项目名称：《和平战场：决斗》(PlayerKnown's battleground)

项目组：

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 手机 | 电子邮箱 |
| 515010910679 | 王浩宇 | 18018596679 | gogowhy@sjtu.edu.cn |
| 517021910705 | 齐澎 | 15201982059 | [757994086@qq.com](mailto:757994086@qq.com) |
| 517021910031 | 谢宜含 | 18817672620 | [xieyihan99@sjtu.edu.cn](mailto:xieyihan99@sjtu.edu.cn) |
| 517021910909 | 周一凡 | 15201952130 | [1774871693@qq.com](mailto:1774871693@qq.com) |

2019年6月

**《和平战场：决斗》(PlayerKnown's battleground)立项建议书**

一、项目的必要性（necessity）

随着应用程序技术的发展，手机游戏作为一个新兴行业正处于蓬勃发展的阶段。现有的手机游戏几乎达成了各种电脑游戏类型的轻量化再现，甚至可以达到不逊于一些主机的游戏性能。

然而绝大多数手机游戏都着力于以至禁锢于这些现有的经典游戏类型，很少有手机游戏能将手机拥有而主机没有的拍照，定位，动作反馈等功能与游戏的主要模块有机结合。本项目创新性的将手机的各种硬件条件大胆结合运用，基于手机的摄像头，GPS，陀螺仪，加速计等特有硬件，实现类似于VR的人机交互体验，让用户的位置移动和肢体动作成为对游戏进行操作控制的主要方法，从而在手机上模拟真人CS游戏，摆脱了定制游戏设备的约束，随时随地用手机即刻开玩。

本项目也可向运动监测等方向拓展延伸应用。

二、项目目标和特性

（一）项目目标

本项目将开发一个手持手机移动设备进行《和平战场：决斗》(PlayerKnown's battleground)射击游戏的系统。该项目在互联网的技术基础上使用gps定位玩家信息，深度学习功能识别敌方人物，点击屏幕进行射击的中大型《和平战场：决斗》(PlayerKnown's battleground)真人射击系统。

（二）功能性需求

1.玩家登陆服务器后选择阵营和游戏模式进行游戏

2.通过GPS获取附近地形生成游戏地图边界和起始点

3.玩家可以通过手机的摄像头组件模拟射击动作，按下快门射击，分析拍得的图像判定是否击中

4.根据玩家的枪械类型对摄像头的缩放效果产生影响，装备放大倍率不足时对强行放大所得画面进行模糊处理

5.玩家可以通过横移手机模拟近战挥砍动作，比对两者GPS确定是否有效击中

6.玩家被击中损失生命值时手机将有长振动提示

7.玩家通过对手机进行特定移动或者操作按键，模拟切换武器和装填弹药

8.玩家生命值归零时手机将通过发声、振动方式提示阵亡

9.某一方胜利或者时间用尽导致游戏结束时全体玩家手机将有发声、振动提示

10.玩家可以使用好友系统进行添加好友操作，使用仓库系统、商场系统进行购买管理枪械道具操作。

（三）进阶需求

使用深度学习技术和gps定位技术对摄像头捕捉到的人物进行识别和区分

使用增强现实技术改变玩家在其他玩家手机中的形象（如手机被替换为枪支等特效）

（四）创新点

1.将手机、互联网技术和《和平战场》技术联系到一起

2.增强现实技术，让游戏界面建立在现实世界的基础上，但比现实世界更加真实

3.深度学习技术的应用完成摄像头中的人物识别

4.通过游戏还可以对运动状态进行记录，对运动健康起检测作用

**1.4 参考资料（References）**

[1]沈备军，陈昊鹏，陈雨亭，《软件工程原理》 北京高等教育出版社

[2]张海藩， 《软件工程导论》，清华大学出版社。

[3]王删著， 《数据库系统原理教程》,清华大学出版社。

[4]陈松桥等著， 《现代软件工程》,北方交通大学出版社。

1.5 相关文档（Related Documents）

[1]《<和平战场：决斗>（Playknown's battleground）需求规约》

1.6 版本更新记录（Version Updated Record）

任何一次版本创建或维护更新，都要追加一条记录。一个版本创建只有一次，但对它的维护更新可能有多次。大版本升级一次，定义为创建一次，如V1.0到V2.0。而V1.0到V1.1，是维护更新一次。版本更新记录格式，如表1-1所示。

