

武汉理工大学

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 2001
姓 名	郝一鸣 钱科均 雷霆杰 张家齐
指导教师	唐祖锴

2023 年 6 月 25 日

目录

1	课程任务	1
2	开发版本计划与任务分派	1
3	程序设计与开发.....	3
4	开发分支模型与代码合并	4
5	开发规范与测试.....	5
6	集成与版本发布.....	6
7	小组成员个人任务.....	7
8	项目展示	12
9	任务总结	14
10	参考文献.....	14

1 课程任务

任务内容

- 创建软件开发小组
- 针对样例代码工程进行小组讨论，确定功能扩充需求点
- 基于 Github 中的 issue 管理功能明确工作任务并为组员分配工作任务
- 基于小组商定的分支模型进行软件功能开发，并按开发流程进行代码测试、提交、归并和同步
- 代码提交到远程仓库后，应进行自动化代码格式规范检查和测试以确保功能符合需求设计
- 完成前述各项任务后，可尝试进行代码自动化打包，自动生成可供执行的 jar 文件

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

2 开发版本计划与任务分派

版本计划

1.0 扩充 zuul 的基本功能，包括 look, go, back, drop, take, give

2.0 更改代码结构，使用策略模式转化为对象之间的交互

3.0 可视化，增加游戏界面

开发计划

1. 设计阶段

分析业务领域的需求，识别系统中的参与者和用例，画出用例图和类图

2. 详细设计阶段

使用 IntelliJ IDEA 进行框架的搭建及具体代码的开发

3. 代码测试及打包阶段

使用 GitHub 的 action 进行自动化代码格式规范检查和用例测试，并尝试进行代码自动化打包，自动生成可供执行的 jar 包

任务分派

张家齐：基础地图和类的设计，游戏地图分布优化，背景图片的优化房间物品等基础属性的设计，人物图片优化，规范化代码注释，报告的编写

雷霆杰：player 类的设计、基础逻辑的实现，包括 look, go, back, drop, take, give 等命令的实现，规范化代码注释；类图的绘制；PPT 绘制，报告的编写

郝一鸣：GUI 层界面的设计，包括界面布局以及事件监听响应；代码冗余检查更新，程序优化，分支的创建，任务的分派，规范化代码注释，PPT 汇报讲解，报告的编写

钱科均：Commands 类的改写与实现，识别 GUI 层的命令并解析，传至 player 层；利用 actions 代码检查与用例测试，结合 maven 打包 jar 包，发布 release，规范化代码注释，报告编写

