

目 录

摘要.....	1
Abstract.....	2
第 1 章 引言.....	3
1.1 研究背景和意义.....	3
1.2 国内外研究现状.....	3
1.3 本设计所做的工作.....	4
1.4 工程复杂性分析.....	4
1.5 环境影响和社会效益.....	5
1.6 结构安排.....	5
第 2 章 开发环境及相关技术.....	6
2.1 开发环境.....	6
2.2 Java 语言.....	6
2.2.1 Java 语言的介绍.....	6
2.2.2 Java 语言的特点.....	6
2.2.3 Java 语言的配置.....	7
2.3 Eclipse.....	7
2.4 Swing.....	8
2.4.1 JFrame.....	8
2.4.2 JPanel.....	8
2.4.3 ActionListener.....	8
2.5 MySQL.....	8
第 3 章 游戏设计.....	9
3.1 游戏可行性分析.....	9
3.1.1 技术可行性.....	9
3.1.2 经济可行性.....	9
3.1.3 操作可行性.....	9
3.2 游戏需求分析.....	9
3.2.1 游戏基本规则.....	9
3.2.2 功能需求.....	10

3.2.3	用户需求.....	10
3.3	游戏流程图.....	13
3.3.1	程序界面流程图.....	13
3.3.2	游戏主体流程图.....	13
第 4 章	程序设计.....	15
4.1	数据库设计.....	15
4.2	模块设计.....	17
4.2.1	登录模块设计.....	17
4.2.2	注册模块设计.....	19
4.2.3	游戏模块设计.....	21
4.2.4	排行榜模块设计.....	26
4.3	关键技术.....	27
4.3.1	布局.....	27
4.3.2	美化.....	27
4.3.3	显示.....	28
4.3.4	进程.....	28
第 5 章	游戏测试.....	29
5.1	注册测试.....	29
5.1.1	用户名重复测试.....	29
5.1.2	手机号验证测试.....	30
5.1.3	注册测试.....	30
5.2	登录测试.....	31
5.2.1	登录失败测试.....	31
5.2.2	登录成功测试.....	31
5.3	游戏测试.....	32
5.3.1	游戏得分测试.....	32
5.3.2	元素下落测试.....	32
5.3.3	全局消除测试.....	34
5.3.4	死阵检测测试.....	34
5.3.5	特效元素测试.....	34
5.4	排行榜测试.....	36

5.4.1 进入排行榜成功测试.....	36
5.4.2 进入排行榜失败测试.....	37
第 6 章 总结与展望.....	39
6.1 总结.....	39
6.2 展望.....	39
参考文献.....	40
致谢.....	41
附录.....	42

基于 Java 的消除游戏的设计与实现

摘要：随着信息技术的高速发展，游戏已经成为人们生活娱乐中重要的一部分。为了开发人们的智力，增强图片识别能力，本毕设采用 Java 语言和 Eclipse 平台实现了一款十分有趣的消除游戏。游戏通过相邻两块元素的交换使三块相同元素形一条直线，然后再将其消除得分。本游戏有简单的登录注册功能，并将用户在游戏的最终得分存入本地数据库以实现排行榜功能。该游戏具有益智性和竞技性等特点，对锻炼大脑和解决娱乐问题有着一定的使用价值。

关键词：Java; Swing; 消除

The Design and Implementation of the Elimination Game Based on Java

Abstract:With the rapid development of information technology, the computer game has become one of the most important parts in people's life. In order to develop people's intelligence and enhance image recognition capability, this project designs a funny elimination game based on Java programming language and Eclipse development platform. In this game a player can earn scores by eliminating two elements which are in a straight line. The game has simple functions such as log-in and register, and the final score can be saved in local database. The game has the characteristics competition, and has certain value in exercising the brain and solving entertainment problems.

Key words: Java; Swing; Eliminate

第 1 章 引言

1.1 研究背景和意义

伴随着生活水平的提高,信息科技的迅猛发展,程序员们为了丰富人们的业余生活、培养人们多方面的综合能力和更好的开发个人智力,所以游戏便成为人们生活中休闲娱乐的必备品。游戏产业不但催动了科技的发展,还成为经济增长的一大重点。为了开发一款老少皆宜,可玩性强的游戏,程序员们可谓是绞尽脑汁,甚至结合了心理学等其他因素,从各方面分析游戏的需求性和可玩性,使人们依赖上、喜欢上玩游戏。

提到消除游戏大家想必都会想到 2015 年乐元素公司开发的《开心消消乐》^[1],此游戏一经发行即获得巨大成功,无论男女老少都喜欢在饭后玩上一盘。其新颖的精力值系统使玩家一直沉迷于此,至此以后很多的游戏都延续了这一“饥饿”特点。

不过对于消除类游戏的起源,普遍的公认是俄罗斯的一位不知名的程序员于 1984 年 6 月开发的休闲游戏《俄罗斯方块》^[2],通过不同形状的方块依次填补以消除得分,它规定了消除类游戏最基本的游戏规则:玩家需要通过不断消除游戏中出现的元素而获得分数。这是一种目标明确,规则简单的游戏模式。此游戏一经发布后就席卷全球,光手机版的游戏就下载超过 1 亿。《俄罗斯方块》定义了模式,那 FC 红白机的《宝石方块》^[3]则定义了消除游戏的规则,不断的交换方块消除方块。

目前消除游戏在手机上较为丰富,但桌面版的游戏并不常见。为满足桌面版游戏的需求,与此同时为了提高本科期间动手能力,巩固所学知识,并进一步了解游戏开发和 Java 的相关知识。此游戏便应运而生,而且开发游戏也是本人未来的梦想所在,可以为将来自己的职业道路铺上一块小小的基石,更好的适应 21 世纪信息时代的发展。

1.2 国内外研究现状

由于各种原因,我国的科技发展大多依赖于国外的科技发明,游戏也是如此。国外的优秀游戏如雨后春笋般遍地开花,但我国目前许多热门游戏都是模仿或者进口,自主创新游戏少之又少^[4]。主要原因还是科技发展能力不足、国内对游戏存在一定的偏见和教育制度的不完善等。虽然目前与国外相比有较大的差距,但随着我国信息产业的迅猛发展,近年来也有一些优秀的国产游戏诞生。为此,应该继续大力扶持信息产业的发展,继续加强自主创新能力,继续加强版权意识。

1.3 本设计所做的工作

①资料的收集

通过任务书所给的参考文献和自己对游戏开发的了解，收集了与 Java 游戏开发^{[5][6]}相关的一系列资料，包括书籍的购买和相关 PDF 的下载。

②平台的搭建

本毕设基于 Java 语言、Eclipse 平台^[7]、MySQL 数据库^[8]和 Swing 安装包^[9]这些在本人的电脑里已经下载安装、配置好了，所以开发平台不需要重新搭建。

③游戏的设计

根据任务书和开题报告的要求，对游戏进行可行性分析和需求分析，设计游戏的流程图和 GUI 面板^[10]。编写了一系列的文档，正所谓工欲善其事必先利其器。

④代码的编写

根据点③的设计和点①所收集的资料，进行游戏代码的编写，设计游戏的数据库。

⑤游戏的测试

根据软件工程测试方法^[11]，分别进行单元测试和集成测试，依次对游戏进行测试并编写测试报告

⑥说明书的编写

编写毕设设计说明书，对以上几点进行总结归纳。

1.4 工程复杂性分析

本项目满足复杂工程问题中的：

①必须要运用深入的工程原理，涉及到数据结构里的图和矩阵^[12]和 Java 面向对象原理^[13]，并且需要相关的算法实现消除机制；

②涉及多方面的技术，如 Java 语言的使用、文档设计、游戏建模以及游戏引擎的实现；

③需要建立合适的 Java 抽象模型^{[14][15]}，对矩阵进行模拟可能出现的情况；

④具有较高的综合性，包含多个相互关联的子问题，如 Java 程序存储原理^[16]和数据库的使用等；

⑤问题相关各方利益不完全一致，年龄不同的玩家对该游戏所持有的态度有所不同；

⑥问题中涉及的因素可能没有完全包含在专业工程实践的标准和规范中，如游戏的规模和游戏的模式。

1.5 环境影响和社会效益

本设计是开发一款游戏，所以对社会环境没有太大的影响，可能会影响到人们的业余生活中的娱乐部分。若本游戏能够联网使用并发布到某小游戏网站，对开发者和网站所有者可能会产生一部分社会效益，有助于游戏的传播与分享。

1.6 结构安排

本毕设按照软件开发标准流程，将本毕设说明书分为以下内容：

第一章：引言

第二章：开发环境和相关技术的介绍

第三章：游戏设计

第四章：程序设计

第五章：游戏测试

第六章：总结

第 2 章 开发环境及相关技术

本游戏使用了 Java 语言和编程功能强大且免费的开发工具 Eclipse，导入 Swing 开发工具包。通过各种组件和模块的搭配使用，集成了 MySQL 数据库完成了这款有趣的游戏。

2.1 开发环境

操作系统：Windows 10

开发语言：Java（JDK 1.8）

编程工具：Eclipse for Java EE

数据库：MySQL 5.7.1；Navicat For MySQL

2.2 Java 语言

伴随着 21 世纪信息网络的快速发展，信息网络给人带来了丰富的机会和挑战。比如网上购物、消息传递、娱乐消遣和资源下载等，这些应用重要的就是技术的开发、开发语言的实用性和普遍性。而 Java 可以基本满足上述所有要求，并且实现了更低的成本。

2.2.1 Java 语言的介绍

Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言（以下简称 Java 语言）和 Java 平台（即 JavaSE，JavaEE，JavaME）的总称。Java 具有优秀的高效性、通用性、安全性和平台移植性，十分广泛的应用于超级计算机、数据中心、个人 PC、游戏控制台、互联网和移动电话，而且拥有全球最多的开发者数量。在全球一体化、云计算和互联网的产业环境下，Java 更具备了广阔的前景和显著的优势。根据最新的编程语言排行榜，Java 语言仍然多年连续蝉联冠军，并超出排行第二的 C 语言数倍，由此可见 Java 的强大和受众之广。

2.2.2 Java 语言的特点

Java 之所以能够受到如此的欢迎，不仅仅是它能够适应形式的发展，它还有以下诸多优点，如面向对象、解释性、强壮、安全、可移植、分布式、高性能、多线程、动态和调用等。下面挑选几个主要特点做些解释。

①简单性

由于设计者的反复考虑，使用者不需要扎实的编程基础就可以进行简单 Java 程序的

开发。原因之一是 Java 的语法格式与 C++ 较为相似，并且 Java 去掉了 C++ 中十分复杂且很容易出问题的部分，比如指针和内存管理等。因为 C++ 使用者如果稍不注意以上问题就会引起程序出错，使得编程门槛变得更高，使用起来也更为困难。Java 就很好的考虑的上述问题，有效的避免了错误的发生，使用起来简单省心。

②面向对象性

真正的面向对象的语言这是 Java 最重要的特点，Java 从开始设计的时候就从面向对象开始设计。它的封装性、多态性和继承性都很完美，最重要的是实现了 C++ 无法实现的动态联编，使得 Java 具有动态应用、扩展和代码重用等优点。在 Java 中，面向对象编程的基本元素是对象，并且它把变量和方法从功能上封装成一个可动态装载、可重用的整体。由此可知，类是对象的集合，在同一个类里的对象有着相同的方法和性质，并且一个类建立后，可以很简单的将其继承，在其基础上增加新的功能，这些就是面向对象的特性。

③可移植性

C 语言的整数类型随着操作系统和机器硬件的变化而变化，在 Intel 648 上是 32 位，在 Windows 上是 16 位，这些变化使得 C 语言编写的程序的可移植性变得很低。而 Java 语言则规定数据类型只有一种标准，从不依赖于机器，很好的避免了上述不可移植性。并且我们经常能看到很多 Java 游戏在不同的平台使用，这就是移植性的体现。