版本更新记录如下表所示：

版本维护更新表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 创建者 | 创建日期 | 维护者 | 维护日期 | 维护纪要 |
| V1.0 | 王浩宇 | 2019.5.12 | — | — | 立项建议书框架构建与分工架— |
| V1.0.1 | — | — | 齐澎 | 2019.5.12 | 完善项目必要性 |
| V1.0.2 | —— | —— | 王浩宇 | 2019.5.15 | 完善项目性能 |
| V1.0.3 | — | — | 谢宜含 | 2019.5.19 | 完善项目概述 |
| V1.0.4 | —— | —— | 王浩宇 | 2019.5.28 | 完善项目运行平台、硬件、系统组成、系统功能、系统部署方案 |
| V1.0.5 | —— | —— | 周一凡 | 2019.5.29 | 完善案例分析 |
| V1.0.6 | —— | —— | 齐澎 | 2019.5.29 | 完善物理架构，逻辑架构 |
| V1.0.7 | —— | —— | 谢宜含 | 2019.6.1 | 完善项目目标和特性 |
| V1.0.8 | —— | —— | 周一凡 | 2019.6.1 | 完善项目风险分析和里程碑计划 |

三．项目技术方案（Project Framework）

2.1 项目概述（Project Summary）

随着应用程序技术的发展，手机游戏作为一个新兴行业正处于蓬勃发展的阶段。现有的手机游戏几乎达成了各种电脑游戏类型的轻量化再现，甚至可以达到不逊于一些主机的游戏性能。本项目创新性的将手机的各种硬件条件大胆结合运用，基于手机的摄像头，GPS，陀螺仪，加速计等特有硬件，实现类似于VR的人机交互体验，让用户的位置移动和肢体动作成为对游戏进行操作控制的主要方法，从而在手机上模拟真人CS游戏，摆脱了定制游戏设备的约束，随时随地用手机即刻开玩。本项目也可向运动监测等方向拓展延伸应用。

2.2 项目架构（Project Framework）

2.2.1运行平台

在 Android 8.0 （Oreo）版本及以上安卓手机上运行客户端。

2.2.2运行需要硬件

手机要求至少ROM 64GB/128GB 海思 麒麟 710 以及其他对应的硬件及以上版本的手机能够流畅运行

2.2.3项目技术概要

项目前端主要由Java语言开发，面向手机native app进行开发，暂不支持web方式的访问，前端使用java接口和react-native的antdesign框架javascript语言。后端使用Spring boot框架，主要采用Java语言编写，数据库采用MySQL和mongodb。

