

# 《王者荣耀》制作技术分享

本节课题: **爆款手游《王者荣耀》制作技术分享**

主讲老师: **博毅创为 Blake 老师**

课时内容:

【第1部分】:《王者荣耀》基本情况介绍;

Moba类型的游戏: 英雄联盟(LOL), 动作即时战略游戏

《王者荣耀》研发团队组成;

产品经理: ---->美术组: 美术主管: 3D游戏: 原画(人物, 风格设计), 3D建模: (角色建模, 场景建模) 3D动作, 特效, UI设计, TA技术美术

---->策划组: 主策划, 系统策划, 数值策划, .....

---->程序组: 客户端(Unity 客户端---->主程, 若干)+服务端(服务端主程 + 程序若干)

---->测试组: 测试人员---->《王者荣耀》稳定性;

---->音乐组: 游戏的音乐, 声音配音, 游戏背景音乐;

【第2部分】:《王者荣耀》客户端开发工具Unity引擎;

Unity 引擎---->手机游戏的时代---->第一款支持跨移动平台的主流3D游戏引擎;

2016年, Unity ---->手机游戏引擎---->《王者荣耀》70%~80% ---->Unity开发的;

[https://unity.cn/releases?tdsourcetag=s\\_pcqq\\_aiomsg](https://unity.cn/releases?tdsourcetag=s_pcqq_aiomsg)

Unity版本的使用情况: 2018, 2017, 2019 5.x

Unity DOTS, ECS;

七麦数据: Unity 开发出来的;

Unity开发的游戏资源: 资源破解工具, 资源解开的;

爬取资源的工具: AssetStudioGUI.exe;

Unity 做游戏 ---->《王者荣耀》C#---->热更新---->lua解释器---->《王者荣耀》Lua写代码---->解释执行;

热更新---->ILRuntime ---->热更新;

其他 VR, AR

【第3部分】: 2小时制作一个Moba《王者荣耀》客户端Demo;

第1课时: 如何来搭建一个客户端的游戏框架;

Unity 引擎---->《王者荣耀》搭建一个框架: 让更多的开发人员能够基于框架并行开发, 协同开发, 积累标准的行业解决方法(可重用);

腾讯: part1: 开发框架: 基于Unity ---->开发流程框架

part2: 定制游戏引擎, Unity公司---->《王者荣耀》源码授权给你, 定制你的项目特殊效果; 渲染效果, 性能优化;

原则1: 运行的时候场景是空的, 场景里面不放置任何内容, 美术和程序更好的协作, 多个程序也可以很好的协作不会冲突;

我们打包运行的时候只要一个场景就可以了。---->打空包,

多新建几个场景给美术编辑;

原则2: 反对手动的去添加代码逻辑组件到节点上, 到预制体上; 写了一个代码, 你要问自己, 这个代码在哪里地方被调用

搜索整个源码---->《王者荣耀》找到调用的地方; 每个节点, 每个预制体都去点, 看看有没有挂这个代码;

A开发: 节点---->《王者荣耀》组件实例

B开发: 节点---->《王者荣耀》手动组件实例; 与视图相关的组件, 特效组件, ---->《王者荣耀》视图:[UI, 角色, 场景, 特效, ...]

美术不用管代码;

原则3: 频繁更新我们的资源, 业务逻辑, 打空包; ---->资源管理;

Resources来做资源管理: Resources文件夹下, 不管你用了还是没有用---》打入包体, 打空包就不方便; 也不方便更新;

AssetsBundle/Addressable 来做我们的资源管理;

step1: 游戏资源---》打包成ab包; ---> StreamingAssets --->

step2: 放到安装包里面,你也可以不放安装包,ab包部署到服务器上;

step3:在开发的过程当中, 每次你改动资源以后, 都要打一下ab包才能使用, 开发的时候非常麻烦;

开发模式: AssetsDatabase来加载我们的资源, 不用每次都打ab包;

发布模式: 正常的走ab包的流程

从头开始, 来搭建一个框架, 展开来, 管理我们的项目;

step1: 维护好你项目的主要的目录结构

step2: 启动代码--->

step3: 做好常用的管理模块: **资源管理**, **事件管理**, 声音管理, **UI管理**, 日志管理, 网络管理, 协议管理, ..

=====

第2课时: 我们程序把场景拼起来

放了地图,

放了游戏逻辑数据 --->场景管理---》角色, 放NPC, 放小兵, 放对方的玩家;

放了游戏的UI

第3课时: 控制我们的角色, 操作我们的角色

step1:定义一些状态: idle, walk, skill

step2: 定义一个变量---》保存当前的状态;

step3: 控制我们玩家的移动

step4:第三人称摄像机的跟随:



【第4部分】: Moba 核心技术 UDP 帧同步的全流程详解;

