影片，仅需要关注自阴影

阴影烘培

- 实现步骤
  - 按黄道面16等分，每个角度烘培一次阴影
  - 恒星系内的建筑确定位置后计算使用哪一张阴影贴图
  - 位置更新时更新阴影贴图



阴影烘培

- 保存方式
  - 使用bit位保存阴影信息
  - Job system实时解压
  - 每张贴图写入两个RG两个通道
  - 分别为当前方向最接近的两个烘培方向
  - 材质中根据当前方向插值混合

陨石带Instancing

- Graphics.DrawMeshInstancedIndirect API
- 构建四叉树进行分块
- 同时使用四叉树做相机剪切
- 根据相机距离选择LOD
- 材质解决陨石自旋
- 内存开销仅有一个2个LOD mesh

恒星系

- 建筑LOD
- 建筑特效优化

Building Mesh x 1 : 6486 Tris
Effect Mesh X 8 : 420 Tris
Particle System X 8

Building Mesh x 1 : 6486 Tris
Effect Mesh X 2 : 104 Tris

Building Mesh x 1 : 2404 Tris
Effect Mesh X 1 : 48 Tris

优化前
Building Mesh x 1 : 6486 Tris
Particle System X 220

战斗优化

- 舰船LOD
- 舰船特效优化
- 舰队移动优化
- 针对常态拖尾的优化
- 本地战斗的分帧计算目标

战斗优化

- 舰船LOD： Impostor
- 舰船特效优化: Impostor Meshes
- Trails : 隐藏时必须关闭 emitting

Mesh x 1 : 3878 Tris
Effect Mesh X 3 : 36 Tris
Trail x 3

Mesh x 1 : 299 Tris
Effect Mesh X 2 : 24 Tris
Trail x 3

Mesh x 1 : 2 Tris

舰队移动优化

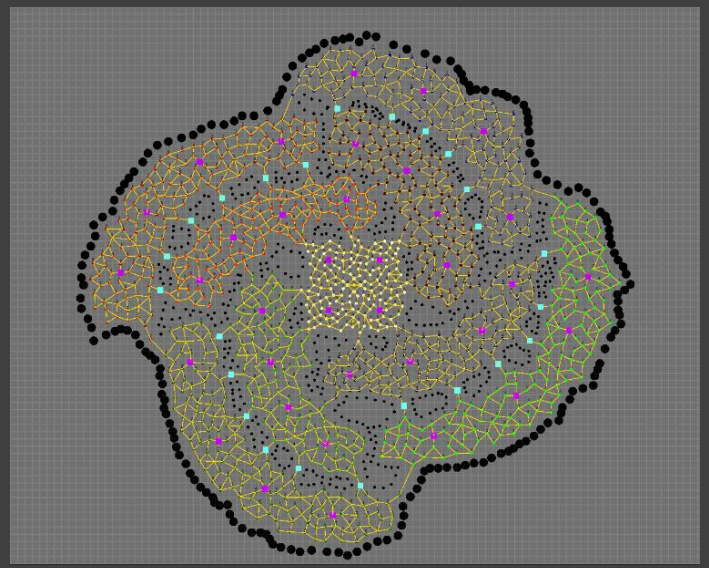
- 匀速运动+曲线拟合
- JobSystem驱动Transform

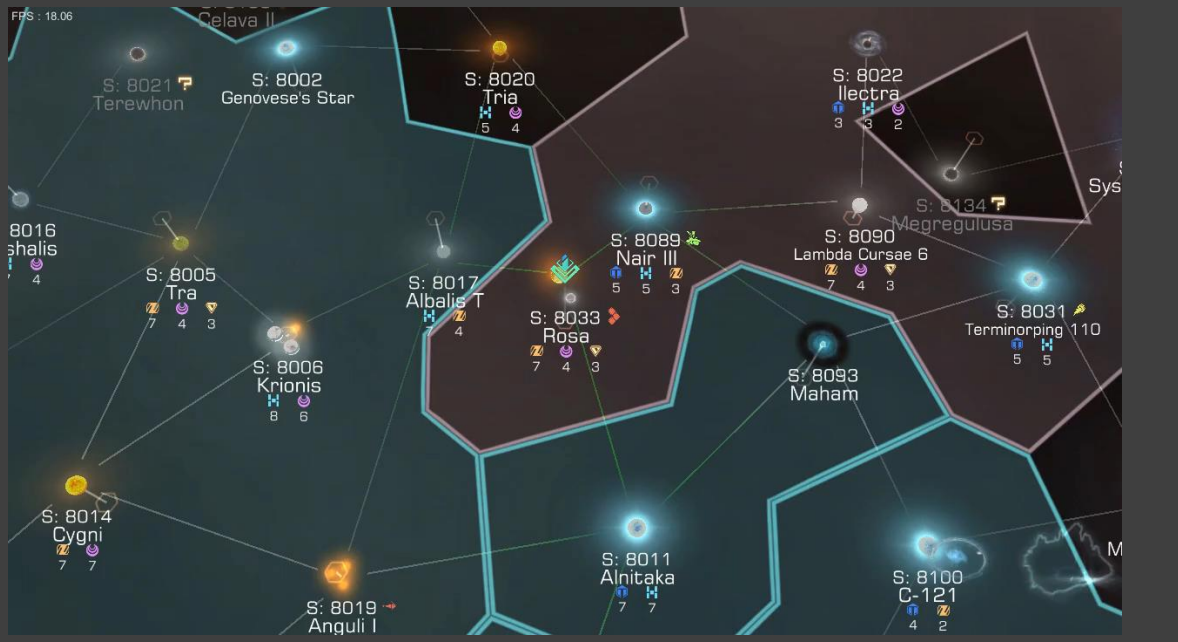
战斗优化

- N vs N的本地战斗
- 遍历寻找目标的时间复杂度O(n^2)
- KDTree查询最近目标O($n^{1-1/k}+m$)
  - m---每次要搜索的最近点个数
- 分帧构建KDTree （复杂度O($\log^2 n$)）
  - 根据不同机型算力分配时间片
  - 时间片内计算KDTree，计算完毕则更新KDTree
  - 未完成则复用上一次计算的结果

