

期中项目说明

期中PJ完成前后端部分，考虑到课堂上尚未讲述Web3D相关内容，放在期末PJ中进行

前端使用Vue完成

后端使用SpringBoot完成，可以参考下发资料的“基于Spring Boot & MyBatis的种子项目”

前端说明

所有标签使用中文

除登陆和注册页面以外，需要提供一个顶部的导航栏，同时需要权限验证，不通过则跳转至登陆界面

导航栏

需要个人中心和3D场景选择两个链接，跳转至对应页面

在最右侧显示用户名，点击弹出下拉框，下拉框中包含退出登录按钮，点击退出登录并跳转至登陆界面

登陆界面(/login)

登陆需要用户名和密码

需要提供链接跳转至注册页面

登陆成功自动跳转至3D场景选择页面

登陆

用户名：

密码：

跳转至注册页面

错误提示

提交

注册页面(/register)

尽可能与登陆页面一致

注册需要用户名和密码

需要再次输入密码检查是否一致

需要提供链接跳转至登陆页面

注册成功自动跳转至登陆页面

注册

用户名：

密码：

跳转至登陆页面

错误提示

提交

个人中心页面(/user/personalCenter)

展示个人信息（内容自定义，参考接口规范中后端返回内容）

同时在一个canvas标签上显示所选化身

化身暂时只需要一个agent_id，以后通过agent_id拉取模型展示在canvas上

暂时不需要提供修改信息功能

导航栏

个人信息

用户名：

密码：

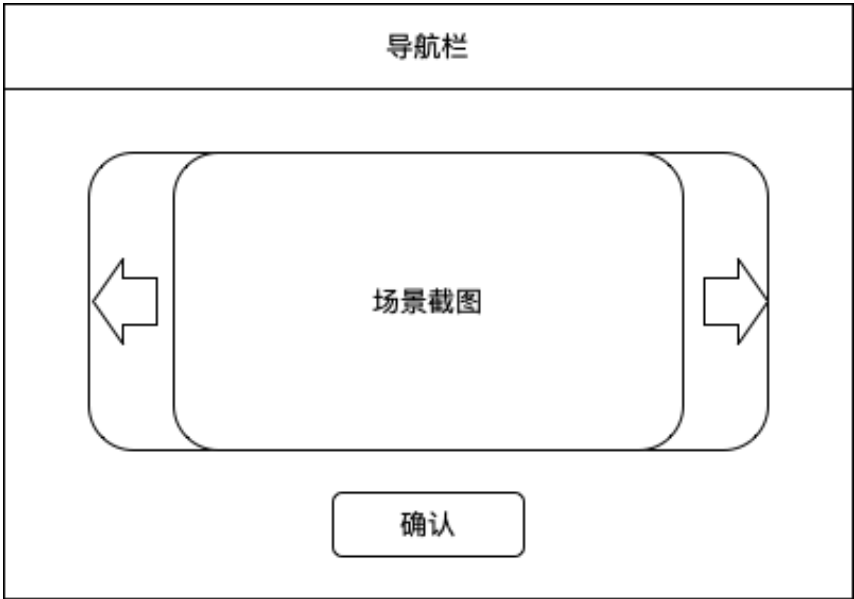
化身选择

canvas

确认

3D场景选择页面(/scene/selectScene)

考虑到项目仅要求几个案例，暂时打算使用轮播图显示，不提供搜索功能

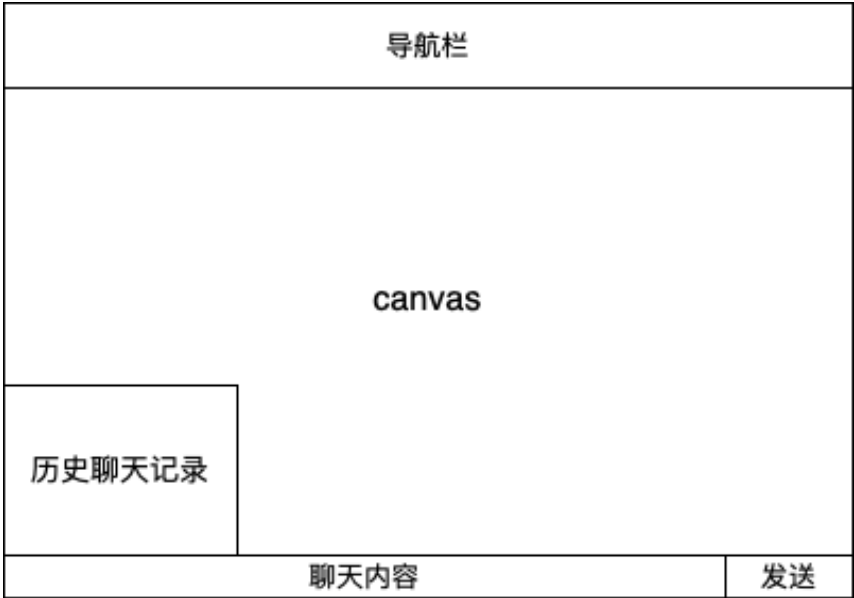


场景页面(/scene/getScene?id={sceneId})

用sceneId表示指定场景

暂时只需要一个canvas(用于显示场景，暂时不需要显示内容)和一个聊天框

历史行为查看和分析功能待定



数据库

User

```
user_id          // 主键
username         // 用户名, 不可重复
password         // 密码
register_time    // 注册时间
```

默认用户: 1 testUser test 2022.5.2

Scene

```
scene_id         // 主键
name            // 场景名称
path            // 场景截图存储路径
```

默认场景: 1 testScene 自己找一张图片写路径

后端说明

使用JWT作权限验证, 在登陆成功时向前端返回一个token

token中数据部分需要包含userId

除注册登陆外, 每次都需要检测token是否存在且合法

接口规范

使用JSON传递数据

除注册登陆以外, header需包含token, 在下文请求规范中省略

后端返回信息需要有两个统一的属性: code和message, 在下文响应规范中省略

code表示请求是否发生错误, 无错误为200, 有错误为400 (注意与是否通过验证区分, 验证使用success)

message表示返回信息, 例如400时为错误原因, 200时为通过/不通过信息

发生错误时后端仅需返回以上两个属性

```
{
  code: int,
  message: string,
  data: {
    ... // 下述响应中的内容在此处获取
  }
}
```

注册

URI: /user/register

方法: POST

请求:

```
{
  username: string,
  password: string,
  repassword: string
}
```

响应:

```
{
  success: boolean // 是否注册成功
}
```

登陆

URI: /user/login

方法: POST

请求:

```
{
  username: string,
  password: string
}
```

响应:

```
{
  success: boolean // 是否登陆成功
  token: string // 仅登陆成功时需要, 作为JWT
}
```

用户信息

URI: /user/get_user

方法: POST

请求:

```
{
}
```

响应：

```
// 根据token中userId确认用户
{
  success: boolean,
  userDetails:
  {
    userId: int,
    username: string,
    registerTime: string
  }
}
```

获取所有场景

URI: /scene

方法: GET

请求：

```
{
}
```

响应：

```
{
  success: boolean,
  sceneList: [
    {
      sceneId: int,
      name: string,
      path: string
    },
    ...
  ]
}
```

获取单个场景

URI: /scene/{sceneId}

方法: GET

请求：

```
{  
}
```

响应:

```
{  
  success: boolean,  
  scene:  
    {  
      sceneId: int,  
      name: string,  
      path: string  
    }  
}
```