**项目背景**

1.公司需要推出虚拟导游服务，具体业务场景是：在一个会议上，有一个主持人，他可以操作显示屏；有许多观众，每个观众自身拥有一个VR头盔。观众听主持人介绍内容时， 可以带上VR头盔，VR头盔里显示了和主持人操作的显示屏上一致的内容，会议开始后观众就可以戴上头盔在主持人的带领下一起观看虚拟世界中的景象。为了支持虚拟现实服务，需要一套稳定的消息同步机制。为了应对将来可能出现的各类需求变更，需要建立一套兼容性很强的控制流程。

系统目标

1. 建立一个稳定的，安全的，低时延的消息同步机制。该机制未来可用于多人互动。
2. MRMixer上做到发送命令和执行命令两个操作之间的完全解耦，实现一个MRTP消息解释器和MRTP消息生成器 。使得MRMixer上大部分的操作都可以由MRTP消息控制。
3. 服务端上做到消息处理与业务逻辑完全解耦，
4. 开发一套小程序，该小程序视为外部设备，也可以发送命令控制MRMixer。
5. 兼容性好，能支持不同版本的MRMixer在进行消息同步时也能够正常同步。

需求分析

基于业务场景，有用户，主持人两类用户角色。对于主持人，他的诉求有：1.他能够将自己在虚拟世界里的所见所闻实时分享给他的观众。2.他能够很方便的在虚拟世界里做出操作3.他能够控制分享的内容和分享的时间。对于观众，他的诉求有：1.他能够进入主持人所在的虚拟世界。2.能够跟随主持人来观察整个虚拟世界。

总体设计

系统架构设计

系统主要有播放端，服务端，控制端组成。3部分各司其职，其主要负责功能如下：

播放端：主要由MRMixer-Pro、MRmixer-Pico两个产品承担播放角色。播放端负责发起消息同步、加入消息同步、退出消息同步、接收从服务端传递的消息、将消息解析成命令、执行命令、修改同步资源。

服务端：服务端负责开辟消息同步需要的资源、变动消息同步所需的资源、接收控制端命令、推送消息给播放端。

控制端：显示可控制内容、发送控制消息、修改同步资源。

他们之间的交互关系如下：

图示

描述已自动生成

关键流程设计

系统关键的流程是消息的传播机制，现在以主持人更换内置场景这个操作为例，描述系统是如何做到场景更换的。

1. MRMixer软件里面有更换场景这一个操作的详细信息，服务器上也相同的操作备份。

2. 主持人在MRMixer上向申请开通一个房间。

3. 服务器自动通过了开通房间申请，将房间信息告知MRMixer。

4. MRMixer接收到房间信息后，将本地资源上传到服务器，然后显示 房间 编号和分享码。

5. 主持人或者观众输入了房间编号与服务器建立长连接。

6. 控制端扫分享码进入页面，在询问服务器房间内容后渲染出更换背景界面。

7. 控制端点击更换背景按钮 ，将消息发送到服务器， 。

8. 服务器处理并记录消息，随后推出命令。

9. MRMixer接收到命令，开始解析命令最后执行命令。

10.MRMixer上用户退出房间

服务端设计

一 服务器架构

【服务器架构图】

二 服务器功能结构

服务端逻辑部分分为消息模块、房间模块、操作模块。

三 使用技术

Websocket ; Netty;定时任务；SpringBoot;maven;

四 接口设计

设计接口时考虑：1.要遵循restful的设计理念，客户端发起请求的都代表客户端需要操作服务器上的资源。2.能满足版本变更和通信安全的需要，对于关键信息，需要使用RSA加密或数字签名。3.加入用户权限校验。4.返回信息简洁，只返回有用的信息。5.接口尽量具有幂等性。以下是申请开通房间接口的详细信息：

|  |  |  |  |
| --- | --- | --- | --- |
| 申请开通房间 | | | |
| 请求地址 | /room/v1 | | |
| 请求类型 | POST | | |
| 返回类型 | JSON | | |
| 请求头参数 | | | |
| Cookie | 用户登录口令 | Content-Type | Application-json |
| 请求参数 | 参数名 | 参数类型 | 约束 |
| 房间名 | name | String | 必填，100字以内 |
| 房间模式 | mode | int | 必填，在0，1中选择 |
| 房间密码 | password | string | 选填 |
| 房间人数限制 | Limits | int | -1默认为无限制 |
| 用途 | purpose | int | 0-投票，1-虚拟导游 |
| 描述 | Description | string |  |
| 加入房间方式 | AttendType | int | 0-输入编号 |
| 返回结果 | | | |
| 返回信息 | data | 房间的内容JSON | |
| 返回状态 | status | 0000代表成功，其他代表失败 | |
| 返回的补充信息 | Msg | 失败的原因 | |