银河系

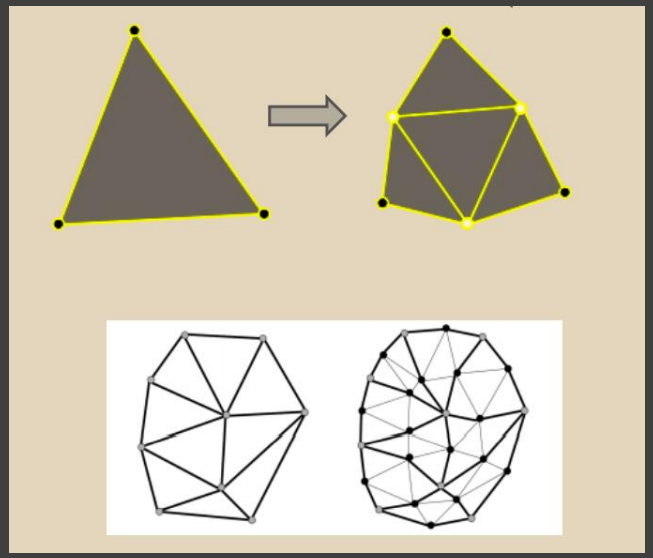
- 基于维罗尼多边形的大地图
- 联盟占领区域的动态生成





联盟占领区域的动态生成

- Delaunay三角动态合并Mesh
- Loop Subdivisoion
- Job system实现细分算法



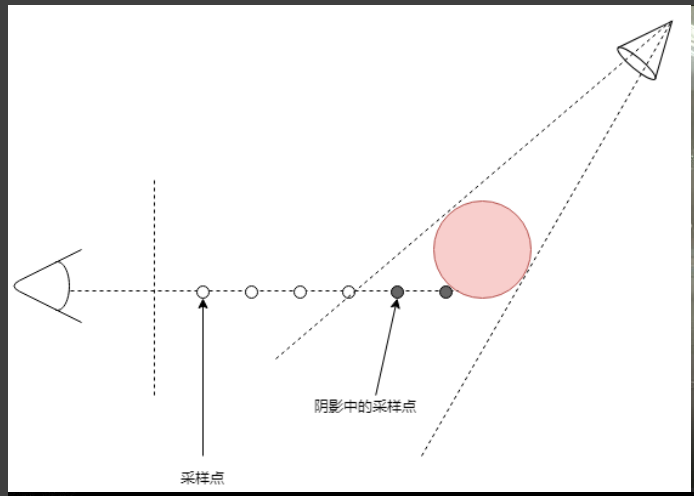
Volume Lights

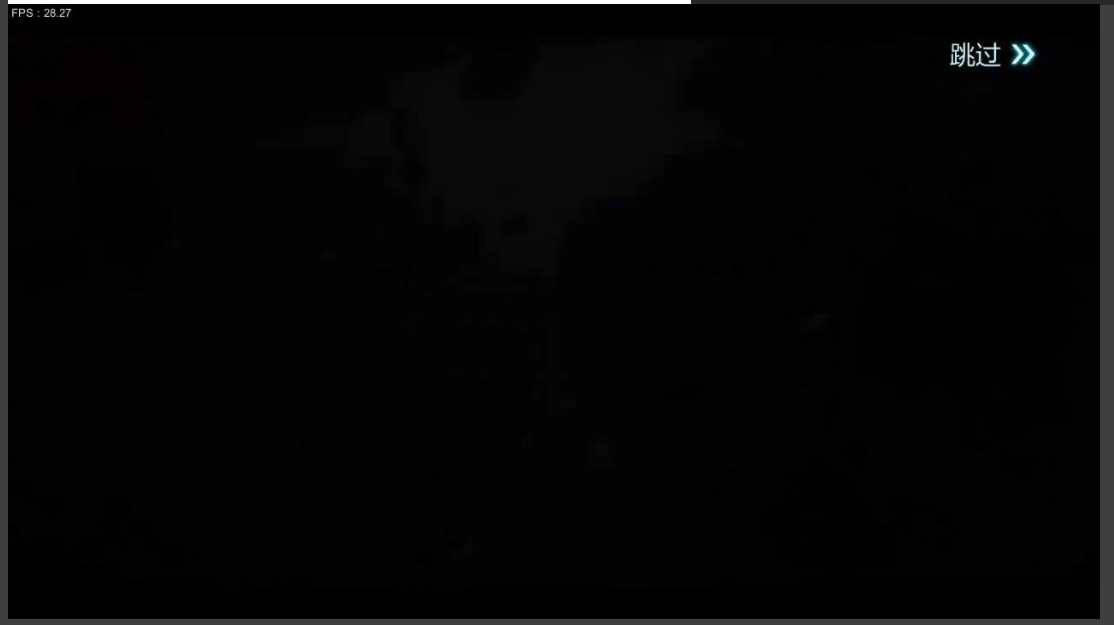
IG Volume Lights

- 需求
  - 模拟丁达尔现像

IG Volume Lights

- 基于物理正确的方式
  - 优点
    - 物理真实，效果好
  - 缺点
    - 精度提升困难
    - 细节表现不光需要提升步进精度，场景的阴影细节也必须有对应的匹配提升