其中java的部分编码规约将完全依据《阿里巴巴Java开发手册》进行，遵循其中的命名风格，常量定义，代码格式等。

视频采用实时追踪模式，暂不提供视频记录存档功能。

2.2.3系统组成

系统由玩家、系统管理员组成。

2.2.4系统功能

管理员：用户管理

普通用户：

1.登录、注册

2.游戏的创建 、加入

3.武器的购买

4.游戏中的对战功能

进阶功能：

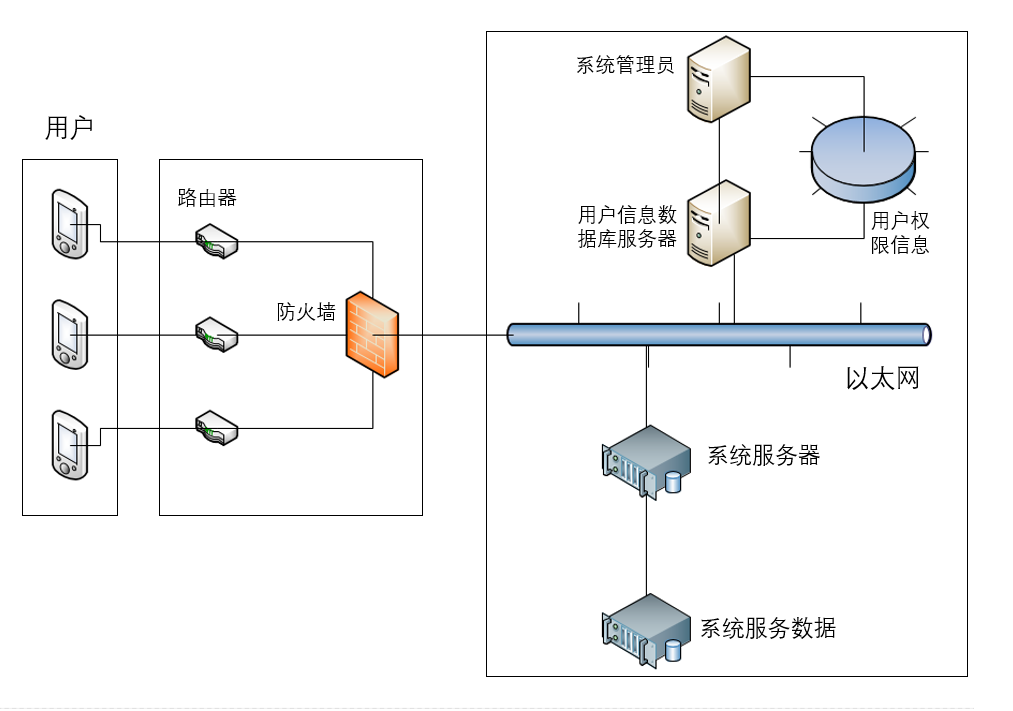
增强现实改变玩家在其他玩家手机摄像头中的形象（穿着和手持武器）

深度学习识别玩家

2.2.5系统部署方案

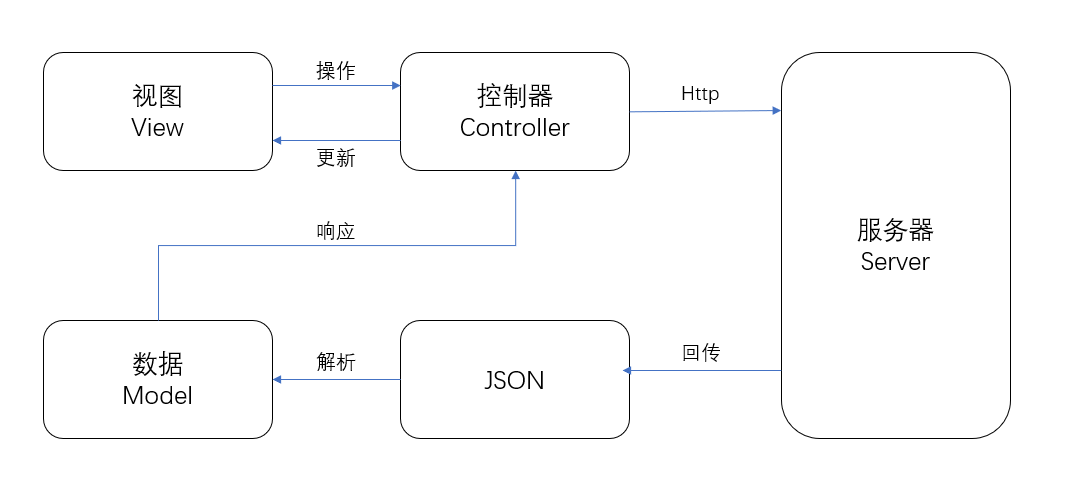
|  |  |  |
| --- | --- | --- |
| 部署地点 | 部署子系统 | 部署模块 |
| 互联网 | 玩家 | 登录 |
| 游戏的创建 |
| 游戏的加入 |
| 武器的购买 |
| 武器的装备 |
| 游戏中的对战功能 |
| 系统管理员 | 注册 |
| 登录 |
| 用户管理 |
| 游戏系统信息管理 |

2.2.6物理架构图



采用C/S架构，用户通过手机APP发送请求，经过路由器和防火墙访问以太网服务器，用户信息数据服务器通过查找用户权限信息对用户开放系统服务器对应功能，系统服务器处理系统服务数据向用户回传运行结果。系统管理员通过用户信息数据库服务器操作用户权限信息，以确定用户可访问的功能部分，对用户进行管理。