Filters ▾

🔍 is:issue is:open

☐ ☒ 4 Open ✓ 0 Closed

☐ ☒ GUI界面实现

#4 opened 5 hours ago by Heart04

☐ ☒ Commands类的改写与实现

#3 opened 5 hours ago by Heart04

☐ ☒ Player类以及方法的实现

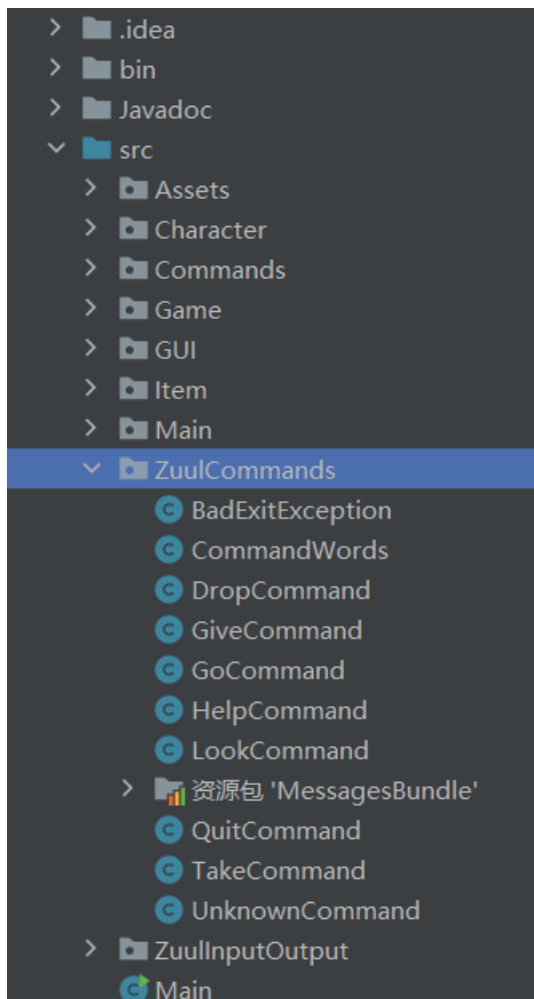
#2 opened 5 hours ago by Heart04

☐ ☒ 基础类的设计

#1 opened 5 hours ago by Heart04

3 程序设计与开发

- 1、基础类的设计：item 类以及 Game 中 room 类
- 2、Player 类的设计：基于 item 以及 room 类的设计，吸收实现第一次实践内容中 Game 类中方法，实现 go、back、take、give、tp 等功能
- 3、Commands 类的设计：包括 Commands 包以及 ZuulCommands 包，沿袭了第一次实践通过 commands 的实现方法，在收到 GUI 层面生成的 Command 消息后，在本层进行 command 分析，并向下通过 Player 对象调用相应方法来实现对应功能
- 4、GUI 层的设计：界面的布置，actionlistener 编写，调用 CommandGUI 中的方法将点击事件转换为自定义的 command 命令，并向下传输到 command 层对应的 execute 方法。



4 开发分支模型与代码合并

分支的创建，各自成员每人一条分支，各自在分支上进行开发，完成后进行合并

Active branches	
Heart04	Updated 5 days ago by Heart04
ltj	Updated 5 days ago by ltj1042410441
zjq	Updated last week by ArcherDongqi
QianKejun	Updated last week by kaiyu1234

因为之前没有了解，所以这次分支的划分不太合理，正确的应该是按功能模块划分，在后续的学习中，我们会吸取教训，并加以改正

Active branches	
class-GUI	Updated 5 days ago by Heart04
class-Play	Updated 5 days ago by Itj1042410441
class-Item	Updated last week by ArcherDongqi
class-Command	Updated last week by kaiyu1234

5 开发规范与测试

注释规范在 java 代码中的方法添加基于 javadoc 规范的注释，保证了程序注释的规范性与严谨性

自动化代码格式规范检查和测试以确保功能符合需求设计；

- 可以利用 github 平台的 actions 功能在代码提交时自动触发代码格式检查，对于不符合规范的代码系统将给出提交失败提示；
- 可以利用 github 平台的 actions 功能在代码提交时自动触发测试用例检查，对于不能通过测试检查的代码系统将给出提交失败提示

GitHub Actions 是一种持续集成和持续交付 (CI/CD) 平台，可用于自动执行生成、测试和部署管道。您可以创建工作流程来构建和测试存储库的每个拉取请求，或将合并的拉取请求部署到生产环境。

```

8
9     name: Java CI with Maven
10
11     on:
12       push:
13         branches: [ "master" ]
14       pull_request:
15         branches: [ "master" ]
16       merge_group:
17         branches-ignore: [checks_requested]
18       create:
19         branches: [ "master" ]
20       delete:
21         branches: [ "master" ]
22     jobs:
23       build:
24         # 配置权限
25         permissions: write-all
26         runs-on: ubuntu-latest
27         steps:

