

****

代码设计文档

|  |  |
| --- | --- |
| **小组名称** | **金东翔小组** |
| **小组成员** | **石远翔，施东成，侯鑫** |
| **编写时间** | **2021年6月1日** |

目录

[第一部分 编写说明 2](#_Toc74144908)

[第二部分 代码思路 2](#_Toc74144909)

[一、 整体思路 2](#_Toc74144910)

[二、 功能实现 3](#_Toc74144911)

[第三部分 代码分析 4](#_Toc74144912)

[一、 编码规范 4](#_Toc74144913)

[二、 命名规范 4](#_Toc74144914)

[三、 模块分析 4](#_Toc74144915)

[四、 源文件分析 5](#_Toc74144916)

# 第一部分 编写说明

本代码设计文档主要面向编程人员，意在规范代码的编写同时展示本项目的代码设计思路，帮助编程人员明晰项目中各个代码模块的作用，便于开发人员对项目的管理，有利于软件的可控性以及后期的维护和更新。本文档将从代码编码、代码思路、实现方法等方面来构析项目的代码设计，为了便于读者阅读，文档会尽量省略细枝末节的说明，主要帮助读者把握整体思路，以及游戏中各个模块的功能实现。

# 第二部分 代码思路

### **整体思路**

1. 整体分为：控制类、展示类；
2. 控制类：实现开启游戏的各个线程（主程、音乐、聊天室等）、游戏对象的属性与加载、玩家操作的监听等；
3. 展示类：可视化游戏对象（玩家、怪物、NPC等）、游戏环境（墙格的布局）、聊天室界面、登录注册页面等；

### **功能实现**

1. 多任务
   1. 类/接口：Runnable
   2. 方式：开启多线程，利用实现Runnable接口创建线程，合理使用sleep()/wait()/interrupt()等方法规范线程；
2. 音乐播放：
   1. 类/接口：AudioPlayer、Applet、AudioStream
   2. 方式：先拿到wav文件的文件类实例，获得其URL，再通过Applet类的newAudioClip(url).play()方法播放；获取wav文件输入流，创建AudioStream 流，最后通过AudioPlayer.player.start()播放
3. 聊天室：
   1. 类/接口：Socket、ServerSocket
   2. 方式：通过Socket网络编程实现服务端和多个客户端，通过new Socket()连接服务器，服务器端通过ServerSocket创建服务器，并且通过不断监视连接来接受客户端请求信息以及发送信息给客户端
4. 存档管理：
   1. 类/接口：输入输出流
   2. 方式：客户端与服务器端通过socket建立连接后，通过socket.getInputStream()/socket.getOutputStream()获取输入输出流，并将其重定向到文件，以这种方式来实现存档在服务器端和客户端的传输
5. 操作监听：
   1. 类/接口：KeyListener
   2. 方式：通过给组件添加KeyListener键盘监听事件来实现对应的操作
6. 可视化对象：
   1. 类/接口：Graphics
   2. 方式：通过覆写paint()方法来将各种游戏对象画到相应的组件上，并且将画笔设置合适的属性达到不同的画图效果
7. 环境布局：布局类
   1. 类/接口: LayoutManager
   2. 方式：通过不同的布局管理器达到不同的布局效果，如GridBagLayout创建网格包布局管理器，再如BorderLayout()创建一个 Border 布局，组件之间没有间隙等等
8. 数据库：
   1. 类/接口：Class、DriverManager
   2. 方式：通过Class.forName()注册Mysql驱动程序，然后通过DriverManager。getConnection()与数据库建立连接

# 第三部分 代码分析

## **编码规范**

1. 文件编码方式：GBK (Chinese Internal Code Specification)
2. 编码说明：GBK的编码避免了在swing组件上显示中文时出现乱码的情况

## **命名规范**

1. 文件命名规范：直接采取文件作用英文名作为文件名
2. 变量/函数命名规范：采取驼峰式(Camel-Case)命名, 混合使用大小写字母来构成变量和函数的名字

## **模块分析**

1. **Controller：存放控制类文件**
2. **Element：存放游戏中的各种对象，如玩家、怪物等**
3. **Main：初始化一些全局属性后开启游戏**
4. **Mysql：存放数据库控制文件**
5. **Session：存放游戏当前玩家的一些信息**
6. **View：视图类，展示游戏内容**

## **源文件分析**

1. **Auto.java：游戏运行控制**
2. **AutoArc.java：判断能否上下楼**
3. **GameKeyListener.java:监听器，监听当用户与道具、商店、NPC交互时对键盘的操作**
4. **Music.java:背景音乐**
5. **Enemy.java:设置敌人的名称与属性**
6. **GameElement.java:设置游戏中每个元素的位置、高度、宽度**
7. **Man.java:设置玩家使用角色的属性值、初始状态；设置玩家触碰到不同事物触发的不同事件**
8. **Manager.java:管理每层的地图**
9. **MazeGameLoad.java:用于怪物手册去重复功能**
10. **Obj.java:将资源中的图片显示到游戏地图中**
11. **MyMain.java:主函数，保存屏幕信息，新建各个类的对象**
12. **DataConnection.java:注册MySQL驱动程序，获取数据库连接**
13. **User.java:用户的注册登录**
14. **Battle.java:与怪物的战斗界面**
15. **Client.java:存档读档功能**
16. **Compass.java:楼层跳跃功能**
17. **EnemyLookPanel.java:查看敌人信息**
18. **ImagePanel.java:制定Jpanel的大小；画出背景**
19. **Login.java:登录界面**
20. **MyFrame.java:初始化；存档到服务器；组件配置及添加；开启线程**
21. **MyPanelleft.java:在游戏主界面左边栏中显示玩家的各项属性**
22. **MyPanelright.java:游戏主界面中显示前奏、通关尾声**
23. **Register.java:设置边框；组件的定义、配置；面板添加组件；容器添加面板**
24. **ScrollEnemyLookPanel.java:显示敌人信息时可以实现下拉**
25. **TalkPanel.java:商店、NPC等触发时弹出的对话框**
26. **Session.java:存储当前登录用户的相关信息**