****

**软件工程实践（二）**

|  |  |
| --- | --- |
| **学 院** | **计算机与人工智能学院** |
| **专 业** | **软件工程** |
| **班 级** | **软件2002** |
| **姓 名** | **李冬俊，任文斌，范春宇，陈奥** |
| **指导教师** | **唐祖锴** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2023 | 年 | 6 | 月 | 26 | 日 |

目录

[Colossal Cave Adventure开发文档 2](#_Toc138546704)

[一、 课程任务 2](#_Toc138546705)

[任务目的 2](#_Toc138546706)

[实践任务 3](#_Toc138546707)

[主要技术与工具 3](#_Toc138546708)

[软件架构设计 3](#_Toc138546709)

[小组管理 4](#_Toc138546710)

[小组软件过程与项目管理 4](#_Toc138546711)

[软件分支管理 4](#_Toc138546712)

[软件代码评审 4](#_Toc138546713)

[持续集成与部署 4](#_Toc138546714)

[二、 程序设计与开发 4](#_Toc138546715)

[扩展功能点的确定 4](#_Toc138546716)

[功能点要求 4](#_Toc138546717)

[功能点确定 5](#_Toc138546718)

[程序开发 7](#_Toc138546719)

[控制台程序 7](#_Toc138546720)

[网页游戏 8](#_Toc138546721)

[数据库设计 10](#_Toc138546722)

[三、 开发版本计划与任务分派 10](#_Toc138546723)

[开发版本计划 10](#_Toc138546724)

[任务分派 12](#_Toc138546725)

[控制台程序 12](#_Toc138546726)

[网页游戏 12](#_Toc138546727)

[四、 开发分支模型与代码合并 13](#_Toc138546728)

[五、 开发规范与测试 15](#_Toc138546729)

[开发规范 15](#_Toc138546730)

[测试 17](#_Toc138546731)

[/games/data 17](#_Toc138546732)

[/games/saveData 18](#_Toc138546733)

[/user 19](#_Toc138546734)

[/users/login 20](#_Toc138546735)

[Action 22](#_Toc138546736)

[六、 集成与版本发布 24](#_Toc138546737)

[七、 小组成员个人任务 25](#_Toc138546738)

Colossal Cave Adventure开发文档

# 课程任务

## 任务目的

* 巩固强化软件编程规范
* 提高面向对象软件建模与抽象能力
* 培养小组协同开发能力
* 掌握基于Maven的软件项目管理机制
* 掌握基于Github的小组协同开发工具和平台
* 了解DevOps软件开发流程

## 实践任务

基于软件工程实践一中完成的项目基础，以小组为开发单位，采用迭代递进的实践过程逐步掌握课程内容

## 主要技术与工具

* 代码规范与注释
* UML与面向对象程序设计
* Git & GitHub
* 个人代码版本管理
* 小组协同开发与代码分支集成
* MarkDown文档编制
* JUnit单元与集成测试
* Maven项目工程管理
* DevOps软件开发过程

