开发文档

### 一、设计文档

#### 前端

技术栈使用Vue + Typescript。

需要实现WebSocket通信（聊天，人物移动，多人互动），基本网页功能（注册登陆功能，选择场景功能），以及导入3D引擎框架并使用（Three.js）。

#### 后端

技术栈使用Springboot + MySQL。

需要实现WebSocket通信的响应（聊天，人物移动，多人互动），基本网页功能的响应（注册登陆功能，选择场景功能）。

#### 美术

使用Blender建模。

需要人物（NPC、玩家）、场景（场馆、天空盒）、物品（画框、冰壶等）的建模。

### 二、实现文档

#### 前端

前端技术栈使用Vue 3.0 + Typescript + Sass + Three.js + Bootstrap。

前端使用观察者（Observer）设计模式来判断用户进入某一矩形区域、用户进入某一圆形区域、前端收到某一websocket消息等。

以用户进入某一矩形区域为例，用户坐标是Observer设计模式中的目标（ConcreteSubject），矩形区域是观察者（ConcreteObserver）。

以用户收到某一websocket消息为例，前端注册的事件是观察者（ConcreteObserver），websocket消息是目标（ConcreteSubject）。

以用户进入某一区域为例，只需要短短的几行代码就可以注册这个事件。

regionDealer.registerEvent(MUSEUM, new class extends UserEnterRegionEvent {

public doEvent() {

that.withinDoor = true

}

}(new RectangleRegion(new Point(-30, CHARACTER\_HEIGHT, -175), new Point(35, CHARACTER\_HEIGHT, -280))))

将重复用到的组件放在了components文件夹中。以聊天框为例，三个场景都需要聊天系统，因而单独编写为一个组件，就可以使用如下的一行代码引入。改善了代码的复用性。

<chat-component ref="chatboard" v-on:msgSend="lockControls()"/>

使用了Sass库，将重复用到的颜色值，属性值等保存为变量，以方便代码复用和主题一致。

$black: #000;

$white: #FFF;

$grey: #777;

$primary-blue: #06F;

$light-blue: #69E;

$dark-yellow: #BB0;

$dark-pink: #F0F;

$half-transparent: rgba(0, 0, 0, 0.5);

$transparent: rgba(0, 0, 0, 0);

#### 后端

后端技术栈使用Springboot + MySQL + Spring JPA。

身份认证采用了jwt技术。登陆时会在响应里返回token。我们自行开发jwt service。它可以从配置文件读取人工设置的有效时间，然后调用alibaba fastjson 和 hutool工具包里的base64编码 sha256签名方法生成token。

多人互动场景和websocket通信使用了springboot带的websocket功能，来完成多人交互功能。用户进入场景的时候会发起websocket连接，接着通过websocket连接发各种控制命令和消息。后端有一个单例维护着所有的websocket连接，并且所有修改它的操作都是被锁保护、线程安全的，所有操作都需要先把锁锁上，做完之后释放锁。

在websocket通信中，每个消息都是一个json串，它必须包含"command"属性。后端为每种command注册了一个处理程序。每次有消息来的时候，会先解析这是什么命令，然后调用对应的处理程序。如果有功能的增加，那么可以新设置一个命令处理程序，不会破坏也不用改动原有代码，可扩展性很强。这是命令模式的设计模式。

#### 美术

全部素材使用Blender建模。分别生成模型、纹理和动画。不在Three.js中使用代码生成任何模型，直接导入模型文件。

### 三、部署文档

服务器位于腾讯云，开放了前端部署端口(80)和后端部署端口(62105)。IP为124.221.255.142。

考虑到数据库需要状态，不能部署在docker里面，我们直接在服务器上安装了mysql-server，并且安装了mysql客户端。

后端DockerFile

FROM java:8

ADD backend.jar backend.jar

ENTRYPOINT ["java","-jar","backend.jar"]

前端DockerFile

FROM nginx:latest

COPY dist/ /usr/share/nginx/html/