2.2.7逻辑架构图



1. View层接受用户的交互请求；

2. View层将请求转交给Controller层，对其进行操作；

3. Controller层通过Http链接Server服务器，回传JSON到Model层，Model层解析回传的JSON对其存储的数据进行更新；

4. 数据更新之后，Model层通知Controller层数据变化；

5. Controller层用Model层的响应更新View层的信息，完成一次交互。

四．项目风险分析和里程碑计划

（Risk Analysis）

3.1 风险（Risk）

**表格名词解释**

|  |  |
| --- | --- |
| ID： | 风险信息表 |
| 优先级：分1-5级，5为优先级最高 | 陈述: |
| 概率：分1-5级，5为可能性最大 |
| 影响：分1-5级，5为对整个开发影响最大 | 起因： |
| 时间框架：分前期（迭代前）、初期（迭代刚开始时）、中期（迭代过程中）、后期（项目收尾时）、远期（项目正式部署运行时） |
| 缓解策略： | |
| 应急计划及触发事件： | |

3.1.1 技术风险

|  |  |
| --- | --- |
| ID：1 | 风险信息表 |
| 优先级：5 | 陈述: 对于web端和手机app端的开发，小组成员开发经验不足，可能遇到较难解决的问题。 |
| 概率：4 |
| 影响：5 | 起因：开发经验不足 |
| 时间框架：初期 |
| 缓解策略：在项目开始前了解及学习相关知识，熟练使用开发工具。 | |
| 应急计划及触发事件：首先开发基本原型。 | |

|  |  |
| --- | --- |
| ID：2 | 风险信息表 |
| 优先级：2 | 陈述: 由于开发工具及开发设备的不可靠性导致项目过程中单Bug。 |
| 概率：2 |
| 影响：4 | 起因：开发工具出现问题，开发者电脑及其他设备出现问题等。 |
| 时间框架：初期至后期 |
| 缓解策略：使用相同的开发工具及设备开发简单程序。 | |
| 应急计划及触发事件：尝试修复开发工具及设备，必要时需更换开发工具或设备。 | |

|  |  |
| --- | --- |
| ID：3 | 风险信息表 |
| 优先级：3 | 陈述: 由于开发中版本管理不当而导致的功能混乱或bug等问题。 |
| 概率：3 |
| 影响：2 | 起因：版本变更管理不当。 |
| 时间框架：中期至后期 |
| 缓解策略：每天检查版本一致性并跟踪和记录版本变更；每次迭代完成后备份可运行版本。 | |
| 应急计划及触发事件：将版本恢复到上一次保存的正确版本，然后重新进行变更。 | |

|  |  |
| --- | --- |
| ID：4 | 风险信息表 |
| 优先级：1 | 陈述: 由于各组员编程语言选择不一致而导致效率低下，进度缓慢。 |
| 概率：1 |
| 影响：2 | 起因：编程语言不一致。 |
| 时间框架：初期 |
| 缓解策略：各组员严格按照需求文档来使用相应的开发语言，在项目开始前统一编程语言。 | |
| 应急计划及触发事件：当发现选择不一致时，及时协调达成一致。 | |

3.1.2 架构风险

|  |  |
| --- | --- |
| ID：5 | 风险信息表 |
| 优先级：4 | 陈述:由于框架自身存在的漏洞bug导致项目失败。 |
| 概率：2 |
| 影响：3 | 起因：搭建框架时存在未预见性的漏洞及bug。 |
| 时间框架：初期 |
| 缓解策略：测试时及时发现问题并解决。 | |
| 应急计划及触发事件：利用框架的可扩展性及时进行补救。 | |

3.1.3 进度风险

|  |  |
| --- | --- |
| ID：6 | 风险信息表 |
| 优先级：3 | 陈述:由于时间紧迫，导致可能无法按期完成项目。 |
| 概率：3 |
| 影响：3 | 起因：时间较紧，开发效率低下。 |
| 时间框架：中期至后期 |
| 缓解策略：提高开发效率。 | |
| 应急计划及触发事件：采用Spring 1\2\3\4增量式开发。 | |