## 软件架构设计

本次实验采用HTML+CSS+JavaScript+Axios+PIXI.JS+SpringBoot+Mybatis进行开发。

HTML：用于网页结构的编写，主要是登录页面和游戏页面。

CSS：用于页面的美化效果。

JavaScript：游戏中实现各种功能的逻辑以及与后端进行交互。

Axios：向后端发送请求。

1. JS：作为游戏引擎加速，用于构建房间，道路，人物等精灵。

SpringBoot：作为后端框架，处理游戏初始化的一些数据。

MyBatis：数据库相关的操作。

## 小组管理

### 小组软件过程与项目管理

主要内容：基于DevOps体系，针对软件工程实践目标采用适当的小组软件过程与项目管理工具进行项目开发与管理。

实践重点：小组软件过程开发阶段的划分、项目管理平台的搭建与使用、开发任务的分派与跟踪。

### 软件分支管理

主要内容：了解并掌握主要软件分支管理模型的概念，及相关工具在软件开发中的应用

实践重点：主分支、发布分支、开发分支、debug分支、代码冲突与解决。

### 软件代码评审

主要内容：了解并掌握代码评审的概念与方法，及相关工具在软件开发中的应用。

实践重点：同行评审、结对编程、code review。

### 持续集成与部署

主要内容：了解并掌握持续集成与部署的概念与方法，及相关工具在软件开发中的应用。

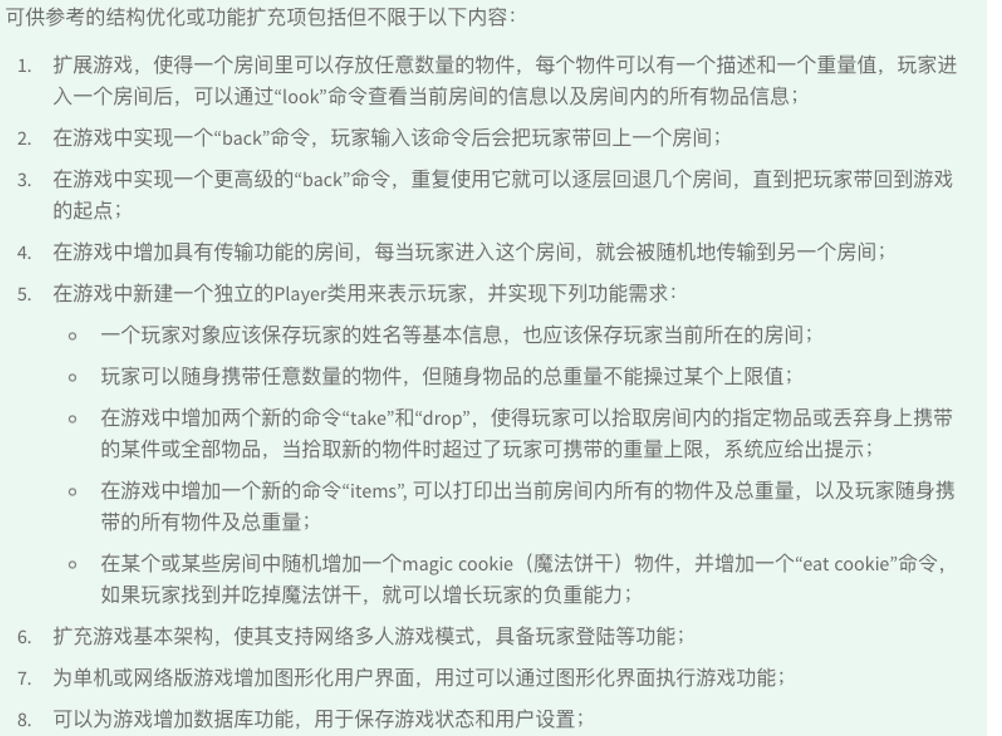
实践重点：自动化测试、持续集成脚本、容器管理与运行、软件打包与发布。

# 程序设计与开发

## 扩展功能点的确定

### 功能点要求

样例工程“world-of-zuul”具备最基本的程序功能，开发小组内可进行沟通讨论，确定系统结构优化需求或功能扩充需求，结构优化或功能扩充项不能少于5项；扩充后的软件应包含用户交互界面及功能服务模块，以提供便捷的功能操作。



### 功能点确定

通过小组讨论，我们小组将开发图形化的游戏界面玩法和控制台玩法。两个版本分开开发，实现下面描述的功能点。

#### 控制台游戏

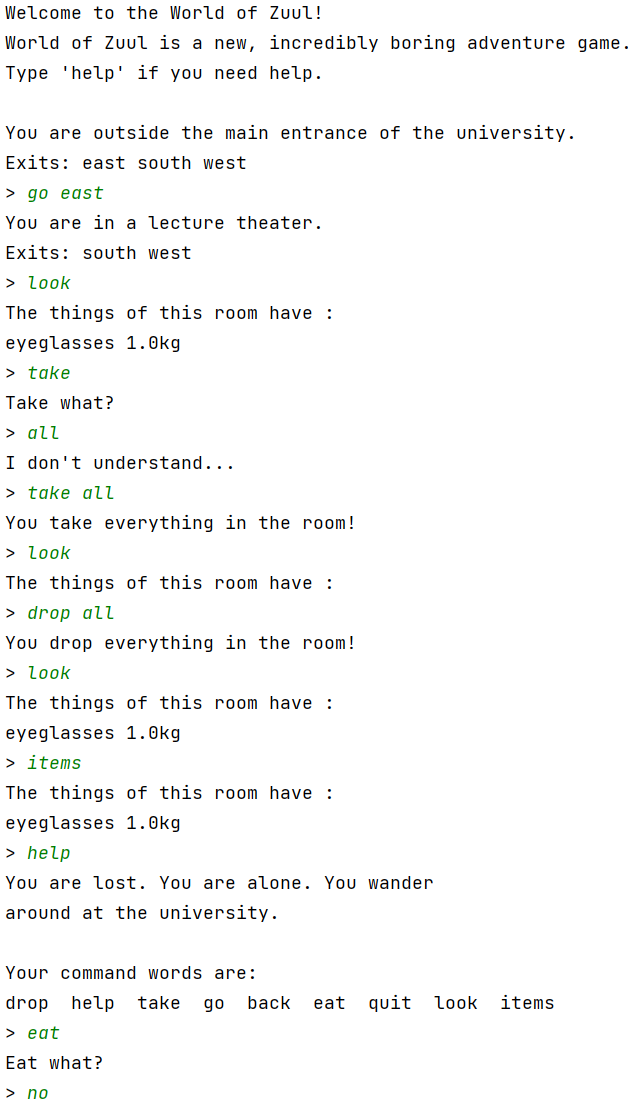
1. look命令。每个房间可以存放任意数量的物件，每个物件有一个描述和一个重量值，玩家进入一个房间之后，可以通过“look”命令查看当前房间的信息以及房间内所有物品信息。
2. 高级的back命令。当用户输入"back"命令之后，可以把玩家待会上一个房间，重复使用可以逐层回退几个房间，直到把玩家带回到游戏的起点。
3. 在游戏中增加具有传输功能的房间，每当玩家进入这个房间之后就会随机传送到另外一个房间。
4. 玩家可以携带任意数量的物件，但是随身携带物件的总重量不能超过某个上限。
5. take命令。玩家可以拾取房间内的指定物品，当拾取的物件超过了玩家可以携带的重量上线，则不能继续拾取，给出提示。
6. drop命令。玩家可以丢弃身上携带的物品，同时丢弃的物件将会放到当前房间中。（即房间中的物件数量将会增多）
7. items命令。打印当前房间内的所有物品信息，包括玩家携带的物件信息。
8. eat cookie命令。在房间中增加magic cookie魔法饼干，当用户吃掉魔法饼干，玩家的负重能力增加。

#### 网页游戏

1. back命令。当用户点击back命令时，将玩家带回上一个房间。连续点击可以逐层返回直到返回起点。
2. look命令。当玩家进入一个房间之后，通过点击look命令可以查看当前房间中的物件数量
3. transmit命令。当玩家点击transmit命令之后，将提供游戏中的所有房间供玩家选择。玩家点击哪个房间将被床送到这个房间中。
4. take命令。玩家可以拾取房间中的物品，但是不能超过玩家背包上限。
5. drop命令。玩家可以丢弃背包中的物品到当前所在的房间，但是丢弃的数量不能多余玩家携带的数量。
6. items命令。玩家可以点击items命令获取当前房间的物品信息和玩家携带的物品信息。
7. magic cookie。随机在两个房间中生成魔法饼干，当玩家找到并点击魔法饼干之后，将增加玩家的背包容量。
8. 应该具备保存游戏功能。保存当前玩家的所在房间。
9. 具备登录和注册功能。
10. 具备退出登录，保存游戏功能