2.2.3 Java 语言的配置

JDK (Java SE Development Kit) 是 Java 软件开发包的简称，开发任何 Java 程序都要先配置 JDK。一般最流行的是 Sun 公司提供的 JDK，可在其官网下载最新版本的 JDK 安装包，然后进入我的电脑-属性-高级系统设置-环境变量配置中的系统变量，新建名为 JAVA_HOME 的变量，加入 JDK 的安装路径，并在系统变量的 Path 中加入 %JAVA_HOME%\bin;%JAVA_HOME%\jre\bin; 在新建 CLASSPATH 变量，填入.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar，至此 Java 语言的配置已全部完成，使用相关的开发平台即可进行 Java 程序的开发。

2.3 Eclipse

Eclipse 是 IBM 公司开发的用来替代商业软件 Visual Age for Java 的下一代免费开源 IDE 开发环境，其本身就是一个框架和一组服务，并且通过各种插件来构建相应的开发环境。最主要的用来 Java 语言的开发，也可以通过安装插件使其作为 C++ 和 Python 的开发工具。

Eclipse 能很好应用于各种程序的开发，如 WEB 开发、数据库的开发以及桌面应用的开发，具有相当强大的功能。

2.4 Swing

Swing 是一个基于 Java 的开发工具包，专门用于开发 Java 应用程序用户界面的，基于其原来的抽象工具包（AWT），可以使应用程序使用任何可插拔的外观风格。只需要很少的代码，程序员就可以利用 Swing 灵活、丰富的功能和模块化的组件来构建优美的用户界面。

2.4.1 JFrame

每个图形界面都至少要有有一个 Swing 容器，JFrame 就是一个框架，Java 开发的应用程序界面都是用 JFrame 作为其基础，将其称为容器（container），将其他的组件放入其中以实现不同的功能界面。它也是屏幕上 Window 的对象，可以最大化最小化。

2.4.2 JPanel

Jpanel 也是 Java 图形用户界面工具包中的面板容器类，但它和 JFrame 不同，它是一种十分轻量的容器，可以加入到 JFrame 容器中，形成一个更小的容器。一个窗口会有很多功能，不同的功能布局也不同，所以不可能将其都放入 JFrame 容器中，因为一个容器只能设置一种布局，所以要另开容器比如 JPanel，相互搭配组合以实现不同的功能界面。

2.4.3 ActionListener

Java 将事件封装成事件类，并且为每一个事件定义了一个事件监听类。如果一个组件注册了事件监听方法，意思是该组件要进行响应指定的事件。所谓响应就是响应用户鼠标键盘事件，然后对用户界面做出一些改变。没有监听的组件是不能和用户交互的，所以 ActionListener 这个接口提供了一系列的响应功能。

2.5 MySQL

MySQL 是一种开发源代码的关系型数据库系统（RDBMS），并且使用结构化查询语言（SQL）进行数据库的管理。MySQL 的主要特点是健壮、快速和易用，相比 ORACLE 来说要小得多而且速度更快，更适合轻量级的应用。

第3章 游戏设计

3.1 游戏可行性分析

3.1.1 技术可行性

本游戏采用 Java 语言编写，在上一章节有介绍 Java 语言相关特点，这里就不再多赘述。游戏主体主要是用 Swing 开发包所提供的界面所设计，并巧妙的结合了按钮 JButton 和定时器 Timer 功能实现游戏本体。并且运用到了 Java 核心知识，包括线程等方面的知识，并应用了 JDBC 连接数据库，对本地数据库进行操作。编程对电脑配置、开发环境等要求不高，所以学生笔记本都能胜任，技术上基本可行。

3.1.2 经济可行性

本游戏设计开发都是基于免费开源软件 Eclipse 和 JDK，可以直接下载安装使用。图片等素材都来自网络上所共享的资源，并不需要支付版权费用。所需要的成本仅是人力和时间的投入，因此经济方面可行性没有问题。

3.1.3 操作可行性

游戏本意是让人们在闲暇时间能够打发时间，休闲娱乐。本游戏的受众人群可以是小学生也可以是上班族，只要你会点击鼠标，对图片有简单的认知能力都能很好的操作本款游戏。不过想要获得高分就需要一定的速度和判断能力，因此在操作上也是可行的。

3.2 游戏需求分析

3.2.1 游戏基本规则

本游戏基于消除类核心原则，并且加入了一定的创新元素：

- ①交换相邻的两个元素后有三个相同元素连成一条直线便可消除得分；
- ②当有四个以上相同元素连成一条直线时会产生特效元素，将特效元素与相邻任一元素交换可全局消除与被交换元素相同的元素；
- ③游戏需要判断当前元素矩阵是否为死阵；
- ④在一定时间内得分越高排行越靠前。

3.2.2 功能需求

游戏分数需要实时显示更新，并加入对数据库的操作，将游戏得分存入本地数据库，通过 SQL 语句查询本地数据库得分前十的玩家显示到排行榜面板。如图 3-1。

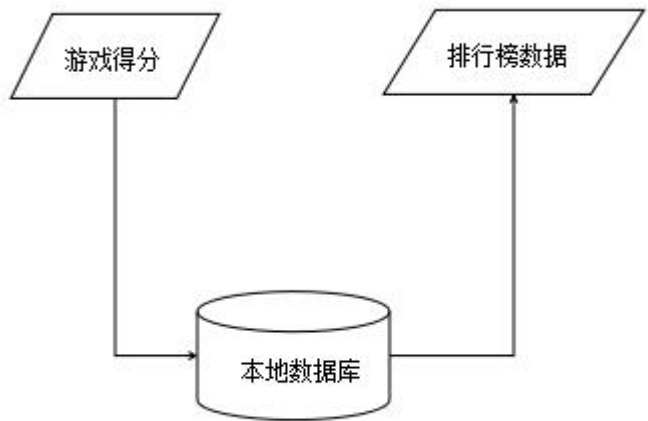


图 3-1 数据流程图

3.2.3 用户需求

游戏基于多用户，每位游戏玩家都有自己的账号，虽然是此游戏目前是单机版，但后续开发可能将其更新为网络版。用户可以注册成为玩家，玩家登录后查看自己分数和查看自己的排行情况。用例图如图 3-2。

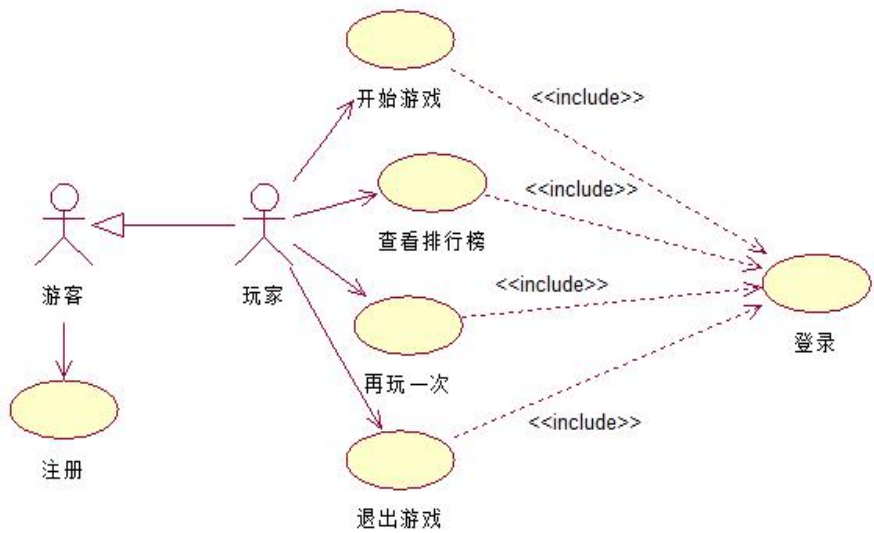


图 3-2 用例图

1. 用例描述：“注册”

简述：此用例允许游客注册账号，游客需要输入用户名、密码和手机号。

参与者：游客。

前置条件：游客打开游戏客户端，点击注册按钮。

主事件流如下：

①系统要求游客输入用户名、密码和手机号；

②点游客击‘注册’按钮，系统会请求数据库检查用户名是否重复。

后置条件：如果注册成功，系统将游客数据存入数据库并跳转到登录窗口，否则提示对应的错误。

2. 用例描述：“登录”

简述：此用例允许注册了账号的游客登录，游客需要输入用户名和密码。

参与者：注册了账号的游客。

前置条件：游客打开游戏客户端，输入用户名和密码，点击登录按钮。

主事件流如下：

①系统要求游客输入用户名和密码；

②游客点击‘登录’按钮，系统请求数据库会检查用户名和密码是否匹配。

后置条件：如果登录成功，跳转到游戏界面，如果用户名和密码不匹配则提示相应的错误，系统状态不变。

3. 用例描述：“开始游戏”

简述：此用例允许登录后成为玩家的游客开始游戏。

参与者：玩家。

前置条件：玩家登录后点击‘开始游戏’按钮。

主事件流如下：

①游戏客户端展示游戏界面，元素按钮变为可点击状态；

②游戏开始倒计时。

4. 用例描述：“查看排行榜”

简述：此用例允许玩家查看排行榜数据。

参与者：玩家。

前置条件：游戏倒计时结束，玩家点击‘查看排行榜’按钮。

主事件流如下：

①游戏客户端请求数据库查询得分前十的玩家；

②将排行榜数据显示在游戏面板上。

5. 用例描述：“再玩一次”

简述：此用例允许玩家重新开始游戏。

参与者：玩家。

前置条件：游戏倒计时结束，玩家点击‘再玩一次’按钮。

主事件流如下：

- ①本次游戏得分数据清零，跳转到全新游戏界面。

6. 用例描述：“退出游戏”

简述：此用例允许玩家退出游戏，关闭游戏客户端。

参与者：玩家。

前置条件：玩家点击‘退出游戏’按钮或关闭游戏窗口。

主事件流如下：

- ①游戏客户端窗口关闭，后台程序关闭。

3.3 游戏流程图

3.3.1 程序界面流程图

界面流程图如图 3-3 所示。玩家需要先进行注册登录，系统会判断用户名是否重复、密码验证是否通过。玩家成功登陆后点击开始游戏进行游戏，游戏结束后系统会判断玩家所得分数是否进入排行榜，然后弹出提示框是否查看排行榜、再玩一次或退出游戏。

