

# web3d项目文档

---

复旦大学2022春季高级Web技术PJ

小组成员：

- 李洋圳
- 袁逸聪
- 赵书誉
- 袁浩

## 概述

### 项目内容

本项目是一个可供多人在线的Web3D教学式场景，参与者进入虚拟场景，与棋盘交互，共同尝试解决八皇后问题。

点击[项目链接](#)进行体验~

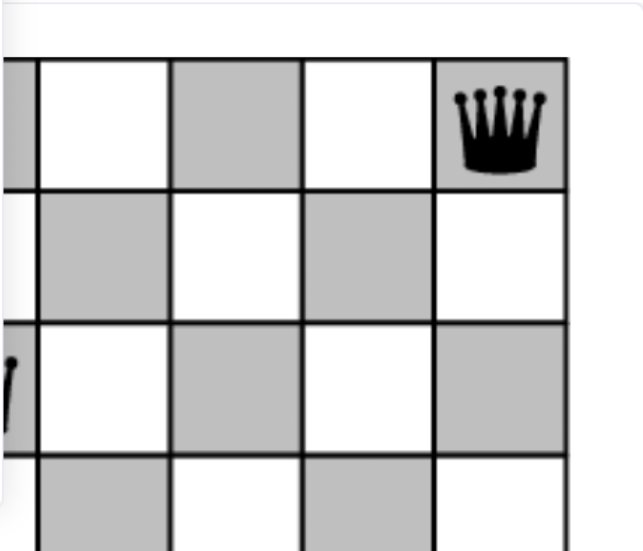
玩家还可以在用户中心选择自己喜欢的虚拟形象，或者听天由命随机！

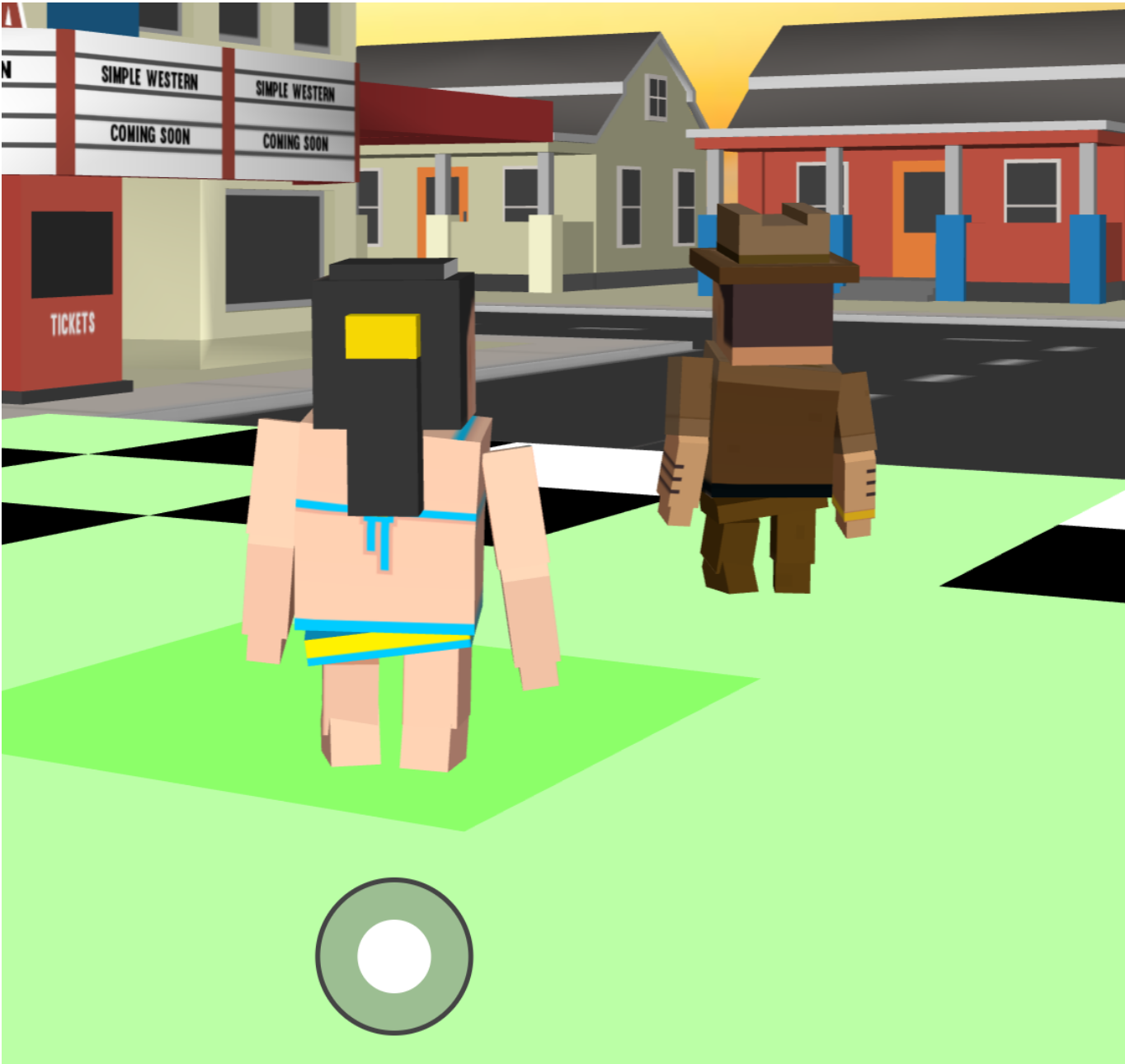
# VirtualFigure

Choose your character!

BeachBabe ^

- Random
- BeachBabe
- BusinessMan
- Doctor
- FireFighter
- Housewife
- Policeman
- Prostitute





工期安排

阶段	计划时间	工作内容	完成情况	备注
需求规划	4/12-4/13	确定项目内容，产出项目文档	已完成	
前台搭建	4/13-4/20	完成账号管理及3D场景选择功能	迟滞	前端进度完成至注册登录跳转、搭建持久化及接口调用工具，需补充用户中心页面；后端进度完成至鉴权以外部分