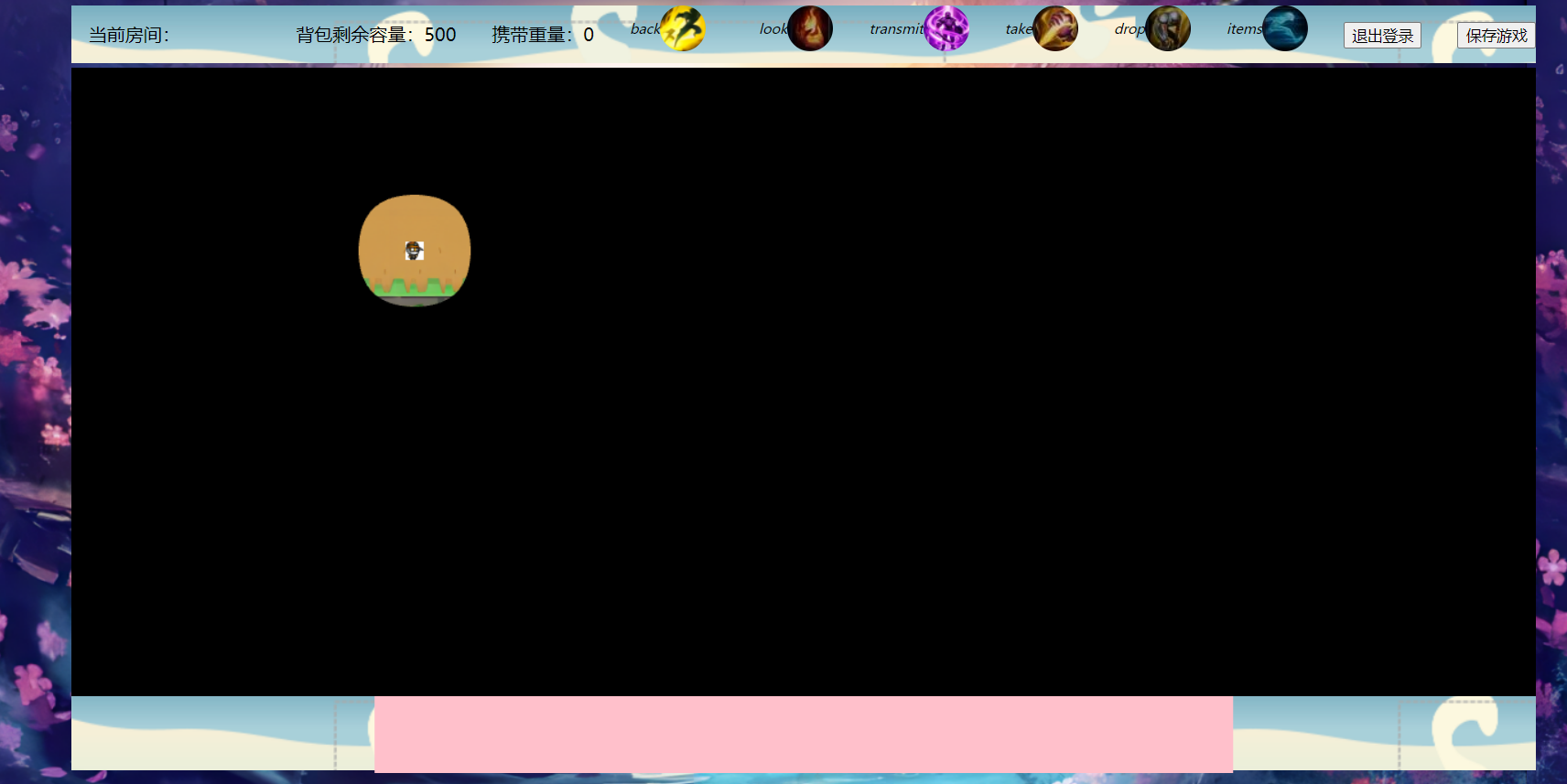
## 程序开发

### 控制台程序

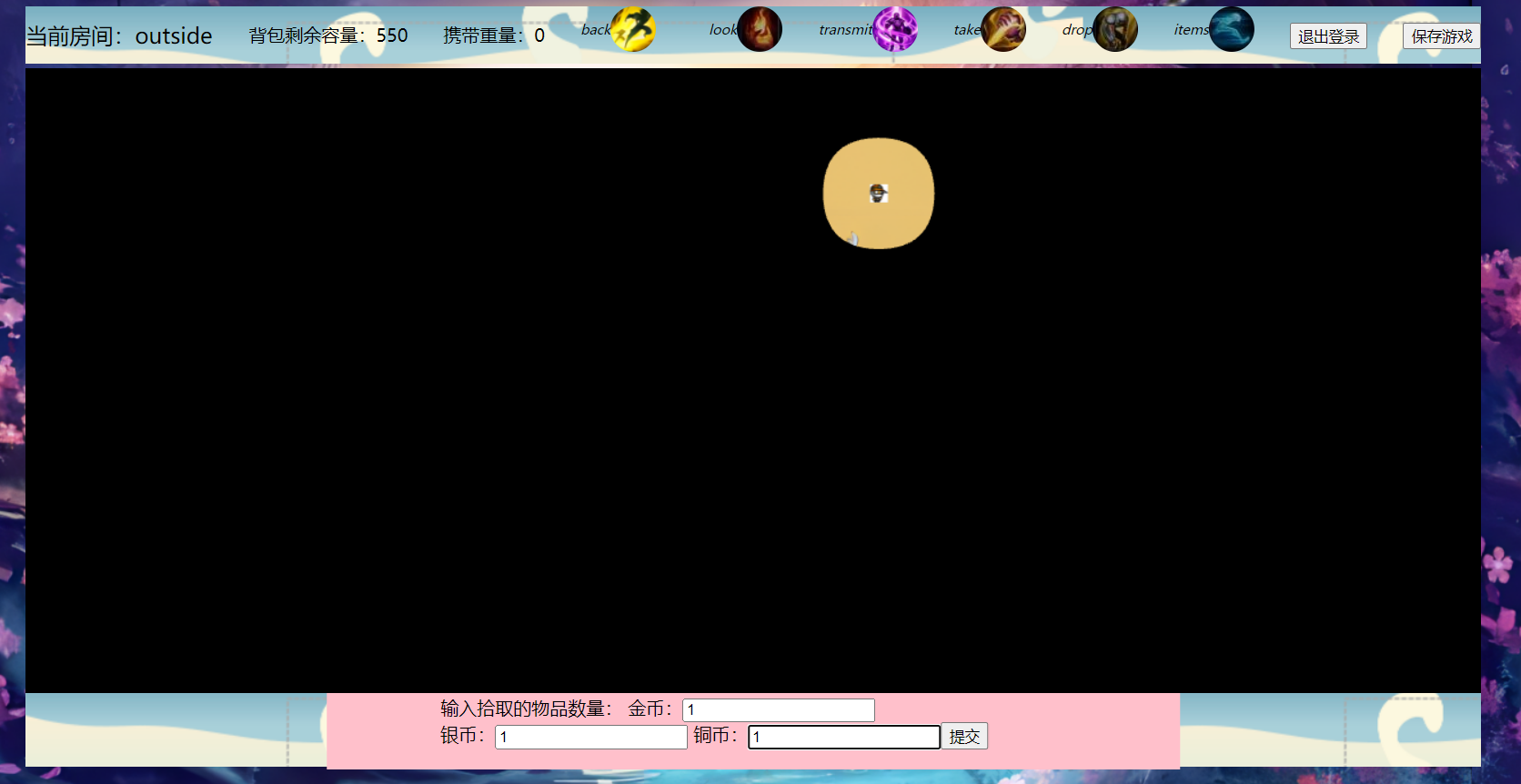


### 网页游戏



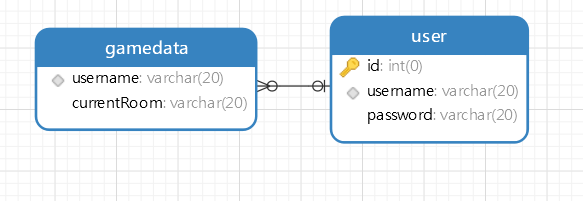








### 数据库设计



# 开发版本计划与任务分派

## 开发版本计划

本次程序将开发控制台游戏和网页游戏，两种类型的游戏分开开发，最后进行功能集成。

1、第一阶段：

第1天：初步建立起项目需要的系统应用与运行机制，完成业务梳理，明确设计模块，开展项目的详细需求分析和设计工作。

2、第二阶段

第2天：在详细设计的基础上，进行项目的实施开发；

控制台程序的开发需要理清代码的逻辑关系，根据规范编写注释。尤其是在设计Player类时，需要综合考虑进行合理的设计，使各个模块之间的调用简单便捷。

网页游戏的开发需要进行前期的地图设计，使用TexturePackerGUI和唤镜引擎进行地图的设计。使用PIXI.JS对游戏的地图，道路和门进行精灵的创建。在这个过程中由于需要直到各个地图元素的坐标信息，需要使用PxCook进行精确的尺寸设计。还需要寻找人物，魔法饼干，房间中的物件的图片素材。

3、第三阶段

第3天：完成控制台游戏的所有功能扩展点的开发，并pull到各自的分支。网页游戏需要对登录界面和游戏界面的主体设计，包含HTML，CSS代码。

4、第四阶段

第4-5天：完成对控制台游戏功能的测试，将其集成到SpringBoot框架下的程序中。网页游戏对游戏界面的逻辑进行编写，主要包括：

* 游戏场景精灵和传送场景精灵的加载
* 人物精灵加载
* 遮罩层设计和人物视野设计
* 加载门，路径，房间等游戏场景数据
* 从后端加载游戏数据，主要包括房间中的物件等信息，根据这些信息创建游戏精灵
* back,look,transmit,take,drop,items命令的逻辑
* 键盘移动的鼠标移动以及约束探险者在道路之内

5、第五阶段