```

All workflows

Showing runs from all workflows

41 workflow runs		Event ▾	Status ▾	Branch ▾	Actor ▾
<div>✔</div> <div>Update REPORT.md</div> <div>Java CI with Maven #41: Commit 74b6740 pushed by kaiyu1234</div>	master		<div>📅 13 hours ago</div> <div>🕒 26s</div>	...	
<div>✔</div> <div>Update REPORT.md</div> <div>Java CI with Maven #40: Commit 1086f13 pushed by kaiyu1234</div>	master		<div>📅 14 hours ago</div> <div>🕒 31s</div>	...	
<div>✔</div> <div>Add files via upload</div> <div>Java CI with Maven #39: Commit c2752d3 pushed by kaiyu1234</div>	master		<div>📅 14 hours ago</div> <div>🕒 25s</div>	...	
<div>✔</div> <div>Update BackCommand.java</div> <div>Java CI with Maven #38: Commit 6a54720 pushed by kaiyu1234</div>	master		<div>📅 5 days ago</div> <div>🕒 28s</div>	...	
<div>✔</div> <div>Update Parser.java</div> <div>Java CI with Maven #37: Commit 0a5b1f3 pushed by kaiyu1234</div>	master		<div>📅 5 days ago</div> <div>🕒 24s</div>	...	
<div>✔</div> <div>Update GamePanel.java</div> <div>Java CI with Maven #36: Commit 1088292 pushed by ArcherDongqi</div>	master		<div>📅 5 days ago</div> <div>🕒 27s</div>	...	
<div>✔</div> <div>Update GamePanel.java</div> <div>Java CI with Maven #35: Commit 5606aea pushed by ArcherDongqi</div>	master		<div>📅 5 days ago</div> <div>🕒 23s</div>	...	

6 集成与版本发布

文件配置


```

# Maven 打包
- name: Build with Maven
  run: mvn -B package --file pom.xml
# 将打包好的 Jar 包 放到 staging 文件夹
- run: mkdir staging && cp target/*.jar staging
# 设置 jobs Maven pom 版本环境变量
- name: Set Release version env variable
  run: |
    echo "RELEASE_VERSION=$(mvn help:evaluate -Dexpression=project.version -q -DforceStdout)" >> $GITHUB_ENV
- name: "Build & test"
  run: |
    echo "done!"
# 上传文件并发布 Release
- uses: "marvinpinto/action-automatic-releases@latest"
  with:
    repo_token: "${{ secrets.GITHUB_TOKEN }}"
    automatic_release_tag: "${{ env.RELEASE_VERSION }}"
    prerelease: false
    title: "Release ${{ env.RELEASE_VERSION }}"
    files: |
      staging/*.jar




```

Release 1.0-SNAPSHOT Latest

Commits

- [74b6740](#) : Update REPORT.md (kaiyu1234)

▼ Assets 3

 zuul-1.0-SNAPSHOT.jar	1.44 KB	13 hours ago
 Source code (zip)		13 hours ago
 Source code (tar.gz)		13 hours ago



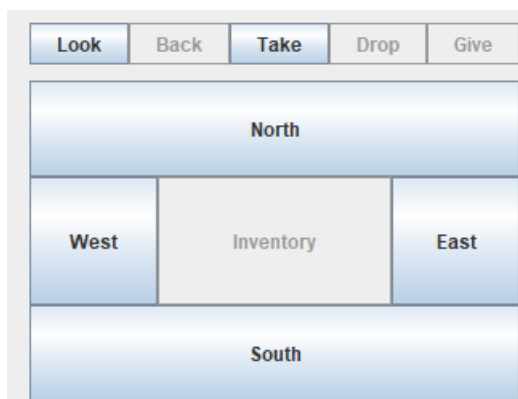
7 小组成员个人任务

①郝一鸣：

- 1、**图片加载方面**：通过 BufferedImage、ImageIO、Graphic2D 等方法首先将图片加载入内存中，利用这些方法可以很方便的布置游戏窗口内的内容
- 2、**窗口设计**：利用 java 中的 Container 容器，整体容器采取 X 横向布置，以此来划分开游戏界面与控制面板界面
- 3、**游戏窗口中**，使用固定窗口值大小以及对应的坐标比例来规划角色、item、方向等图片，利用 graphic.drawImage 方法将图片类加载。控制窗口中，通过 BoxLayout 调整为 Y 纵向方向布置，在纵向的基础之上通过空的刚性区域来划分不同的控制面板

集合，集合中再用 X 横向布置 look、back、take、give、drop 或是 page.start、page.end, center 的方形布局方式。

- 4、**控制面板方面的其他设计**：由于我们引入了 item 以及 drop 的内容，并且实现了背包重量存在上限相关内容，于是我们新增加了布置于中心的 Inventory 按钮，用于打印背包内所有物品以及总重量；我们实现了地图上存在其他角色以及 give 功能，我们拓展了 command 方法中的解析命令，选取背包中物品给予给定的角色，两次的选择我们都基于 java 滚动条面板来实现；通过 button 的 setenable 功能验证当前房间的 direction 那些存在或是 back 记录是否被设置为空，在 GUI 层面将这些复杂的验证情况 ban 掉，避免设计过多的 try catch 的验证以及太多 exception 报错内容的编写