前台搭建-补	4/21-4/24	补完前台搭建需求及双端测试	已完成	还需要将项目部署到云端

阶段	计划时间	工作内容	完成情况	备注
Web3D探索	4/25-5/1	探索技术栈并确定虚拟场景的同步方案	已完成	ThreeJs学习, 讨论虚拟场景方案
虚拟场景	5/2-6/12	完成虚拟场景搭建	已完成	棋盘交互逻辑、多人同步、聊天室及场景搭建
功能补充及文档完善	6/13-6/18	完善场景交互和统计功能	已完成	

## 分工

- 需求规划：共同完成
- 前台搭建
  - 前端：李洋圳 袁逸聪（需求对接）
  - 后端：赵书誉 袁浩（需求对接）
- 虚拟场景：袁浩 袁逸聪
- 具体负责模块
  - 李洋圳：前端搭建，前端页面美化
  - 赵书誉：数据库搭建，后端搭建，后端结构图绘制
  - 袁浩：虚拟场景环境与人物模型，socketio同步，后端搭建，项目部署
  - 袁逸聪：虚拟场景棋盘，socketio同步，前端搭建，前端页面美化，文档与接口管理

## 目录结构

项目共分为3个部分

- backend：springboot后端，管理持久化数据
- frontend：vue前端，包含threejs页面渲染
- socketio：socketio的server部分，转发threejs数据进行同步