第6-7天：完成对网页游戏数据库功能的编写，主要对接口进行测试。将代码集成之后进行整体测试。

## 任务分派

### 控制台程序

任文斌

* look命令。每个房间可以存放任意数量的物件，每个物件有一个描述和一个重量值，玩家进入一个房间之后，可以通过“look”命令查看当前房间的信息以及房间内所有物品信息。
* 高级的back命令。当用户输入"back"命令之后，可以把玩家待会上一个房间，重复使用可以逐层回退几个房间，直到把玩家带回到游戏的起点。
* 具有传输功能的房间。每当玩家进入这个房间之后就会随机传送到另外一个房间。
* 实现Player类，玩家可以携带任意数量的物件，但是随身携带物件的总重量不能超过某个上限。
* take命令。玩家可以拾取房间内的指定物品，当拾取的物件超过了玩家可以携带的重量上线，则不能继续拾取，给出提示。
* 范春宇

1. drop命令。玩家可以丢弃身上携带的物品，同时丢弃的物件将会放到当前房间中。（即房间中的物件数量将会增多）
2. items命令。打印当前房间内的所有物品信息，包括玩家携带的物件信息。
3. eat cookie命令。在房间中增加magic cookie魔法饼干，当用户吃掉魔法饼干，玩家的负重能力增加。
4. 控制台玩法应该具备和前端交互的功能。应该对控制台程序中的命令解析器(Parse类)进行适当的修改，使得当在前端页面上提交命令之后，该命令能被正确的解析。
5. 程序的输出应该能提交到前端页面展示。
6. 应该将原始的在控制台界面输入输出的方式修改为在前端页面上进行输入输出，以期提供友好的用户交互界面。网页玩法应该实现的功能

### 网页游戏

* 陈奥

1. 设计游戏地图，包括道路，房间，墙
2. 设计游戏技能图标
3. 设计游戏精灵（包括人物，房间中的物件，魔法饼干，房间等），并使用Pixi.js游戏引擎。其中房间中的物件有：金币，银币，铜币。
4. 设计人物的属性，包括携带的金币数。银币数，铜币数，携带的重量，背包的容量。
5. 用户可以通过鼠标点击移动玩家，也可以通过键盘移动玩家。
6. 游戏地图，人物，物件的设计使用工具TexturePackerGUI设计。