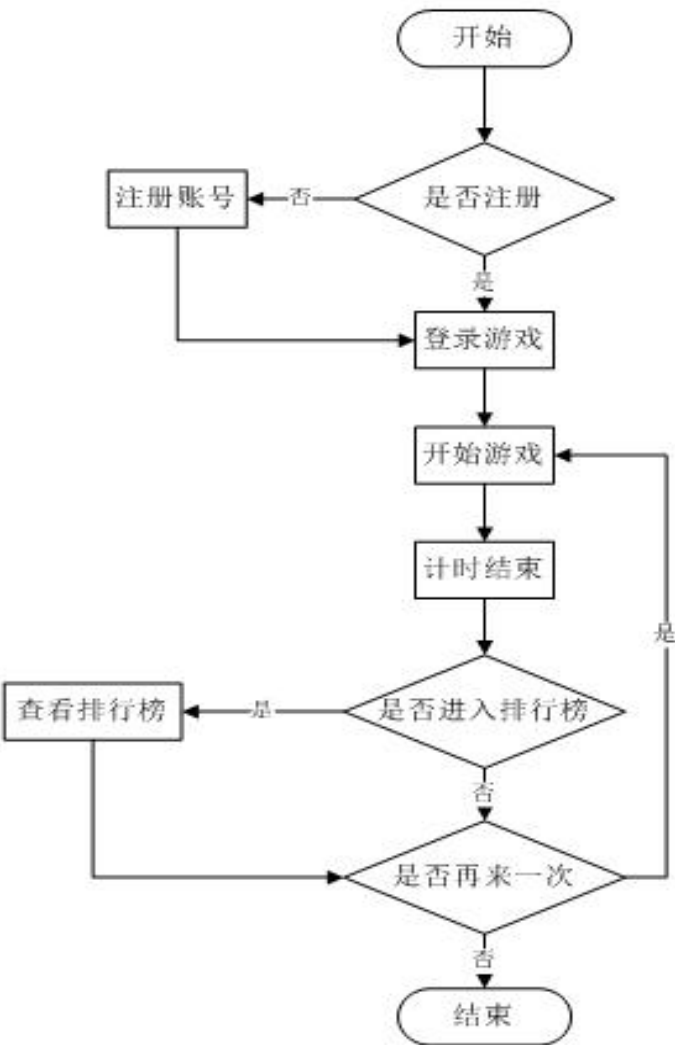


图 3-3 界面流程图

3.3.2 游戏主体流程图

游戏的判定过程是一个很关键的过程，也是游戏的核心思想。消除游戏基于二维矩阵，矩阵里不同的行列值代表着不同的元素。每次游戏的初始化都要遍历矩阵看其中是

否有三个以上行列值相同且相邻存在于同一列或同一行，同时要判断矩阵是否为死阵，死阵则指矩阵任何相邻两个元素交换后都不能形成三连状态，则要重新初始化矩阵。交换元素时要判断是否相邻，若交换后能形成三个以上行列值相同则将消除，否则将元素交换回来。被消除的元素的上方元素向下移动，最上面空白的位置进行随机填补，填补后全局遍历矩阵，再判断是否有三连，重复以上行为直至整个矩阵没有三连元素存在。游戏主体流程图如图 3-4 所示。

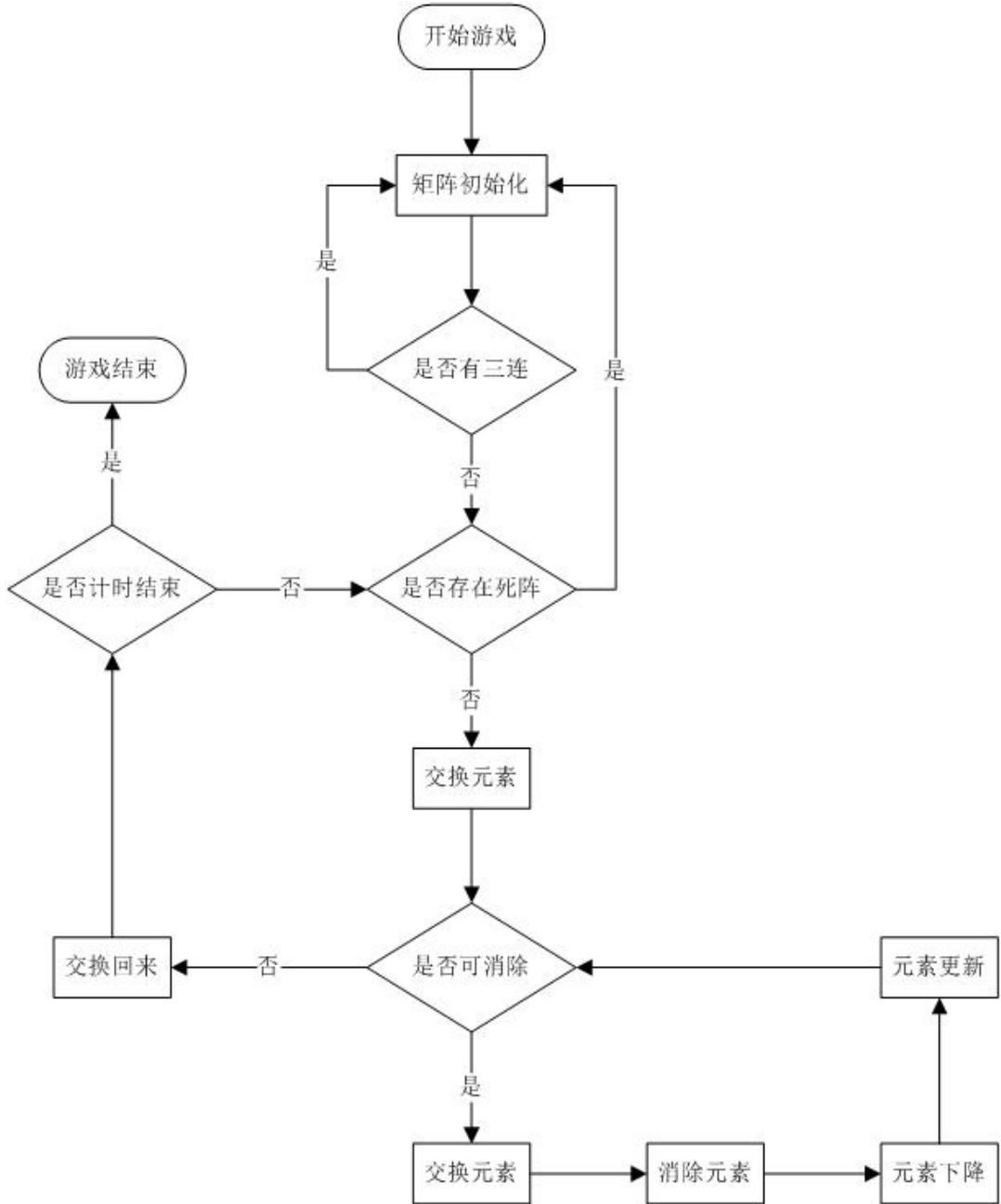


图 3-4 游戏主体流程图

第 4 章 程序设计

4.1 数据库设计

根据第三章的游戏设计可知，游戏包含用户登录模块和排行榜模块等，这两个模块需要用到数据库方面的功能，且数据库的设计满足 3NF 范式。本游戏使用的是 MySQL 数据库，并且使用 Navicat For MySQL 操作数据库。相关的表如下：

表 4-1 User 表

字段名	数据类型	字段长度	允许空值	备注
Id	int	10	否	主键，自增
Name	Varchar	20	否	姓名
Password	Varchar	20	否	密码
Phone	Varchar	20	否	手机号

表 4-2 Rank 表

字段名	数据类型	字段长度	允许空值	备注
Id	Int	10	否	主键，自增
Name	Varchar	20	否	姓名
Score	Int	10	否	得分
Time	Date	0	否	时间

数据库建表语句如下：

User 表：

```
CREATE TABLE `user` (  
    `id` int(10) NOT NULL AUTO_INCREMENT,  
    `name` varchar(20) NOT NULL,  
    `password` varchar(20) NOT NULL,  
    `phone` varchar(20) NOT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
```

Rank 表:

```
CREATE TABLE `rank` (  
  `id` int(10) NOT NULL AUTO_INCREMENT,  
  `name` varchar(20) NOT NULL,  
  `score` int(10) NOT NULL,  
  `time` date NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=51 DEFAULT CHARSET=utf8;
```

实体类图如图 4-1，一个 User 对应多个 Rank，因为一个人也可以占据排行榜前十的全部名额；一个 Rank 只能对应一个 User。

User 类里有 Login() 方法，请求数据库验证用户登录帐号和密码，匹配则返回真，否则假；Register() 方法，向数据库插入一条用户注册信息，成功插入则返回真，否则假；Query() 方法，用于查询用户注册的帐号是否重复，返回一个 User 类。

Rank 类里有 insert() 方法，向数据库插入一条游戏得分记录，成功插入则返回真，否则假；Query() 方法，查询数据库得分前 10 的用户，返回一个 Rank 的 List 类。

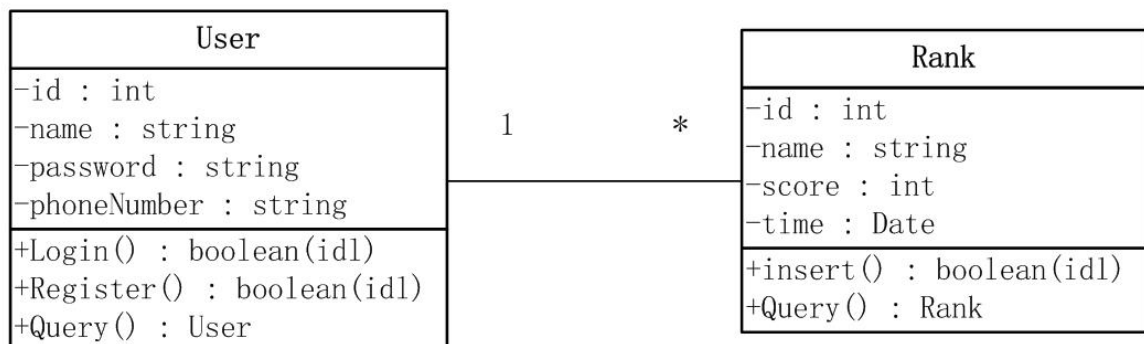


图 4-1 类图

4.2 模块设计

4.2.1 登录模块设计

窗口程序基于 Swing 组件，在 JFrame 的基础上增加了多个 JPanel 用来存放用户名和密码输入框等相应组件，并对所有 JButton 实施监听响应对应的操作和实时跳转。点击登录按钮后系统查询后台数据库与输入值进行匹配，成功则跳转到开始游戏画面，失败则做出相应的错误提示。

该模块由 Login.java 文件实现，是游戏启动的第一步，简单的实现了登录功能，主要成员变量如表 4-3 所示。

表 4-3 Login 类中主要成员变量表

名称	变量类型	变量描述
Buttona	JButton	登录按钮
Buttonb	JButton	注册按钮
Name	JTextField	登录文本输入框
Password	JPasswordField	密码文本输入框

类中包含如下方法，见表 4-4。

表 4-4 Login 类中主要方法

方法名	功能
Login()	登录窗口的绘制
clearText()	清空文本框

Login 类登录界面如图 4-2，登录注册按钮两个按钮都加入监听，在继承了 ActionListener 的类中进行判断按钮所按的是哪个，然后进行后续的判断操作。