## 后端结构

核心代码在/src/main/java/com/company/project下

- /configurer：项目初始化配置
- /core：共同核心功能
- /dao：数据库接口
- /model：数据模型

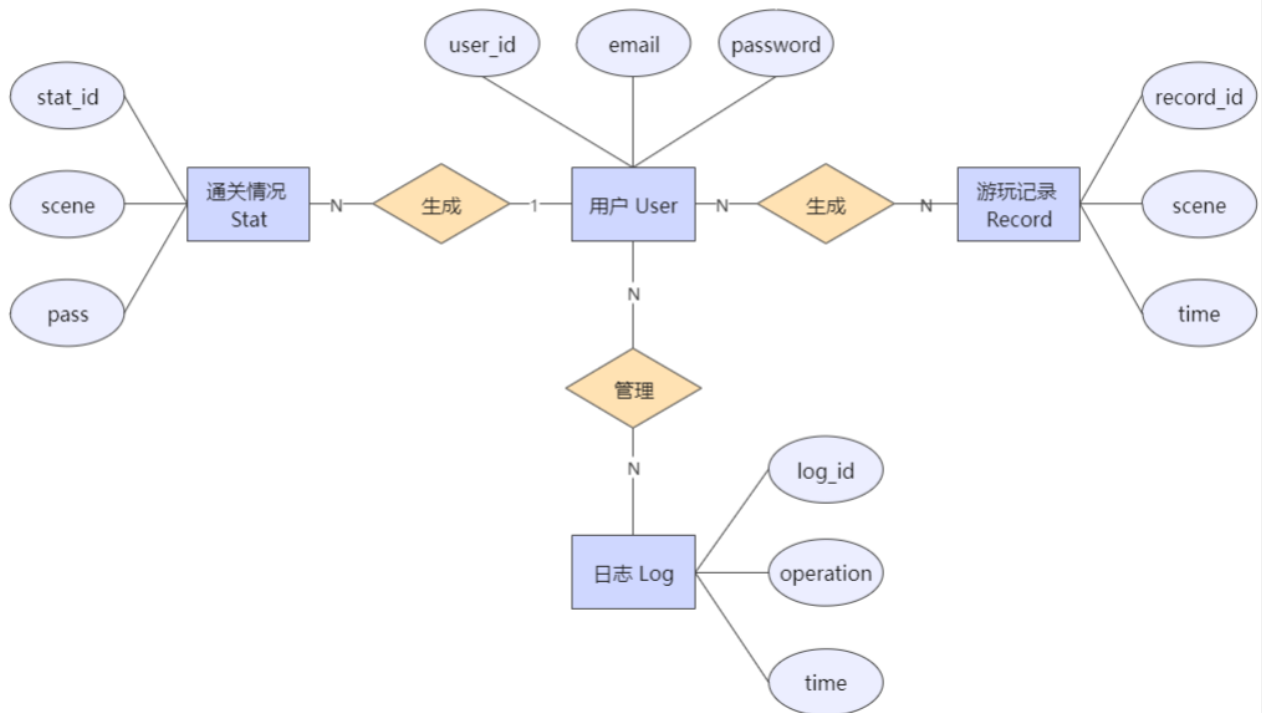
- /service: 功能实现类
- /util: 工具
- /web: controller类, 对外提供接口
- Application.java:项目启动入口

本项目的核心在于虚拟场景, 因此后端采用简单设计, 有待增加虚拟场景数量后拓展。

后端提供的主要功能是用户管理与虚拟场景使用情况统计。

- 用户
  - 身份验证
    - 使用临时生成的token, 后端存储token-用户对应关系, 前端发送请求携带token以表明身份
- 虚拟场景
  - 使用情况
    - 记录虚拟场景的使用次数, 以表征其热度
  - 通过情况
    - 记录虚拟场景的通关比例, 以表征其难度