* 李冬俊

1. back命令。当用户点击back命令时，将玩家带回上一个房间。连续点击可以逐层返回直到返回起点。
2. look命令。当玩家进入一个房间之后，通过点击look命令可以查看当前房间中的物件数量
3. transmit命令。当玩家点击transmit命令之后，将提供游戏中的所有房间供玩家选择。玩家点击哪个房间将被床送到这个房间中。
4. take命令。玩家可以拾取房间中的物品，但是不能超过玩家背包上限。
5. drop命令。玩家可以丢弃背包中的物品到当前所在的房间，但是丢弃的数量不能多余玩家携带的数量。
6. items命令。玩家可以点击items命令获取当前房间的物品信息和玩家携带的物品信息。
7. magic cookie。随机在两个房间中生成魔法饼干，当玩家找到并点击魔法饼干之后，将增加玩家的背包容量。
8. 应该具备保存游戏功能。保存当前玩家的所在房间。
9. 具备登录和注册功能。
10. 具备退出登录，保存游戏(玩家名和当前所在房间)功能

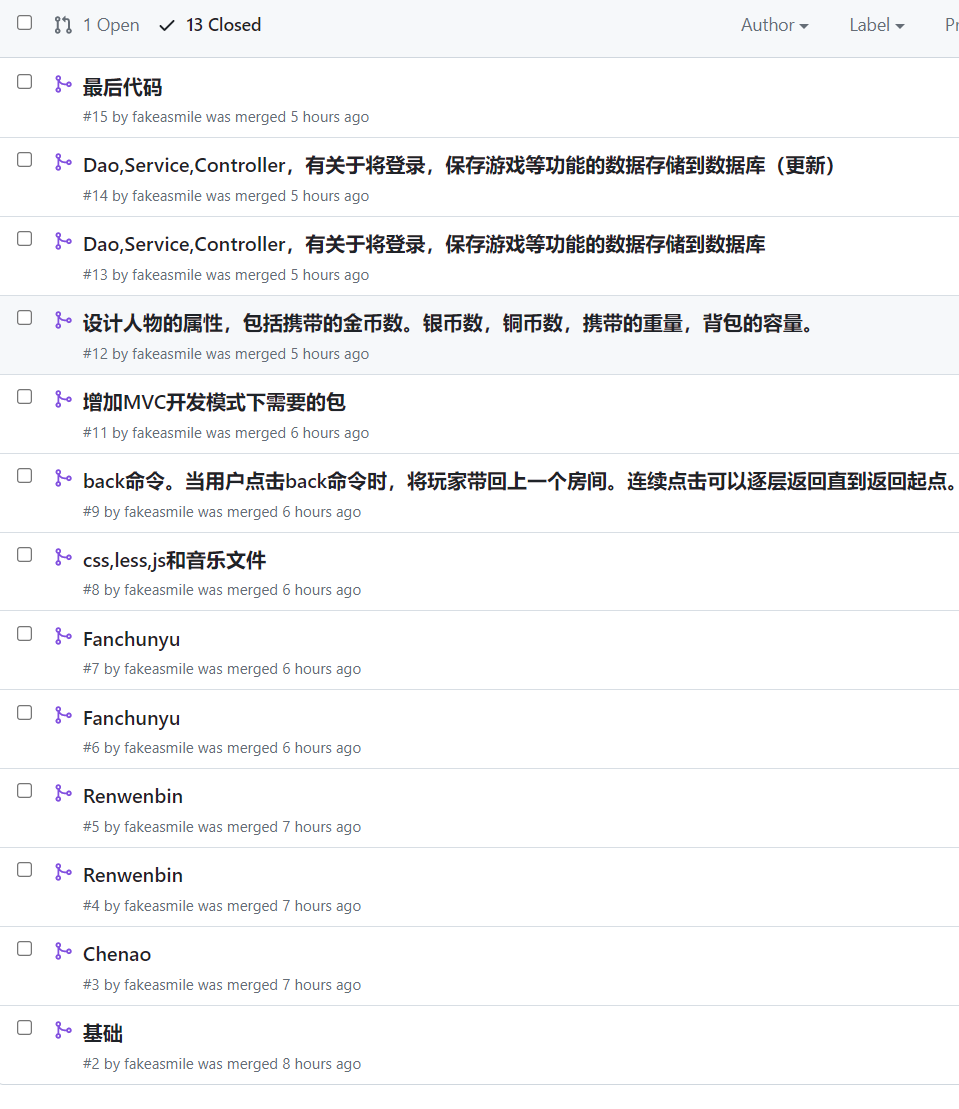
# 开发分支模型与代码合并

在我们小组的GitHub 的团队协作中，分支模型采用 Git Flow 模型。将开发过程分为两个主要分支：主分支（master）和开发分支（develop）。主分支用于发布稳定版本，开发分支用于日常开发和提交。每个提交到开发分支上的代码都将进行测试之后才能最终合并到主分支上，发布一个稳定的版本。每个新功能都应该在一个新的分支上进行开发，然后将其合并到 develop 分支中。当 develop 分支中的代码稳定后，可以将其合并到 master 分支中发布一个新的稳定版本。