图 4-2 登录框

模块流程图如图 4-3。

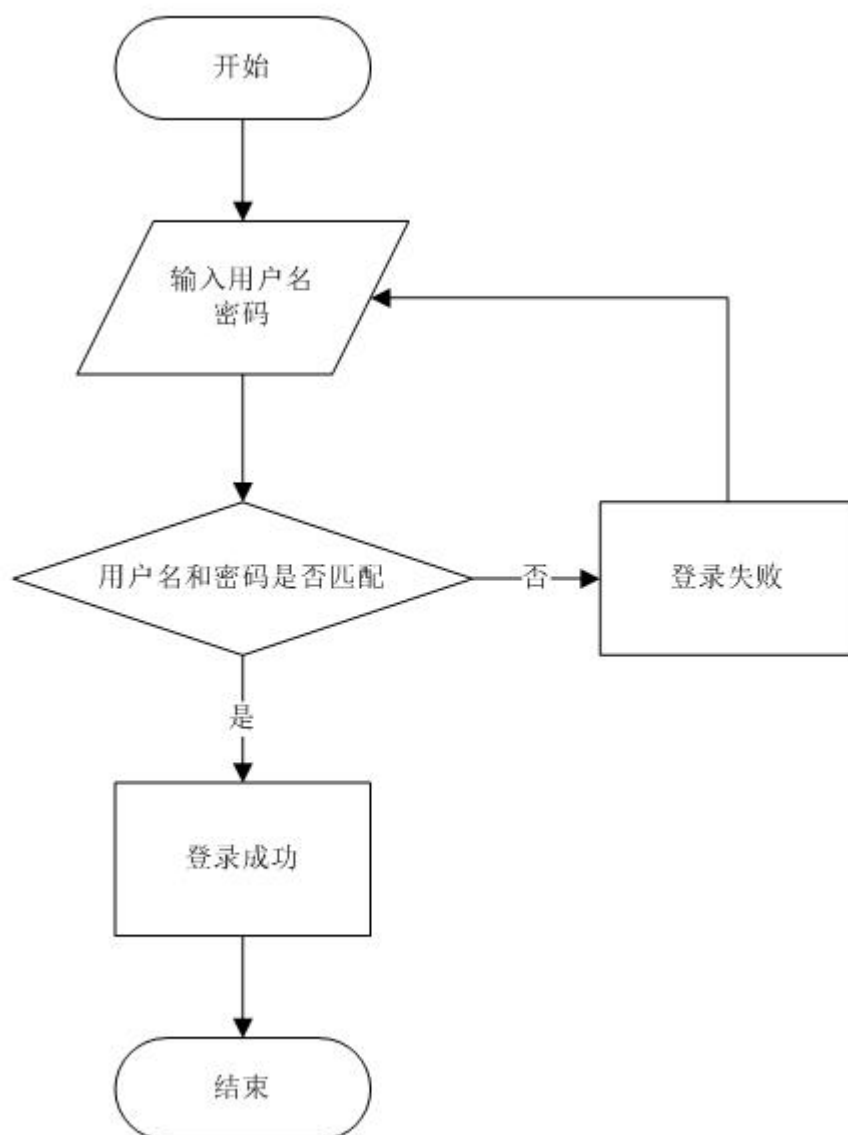


图 4-3 登录流程图

源代码见附录 40 页。

4.2.2 注册模块设计

同样基于 Swing 组件，将组件以 FlowLayout()方法上下分布。点击注册按钮时会查询数据库验证用户名是否重复，若无重复则向本地数据库插入一条用户记录，否则提示相应的错误信息。

该模块由 Register.java 实现，实现了简单的注册功能和密码校验，成员变量表如表 4-5。

表 4-5 Register 类中主要成员变量表

名称	变量类型	变量描述
Buttona	JButton	注册按钮
Buttonb	JButton	取消按钮，返回登录窗口
Name	JTextField	登录文本输入框
Password	JTextField	密码文本输入框
PhoneNumber	JTextField	手机号文本输入框

类中包含的方法如表 4-6。

表 4-6 Register 类主要方法

方法名	功能
Register()	注册窗口的绘制
clearText()	清空文本框

Register 类程序窗口如图 4-4，同样所有按钮都加入监听。

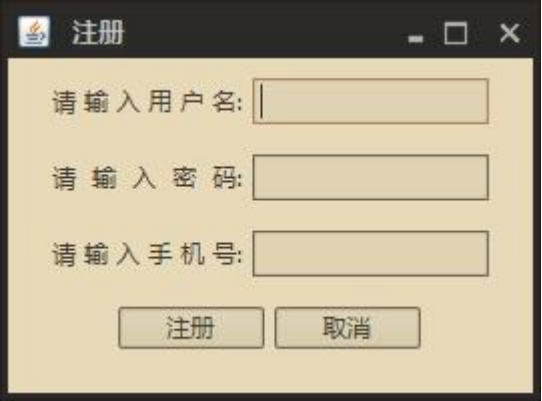


图 4-4 注册界面

注册模块流程图和登陆模块相差不大，如图 4-5。

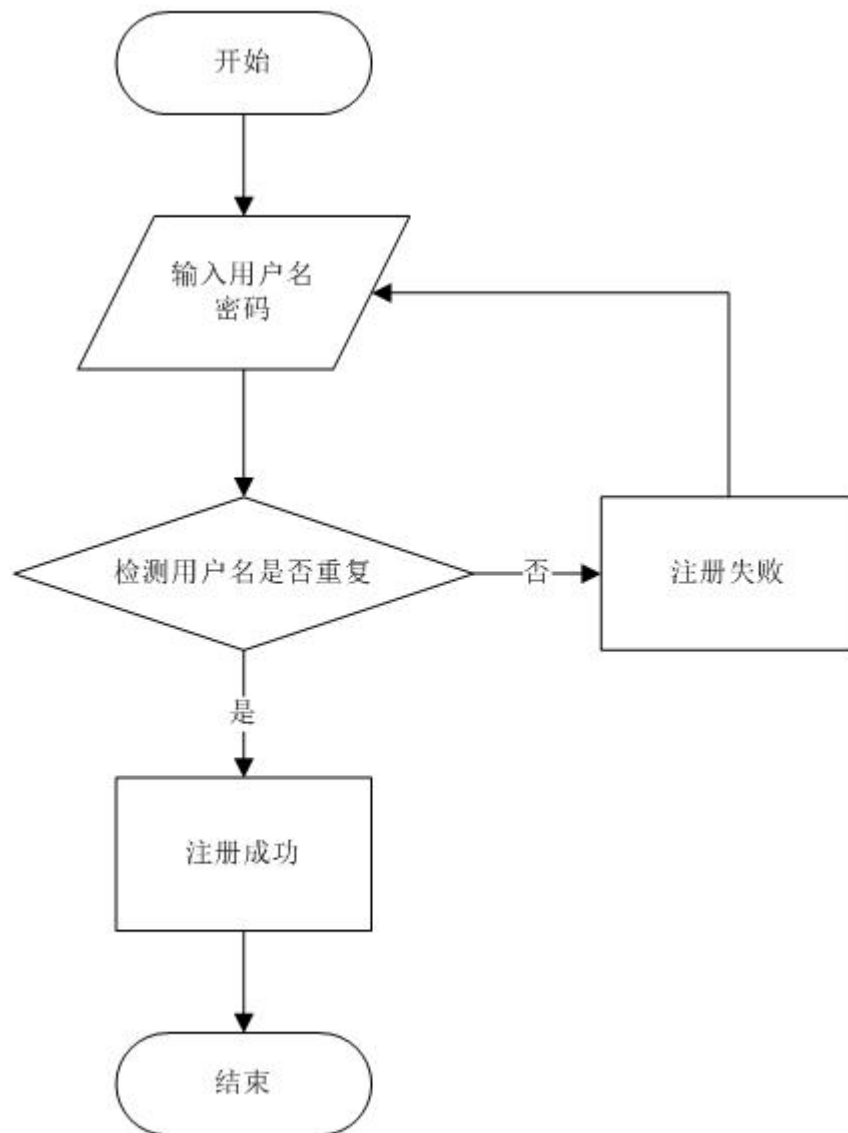


图 4-5 注册模块流程图

源代码见附录 41 页。

4.2.3 游戏模块设计

由第三章的游戏设计可知，本游戏是基于 Java Swing 工具包开发，所填充的元素都是基于按钮 JButton，系统同时遍历所有按钮来判断哪个位置的按钮被按到，可以进行点击交换并且判断。根据第三章的游戏流程图可知本游戏有多个函数同时进行，并且有一定的顺序和判断。本模块流程图可见第 3 章 3.3.2 游戏主体流程图。

游戏模块由 MyGame.java 实现，相关的成员变量如表 4-7，主要方法如表 4-8 所示。

表 4-7 MyGame 中主要成员变量表

名称	变量类型	变量描述
Buttona	JButton	游戏开始按钮
Buttonb	JButton	游戏退出按钮
Jindu	JProgeressBar	游戏进度条
Button[][]	JButton	二维数组的元素按钮
Iocn[]	ImageIcon	元素对应的值
Empty	Int	值为0时元素为空
IsThreeLinked	Boolean	为真时有三连，假则无
Grade	Int	分数

表 4-8 MyGame 类中主要方法

方法名	功能
MyGame()	游戏窗口的绘制
initAnimalMatrix()	初始化动物矩阵，将其随机赋值
removeLinked(int x, int y)	消除函数，以(x,y)为基点将相同的元素置为空
removeAll(int x ,int y,int x1,int y1)	消除函数，将(x,y),(x1,y1)相同的元素都置为空
globalSearch(int flag)	遍历函数，flag为1时判断，为2时消除
isThreeLinked(int x, int y)	判断(x, y)点是否有三个以上的相同元素
isDeadMatrix()	判断当前元素矩阵是否为死阵
updateAnimal()	更新元素矩阵，将元素值为空的随机赋值
downAnimal()	消除的元素上方的元素向下移动
print()	绘制元素图片
swapAnimal(int x, int y)	交换元素
init()	重复初始化，直到元素矩阵没有三连情况

游戏涉及到众多方法函数，下面依次介绍表 4-8 中方法的具体实现。

(1) MyGame()

游戏主体面板，通过 JPanel 的组合绘制了游戏面板包括开始游戏和退出游戏按钮，并且将分数和倒计时通过 Timer 类实时显示在面板上。元素矩阵由 N*N 个 JButton 组成，通过 GridLayout(N,N,1,1)布局按矩阵均匀排列，元素初始值随机赋值，并且全部加入监听，元素的图片通过 ImageIcon 类从本地图片进行设置，locn 数组的值从 1 到 7 代表不同的元素，8 号元素则为特效元素。

源代码见附录 45 页。

(2) initAnimalMatrix()

单独的初始化动物矩阵，两个 for 循环依次遍历，利用 math.random 进行随机赋值。在最后的 init()需要用到。

源代码见附录 46 页。

(3) removeLinked(int x, int y)

消除函数，游戏的核心所在，所传入的 (x, y) 会依次判断其上下左右是否有与相同元素值的元素，若相同的值大于等于三则将其置为空。同时会判断相同值是否大于等于四，若是则元素消除后将 (x, y) 置为特效元素（交换其相邻的任一元素可以全局消除与被交换的元素相同值的元素）。分数的计算由被消除值的个数*10 相加，实时显示在游戏面板里。

源代码见附录 47 页。

(4) removeAll(int x, int y, int x1, int y1)

特效元素所触发的函数，当有四个以上相同元素排列成行列消除时会产生特效元素，元素与特效元素交换后，将全矩阵与被交换的元素值相同的元素消除，能够得到很高的一个分数。

源代码见附录 50 页。

(5) isThreeLinked(int x, int y)

判断 (x, y) 点是否有三个以上相连，和 removeLinked (int x, int y) 函数差不多，如果有则返回真否则返回假，单独列出来是因为全局搜索和判断“死阵”的需要。

源代码见附录 50 页。

(6) isDeadMatrix()

判断元素矩阵当前是否为“死阵”，所谓“死阵”则是任意两个元素交换后不能形成三连以上元素序列，此时游戏将无法进行，需要重新打乱元素排列。判断过程类似于回溯法，先交换后，依次遍历全局，若有三连则将元素交换回来并返回 **Ture**，否则返回 **False** 调用 **init()**函数重新打乱元素排列。

源代码见附录 51 页。

(7) updateAnimal()

全局判断所有元素值为空的按钮，将其随机赋值更新，并在面板上重新显示按钮图片，并使其可点击，简单的遍历过程。

源代码见附录 52 页。

(8) downAnimal()

从元素矩阵底部（行列值从大到小）遍历，遇到元素值为 0 的按钮则将空值元素与其上方的元素交换，依次进行上述操作达到元素移动的目的。

源代码见附录 54 页。

(9) print()

设置图片函数，利用 **JButton** 的 **setIcon** 方法，通过 **Icon** 数组将元素图片显示到按钮上。每一次元素的变化都要调用该函数以实现实时更新和动态刷新的视觉效果。

源代码见附录 55 页。

(10) swapAnimal(int x, int y)

核心函数之一，有一个标记值（**isDoubleClicked**）记录此次是第一次点击还是第二次点击，若第二次则判断两次标记的元素是否相邻，也就是二者的坐标值相减的绝对值不超过 1。若是则交换二者元素的值，再进入判断是否形成三连等一系列动作，否则交换回来。在此基础上加入了特效元素的判断，若第一次点击的是特效元素，则第二次点击判断完相邻后调用 **removeAll(x, y)**函数。