#### 数据库结构

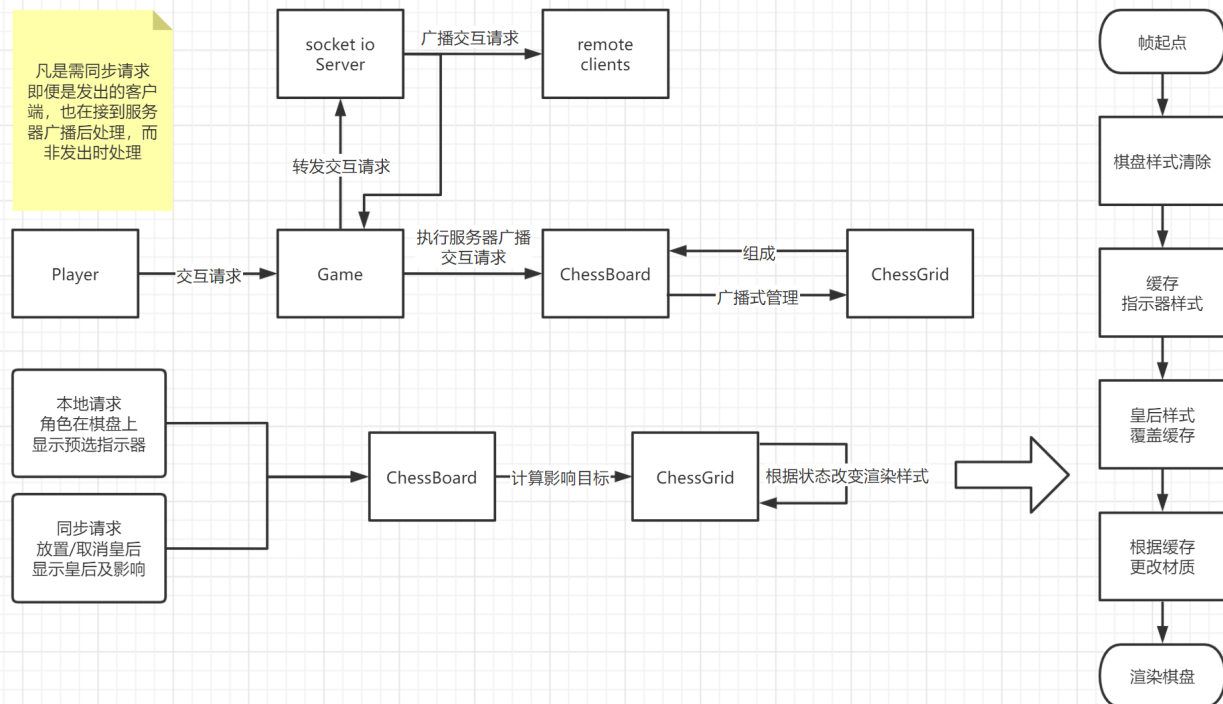


#### 前端结构

- ChessBoard
  - 由ChessGrid组成, 根据状态有不同外显
    - 皇后: 深紫色
    - 皇后影响范围: 淡紫色
    - 预选指示器中心: 深绿色
    - 预选指示器影响范围: 浅绿色
  - 棋盘接受外部请求, 管理所有格子的状态
    - 本地请求: 预选则指示器, 本地渲染, 指示将要覆盖的格子

- 同步请求：放置或拿走皇后，发送请求到socketio server，server转发给所有客户端一起处理

## 设计概览



## socketIO结构

核心代码见/node/app.js

socket io开发非常方便！内容全部写在一个文件也不显得臃肿

```
> socket io server: 监听时间并处理，如转发等
io.sockets.on('connection', function(socket){
  socket.on('update', function(data){
    socket.userData.x = data.x;
    socket.userData.y = data.y;
    socket.userData.z = data.z;
    socket.userData.heading = data.h;
    socket.userData.pb = data.pb,
    socket.userData.action = data.action;
  });
});
```

```
> socket io client:发送与监听事件
socket.emit('update',...)
socket.on('update',...)
```

## 项目需求

## 前台

### 1. 注册与登录

1. 账号由邮箱、密码组成
2. 密码要求6-18位，由数字、大写字母、小写字母组成，至少包含其中两种
3. 邮箱不可重复注册，提供重复检验与回显提示

### 2. 用户后台

1. 用户后台提供选择进入虚拟场景的链接
2. 用户后台显示邮箱
3. 用户后台可预览进入虚拟场景所用的虚拟形象
4. 用户后台显示各场景使用历史

### 3. 虚拟场景入口

1. 虚拟场景提供回到用户后台的链接
2. 本阶段不必实现场景

## 虚拟场景

虚拟场景为多人在线可交互的八皇后问题

### 1. 棋盘

1. 黑白相间的国际象棋棋盘
2. 本地功能：本地玩家角色处在棋盘格子上时，显示预选指示器，表明选中的格子与选中后将被覆盖的格子范围，指示器仅在本地生效
3. 同步功能：玩家角色处在棋盘格子上时，可以放置或拿走皇后
  1. 皇后覆盖范围内的格子无法交互
  2. 非皇后格子交互后放置皇后，同步给所有玩家
  3. 皇后格子交互后拿走皇后，同步给所有玩家

### 2. 玩家角色

1. 玩家角色只能在平面上移动，通过改变自身位置选中脚下的棋盘格
2. 本地玩家角色
  1. 本地玩家角色附带本地相机，相机以第三人称视角围绕角色转动
  2. 需要限制相机的转动范围，使其不越过地面
3. 同步玩家角色
  1. 客户端还需要同步显示其他玩家角色的位置，但不需要渲染相机