②钱科均：

***Acommand**: 抽象命令父类，定义了基本的属性以及获取命令单词的方法，后续命令实体均继承此类

***Parser 类**: Parser 类负责解析游戏中的用户命令。它对输入进行标记化，检查命令是否有效，并创建相应命令类的实例。如果命令无法识别，则返回 UnknownCommand 类的实例。该类还具有一个用于初始化必要变量的构造函数，这次试验添加了三个单词，主要是为 give 功能服务，因为要判断给的物品是什么以及给的人是谁

***合法命令判断**：对获取的命令进行合法性判断，看是否是属于已经定义的命令，若不是则予以提示

***Zuulcommand**: 充当一个命令接口的作用，继承自 Acommand 父类，作用是将 GUI 层面的命令转化为命令对象，并传至下层 player 执行

***控制错误推出异常：**若房间不存在或方向错误，则进行错误提醒

***GUI 层的命令响应传递：**当 GUI 界面点击某个命令的按钮之后，可以通过 parser 类中的 getcommandGUI()方法识别对应的命令，具体为，通过使用 Class 类中的静态 forName()方法获得与字符串命令相应的 Class 并利用 newInstance() 方法创建对象，并将该对象赋值给父类 Acommand 的对象，通过向上转型，该命令对象调用 execute 方法，去调用 player 类的具体实现方法，这样就实现了将 GUI 层的命令传递给 player 的底层逻辑实现

```
try
{
    ACommand cmd = (ACommand) Class.forName(cmdString).newInstance()
    cmd.addWords(word1, word2, word3);
    /*使用Class类中的forName ()方法获得与字符串相应的Class对象，
    动态加载和创建Class 对象，根据用户输入的字符串来创建对象
```

***actions：**利用 github 平台的 actions 功能在代码提交时自动触发测试用例检查，对于不能通过测试检查的代码系统将给出提交失败提示

All workflows

Showing runs from all workflows

Q Filter workflow runs

41 workflow runs			Event ▾	Status ▾	Branch ▾	Actor ▾
✓	<div>Update REPORT.md</div> <div>Java CI with Maven #41: Commit 74b6740 pushed by kaiyu1234</div>	master	13 hours ago	26s
✓	<div>Update REPORT.md</div> <div>Java CI with Maven #40: Commit 1086f13 pushed by kaiyu1234</div>	master	14 hours ago	31s
✓	<div>Add files via upload</div> <div>Java CI with Maven #39: Commit c2752d3 pushed by kaiyu1234</div>	master	14 hours ago	25s
✓	<div>Update BackCommand.java</div> <div>Java CI with Maven #38: Commit 6a54720 pushed by kaiyu1234</div>	master	5 days ago	28s
✓	<div>Update Parser.java</div> <div>Java CI with Maven #37: Commit 0a5b1f3 pushed by kaiyu1234</div>	master	5 days ago	24s
✓	<div>Update GamePanel.java</div> <div>Java CI with Maven #36: Commit 1088292 pushed by ArcherDongqi</div>	master	5 days ago	27s
✓	<div>Update GamePanel.java</div> <div>Java CI with Maven #35: Commit 5606aea pushed by ArcherDongqi</div>	master	5 days ago	23s

***打包发布：**

利用 actions 功能和 maven 编译脚本，在代码通过规范性检查和测试用例后，进行自动化打包，生成可供直接执行的 jar 文件

③雷霆杰：Play 类的实现思路

首先我的核心是实现 `take,go,drop,give,look,back` 等底层的动作命令,先定义一个接口,然后去实现这个接口,要理清每个命令的逻辑然后再去思考如何设计。比如 `take` 命令,拿起物品时需要判断当前房间是否有这个物品,以及是否拿了之后超重,若都满足,则可以实现拿起这个命令,否则输出相应的语句表示给予失败;

back 命令, 回退到上一个房间东边西, 西变东, 南变北, 北变南;

go 命令, 首先判断当前房间是否是 `TPRoom`,若是则随机生成一个 `TP` 房间让人物进入一个普通房间后再进一个房间, 若不是则输出相应的语句表示给予失败;