源代码见附录 56 页。

(11) init()

初始化函数，重复调用 `globalSearch` 函数，若有三连则调用 `initAnimalMatrix()`，直到没有三连为止。由于程序运行速度很快，所以玩家在游戏开始的时候并不会发觉有此函数在运作。

源代码见附录 58 页。

以上为游戏模块的基础功能介绍，下面介绍一下游戏的重要部分，游戏动画实现。

(1) Timer 类

`Timer` 类是一种线程的设施，可以用来实现某一个时间后或一定的时间段内安排某个任务的执行或重复执行，这个任务可以是方法等任何事件。但是 `Timer` 的所有任务都是由用一个线程调度的，所以所有的任务都是串行执行的。同一时间只能有一个任务执行，前一个任务的异常或延迟都会影响下一个任务的执行。

在这里为了实现动画效果，定义了两个 `Timer` 定时任务，运行时间都是 1S 并且在方法里结束。将 `downAnimal()` 和 `updateAnimal()` 放入所设置的 `Timer` 定时器任务里，在消除元素后等待 1s 再进行元素的下降和填充，使得游戏过程变得更加直观。通过 `Timer` 的 `start()` 方法和 `stop()` 方法来实现任务的调度。

源代码见附录 60 页。

(2) 事件流

游戏倒计时也是一个 `timer` 类，每 1S 运行一次，同时有 `int` 型的 `times` 变量。`Timer` 任务每秒运行一次则 `times` 加 1，当 `times` 大于等于 100 时倒计时结束，系统会自动跳出游戏结束提示框，里面包含玩家所得分数和相应的语句提示包括再玩一次和查看排行榜等，该判断由 `Over.java` 实现，然后将本次游戏数据插入本地数据库。

源代码见附录 59 页。

(3) 游戏结束

游戏结束框由 `over.java` 类实现，系统会请求数据库查询分数前 10 名的玩家，若查询结果小于 10 则直接提示玩家进入排行榜，若查询结果等于 10 则从小到大依次与玩家本次所得分数进行比较，若大于所查数据中其中一条则提示玩家进入排行榜，否则提示玩家未能进入排行榜。

游戏运行截图如图 4-4，进入排行榜如图 4-5，未进入排行榜如图 4-6。

源代码见附录 61 页。



图 4-4 游戏截图

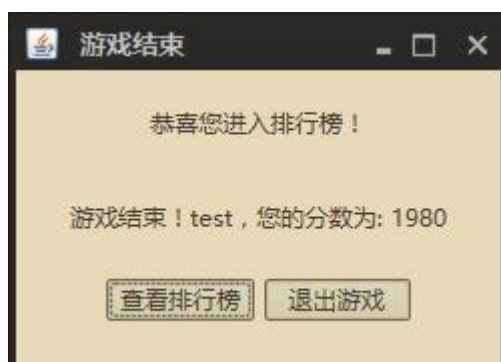


图 4-5 进入排行榜



图 4-6 未进入排行榜

4.2.4 排行榜模块设计

主界面基于 JTable，其提供表格类的面板工具，并定义了一个二维数组对象 Object [][]，Object 是任何对象的父类，可以存放任何类型的变量，所以将其值定义为数据库查询所得数据。因为在 Swing 下的 JTable 中载入图片会错误的显示成其路径，所以重写了 DefaultTableCellRenderer 类使其实现显示冠亚季军图片。若玩家本次游戏所得分数进入了排行榜，则系统记录下 id 并且标识其排行位置。源代码见附录 64 页。

本功能由 RankList.java 实现，主要成员变量如表 4-9 所示。

表 4-9 RankList 中主要成员变量表

名称	变量类型	变量描述
Buttona	JButton	再来一次按钮
Buttonb	JButton	退出游戏按钮
data	Object[][]	存放图片和排行榜信息
Columns	String[]	存放表头

类中包含如下方法，见表 4-10。

表 4-10 RankList 类中主要方法

方法名	功能
RankList()	排行榜窗口的绘制
Main()	调用窗口

排行榜截图如图 4-7。



排名	姓名	分数	时间
1	test	1980	2017-05-06
2	TWB	280	2017-04-18
3	213	160	2017-04-16
4	里里	150	2017-04-15
5	谭文波	120	2017-04-13
6	123	120	2017-04-18
7	213	120	2017-04-16
8	213	120	2017-04-16
9	123	100	2017-04-18
10	123	70	2017-04-18

再玩一次 退出游戏

图 4-7 排行榜

4.3 关键技术

在第一章介绍了 Swing 的几个基本组件等，下面详细的介绍本游戏使用的关键技术，包括 Swing 里一些关键组件的介绍，如何布局等。

4.3.1 布局

Swing 是如何让组件按照我们想要的位置分布排放的呢？这里就要提到界面布局，Swing 有多种布局，统一以***Layout()命名，不同的布局有不同的显示排版效果。本游戏主要用了以下几种布局。

(1) FlowLayout（流式布局）

顾名思义，将组件按本身大小和窗口大小从左到右依次布局，若组件大小超出窗口大小则换行显示，它是默认居中对齐的。

(2) GridLayout（网格布局）

将组件按 M 行 N 列网格格式依次布局，可以设置每格之间的间隙。本游戏主体元素的布局就是应用了这种布局 GridLayout(8,8,1,1)指 8 行 8 列的网格布局上下左右间隙为 1。

其他的就不再详细介绍。

4.3.2 美化

Swing 的直观画面给人的感觉是比较简陋的，为了让玩家感到游戏优美则需要对游戏窗体进行美化操作，在这里导入了 Substance.jar 工具包。Substance 提供了一系列的窗口皮肤和主题，导入后简单的调用后就能让界面变得更加美观。美观前后图见图 4-8 和图 4-9。



图 4-8 美化前

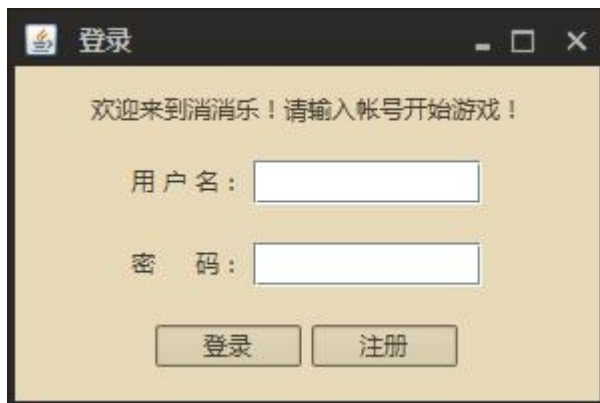


图 4-9 美化后

具体美化代码见附录 41 页。

4.3.3 显示

此代码可以让游戏窗口能在任何电脑中居中显示，通过 awt 中 Dimension 的 `getScreenSize()` 方法可以获取屏幕尺寸，然后将屏幕的长宽减去游戏窗口的长宽再除 2 就可以使游戏窗口在任何屏幕居中显示。

部分代码如下：

```
Dimension screen = frame1.getToolkit().getScreenSize();  
frame1.setLocation((screen.width - frame1.getSize().width) / 2,  
                    (screen.height - frame1.getSize().height) / 2);
```

