

2048游戏的设计与实现

《面向对象程序设计》大作业说明书



2021-12-9

中北大学 2013040111 田梓萱

**基于Swing的2048小游戏设计与实现**

**摘要**

在科技飞速发展的今天，在越来越多的前沿技术为人们的生活提供便利的同时，也不能忘记以前的一些传统小游戏。2048小游戏的开发就是为使大家可以在课余休息时间重温经典小游戏。

本学期开设了面向对象程序设计课程，设计此游戏主要想熟练运用所学知识以解决生活中的问题，践行中北大学“致知于行”的校训。本项目在此背景下，设计和研究了基于Swing的2048小游戏，该系统使用IDEA集成环境进行开发，对2048小游戏问题的研究主要包括4个方面：分别是游戏原理分析、界面设计、功能设计与算法设计。

本文主要是针对2048小游戏中的界面创建、功能实现等部分进行了描述。主要工作有：

* 通过分析系统的需求，将系统分为用户信息数据和游戏数据实现两大模块。
* 开发核心模块，实现了相关功能。
* 开发相关接口，方便未来扩展新的功能。
* 对系统进行测试，达到了系统开发工作的预期目的。

总体上来讲，2048小游戏具有综合性，“以Java面向对象编程思想为灵魂、以Swing作为交互界面”，使我在开发时不仅要确保游戏结果的正确性、稳定性，同时需要兼顾良好的用户交互，本系统具有现实使用价值。

**关键词**：2048小游戏

目录

[1.绪论 2](#_Toc7434)

[2.需求分析 2](#_Toc31241)

[2.1 用户需求分析 2](#_Toc19115)

[3.概要设计 3](#_Toc21828)

[3.1 创建界面模块 4](#_Toc20629)

[3.2 信息存储模块 4](#_Toc7314)

[3.3 问题求解模块 5](#_Toc10044)

[3.4 非功能需求分析 5](#_Toc522)

[4.详细设计 5](#_Toc27953)

[4.1 GBackGroup.java 7](#_Toc27953)

[4.2 GStart.java 11](#_Toc27953)

[4.3 IData.java 14](#_Toc27953)

[4.4 IGView.java 15](#_Toc27953)

[4.5 GView4.java 17](#_Toc27953)

[4.6 Login.java 45](#_Toc27953)

[4.7 Main2048.java 53](#_Toc27953)

[4.8 Register.java 54](#_Toc27953)

[4.9 ILogin.java 67](#_Toc27953)

[4.10 UserInfo.java 68](#_Toc27953)

[4.11 User.java 72](#_Toc27953)

[5.2048小游戏程序使用说明 76](#_Toc13625)

[6.心得体会 82](#_Toc22577)

[7.参考文献 84](#_Toc10411)

# 1.绪论

随着移动互联网的兴起，为了让人们可以重温以前的经典小游戏，像习总书记说的那样：不忘初心，牢记使命。2048小游戏应运而生，让老师同学在休息之时可以体验含有现在科技的2048无广告版小游戏。

本游戏使用了《面向对象程序设计》和上学期学的《C语言基础》的知识，在开发的过程中，提高了动手能力，而且夯实了基础，巩固了所学知识。

# 2.需求分析

本游戏为了使用户可以游玩2048，系统模块设计高内聚、低耦合，游戏展示界面简洁直观大方，用户上手成本较低，且符合国家游戏相关政策。因为当前市面上的2048小游戏大多较为封闭且服务单一，本游戏设计的总目标如下：

1. 界面直观，方便用户操作，同时结果的展示易于理解。
2. 操作容易，尽最大可能降低用户使用软件带来的成本。
3. 算法高效，能够在较短时间内判断用户操作，并给出正确的显示。
4. 数据安全，能够检测并告知用户数据文件的相关问题，避免产生错误。
5. 结构灵活，可以方便后期项目的扩展。
6. 通用性、稳定性、安全性并有一定的容错性。
7. 能够防御不法分子的攻击。