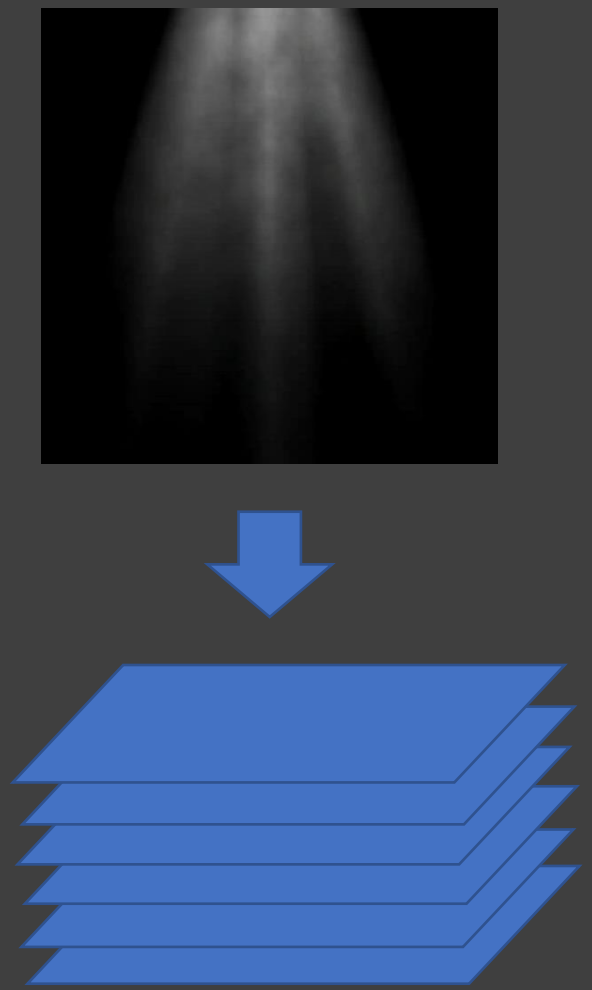




IG Volume Lights

- 模拟方式
  - 使用贴图作为输入
  - 模拟3d Texture，可以认为是烘培好的体积光

```
half3 RayMarching(float3 start, float3 end, int step, half noise)
{
    half maxLen = length(end - start);
    float stepsize = max(_MinStepSize, maxLen / max(1, step));
    float3 marchingDir = normalize(end - start);
    half marchingLen = 0;
    half3 lightColor = 0;
    [loop]
    for (int i = 0; i < step; ++i) {
        marchingLen = (i + noise) * stepsize;
        if (marchingLen <= maxLen)
        {
            float3 samplePos = start + marchingLen * marchingDir;
            float4 localPos = mul(unity_WorldToObject, float4(samplePos, 1));