4.3.4 进程

每个 Swing 窗口都是一个进程，开启时会占用计算机 CPU 的资源，如果直接关闭窗口是无法关闭后台进程的，需要在代码里加入关闭进程代码。这样再关闭窗口的同时可以将后台进程关闭。所以每个窗口类都要加入以下该代码：

```
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

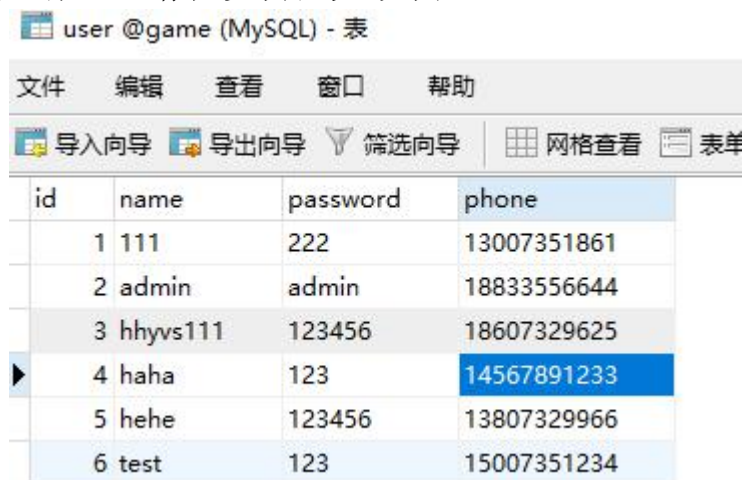
第 5 章 游戏测试

5.1 注册测试

游戏要求用户在进行游戏前先要登录，但是没有账号的玩家需要进行注册。下面就注册模块进行单元测试。

5.1.1 用户名重复测试

在数据库预先可知已经存在以下账号，如图 5-1。



id	name	password	phone
1	111	222	13007351861
2	admin	admin	18833556644
3	hhyvs111	123456	18607329625
4	haha	123	14567891233
5	hehe	123456	13807329966
6	test	123	15007351234

图 5-1 玩家信息

我们选取账号为：hhyvs111 来进行测试，测试过程如图 5-2 和图 5-3。



图 5-2 输入注册信息



图 5-3 用户名已存在

由上图可知测试成功。

5.1.2 手机号验证测试

手机号码由 11 位组成。目前的本游戏暂未能验证手机号码的真实性，只能验证其是否符合长度。测试过程如图 5-4 和图 5-5。



图 5-4 输入注册信息



图 5-5 手机号不正确

测试成功。

5.1.3 注册测试

按注册要求进行正确性校验，查看注册信息是否存入本地数据库。测试过程如图 5-6 和图 5-7 所示。



图 5-6 输入注册信息



图 5-7 注册成功

查看数据库是否存在账号为：hhyvs520 的数据条目。如图 5-8。

	5	hehe	123456	13807329966
	6	test	123	15007351234
➤	7	hhyvs520	a13007351861	18607329625

图 5-8 数据库信息

测试成功。

5.2 登录测试

5.2.1 登录失败测试

用 5.1.3 注册的账号进行登录测试，将输入的密码改为：123456。看登录是否能成功，测试过程如图 5-9 和图 5-10 所示。



图 5-9 用户登录



图 5-10 登录失败

测试成功。

5.2.2 登录成功测试

同理，用 5.1.3 注册的账号进行测试，输入正确的用户名和密码。测试过程如图 5-11 和图 5-12 所示。



图 5-11 用户登录

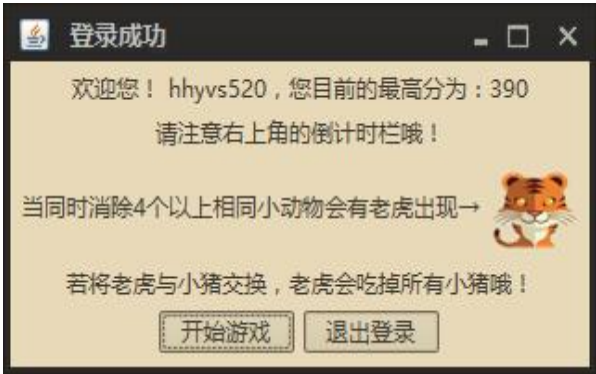


图 5-12 登录成功

测试成功。

5.3 游戏测试

5.3.1 游戏得分测试

由第四章的 4.2.3 游戏模块设计可知,游戏的得分以消除的元素个数*10 为本次交换得分,所以我们对游戏得分进行一下验证看是否得分正确。不过在测试的过程中元素的消除后下降更新的元素矩阵经过全局遍历后可能会进一步消除,所以结果可能不是预测的那么准确。

具体过程因篇幅有限就不一一截图了,经过一系列测试,得分与消除的元素准确对应。测试成功。

5.3.2 元素下落测试

由第四章的 4.2.3 游戏模块设计可知,元素消除后其上方的元素要向下移动,并且空白的元素要随机填补。测试过程可见图 5-13, 图 5-14 和图 5-15。



图 5-13 交换元素



图 5-14 元素消除



图 5-15 元素下落更新

测试成功。

5.3.3 全局消除测试

全局消除顾名思义，全局遍历元素矩阵，将有三连以上的元素序列消除。因为测试过程中显示的速度较快，全局消除的测试不能较好用图片来显示其过程，这里就省略了上传图片。

经一系列测试，每次消除矩阵后检查元素矩阵后没有发现有三连的状况。测试成功。

5.3.4 死阵检测测试

死阵意为交换任一相邻两个元素后元素矩阵没有可消除的三连元素序列。同 5.3.3，本测试不能较好的用图片展示，且在一个 8*8 的元素矩阵中，出现死阵的概率大概为 1/256，测试过程较为繁琐。在这里省略了其测试过程。

经一系列测试和多次试玩游戏，在出现了死阵的情况下元素矩阵重置成功。

5.3.5 特效元素测试

特效元素为四个以上相连元素消除后所得，标志为老虎。将其与任一元素交换可以消除全矩阵与被消除元素相同的元素。测试过程如图 5-16 至图 5-18 所示。



图 5-16 元素交换



图 5-17 特效元素

将其与上方的小狗交换，可见全局的小狗都成功消除。



图 5-18 全局消除

5.4 排行榜测试

5.4.1 进入排行榜成功测试

由图 5-19 可知，目前的排行榜的分数。



排名	姓名	分数	时间
1	test	1980	2017-05-06
2	hhyvs520	390	2017-05-08
3	TWB	280	2017-04-18
4	213	160	2017-04-16
5	里里	150	2017-04-15
6	谭文波	120	2017-04-13
7	123	120	2017-04-18
8	213	120	2017-04-16
9	213	120	2017-04-16
10	123	100	2017-04-18

再玩一次 退出游戏

图 5-19 排行榜

测试目标：分数达到 100 以上。可见图 5-20 和图 5-21。



图 5-20 进入排行榜

排行榜, hhyvs520 您的分数为 : 300			
排名	姓名	分数	时间
1	test	1980	2017-05-06
2	hhyvs520	390	2017-05-08
3	hhyvs520	300	2017-05-08
4	TWB	280	2017-04-18
5	213	160	2017-04-16
6	里里	150	2017-04-15
7	谭文波	120	2017-04-13
8	123	120	2017-04-18
9	213	120	2017-04-16
10	213	120	2017-04-16
再玩一次 退出游戏			

图 5-21 排行榜

测试成功。

5.4.2 进入排行榜失败测试

由图 5-21 的最新排行榜数据，所得的分数不能超过 120 分。测试过程如图 5-22 和图 5-23 所示。



图 5-22 未进入排行榜

排行榜, hhyvs520 您的分数为 : 90			
排名	姓名	分数	时间
1	test	1980	2017-05-06
2	hhyvs520	390	2017-05-08
3	hhyvs520	300	2017-05-08
4	TWB	280	2017-04-18
5	213	160	2017-04-16
6	里里	150	2017-04-15
7	谭文波	120	2017-04-13
8	123	120	2017-04-18
9	213	120	2017-04-16
10	213	120	2017-04-16
再玩一次 退出游戏			

图 5-23 排行榜

未进入排行榜，测试成功。

第 6 章 总结与展望

6.1 总结

通过这次毕业设计的动手编程和撰写毕业设计说明书，不仅巩固了本科期间学习的专业知识，而且加强了自己的动手能力。学习了一些编程的技巧，如静态变量的使用、线程的机制和数据库的设计等，进一步的了解了编程的核心知识，体会了做项目的不易。有时候一个错误就要 debug 很久，修改后又会有一系列的错误。因此学会了：在写代码的时候多加上注释，加强代码的可读性；多使用局部变量，防止变量的误用；多对代码进行思考，想象其步骤过程的实施等。而且本毕业设计是大学期间最后一次作业，从选题、收集资料和查阅相关书籍资料到最后的实际动手编程，遇到了一系列的困难与障碍，但是经过自己的努力和请教同学老师，最后问题都迎刃而解了。

毕设应用了《Java 面向对象编程》、《Java 游戏开发与编程》和《数据结构》等课程知识，解决了 Java 游戏开发中相关的复杂性工程问题，具体体现在：游戏开发种类多种多样，最后经研究对比确定了用 Swing 工具包进行消除游戏窗体的开发，并且应用相关组件实现简单的游戏交互，通过本地数据库实现数据的存储与更新；本游戏的设计需要把 Java 语言和 Swing 的特点综合使用，具有一定的综合性；游戏设计中存在成本与技术之间的冲突，游戏结合了开发者的技术实际情况，设计了基于 Java 的消除游戏。

游戏基于 Swing 工具包，但美观程度依旧不够，其样式格式较为古板。且因为时间和技术上的问题，游戏尚存在 Bug 未能解决，在元素交换后的全局遍历的过程中，有些不该消除的元素也被消除了。在思考了很长一段时候和查阅了较多资料后仍得不到解决，这是游戏的其中一不足之处。另外游戏的联网功能也未能实现，因为是一个桌面应用，与网页直接部署在服务器上不同，需要进行下载操作。此技术以本人目前的水平暂不能解决。

6.2 展望

游戏虽说不是生活的必需品，但却是人生必不可少的调味品。在未来，越来越多的游戏会被设计出来，而且人们参与游戏的方式也不仅仅是电脑和手机。现在的 VR 技术使游戏变得更具趣味性，人们不仅在视觉上充分参与，在感官上也融入游戏。并且游戏也带动了很多经济产业的发展，产生了一系列职业需求。本游戏设计虽然简单，但是也算是未来技术发展的一个缩影。我相信在将来的日子，随着生活品质要求变得更高，游戏开发的道路会越走越远。

参考文献

- [1] 百度百科.开心消消乐.<http://baike.baidu.com/item/开心消消乐?sefr=ps>.2017-01-10.
- [2] 百度百科.俄罗斯方块.<http://baike.baidu.com/item/俄罗斯方块/535753?sefr=ps>.2016-04-10.
- [3] 百度百科.宝石方块.<http://baike.baidu.com/item/宝石方块?sefr=ps>.2015-04-12.
- [4] 许强.我国网络游戏产业的特点和发展趋势[J].经济管理,2007(9):69-71.
- [5] 何青.Java 游戏程序设计教程[M].北京:人民邮电出版社,2014.
- [6] 陈锐,夏敏捷,葛丽平.Java 游戏编程原理与实践教程[M].北京:人民邮电出版社,2013.
- [7] 李松林,陈华清,任鑫.Eclipse 宝典[M].北京:电子工业出版社.2007.9.
- [8] 陶宏才.数据库原理与设计[M].北京:清华大学出版社,2005.
- [9] 王鹏.Java Swing 图形界面开发与案例详解[M].北京:清华大学出版社 2008.33.
- [10] 赵满来.可视化 Java GUI 程序设计[M].北京:清华大学出版社.2010.11.
- [11] 张海藩. 软件工程导论[M]. 北京:清华大学出版社,2003.
- [12] 严蔚敏,吴伟民.数据结构[M].北京:清华大学出版社,2007.
- [13] 王保罗 .Java 面向对象程序设计[M].北京:清华大学出版社,2003.
- [14] S.Vidal,.Over-exposedclasses in Java: An empirical study[J]. Computer Languages, Systems & Structures,2016.13.
- [15] Yu Lin,Danny Dig. A study and toolkit of CHECK - THEN - ACT idioms of Java concurrent collections[J]. Softw. Test. Verif. Reliab.,2015.45.
- [16] Abdelhak Mesbah,Jean-Louis Lanet,Mohamed Mezghiche. Reverse engineering a Java Card memory management algorithm[J]. Computers & Security,2017.25.

致谢

本毕设的顺利完成，首先要感谢我的指导老师申冬苏和本组其他老师，申老师在毕设期间给予了我很多帮助和指点。并且在本组老师的悉心指导下，我不但学到了丰富的专业知识，也为以后的工作打下了坚实的基础。老师们一丝不苟的工作作风和诲人不倦的育人精神让我受益匪浅。在此再一次向申老师和本组老师表示衷心的感谢。

大学四年转瞬即逝，在此期间认识了很多同学、朋友和老师，他们给予了我一系列的帮助和扶持使我圆满的完成了学业，在此我深深地表示由衷的感谢之情。

附录

(1) Login.java

```
import java.awt.*;
import javax.swing.*;

public class Login extends JFrame{
    JButton buttona = new JButton("登录");
    JButton buttonb = new JButton("注册")
    public static JTextField name = new JTextField();
    public static JPasswordField password = new JPasswordField();
    public Login(){
        this.setSize(300, 200);
        Dimension screen = this.getToolkit().getScreenSize();
        this.setLocation((screen.width - this.getSize().width) / 2,
            (screen.height - this.getSize().height) / 2);
        this.setTitle("登录");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setLayout(new GridLayout(4,0));
        JLabel label1 = new JLabel("欢迎来到消消乐！请输入帐号开始游戏！",JLabel.CENTER);
        JLabel label2=new JLabel("用 户 名： ");
        JLabel label3=new JLabel("密      码： ");
        name.setColumns(10);
        password.setColumns(10);
        JPanel panel1 = new JPanel();//放用户名和输入框
        panel1.setLayout(new FlowLayout());
        panel1.add(label2);
        panel1.add(name);
        JPanel panel2 = new JPanel();//放密码和输入框
        panel2.setLayout(new FlowLayout());
        panel2.add(label3);
        panel2.add(password);
```

```

        JPanel panel3 = new JPanel();    //放登录
        panel3.setLayout(new FlowLayout());
        panel3.add(buttona);
        panel3.add(buttonb);
        this.add(label1);
        this.add(panel1);
        this.add(panel2);
        this.add(panel3);
        MyListener mylisten = new MyListener();
        buttona.addActionListener(mylisten);
        buttonb.addActionListener(mylisten);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    public static void main(String[] args){
        EventQueue.invokeLater(new Runnable() {
            @Override
            public void run() {
                JFrame.setDefaultLookAndFeelDecorated(true);
                JDialog.setDefaultLookAndFeelDecorated(true);
                try {
                    UIManager
                        .setLookAndFeel(new
org.jvnet.substance.skin.SubstanceDustCoffeeLookAndFeel());
                    org.jvnet.substance.skin.SubstanceOfficeBlue2007LookAndFeel();
                } catch (Exception e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
                Login login=new Login();
                login.setVisible(true);
            }
        })
    }

```

```

    });
}
private void clearText() { //清空文本框, 密码框的输入
    password.setText("");
}
class MyListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == buttona) {
            List<User> list = UserOperate.queryUser(name.getText());
            if(list.isEmpty()){
                Message message = new Message(1); //调用 message 弹出消息窗口
                System.out.println("用户不存在");
                clearText(); //清空文本框 密码框的输入
            }
            else{
                String key = new String(password.getPassword()); // 取得密码
                User user = list.get(0);
                if(user.getPassword().equals(key)){
                    setVisible(false);
                    Message message = new Message(2); //调用 message 弹出消息窗口
                    System.out.println("登录成功");
                }
                else{
                    Message message = new Message(1);
                    System.out.println("密码错误");
                    clearText(); //清空文本框 密码框的输入
                }
            }
        }
        if(e.getSource() == buttonb){
            Register re = new Register();
            System.out.println("前往注册");
        }
    }
}

```