### 3. 聊天室

1. 聊天界面固定显示在屏幕下方，打开时遮盖游戏场景，不改变场景大小
2. 按下回车打开聊天界面，打开时输入框获取焦点
3. 输入内容再按下回车即发送，信息同步给所有在线玩家
4. 输入框中无内容时按下回车关闭输入框

## 前端页面

1. 主页：供未登录用户浏览，介绍项目情况，并提供注册、登录页面的链接
2. 注册、登录：提供注册、登录功能，成功后跳转到用户后台
3. 用户后台：提供虚拟场景入口，可以查看虚拟场景使用历史
4. 虚拟场景

## 规范及接口

编码规范

本项目统一采用小驼峰法传输变量，对前端、后端及数据库都适用

对常用变量命做如下规定

变量	变量名
邮箱	email
密码	password
鉴权	token

前后端接口

本项目采用RESTFUL风格API规范

HTTP Method	含义
GET	获取
POST	创建
PUT	更新
DELETE	删除

接口一览

- 注册登录
  - 注册，创建账号
  - 查看邮箱重复性，查看账号
  - 登录，创建token
  - 登出，删除token
- 用户后台
  - 创建虚拟场景进入记录
  - 查看虚拟场景进入记录
- 虚拟场景
  - 用户进入虚拟场景
  - 用户通过虚拟场景
  - 查看虚拟场景进入情况
  - 查看虚拟场景通过情况

接口详情

此处展示所有接口通用的返回参数

接口实际返回参数为通用参数加上各自的输出参数

参数	类型	含义	必填	示例
----	----	----	----	----



参数	类型	含义	必填	示例
alert	string	需要回显的内容	否 需要回显时给出	服务器繁忙，请重试

查看邮箱重复性

`${ip}:${port}/api/user/account GET`

输入参数

参数	类型	含义	必填	示例
email	string	所查看邮箱	是	19302010020@fudan.edu.cn

输出参数

参数	类型	含义	必填	示例
email	string	查看的邮箱	否 存在则有	19302010020@fudan.edu.cn

注册

`${ip}:${port}/api/user/account POST`

输入参数

参数	类型	含义	必填	示例
email	string	注册邮箱	是	19302010020@fudan.edu.cn
password	string	注册密码	是	abcd1234

登录

`${ip}:${port}/api/user/token POST`

输入参数

参数	类型	含义	必填	示例
email	string	登录邮箱	是	19302010020@fudan.edu.cn
password	string	登录密码	是	abcd1234

输出参数

参数	类型	含义	必填	示例
token	int	鉴权信息	否 成功则有	114514

登出

`${ip}:${port}/api/user/token DELETE`

输入参数

参数	类型	含义	必填	示例
token	int	鉴权信息	是	114514

创建用户历史记录

`${ip}:${port}/api/history POST`

输入参数

参数	类型	含义	必填	示例
token	int	鉴权信息	是	114514
scene	string	使用的虚拟场景	是	汉诺塔

获取用户的历史记录

`${ip}:${port}/api/history GET`

输入参数

参数	类型	含义	必填	示例
token	int	鉴权信息	是	114514

用户进入虚拟场景

`${ip}:${port}/api/record POST`

输入参数

参数	类型	含义	必填	示例
token	int	鉴权信息	是	114514
scene	string	使用的虚拟场景	是	汉诺塔

查询虚拟场景进入记录 TODO

`${ip}:${port}/api/record/all GET`

输入参数

参数	类型	含义	必填	示例
token	int	鉴权信息	是	114514
scene	string	使用的虚拟场景	是	汉诺塔

输出参数

参数	类型	含义	必填	示例
----	----	----	----	----

用户通过虚拟场景

`${ip}:${port}/api/stat` POST

输入参数

参数	类型	含义	必填	示例
token	int	鉴权信息	是	114514
scene	string	使用的虚拟场景	是	汉诺塔

查询虚拟场景游玩通过情况

`${ip}:${port}/api/stat/all` GET

输入参数

参数	类型	含义	必填	示例
scene	string	使用的虚拟场景	是	汉诺塔

输出参数

参数	类型	含义	必填	示例
data	array of integer	游玩人数与通过人数	是	[7,1]