            half4 c_xy = SAMPLE_TEXTURE2D_X(_BaseMap, sampler_BaseMap, localPos.xy + 0.5);
            half4 c_zy = SAMPLE_TEXTURE2D_X(_BaseMap, sampler_BaseMap, localPos.zy + 0.5);
            float4 shadowCoord = TransformWorldToShadowCoord(samplePos);
            half shadow = MainLightRealtimeShadow(shadowCoord);
            lightColor += c_xy.rgb * c_zy.rgb * shadow;
        }
    }
    return lightColor;
}
```



IG Volume Lights

- 模拟方式
  - 使得体积光的形态更可控
  - 独立于场景的其他设置，计算简单

Face Sync Solution

IG Face Sync solution

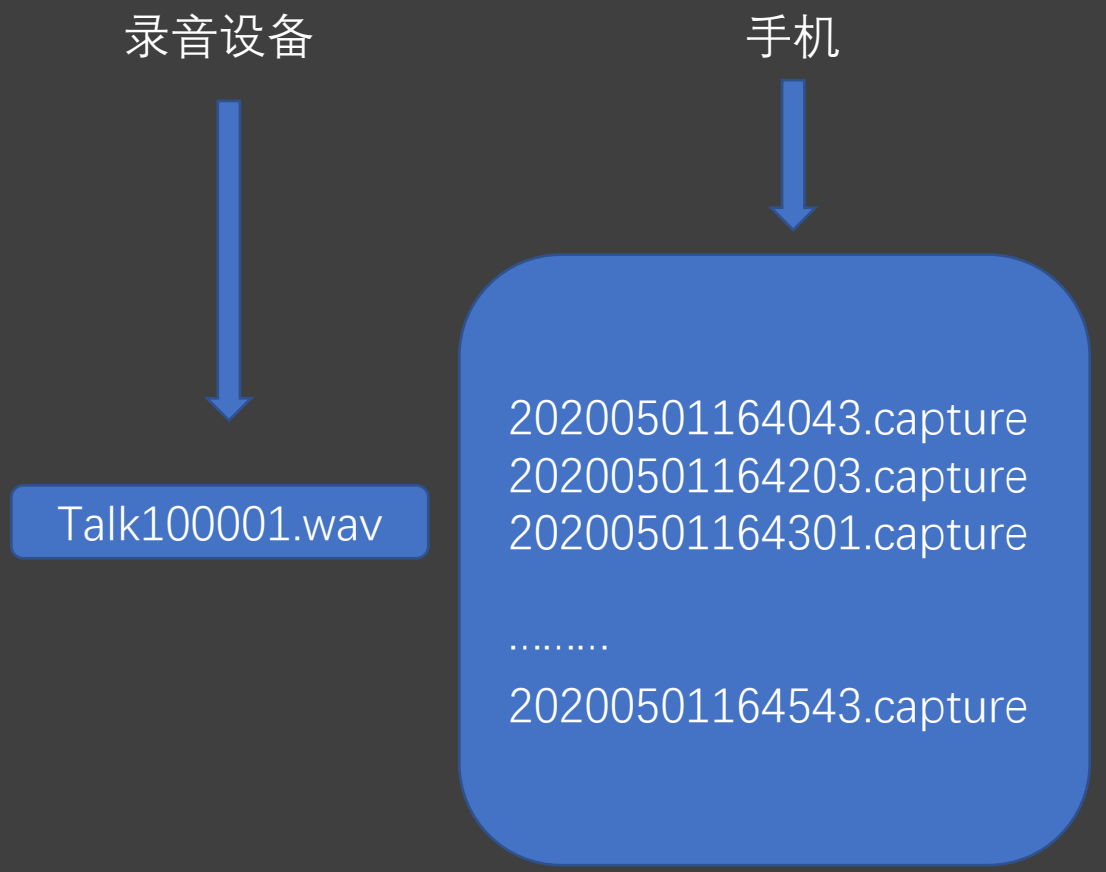
- 需求
  - 更生动的剧情表现
  - 声画同步
- 目标
  - 完备的工具流水线
  - 支撑快速制作多语言面部动作与语音配合

