|  |  |
| --- | --- |
| **学 号：** | 0122010870308 |

****

**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** | **曾俊博** |
| **小组成员** | **曾俊波 周盟 毕列祥 曾梓迪** |
| **指导教师** | **唐祖锴** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2023 | 年 | 6 | 月 | 18 | 日 |

# 1 课程任务

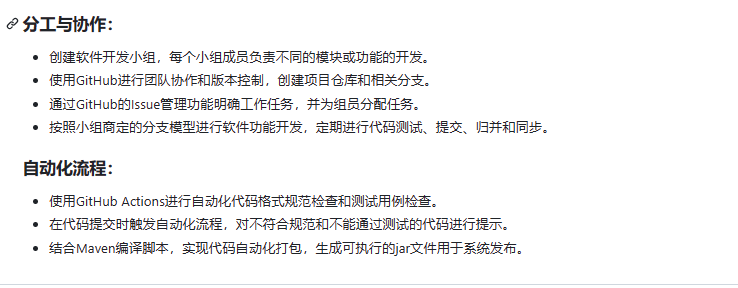






# 2 程序设计与开发





# 3 开发版本计划

版本计划：

Version 0.1：

基本界面

创建基本UI布局，并实现角色移动功能；

设计简单的游戏场景和游戏角色；

基本框架

搭建基本的服务器框架；

创建玩家账号系统和角色信息管理系统；

实现基本数据库功能；

Version 0.2：

地图和角色展示：

改进游戏场景和游戏角色，提高质量；

支持大规模游戏世界，实现多种地图效果；

添加角色定制功能；

游戏逻辑：

添加游戏中心逻辑处理；

实现简单的游戏规则系统；

添加角色进入游戏和出游戏等操作；

Version 0.3：

交互系统

添加交互选项，让玩家能够快速地探索游戏世界；

支持玩家交流和动态信息显示；

增加任务管理中心；

数据管理

实现人物和装备等数据管理；

增加在线交互功能；

建立基础架构，可支持大规模游戏操作；

Version 0.4：

安全和稳定性

增加异常捕获和报告机制；

强化数据安全和游戏安全；

实现玩家数据备份和恢复机制；

Version 1.0：

正式版本

优化游戏服务器性能；

实现多服务器分布式处理；

提供API接口，便于二次开发和扩展。

# 4 任务分派

版本号：v1.0.0

任务分派如下：

1. 架构设计文档编写

- 负责人：周盟

- 描述：根据系统需求和技术方案，编写系统的整体架构设计文档，包括前后端架构、数据交互方式、数据库设计等。

2. 前端界面编写 - 登录、注册、忘记密码部分

- 负责人：毕列祥

- 描述：实现游戏的前端界面，包括登录、注册、忘记密码等功能，并实现对表单字段的校验。

3. 前端界面编写 - 游戏界面部分

- 负责人 ：曾俊博

- 描述：实现游戏的前端界面，包括游戏场景展示、角色移动等功能。

4. 后端鉴权认证功能实现

- 负责人：曾梓迪

- 描述：编写后端的鉴权认证功能，包括登录、注册、忘记密码等接口的实现，确保用户的身份验证和安全性。

5. 后端核心游戏逻辑编写

- 负责人：周盟

- 描述：根据游戏设计要求，编写后端的核心游戏逻辑，包括游戏进程控制、事件处理等功能。

6. 数据库设计与前后端数据交互格式设计

- 负责人：周盟

- 描述：设计游戏所需的数据库结构，确定表的字段和关系，同时定义前后端数据交互的格式，确保数据的一致性和有效性。

7. 测试与问题记录

- 负责人：全体成员

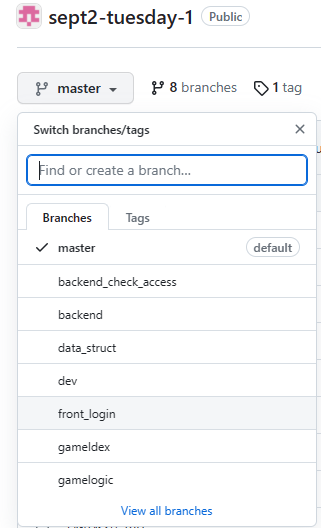
- 描述：进行测试并记录系统的功能、性能、安全等方面的问题，及时反馈并解决问题，确保系统的稳定性和质量。