小组分支模型：



分支与代码合并：



# 开发规范与测试

## 开发规范

本次实验全部代码都基于javadoc规范，其中要求：

* 排版：

1.关键词和操作符之间加适当的空格。

2.相对独立的程序块与块之间加空行

3.较长的语句、表达式等要分成多行书写。

4.划分出的新行要进行适应的缩进，使排版整齐，语句可读。

5.长表达式要在低优先级操作符处划分新行，操作符放在新行之首。

6.循环、判断等语句中若有较长的表达式或语句，则要进行适应的划分。

7.若函数或过程中的参数较长，则要进行适当的划分。

8.不允许把多个短语句写在一行中，即一行只写一条语句。

9.函数或过程的开始、结构的定义及循环、判断等语句中的代码都要采用缩进风格。

* 命名：

1.较短的单词可通过去掉“元音”形成缩写；

2.较长的单词可取单词的头几发符的优先级，并用括号明确表达式的操作顺序，避免使用默认优先级。

3.采取驼峰命名法

* 可读性：

1.避免使用不易理解的数字，用有意义的标识来替代。

2.不要使用难懂的技巧性很高的语句。

3.源程序中关系较为紧密的代码应尽可能相邻。

* 变量：

1.去掉没必要的公共变量。

2.构造仅有一个模块或函数可以修改、创建，而其余有关模块或函数只访问的公共变量，防止多个不同模块或函数都可以修改、创建同一公共变量的现象。

3.仔细定义并明确公共变量的含义、作用、取值范围及公共变量间的关系。

4.明确公共变量与操作此公共变量的函数或过程的关系，如访问、修改及创建等。

5.当向公共变量传递数据时，要十分小心，防止赋与不合理的值或越界等现象发生。

6.防止局部变量与公共变量同名。

7.仔细设计结构中元素的布局与排列顺序，使结构容易理解、节省占用空间，并减少引起误用现象。

8.结构的设计要尽量考虑向前兼容和以后的版本升级，并为某些未来可能的应用保留余地（如预留一些空间等）。

9.留心具体语言及编译器处理不同数据类型的原则及有关细节。

10.严禁使用未经初始化的变量。声明变量的同时对变量进行初始化。

1. 编程时，要注意数据类型的强制转换。

* 程序效率：

1.编程时要经常注意代码的效率。

2.在保证软件系统的正确性、稳定性、可读性及可测性的前提下，提高代码效率。

3.不能一味地追求代码效率，而对软件的正确性、稳定性、可读性及可测性造成影响。

4.编程时，要随时留心代码效率；优化代码时，要考虑周全。

5.要仔细地构造或直接用汇编编写调用频繁或性能要求极高的函数。

6.通过对系统数据结构划分与组织的改进，以及对程序算法的优化来提高空间效率。

7.在多重循环中，应将最忙的循环放在最内层。

8.尽量减少循环嵌套层次。

9.避免循环体内含判断语句，应将循环语句置于判断语句的代码块之中。

10.尽量用乘法或其它方法代替除法，特别是浮点运算中的除法。

* 代码测试：

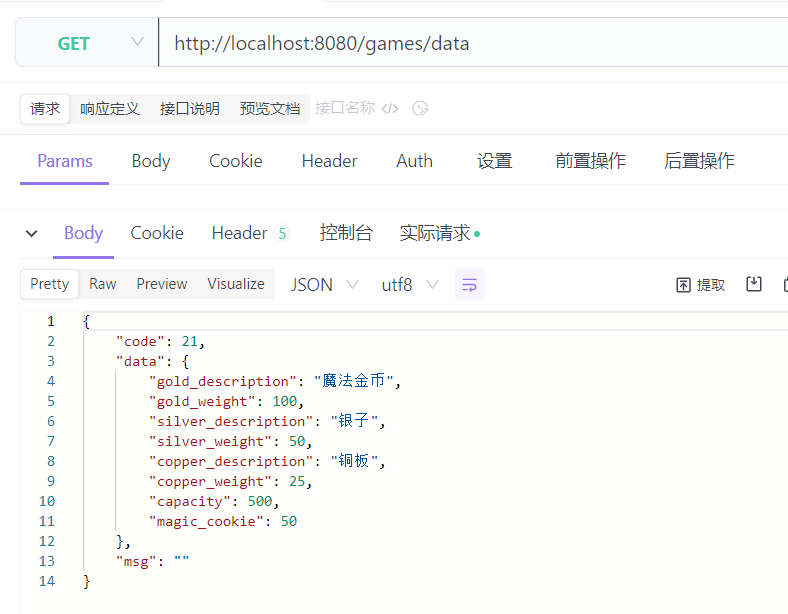
1. 单元测试要求至少达到语句覆盖。
2. 单元测试开始要跟踪每一条语句，并观察数据流及变量的变化。
3. 清理、整理或优化后的代码要经过审查及测试。
4. 采用黑盒测试，白盒测试