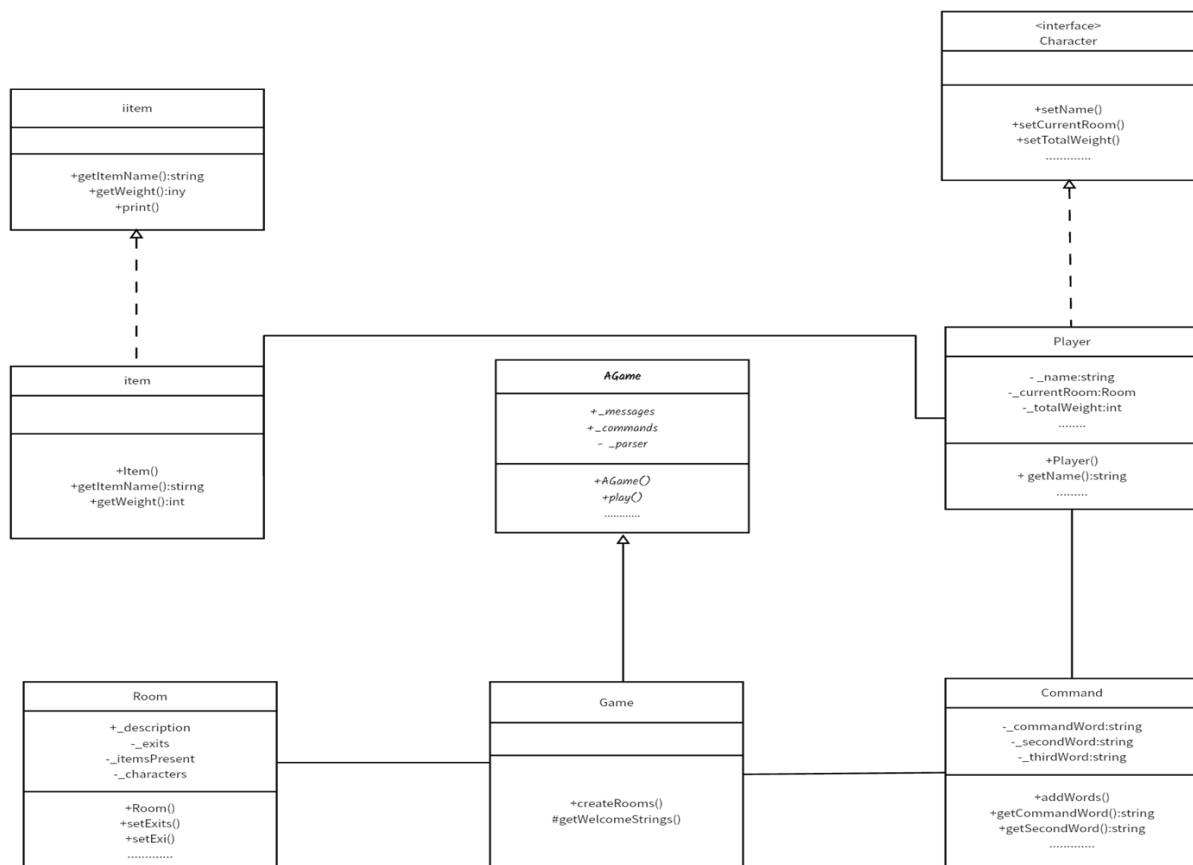
take 命令: 在游戏中增加具有拾取物品的功能, 用 `hashmap` 存放所拾取的物品, 玩家进入房间后可先查看当前房间具有的物品, 执行拾取某物品, 首先要判断该房间内是否存在该物品

然后为该物品初始化一个对象, 判断该物品是否超重, 若不超重, 则调用 `remove` 方法移除该物品, 并加入到 `inventory` 的 `hashmap` 中去

drop 命令, 丢弃之前判断人物仓库是否有这个物品, 以及当前房间丢弃后会不会超重;

give 命令, 先判断当前房间是否有人物, 且是否有要给予的物品, 以及给予后人物是否超重, 若都满足, 则可以实现给予这个动作, 否则输出相应的语句表示给予失败。