IG Face Sync solution

- 录音室同步录制声音和演员表情
  - ARKit XR Plugin- Unity
  - ARKit支持头部偏转信息，这点对于语言表达很重要
  - 记录头部偏转信息，在游戏中与角色自身的动作进行融合

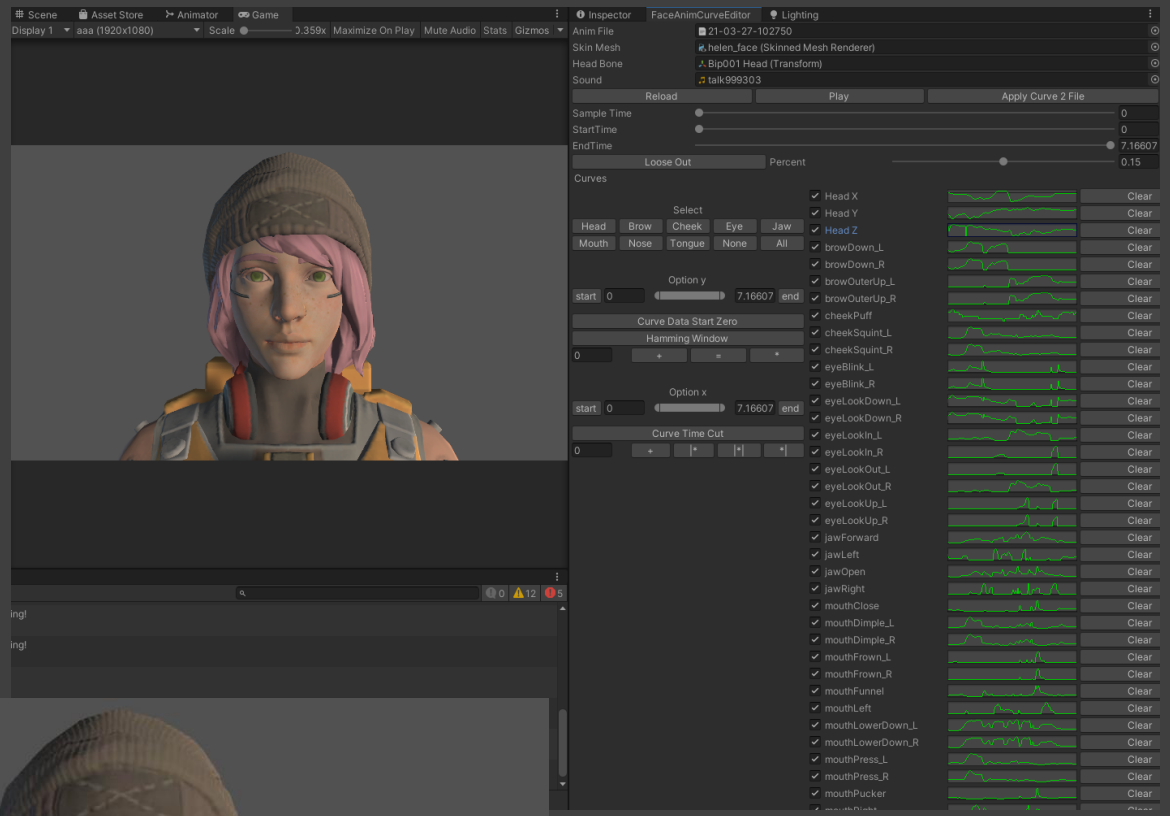
IG Face Sync solution

- 最终音频与录制文件匹配
  - 面部表情动画录制设备与录音设备独立
  - 使用Speech to Text服务解析动画对应的文字
  - 精确匹配动画与录音文件



IG Face Sync solution

- 音频与录制动作数据精修调整



IG Face Sync solution

- 画面表现
  - URP多光源
  - 预积分皮肤（Pre-Integrated Skin ）材质