```

        setVisible(false);
    }
}
}
}

```

(2) Register.java

```

import java.awt.*;
import java.awt.event.*;
import java.util.List;
import javax.swing.*;

public class Register extends JFrame implements ActionListener {
    // 申明需要的组件
    public JTextField name = new JTextField(); //账户
    public JTextField password = new JTextField(); //密码
    public JTextField phone = new JTextField(); //手机号
    public JButton buttona = new JButton("注册");
    public JButton buttonb = new JButton("取消");
    JPasswordField jpfl;//密码框
    public Register(){
        this.setTitle("注册");// 窗口标题
        this.setLayout(new FlowLayout());
        this.setSize(270, 200);
        Dimension screen = this.getToolkit().getScreenSize();
        this.setLocation((screen.width - this.getSize().width) / 2,
            (screen.height - this.getSize().height) / 2);
        JPanel panel1 = new JPanel();//放用户名和输入框
        panel1.setLayout(new FlowLayout());
        JLabel JL1 = new JLabel("请 输 入 用 户 名:");
        panel1.add(JL1);
        name.setColumns(10);
        panel1.add(name);
    }
}

```



```

JPanel panel2 = new JPanel();//放密码和输入框
panel2.setLayout(new FlowLayout());
JLabel JL2 = new JLabel("请 输 入 密 码:");
panel2.add(JL2);
password.setColumns(10);
panel2.add(password);

JPanel panel3 = new JPanel();//放密码和输入框
panel3.setLayout(new FlowLayout());
JLabel JL3 = new JLabel("请 输 入 手 机 号:");
panel3.add(JL3);
phone.setColumns(10);
panel3.add(phone);

JPanel panel4 = new JPanel();//放注册和返回
panel4.setLayout(new FlowLayout());
panel4.add(buttona);
panel4.add(buttonb);
buttona.addActionListener(this);// 添加动作响应器
buttonb.addActionListener(this);// 添加动作响应器
this.add(panel1);
this.add(panel2);
this.add(panel3);
this.add(panel4);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

private void clearText() {//清空文本框, 密码框的输入
    name.setText("");
    password.setText("");
    phone.setText("");
}

//动作相应

```

```

@Override
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    if(e.getSource() == buttona){
        List<User> list = UserOperate.queryUser(name.getText());
        //检验用户名是否重复
        if(phone.getText().length() != 11){
            JOptionPane.showMessageDialog(this, "请输入正确的手机号码", "通知",
JOptionPane.ERROR_MESSAGE);
        }
        else{
            if(list.isEmpty()){
                User user = new User();
                user.setName(name.getText());
                user.setPassword(password.getText());
                user.setPhone(phone.getText());
                if(UserOperate.add(user)){
                    System.out.println("注册成功");
                    JOptionPane.showMessageDialog(this, "注册成功！");
                    setVisible(false);
                    Login login = new Login();
                }
            }
            // message.main(0);
        }
        else{
            JOptionPane.showMessageDialog(this, "用户名已存在.", "通知",
JOptionPane.ERROR_MESSAGE);
            clearText();
        }
    }
}

if(e.getSource() == buttonb){

```

```

        setVisible(false);
        Login login = new Login();
    }
}
}

```

(3) MyGame.java

```

import java.awt.*
import java.util.*
import javax.swing.*;

public class MyGames extends JFrame {
    int num = Main.num; // NUM 阶矩阵
    public static int id ;
    private JPanel panel1 = new JPanel(); // 存放功能栏
    private JButton buttona = new JButton("开始");
    private JLabel label1 = new JLabel("分数");
    private JTextField textarea1 = new JTextField(10); // 文本框
    private JLabel buttonc = new JLabel("时间");
    private JProgressBar jindu = new JProgressBar();
    private Timer timer;
    private Timer downAnimal = new Timer(1000, new TimeListener1());
    private Timer updateAnimal = new Timer(1000, new TimeListener2());
    private JButton buttonb = new JButton("退出");
    private JPanel panel2 = new JPanel(); // 存放代码
    private JButton button[][] = new JButton[num][num];
    private int animal[][] = new int[num][num];
    private ImageIcon Iocn[] = new ImageIcon[9];
    // 标记是否有三个以上的连接
    private final int EMPTY = 0; // 无为 0, 有为 1
    // 随机数
    // 标记是否有三个以上的连接

```

```

private boolean isThreeLinked;
// 标记单击次数
private boolean isDoubleClicked = false;
private boolean isDoubleClickedTiger = false;
private int x1;// 记录第一次被点击按钮的 X 坐标
private int y1;// 记录第一次被点击按钮的 Y 坐标
public static int grade = 0; //得分 ,static 可以传值
private boolean isDownAnimal = false;
MyGames() {
    // 加载图片
    locn[1] = new ImageIcon("image//cat.png");
    locn[2] = new ImageIcon("image//cattle.png");
    locn[3] = new ImageIcon("image//chicken.png");
    locn[4] = new ImageIcon("image//fox.png");
    locn[5] = new ImageIcon("image//monkey.png");
    locn[6] = new ImageIcon("image//panda.png");
    locn[7] = new ImageIcon("image//frog.png");
    locn[8] = new ImageIcon("image//tiger.png");
    panel1.setLayout(new FlowLayout());
    panel1.add(buttona);
    panel1.add(label1);
    panel1.add(textarea1);
    textarea1.setEditable(false);
    textarea1.setText(Integer.toString(grade));//
    panel1.add(buttonc);
    jindu.setMaximum(100);
    panel1.add(jindu);
    panel1.add(buttonb);
    this.setLayout(new BorderLayout());
    this.add(panel1, BorderLayout.NORTH); //将功能栏放在最上面 NORTH
    panel2.setLayout(new GridLayout(num, num, 1, 1));
    MyListener mylisten = new MyListener();

```

```

int m;
// 初始化动物数组
for (int i = 0; i < num; i++)
    for (int j = 0; j < num; j++) {
        m = (int) (Math.random() * 7) + 1 ; //产生
        button[i][j] = new JButton(Icon[m]);
        animal[i][j] = m;
        button[i][j].setSize(50, 50);
        button[i][j].addActionListener(mylisten);
        button[i][j].setEnabled(false); //图形按钮无效
        panel2.add(button[i][j]);
    }
for(int i = 0;i < num;i++){
    for(int j = 0;j < num;j++){
        System.out.print(animal[i][j]+" ");
        System.out.println();
    }

    this.add(panel2, BorderLayout.CENTER); //将放在中间
    buttona.addActionListener(mylisten);
    buttonb.addActionListener(mylisten);
}
// 初始化动物数组
private void initAnimalMatrix() {
    for (int i = 0; i < num; i++) {
        for (int j = 0; j < num; j++) {
            // 随机选取动物
            animal[i][j] = (int) (Math.random() * 7) + 1 ;
        }
    }
}
//消除函数，消去连接

```

```

private void removeLinked(int x, int y) {
    if(animals[x][y] == EMPTY)return; //若空返回
    int n = 0;
    int tmp;
    int linked = 1;           //相同的动物数
    //选定的点不超过下界，如果选点下方的动物与选点相同则 linked++
    if (x + 1 < num) {
        tmp = x + 1;
        while (tmp < num && animals[x][y] == animals[tmp][y]) {
            linked++;
            tmp++;
        }
    }
    //选定的点不超过上界，如果选点上方的动物与选点相同则 linked++
    if (x - 1 >= 0) {
        tmp = x - 1;
        while (tmp >= 0 && animals[x][y] == animals[tmp][y]) {
            linked++;
            tmp--;
        }
    }
    //这是判断上下动物数是否有 3 个以上相同的，相同的消除也就是置为空
    if (linked >= 3) {
        System.out.println("可消除的点为:" + x + "," + y);
        n = n+linked;
        tmp = x + 1;
        //向 X,Y 的下面找
        while (tmp < num && animals[tmp][y] == animals[x][y]) {
            animals[tmp][y] = EMPTY;
            tmp++;
        }
        tmp = x - 1;
    }
}

```

```

//向 X,Y 的上面找
while (tmp >= 0 && animal[tmp][y] == animal[x][y]) {
    animal[tmp][y] = EMPTY;
    tmp--;
}
// 当前交换过来的点，自己也变为空
if(linked >= 4){
    animal[x][y] = 8; //变为老虎可以消除任意点击的
}
else{
    animal[x][y] = EMPTY;
}
print();
downAnimal.start();
}
tmp = 0;
linked = 1;
//选定的点不超过右界，如果选点右方的动物与选点相同则 linked++
if (y + 1 < num) {
    tmp = y + 1;
    while (tmp < num && animal[x][y] == animal[x][tmp]) {
        linked++;
        tmp++;
    }
}
//选定的点不超过左界，如果选点左方的动物与选点相同则 linked++
if (y - 1 >= 0) {
    tmp = y - 1;
    while (tmp >= 0 && animal[x][y] == animal[x][tmp]) {
        linked++;
        tmp--;
    }
}

```

```

    }
    //再次判断左右相连动物数
    if (linked >= 3) {
        System.out.println("可消除的点为:" + x + "," + y);
        n = n+linked;
        tmp = y + 1;
        while (tmp < num && animal[x][y] == animal[x][tmp]) {
            animal[x][tmp] = EMPTY;
            tmp++;
        }
        tmp = y - 1;
        while (tmp >= 0 && animal[x][y] == animal[x][tmp]) {
            animal[x][tmp] = EMPTY;
            tmp--;
        }
        // 当前交换过来的点
        if(linked == 4){
            animal[x][y] = 8; //变为老虎可以消除任意点击的
        }
        else{
            animal[x][y] = EMPTY;
        }
        print();
        downAnimal.start();
    }
    //加分
    grade += n*10;
    textarea1.setText(Integer.toString(grade));
}

```

```

private void removeAll(int x ,int y,int x1,int y1){

```



```

int all = animal[x][y];
int n = 1;
for(int i = 0;i < num;i++){
    for(int j = 0;j < num;j++){
        if(animal[i][j] == all){
            n++;
            animal[i][j] = EMPTY;
        }
    }
}
animal[x1][y1] = EMPTY;
print();
downAnimal.start();
grade += n*10;
textarea1.setText(Integer.toString(grade));
System.out.println(grade + " ====="+x+" "+y);
}
//全部搜索一遍，如果有相同的则消除
private boolean globalSearch(int flag) {
    if (1 == flag) {
        for (int i = 0; i < num; i++) {
            for (int j = 0; j < num; j++) {
                if (isThreeLinked(i, j)) {
                    return true;
                }
            }
        }
    }
}
else if (2 == flag) {
    for (int i = 0; i < num; i++) {
        for (int j = 0; j < num; j++) {
            // 删除三个相连的点

```

```

        removeLinked(i, j);
    }
}

return false;
}

// 是否有三个以上连接的点
private boolean isThreeLinked(int x, int y) {
    int tmp;
    int linked = 1;
    //向下
    if (x + 1 < num) {
        tmp = x + 1;
        while (tmp < num && animal[x][y] == animal[tmp][y]) {
            linked++;
            tmp++;
        }
    }
    //
    if (x - 1 >= 0) {
        tmp = x - 1;
        while (tmp >= 0 && animal[x][y] == animal[tmp][y]) {
            linked++;
            tmp--;
        }
    }
    if (linked >= 3) {
        return true;
    }
    linked = 1;
    if (y + 1 < num) {
        tmp = y + 1;

```

```

        while (tmp < num && animal[x][y] == animal[x][tmp]) {
            linked++;
            tmp++;
        }
    }
    if (y - 1 >= 0) {
        tmp = y - 1;
        while (tmp >= 0 && animal[x][y] == animal[x][tmp]) {
            linked++;
            tmp--;
        }
    }
    if (linked >= 3) {

        return true;
    }
    return false;
}

private boolean isDeadMatrix(){
    for(int i = 0; i < num ; i++){
        for(int j = 0; j < num; j++){
            int tmp;
            //向右交换
            if(j < num - 1){
                tmp = animal[i][j];
                animal[i][j] = animal[i][j+1];
                animal[i][j+1] = tmp;
                if(isThreeLinked(i,j) || isThreeLinked(i,j+1)){
                    return true;
                }
            }
            else{
                tmp = animal[i][j];

```