类图:



④张家齐：

游戏类（Game 类）：

- 创建一个名为 Game 的类，继承自 AGame 类。
- 在`Game`类中定义私有变量，如房间列表（List<Room>）和玩家对象（Player）。
- 实现构造函数，调用父类构造函数并初始化房间列表和玩家对象

房间类（Room 类）：

- 创建一个名为 Room 的类。
- 在 Room 类中定义私有变量，如房间描述（String）、出口（Map<String, Room>）、物品表（HashMap<String, Item>）和角色表（HashMap<String, Player>）。
- 实现构造函数，用于初始化房间对象和相关变量。
- 实现 setExits 方法，根据输入参数设置房间的出口。
- 实现 setExit 方法，用于设置单个出口。
- 实现 getExit 方法，根据输入方向获取对应的出口。
- 实现 getDescription 方法，返回房间的描述信息。
- 实现 addItem 方法，添加物品到房间的物品表中。

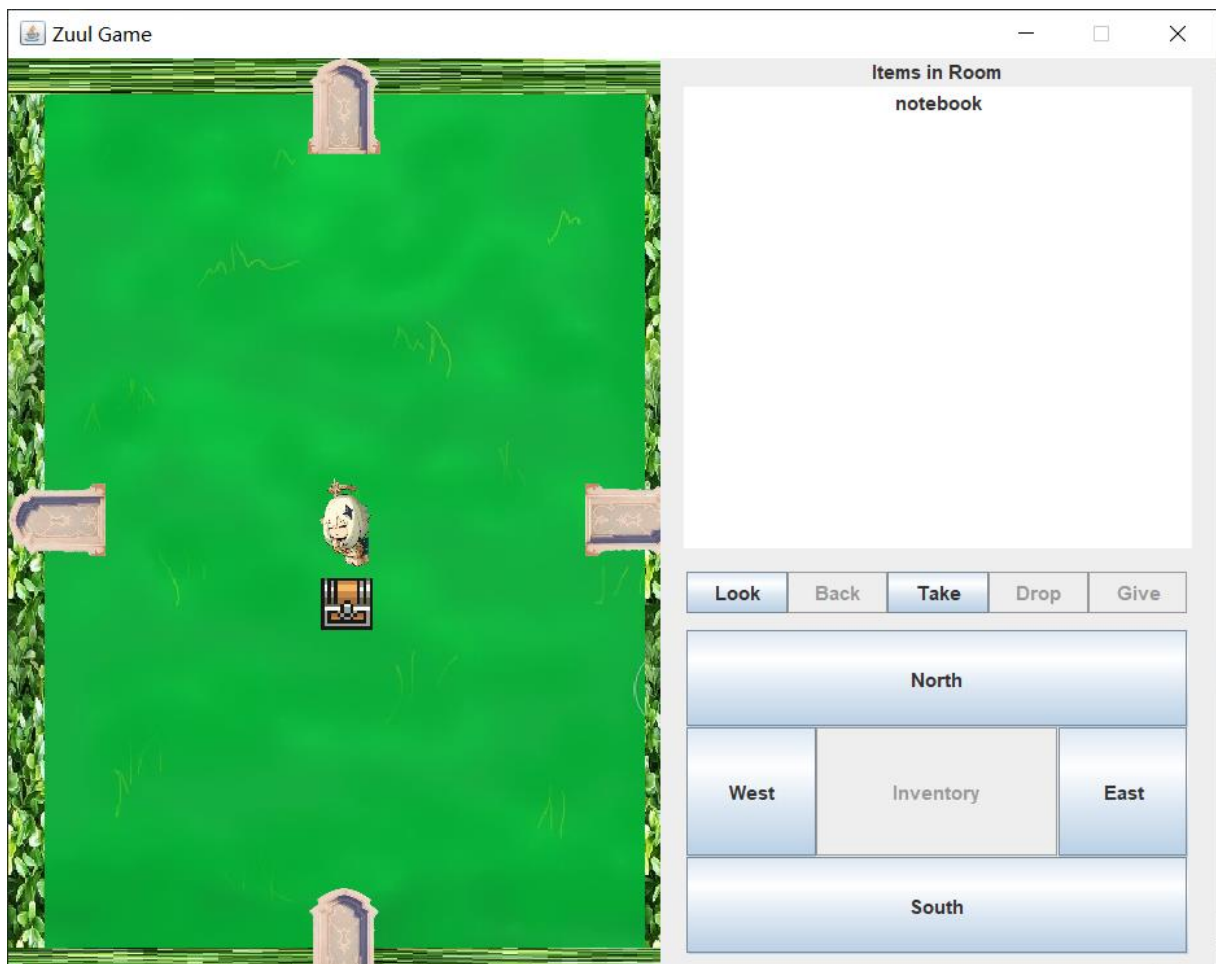
物品接口 (IItem 接口):