任务分派遵循每个人领取相应的issue并在自己的分支上进行开发的方式，完成阶段性开发任务后提交相应的PR到dev分支，同时每次提交需要简要说明本次提交的内容。

请根据以上计划和任务分派，合理安排自己的时间和工作，确保按时完成任务并达到预期的质量要求。

# 5 开发分支模型与代码合并

模型名： master-dev 模型



**代码合并过程：**

1. 主要分支：

- master: 主分支，用于存放发布版本的稳定代码。

- dev: 开发分支，用于集成各个成员的开发成果，并进行整体功能测试。

2. 功能分支：

- backend\_check\_access: 后端登录注册功能开发分支，由周盟负责。

- game\_logic: 后端游戏逻辑开发分支，由曾梓迪负责。

- frontend\_login: 前端登录注册功能开发分支，由毕列祥负责。

- gameIndex: 前端游戏界面开发分支，由曾俊博负责。

3. 代码合并策略：

- 曾梓迪和毕列祥负责的功能分支（backend\_\_check\_access和frontend\_login）开发完成后，将代码提交到对应分支，并创建Pull Request（PR）到dev分支。经过代码审查和测试后，由团队成员合并到dev分支。

- 周盟负责的功能分支（game\_logic）开发完成后，同样提交代码到对应分支，并创建PR到dev分支。经过审查和测试后，合并到dev分支。

- 曾俊博负责的功能分支（gameIndex）开发完成后，提交代码到对应分支，并创建PR到dev分支。经过审查和测试后，合并到dev分支。

- dev分支经过一段时间的集成和测试后，当功能达到一定的阶段性成果，进行版本发布，将dev分支合并到master分支，生成新的发布版本Version 1.0。

# 6 开发规范与测试

**开发规范：**

**1. 代码规范：**

**- 遵循统一的命名规范，包括变量、函数、类等命名方式。**

**- 保持代码的可读性和可维护性，使用合适的缩进、注释和代码结构。**

**- 遵循团队约定的编码风格，如代码缩进、命名规则等。**

**2. 分支管理：**

**- 每个成员负责的任务在独立的分支上开发，确保代码的隔离性和独立性。**

**- 分支命名规范：按照任务名称或功能进行命名，避免冲突和混淆。**

**3. 提交规范：**

**- 每次提交需要简要说明本次提交的内容，以便于其他成员了解代码的改动和目的。**

**- 使用有意义的提交消息，清晰描述本次提交的目的和影响。**

**4. 代码审查：**

**- 团队成员之间进行代码审查，确保代码质量和一致性。**

**- 审查过程中提供有建设性的反馈和建议，帮助改进代码。**

**测试策略：**

**1. 单元测试：**

**- 编写针对各个模块的单元测试用例，验证各个函数和方法的功能和边界条件。**

**- 执行单元测试并确保测试覆盖率达到预期水平。**

**2. 集成测试：**

**- 在dev分支进行集成测试，验证各个模块之间的协作和交互是否正常。**

**- 模拟真实场景，测试各个功能模块的整体性能和稳定性。**

**3. 用户验收测试：**

**- 在开发阶段的某个时点，邀请用户参与测试，收集用户反馈和建议。**

**- 验证系统是否符合用户需求，修复和改进系统中发现的问题。**

**4. 性能测试：**

**- 针对关键功能和系统瓶颈点进行性能测试，评估系统的承载能力和响应时间。**

**- 优化性能瓶颈，提升系统的性能和用户体验。**

# 7 集成和版本发布

集成：