```

        animal[i][j] = animal[i][j+1];
        animal[i][j+1] = tmp;
    }
}
//向左交换
if(j > 0){
    tmp = animal[i][j];
    animal[i][j] = animal[i][j-1];
    animal[i][j-1] = tmp;
    if(isThreeLinked(i,j) || isThreeLinked(i,j-1)){
        return true;
    }
    else{
        tmp = animal[i][j];
        animal[i][j] = animal[i][j-1];
        animal[i][j-1] = tmp;
    }
}
//向下交换
if(i < num - 1){
    tmp = animal[i][j];
    animal[i][j] = animal[i+1][j];
    animal[i+1][j] = tmp;
    if(isThreeLinked(i,j) || isThreeLinked(i+1,j)){
        return true;
    }
    else{
        tmp = animal[i][j];
        animal[i][j] = animal[i+1][j];
        animal[i+1][j] = tmp;
    }
}
}

```

```

        if(i > 0){
            tmp = animal[i][j];
            animal[i][j] = animal[i-1][j];
            animal[i-1][j] = tmp;
            if(isThreeLinked(i,j) || isThreeLinked(i-1,j)){
                return true;
            }
            else{
                tmp = animal[i][j];
                animal[i][j] = animal[i-1][j];
                animal[i-1][j] = tmp;
            }
        }
    }
}

return false;
}

// 更新动物矩阵
private void updateAnimal() {
    for (int i = 0; i < num; i++) {
        for (int j = 0; j < num; j++) {
            if (animal[i][j] == EMPTY) {
                // 将矩阵中为空值的点再进行随机赋值 1 到 7
                animal[i][j] = (int) (Math.random() * 7) + 1 ;
                button[i][j].setEnabled(true);
            }
        }
    }
    System.out.println("更新完了");
}

```

// 动物下降，全局扫描，如果为空则将其上方的元素下降

```
private boolean downAnimal() {  
    int tmp;  
    int flag = 0;  
    //从下往上找  
    for (int j = num - 1; j >= 0; j--) {  
        for (int i = 0; i < num; i++) {  
            if (animal[j][i] == EMPTY) {  
                button[j][i].setEnabled(true);  
                for (int k = j - 1; k >= 0; k--) {  
                    if (animal[k][i] != EMPTY) {  
                        tmp = animal[k][i];  
                        animal[k][i] = animal[j][i];  
                        animal[j][i] = tmp;  
                        flag = 1;  
                        break;  
                    }  
                }  
            }  
        }  
    }  
    if(flag == 1){  
        System.out.println("下降完了");  
        return true;  
    }  
    else  
        return false;  
}  
//重新显示按钮图形  
private void print() {  
    for (int i = 0; i < num; i++) {  
        for (int j = 0; j < num; j++) {
```

```

        button[i][j].setIcon(Iocn[animal[i][j]]);
    }
}

int flag = 0; //老虎标志
//交换函数
private void swapAnimal(int x, int y) {
    if ((x >= 0 && x <= num) && (y >= 0 && y <= num)) {
        // 被点击的动物的坐标
        int x2;
        int y2;
        //如果被单击的是老虎，它可以和它上下左右任意一个交换然后爆炸
        if(animal[x][y] == 8 || flag == 1){
            if(!isDoubleClickedTiger){
                isDoubleClickedTiger = true;
                flag = 1;
                x1 = x;
                y1 = y;
                System.out.println("老虎第一次被单击的点的坐标 = " + x1 + "," + y1);
            }
            else{
                x2 = x;
                y2 = y;
                flag = 0;
                isDoubleClickedTiger = false;
                System.out.println("老虎第二次被单击的点的坐标 = " + x2 + "," + y2);
                //两点的坐标相减的绝对值等于 1 时视为相邻的两点
                if (1 == Math.abs(x2 - x1) && y2 == y1
                    || 1 == Math.abs(y2 - y1) && x2 == x1){
                    removeAll(x2,y2,x1,y1);
                    updateAnimal.start();
                }
            }
        }
    }
}

```

```

    }
}
else{
    if (!isDoubleClicked) { //第一次单击
        isDoubleClicked = true;
        x1 = x;
        y1 = y;
        System.out.println("第一次被单击的点的坐标 = " + x1 + "," + y1);

    } else{                //第 2 次单击
        x2 = x;
        y2 = y;
        isDoubleClicked = false;
        System.out.println("第二次被单击的点的坐标 = " + x2 + "," + y2);
        //两点的坐标相减的绝对值等于 1 时视为相邻的两点
        if (1 == Math.abs(x2 - x1) && y2 == y1
            || 1 == Math.abs(y2 - y1) && x2 == x1) {
            //-----交换矩阵中相邻的两点的值-----
            int tmp;
            tmp = animal[x2][y2];
            animal[x2][y2] = animal[x1][y1];
            animal[x1][y1] = tmp;
            System.out.println("已经交换");
            //-----
            if (isThreeLinked(x2, y2) || isThreeLinked(x1, y1)) {
                System.out.println("可以消除");
                if (isThreeLinked(x2, y2)) {
                    removeLinked(x2, y2);
                }
                if (isThreeLinked(x1, y1)) {
                    removeLinked(x1, y1);
                }
            }
        }
    }
}

```



```
}
```

```
//主函数
```

```
// 监听器,鼠标事件
```

```
class MyListener implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
        if (e.getSource() == buttona) {  
            buttona.setEnabled(false);  
            jindu.setStringPainted(true);  
            jindu.setMaximum(100);  
            jindu.setMinimum(0);  
            timer = new Timer(800, new TimeListener());  
            timer.start();  
            grade = 0;  
            textarea1.setText(Integer.toString(grade));  
            for (int i = 0; i < num; i++)  
                for (int j = 0; j < num; j++) {  
                    button[i][j].setEnabled(true); //图形按钮有效  
                }  
        }  
        if (e.getSource() == buttonb) {  
            System.out.println("end");  
            System.exit(1);  
        }  
        for (int i = 0; i < num; i++) {  
            for (int j = 0; j < num; j++) {  
                if (e.getSource() == button[i][j]) {  
                    System.out.println("第" + i + " " + j + "键");  
                    swapAnimal(i, j);  
                }  
            }  
        }  
    }  
}
```

```

        }
    }
}

class TimeListener1 implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        System.out.println("time1");
        downAnimal();
        updateAnimal();
        print();
        downAnimal.stop();
    }
}

class TimeListener2 implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        System.out.println("time2");
        if(isDeadMatrix()){
            System.out.println("不是死阵");
            while(globalSearch(1)){
                //全局扫描消除三个以上相连的点
                System.out.println("有全局三连");
                globalSearch(2);
                downAnimal();
                updateAnimal();
                //再次更新动物矩阵
                print();
            }
        }
        else{
            System.out.println("死阵");
        }
        updateAnimal.stop();
    }
}

```

```

}

class TimeListener implements ActionListener {
    int times = 0;

    public void actionPerformed(ActionEvent e) {
        jindu.setValue(times++);
        if (times > 100) {
            timer.stop();
            //定时器结束
            for (int i = 0; i < num; i++)
                for (int j = 0; j < num; j++) {
                    button[i][j].setEnabled(false); //图形按钮无效
                }
            buttona.setEnabled(true);
            Rank rank = new Rank();
            String name = Main.Name;
            Date time = new java.sql.Date(new java.util.Date().getTime());
            rank.setName(name);
            rank.setScore(grade);
            rank.setTime((java.sql.Date) time);
            if(RankOperate.add(rank)){
                System.out.println("插入成功！");
                //id 自增，最后插入的就是 id 最大的那个
                List<Rank> list = RankOperate.queryMax();
                Rank ra = list.get(0);
                System.out.println(ra.getId());
                id = ra.getId();
                //游戏结束调用面板！
                setVisible(false);
                Over.main(null);
            }
            else
                System.out.println("插入失败");
        }
    }
}

```

```

        }
    }
}
}

```

(4) RankList.java

```

import java.awt.*;
import java.text.SimpleDateFormat;
import java.util.List;
import javax.swing.*;

public class RankList extends JFrame{
    private static JTable table;
    private static DefaultTableModel model;
    JButton buttona = new JButton();
    JButton buttonb = new JButton();
    String name = Main.Name;
    int grade = MyGames.grade;
    static int Id = MyGames.id;
    static int Row = -1;
    SimpleDateFormat sdf = new SimpleDateFormat( "yyyy-MM-dd" );
    RankList(){
        Id = MyGames.id;
        System.out.println("当前 id 为"+ Id);
        this.setTitle("排行榜, "+name+" 您的分数为: "+grade);
        this.setSize(800, 400);
        String[] columns = { "排名", "姓名", "分数", "时间" };
        //数据对象,10 行, 4 列
        Object[][] data = new Object[10][4];
        // 把数据库查出来的数据放到表格里
        List<Rank> list = RankOperate.queryTen();
        for (int i = 0; i < 10; i++) {

```

```

        if(i < list.size()){
            Rank ra = list.get(i);
            System.out.println(ra.getId() + " "+ ra.getName() + " "+ ra.getScore());
            if(Id == ra.getId()){
                Row = i;
                System.out.println(Row+" 变红"+Id+" "+ra.getId());
            }
            data[i][0] = i+1;
            data[i][1] = ra.getName();
            data[i][2] = ra.getScore();
            data[i][3] = ra.getTime();
        }
        else{
            data[i][0] = i+1;
            data[i][1] = data[i][2] = data[i][3] = "-";
        }
    }
}

model = new DefaultTableModel(data, columns);
table = new JTable(model){
    public boolean isCellEditable(int row, int column) {
        return false;
    }
    Color color = getForeground();
    //本次游戏的排名，查看自己的位置！
    public Component prepareRenderer(TableCellRenderer renderer, int row, int column) {
        Component component = super.prepareRenderer(renderer, row, column);
        //当某一行与 id 匹配则单元格的背景 字体设为红色
        if (row == Row) {
            component.setForeground(Color.RED);
        }else{ //否则 则为之前的颜色
            component.setForeground(color);
        }
    }
}

```

```

        return component;
    }
};

//内容居中
SupportIconTableCellRenderer r = new SupportIconTableCellRenderer();
r.setHorizontalAlignment(JLabel.CENTER);
table.setDefaultRenderer(Object.class,r);
table.setRowHeight(30);
JScrollPane panel1 = new JScrollPane(table);
this.setLayout(new BorderLayout(1,1));
this.add(panel1, BorderLayout.CENTER);
JPanel panel2 = new JPanel();
panel2.setLayout(new FlowLayout());
buttona = new JButton("再玩一次");
buttonb = new JButton("退出游戏");
panel2.add(buttona);
panel2.add(buttonb);
this.add(panel2, BorderLayout.SOUTH);
MyListener mylisten = new MyListener();
buttona.addActionListener(mylisten);
buttonb.addActionListener(mylisten);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}

public static void main(String[] args) {
    RankList ranklist = new RankList();
    Dimension screen = ranklist.getToolkit().getScreenSize();
    ranklist.setLocation((screen.width - ranklist.getSize().width) / 2,
        (screen.height - ranklist.getSize().height) / 2);
}

class MyListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {

```

```
        if (e.getSource() == buttona) {  
            System.out.println("再来一次");  
            setVisible(false);  
            Main.main(name);  
        }  
        if(e.getSource() == buttonb){  
            System.out.println("结束游戏");  
            System.exit(1);  
        }  
    }  
}
```