- 创建一个名为 IItem 的 Java 接口，定义获取物品名称和重量的方法。

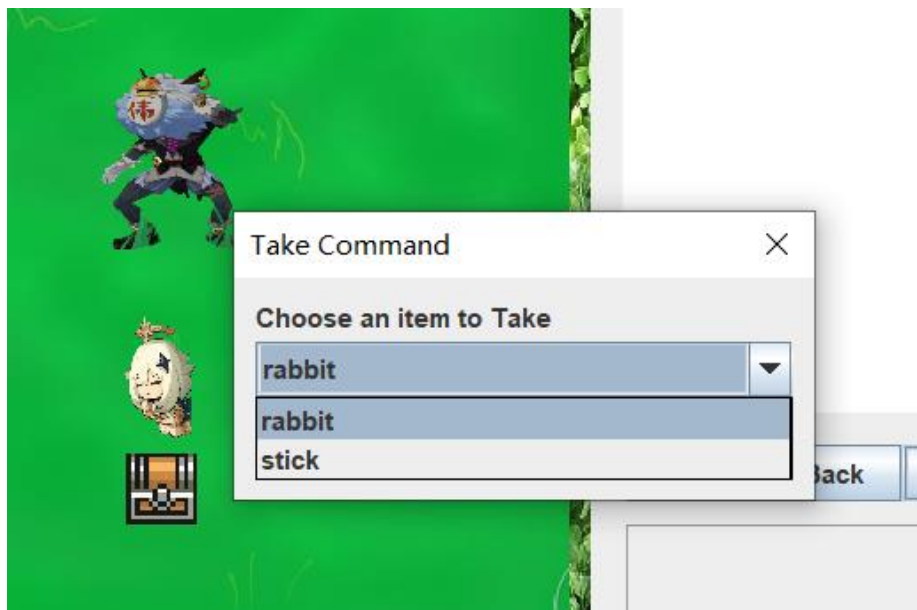
物品类 (Item 类):

- 创建一个名为 Item 的类，实现 IItem 接口。
- 在 `Item` 类中定义私有变量，如物品名称 (String) 和物品重量 (int)。
- 实现构造函数，用于初始化物品对象和相关变量。
- 实现 getItemName 方法，返回物品的名称。
- 实现 getWeight 方法，返回物品的重量。