3.1.4 需求风险

|  |  |
| --- | --- |
| ID：7 | 风险信息表 |
| 优先级：4 | 陈述:由于需求分析不到位或计划安排不妥当，导致无法在一次迭代结束时完成应用的功能。 |
| 概率：4 |
| 影响：3 | 起因：需求分析不到位或迭代计划安排不妥当。 |
| 时间框架：初期至后期 |
| 缓解策略：对各个需求或功能的开发难度及时间做出较准确的估计，按照项目组实际情况合理划分各个迭代的任务。 | |
| 应急计划及触发事件：紧急更改迭代的任务描述，首先完成必要功能。 | |

|  |  |
| --- | --- |
| ID：8 | 风险信息表 |
| 优先级：4 | 陈述:由于需求分析不到位或需求的不断变更，导致数据模型的重建、设计与开发的重做等。 |
| 概率：4 |
| 影响：4 | 起因：需求分析不到位和需求变更。 |
| 时间框架：前期至中期 |
| 缓解策略：在项目开展前期多做调研，尽量确定主要需求及需求功能的实现。在开发过程中相互确认功能需求。 | |
| 应急计划及触发事件：及时根据变更或修正的需求进行编程。 | |

3.2 迭代以及任务描述

|  |  |  |
| --- | --- | --- |
| 迭代 | 任务描述 | 成果 |
| 项目启动（11天）  6月4日-6月14日 | 与用户进行沟通，并确定初始方案；  进行需求规约、立项答辩；  项目立项。 | 确定用户对软件的需求 |
| Sprint1（11天）  7月1日-7月11日 | 架构分析与设计；  架构实现与搭建；  R1的需求分析、设计与实现；  系统测试，并获得用户反馈，进行缺陷修复与改进。 | 完成架构原型的搭建，完成系统版本1（R1）的开发 |
| Sprint2（10天）  7月12日-7月21日 | 在R1的基础上进行R2的需求分析、设计与实现；  系统测试，并获得用户反馈，进行缺陷修复与改进。 | 完成系统版本2（R2）的开发 |
| Sprint3（10天）  7月22日-7月31日 | 在R2的基础上进行R3的需求分析、设计与实现；  系统测试，并获得用户反馈，进行缺陷修复与改进。 | 完成系统版本3（R3）的开发 |
| Sprint4（40天）  8月1日-9月9日 | 安装R3，撰写用户手册，对用户进行培训，系统上线运行，并根据系统运行反馈进行缺陷修复与改进。 | 完成系统版本3的发布 |

l  架构原型必须实现的功能：完成React Native,Spring Boot等前后端框架的搭建，完成数据库的创建。

l  系统版本1（R1）必须实现的功能：完成用户的注册/登录功能，用户的创建房间功能，房间内用户交流功能

l  系统版本2（R2）必须实现的功能：完成用户的gps定位功能、拍照用户识别任务功能

l  系统版本3（R3）必须实现的功能：完成用户增强现实显示装备功能、用户实时对战功能

五．项目预期成果（Project achievement）

项目验收时至少应交付以下成果：

  《立项建议书》

  《迭代计划》（每个迭代开始前编写迭代计划）

  《迭代评估报告》（每个迭代结束后编写迭代评估报告）

  《SRS文档》和用例模型（.oom）

  《软件架构文档》和分析设计模型（.oom）

  《测试用例》和《测试报告》

  《项目总结报告》

  源代码和可执行代码

   演示视频文件（包括安装、运行、功能等）

   演示PPT

六.项目性能（Project Performance）

5.1 响应时间（Response Time）

C/S结构架构的终端数量为100台时，游戏外操作要求响应时间小于1秒，游戏内操作（在玩家网络正常的情况下）要求延迟小于100毫秒。

5.2 处理速度（Disposal Speed）

C/S结构、B/S结构两种架构的后台结算方式，结算速度应尽可能小。

在同时处理1000个人的用户请求时，系统的响应时间最多为1s。

系统在8:00-22:00，正常运行时间至少占据该时间段的百分之九十。

七.案例分析（Cases Analysis）

当前的火爆fps游戏pubg、《和平战场：决斗》(PlayerKnown's battleground)go和守望先锋等让玩家体验了枪战的快感，大型多人在线的枪战画面异常火爆，在整个游戏市场也是占据了半壁江山的份额。

而增强现实技术为基础的pokeman等基于gps的实景拍照游戏更是在早些时间火遍全球。

分析这两个案例，我们打算结合当下最流行的第一人称视角风格和当下最火爆的实时真实场景游戏技术，制作出与众不同的《和平战场：决斗》(PlayerKnown's battleground)对战系统。