## 测试

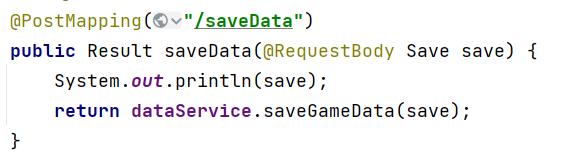
对Controller类中的接口使用Apifox进行测试

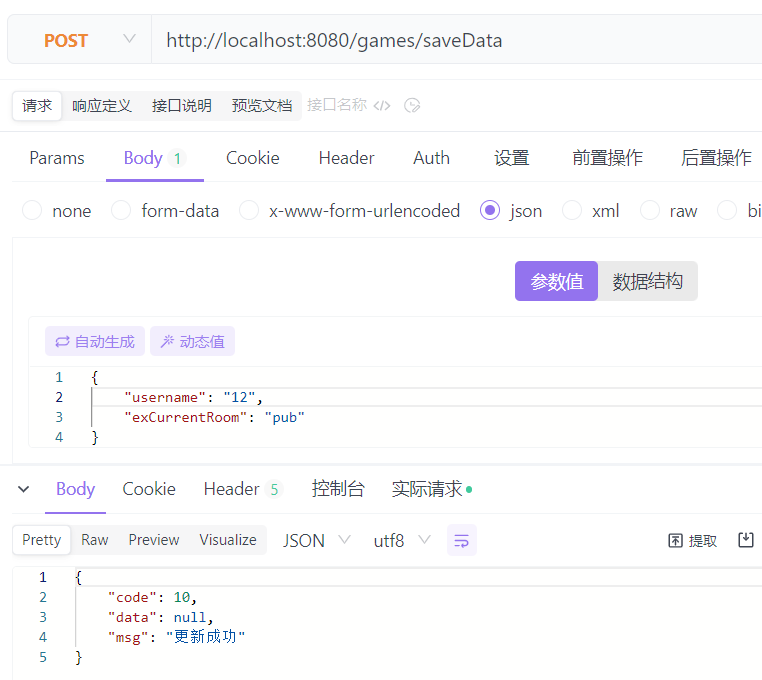
### /games/data



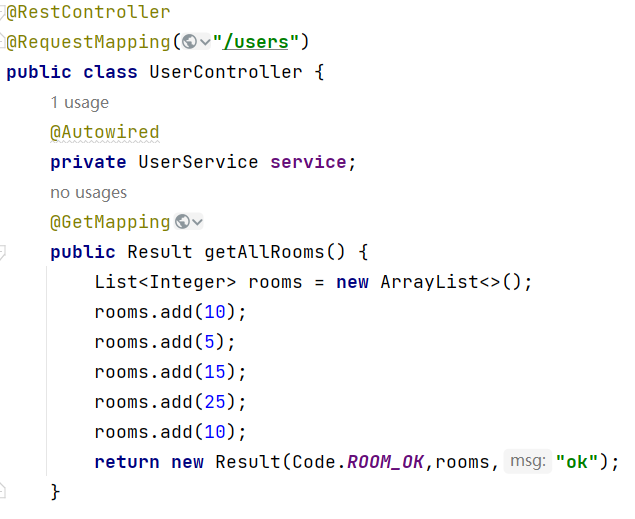
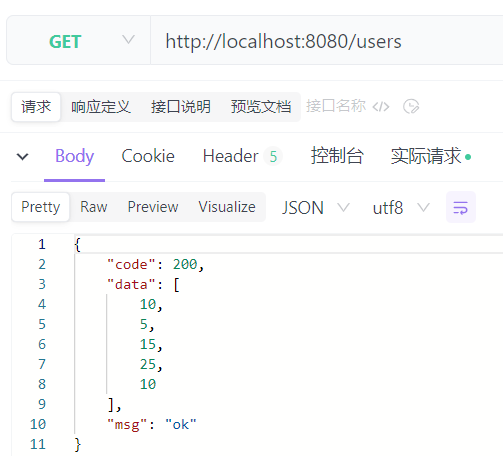