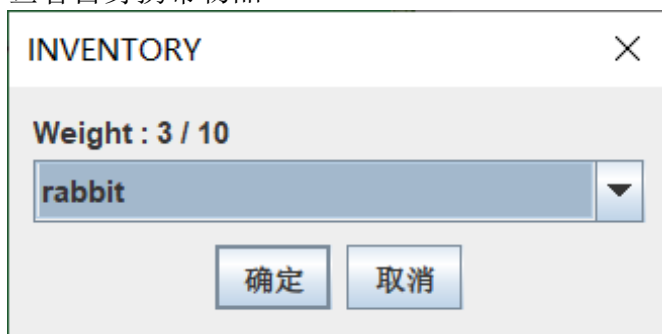
8 项目展示



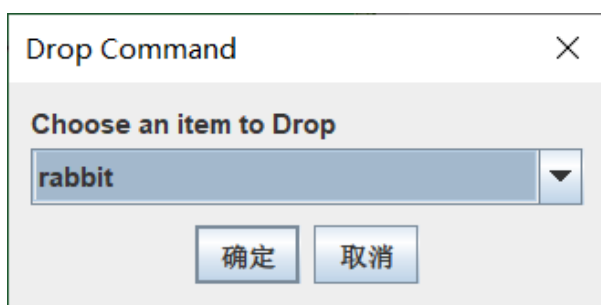
Take 指令



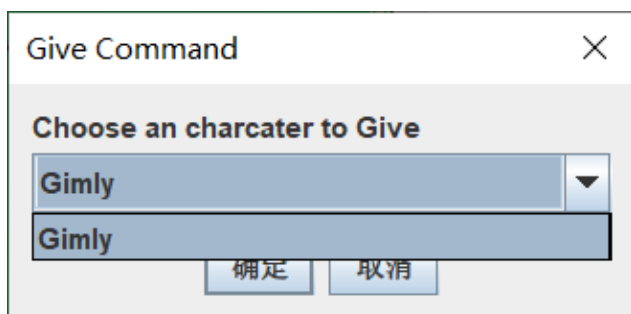
查看自身携带物品



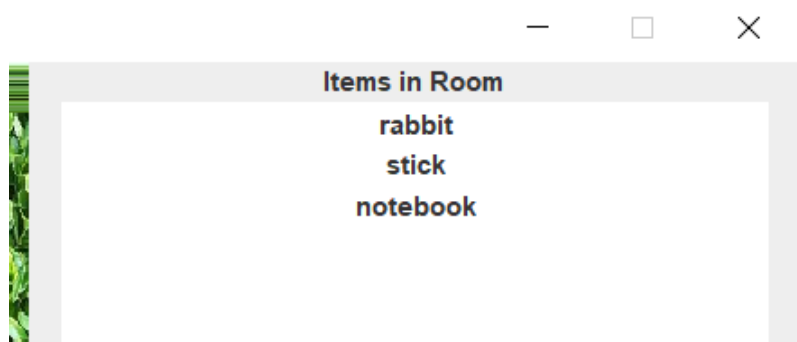
Drop



Give



查看当前房间所存在的物品



随机房间

```
You are in TRoom and you will tp to a random room
Exits: east south north west
Items:
Characters:
You are in in the computing admin office
Exits: west
```

9 任务总结

本项目的开发中，对于各个流程的运作和意义也有了更加深入的了解，本次项目在锻炼代码编写能力的基础上，以工程化的手段培养了我们的作为编程者在实际代码代码编写时应具备的能力。相信之前很多人和我会有一样的误区，在软件开发过程只有代码的编写最为重要，但事实并非如此，在软件开发过程中，需求分析、设计、开发、测试、维护等步骤都必不可少，甚至比重优于代码的编写，在系统开发过程中设计作为整个软件的骨骼，是软件开发的指南，环环相扣。

本次实践中存在一些不足之处，对 GitHub 的一些功能并不是特别熟练，不过通过实践，我们也明白了协作开发的重要性，在此后的实践中，会加强对 GitHub 功能的学习和应用

10 参考文献

[1]命名规范: <https://blog.csdn.net/fightfa/article/details/77899695>

[2]分支管理规范: <https://cloud.tencent.com/developer/article/1592957>

[3]提交规范: <https://zhuanlan.zhihu.com/p/90281637>

《软件工程实践（二）》成绩评定表

姓 名	郝一鸣, 钱科均, 雷霆杰, 张家齐	学 号		
专业、班级	软件 2001			
成绩评定:				
评价内容		满分	实得分	
			得分	小计
实践任务 完成情况	软件项目设计、改进与扩充	20		
	个人软件过程与项目管理	15		
	代码版本管理	25		
	代码注释与编码规范	25		
	单元测试	15		
实践报告 总评情况	学习态度与考勤	10		
	报告格式的规范性	10		
	报告的逻辑结构与语言表达	15		
	实践内容的正确性与合理性	60		
	文献引用及标注	5		
总分		100		
最终评定成绩（以优、良、中、及格、不及格评定）				

指导教师签字:

年 月 日