前后端分离：开发人员可以将前端代码和后端代码分别上传到github版本管理工具，在项目团队内部进行协作；

接口文档：开发人员需要编写接口文档，明确前后端数据交互的具体规范；

联调测试：前后端在自己本地进行单元测试，在完成后进行联调测试；

部署环境：部署至测试环境，完成全面测试后上线。

版本发布：

打标签：在代码上传至版本管理工具后，为每个重要版本打上tag。例如：v0.1, v0.2等等；

版本控制：使用版本控制工具（例如Jenkins）来进行版本管理,每个版本的代码需要在控制工具中单独设置一个编译任务；

发布环境：在测试环境测试完成后，可以将代码部署至生产环境；

发布通知：发布新版本之后，包括开发人员在内的所有使用人员需要得到相应的通知。

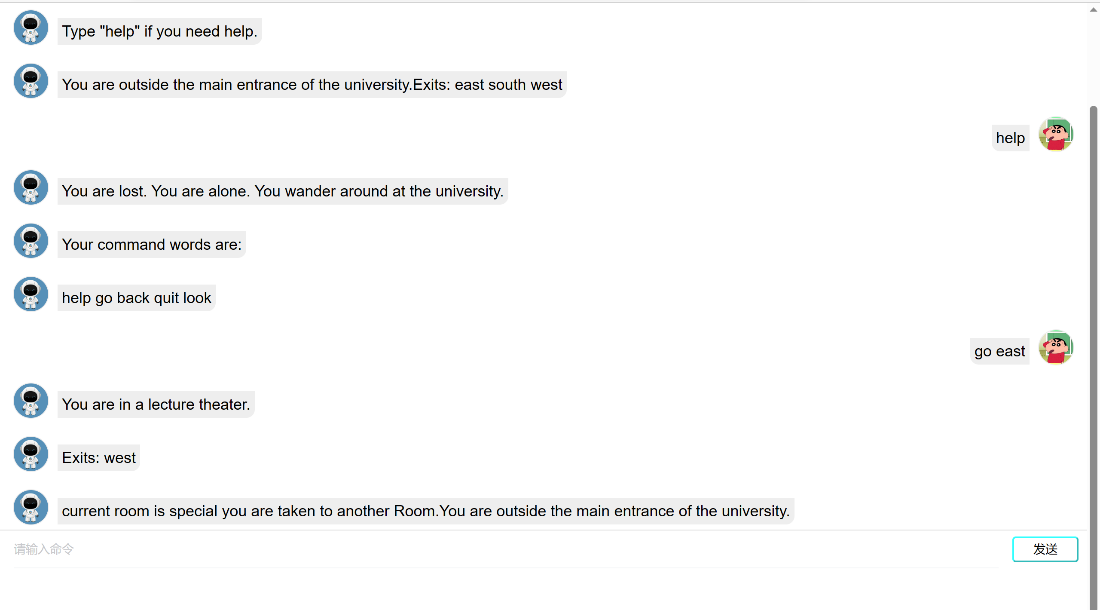
# 8 小组成员个人任务

**我的任务和工具：**

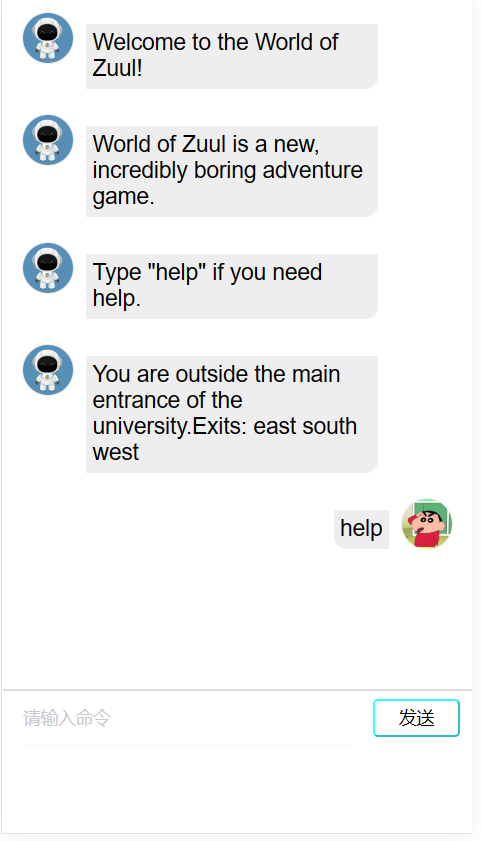
我在github上承接的issue 是编写文字游戏的主页面，对于这个任务，我选择的开发工具是 vue 3 和 vant 等 vue 的一些插件。项目的主要管理方法是在本地建立新的分支进行开发，并使用git管理，在本地测试通过之后上传到远程仓库中。等待另一位前端负责人也提交后，申请合并并处理关键冲突，再将合并的代码pull 到本地进行进一步完善。等待后端人员git之后，使用本地的服务器进行交互测试。

**功能设计方案：**

**前端页面**：采用聊天框的形式，使用vant插件将用户的命令和后端的返回结果显示在同一个页面中，在vue 中定义一个列表来存放当前页面的消息（包括用户输入和后端返回），使用v-for指令将其显示在页面中。以下为效果截图：



**响应式：**使用flex布局方式 实现响应式页面效果，结果如图：



**向后端发送请求：** 使用axios 向后端发送请求，并处理数据，代码和debug截图：





**开发中主要的问题：**

项目中遇到的主要的问题有两个。

一个是html中使用vue 的{{str}} 显示内容时，str中的\n不会生成回车的问题，在本项目中，我的解决方案是使用 split 将str 按照\n 分成多个字符串组成的数组，使用for循环将数组封装成多个 data 对象，每个data对象对应 msgList 中的一个元素，这样需要换行的消息就分为了多条消息，不会出现文字密集的情况。

另一个问题是前端初步实现之后，进行前后端交互测试时，**后端人员提议在接到消息时能不能像聊天软件一样自动地显示最新的消息**。一开始使用网络上的方法，利用nextTick 进行回调，在Dom元素更新完成后将整体scroll （下滑），但是一直没效果，后来采用了 directive 的方式自定义了一个指令 v-scroll （使用 scrollIntoView 方法实现），在每个消息后加上标记 v-scroll指令的高度为0的div，完成了该功能。

**《软件工程实践（一）》成绩评定表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓 名** | | **曾俊博** | **学 号** | **0122010870308** | |
| **专业、班级** | | **软件2002** | | | |
| **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：**

**2023年 6月 25日**