### /games/saveData

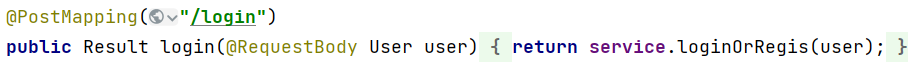




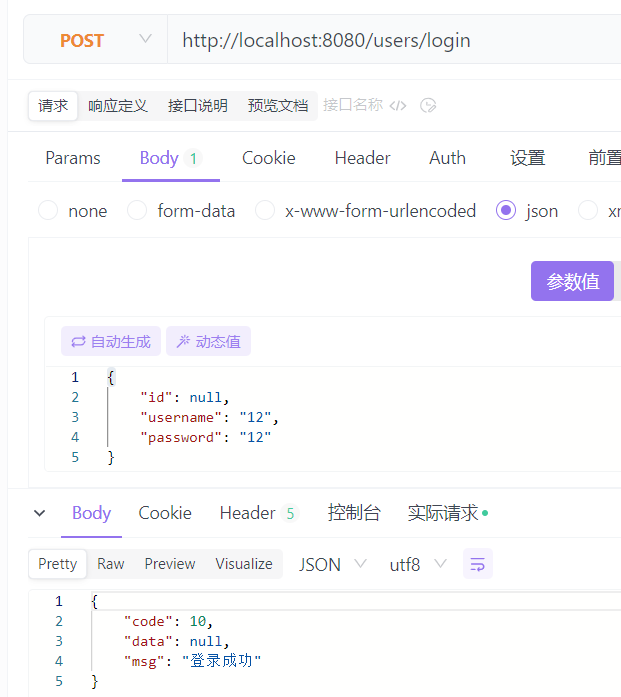
### /user



### /users/login



如果登录已注册的账号：



如果登录未注册的账号：



## Action

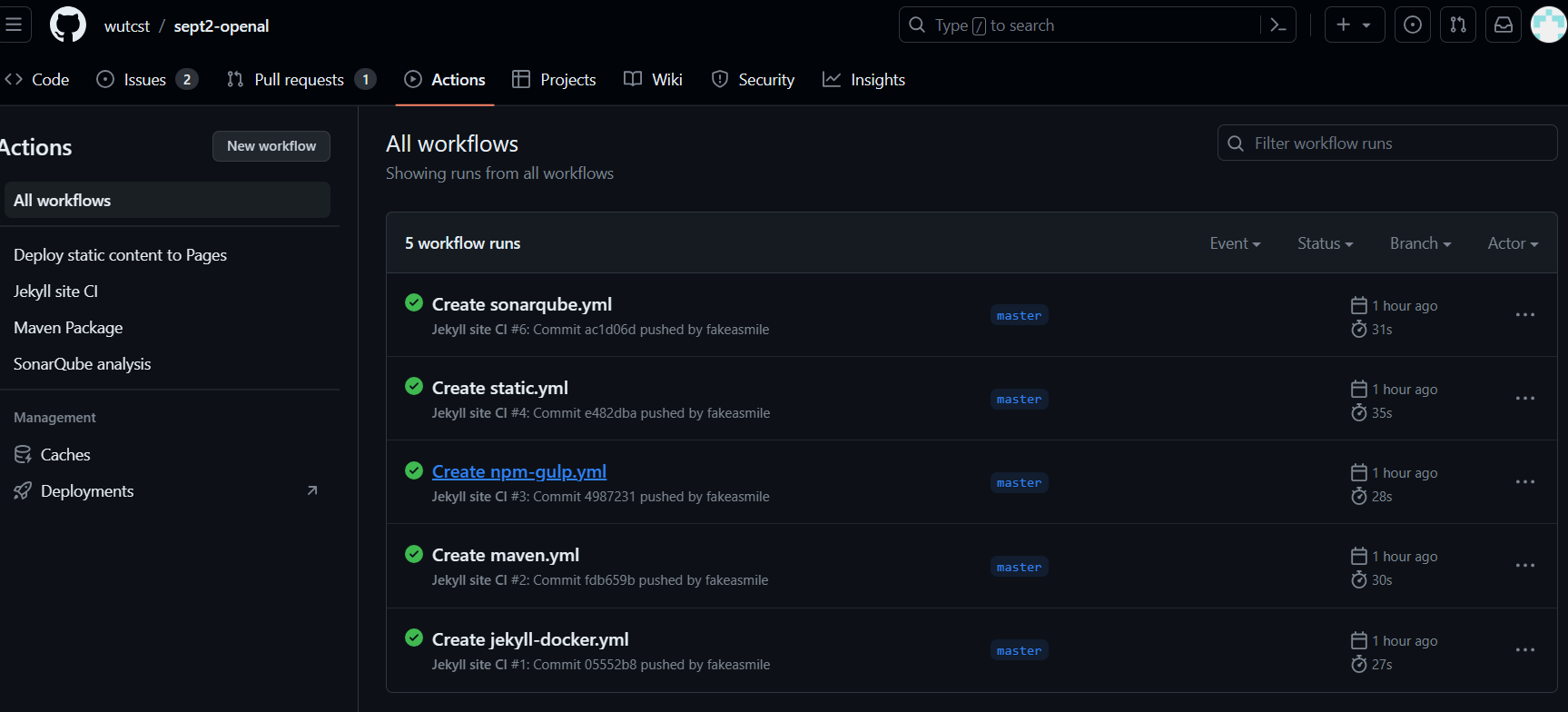
为了保证代码质量和可维护性，我们制定了以下开发规范：

在代码提交通过github里的actions工具，在.github/workflows目录下，通过不同的yml类文件，来对不同的代码进行代码风格检查。

- 代码注释清晰明了，变量和函数命名规范。

- 代码风格统一，遵循团队约定。

- 代码提交前进行自测和代码审查。



测试方面，我们采用了以下方法：

- 单元测试：对每个模块进行单元测试，保证功能正确性。

- 集成测试：对整个游戏进行测试，保证各个模块之间的协作和兼容性。

- 用户体验测试：邀请用户进行测试和反馈，改进游戏体验和功能。

测试：

|  |  |  |
| --- | --- | --- |
| 测试命令 | 期望输出 | 实际输出 |
| go east | You are in a lecture theater. | You are in a lecture theater. |
| go south | You are in the random room | You are in the random room.  Exits: north  You are teleported to in the library  You are in the library.  Exits: east south north |
| go west | There is no door! | There is no door! |
| go north | You are in the canteen | You are in the canteen |
| east | I don't understand... | I don't understand... |
| west | I don't understand... | I don't understand... |
| south | I don't understand... | I don't understand... |
| north | I don't understand... | I don't understand... |
| 空 | I don't understand... | I don't understand... |
| go | Go where? | Go where? |
| look | The things of this room have :  cookie 0.5kg  chopsticks 0.2kg  bowl 0.8kg | The things of this room have :  cookie 0.5kg  chopsticks 0.2kg  bowl 0.8kg |
| take chopsticks | You take chopsticks | You take chopsticks |
| items | The things of this room have :  cookie 0.5kg  bowl 0.8kg  The things of this player have :  chopsticks 0.2kg | The things of this room have :  cookie 0.5kg  bowl 0.8kg  The things of this player have :  chopsticks 0.2kg |
| back | You are teleported to in the library  You are in the library. | You are teleported to in the library  You are in the library. |
| drop chopsticks | You drop chopsticks | You drop chopsticks |
| drop | Drop what? | Drop what? |

# 集成与版本发布

在控制台程序整体开发完成之后，发布控制台版本。



在网页游戏程序开发完之后，发布网页游戏版本。



集成控制台和网页游戏两种玩法之后，发布最终版本。



# 小组成员个人任务

|  |  |
| --- | --- |
| 成员 | 任务 |
| 李冬俊 | 1. back命令。当用户点击back命令时，将玩家带回上一个房间。连续点击可以逐层返回直到返回起点。 2. look命令。当玩家进入一个房间之后，通过点击look命令可以查看当前房间中的物件数量 3. transmit命令。当玩家点击transmit命令之后，将提供游戏中的所有房间供玩家选择。玩家点击哪个房间将被床送到这个房间中。 4. take命令。玩家可以拾取房间中的物品，但是不能超过玩家背包上限。 5. drop命令。玩家可以丢弃背包中的物品到当前所在的房间，但是丢弃的数量不能多余玩家携带的数量。 6. items命令。玩家可以点击items命令获取当前房间的物品信息和玩家携带的物品信息。 7. magic cookie。随机在两个房间中生成魔法饼干，当玩家找到并点击魔法饼干之后，将增加玩家的背包容量。 8. 应该具备保存游戏功能。保存当前玩家的所在房间。 9. 具备登录和注册功能。 10. 具备退出登录，保存游戏(玩家名和当前所在房间)功能 11. 文档编写 |
| 任文斌 | 1. look命令。每个房间可以存放任意数量的物件，每个物件有一个描述和一个重量值，玩家进入一个房间之后，可以通过“look”命令查看当前房间的信息以及房间内所有物品信息。 2. 高级的back命令。当用户输入"back"命令之后，可以把玩家待会上一个房间，重复使用可以逐层回退几个房间，直到把玩家带回到游戏的起点。 3. 在游戏中增加具有传输功能的房间，每当玩家进入这个房间之后就会随机传送到另外一个房间。 4. 玩家可以携带任意数量的物件，但是随身携带物件的总重量不能超过某个上限。 5. take命令。玩家可以拾取房间内的指定物品，当拾取的物件超过了玩家可以携带的重量上线，则不能继续拾取，给出提示。 6. 文档编写 |
| 范春宇 | 1. drop命令。玩家可以丢弃身上携带的物品，同时丢弃的物件将会放到当前房间中。（即房间中的物件数量将会增多） 2. items命令。打印当前房间内的所有物品信息，包括玩家携带的物件信息。 3. eat cookie命令。在房间中增加magic cookie魔法饼干 4. 控制台玩法应该具备和前端交互的功能。 5. 程序的输出应该能提交到前端页面展示。 6. 应该将原始的在控制台界面输入输出的方式修改为在前端页面上进行输入输出，以期提供友好的用户交互界面。 7. PPT制作 |
| 陈奥 | 1. 设计游戏地图，包括道路，房间，墙 2. 设计游戏技能图标 3. 设计游戏精灵（包括人物，房间中的物件，魔法饼干，房间等），并使用Pixi.js游戏引擎。其中房间中的物件有：金币，银币，铜币。 4. 设计人物的属性，包括携带的金币数。银币数，铜币数，携带的重量，背包的容量。 5. 用户可以通过鼠标点击移动玩家，也可以通过键盘移动玩家。 6. 游戏地图，人物，物件的设计使用工具TexturePackerGUI设计。 7. PPT制作 |