## 2.1 用户需求分析

用户在使用软件时需要先注册用户的实名信息，之后就可以进入游戏游玩，同时可以查看排行榜、目前最高分。

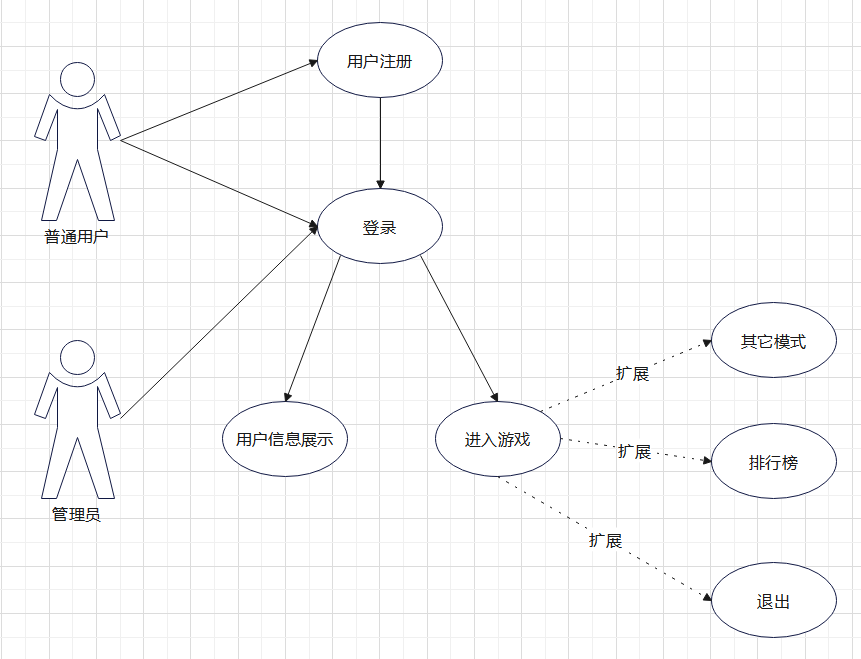


图2.1 用户用例图

|  |  |
| --- | --- |
| **用例名** | 2048小游戏 |
| **主要参与者** | 任意人员 |
| **描述** | 本用例主要描述2048小游戏的简单流程 |
| **前置条件** | 有一个默认管理员 |
| **后置条件** | 存储生成的各种数据 |
| **触发条件** | 登录、注册 |
| **主事件流** | 1. 用户使用自己真实用户信息注册 2. 管理员登录后可查看用户信息 3. 将产生的数据存盘 |
| **扩展流** | 1a. 提供的文件路径无效或者无读写权限  3a. 磁盘控件不足或者目标目录没有写入权限  4a. 数据文件损坏，用户数据过于庞大 |

# 表2.1 用例图说明

# 3.概要设计

根据游戏的各个功能模块的不同需求进行划分。

## 3.1 创建界面模块

此模块用于初始化游戏界面信息，因为读写创建文件是耗时工作，所以对人机交互有以下要求：

1. 操作流畅自然，符合人的一般认知与习惯。
2. 符合中国法律法规
3. 游戏稳定，不会因为崩溃而丢失数据。
4. 提供丰富的操作功能。

## 3.2 信息存储模块

此模块能够将用户创建的所有相关数据，包括游戏数字位置和描述等信息妥善保存，增加系统的安全性。

因为数据能够顺利读入、写出直接影响了用户体验，于是需要游戏有以下特性：

1. 保存数据快速，数据中冗余信息较少。
2. 在必要时使用压缩技术保证传输效率。
3. 有数据校验功能，能够对错误提前预知。
4. 兼容性较好，能够识别并转换老版本创建的数据文件。

**3.2.1用户基础信息存储模块**

此模块能够将用户注册的相关数据，包括用户昵称、姓名、性别、密码等信息妥善保存，增加系统的安全性。

因为数据能够顺利读入、写出直接影响了用户体验，于是需要游戏有以下特性：

1.保存数据快速，数据中冗余信息较少。

2.在必要时使用压缩技术保证传输效率。

3.兼容性较好，能够识别并转换老版本创建的数据文件。

**3.2.2用户游戏数据存储模块**

此模块能够将用户创建的所有相关数据，包括游戏数字位置和最好成绩等信息妥善保存，增加系统的安全性。

因为数据能够顺利读入、写出直接影响了用户体验，于是需要游戏有以下特性：

1.保存数据快速，数据中冗余信息较少。

2.在必要时使用压缩技术保证传输效率。

3.有数据校验功能，能够对错误提前预知。

4.兼容性较好，能够识别并转换老版本创建的数据文件。

## 3.3 问题求解模块

用户在此游戏中可以使用预置4\*4游戏游玩。游戏内置以下功能：

1. 存储用户每一次操作后的步骤数据。
2. 保证用户是经过实名认证的人
3. 用户的最好成绩会自动记录

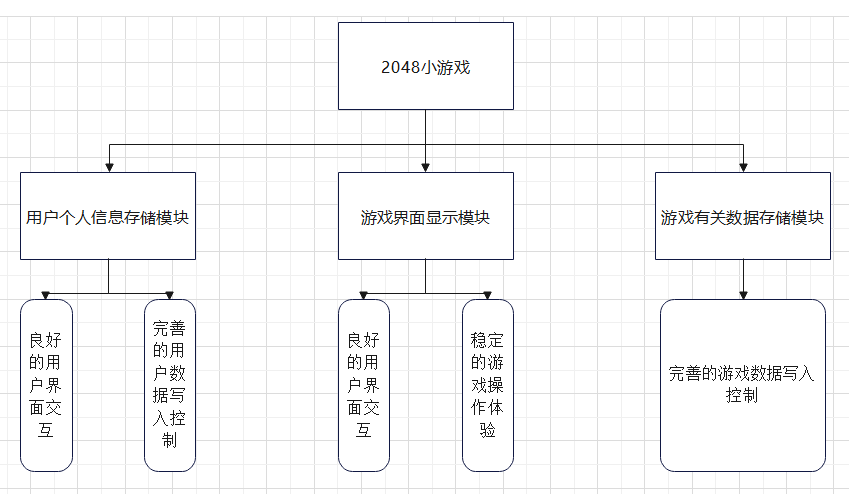


图3.1 系统功能模块图

## 3.4 非功能需求分析

游戏的非功能性需求主要是指游戏除满足基本的功能性需求之外，还要保证游戏的可靠性、可支持性等需求。游戏的非功能性需求保证了游戏的稳定运行和用户玩游戏的良好体验。

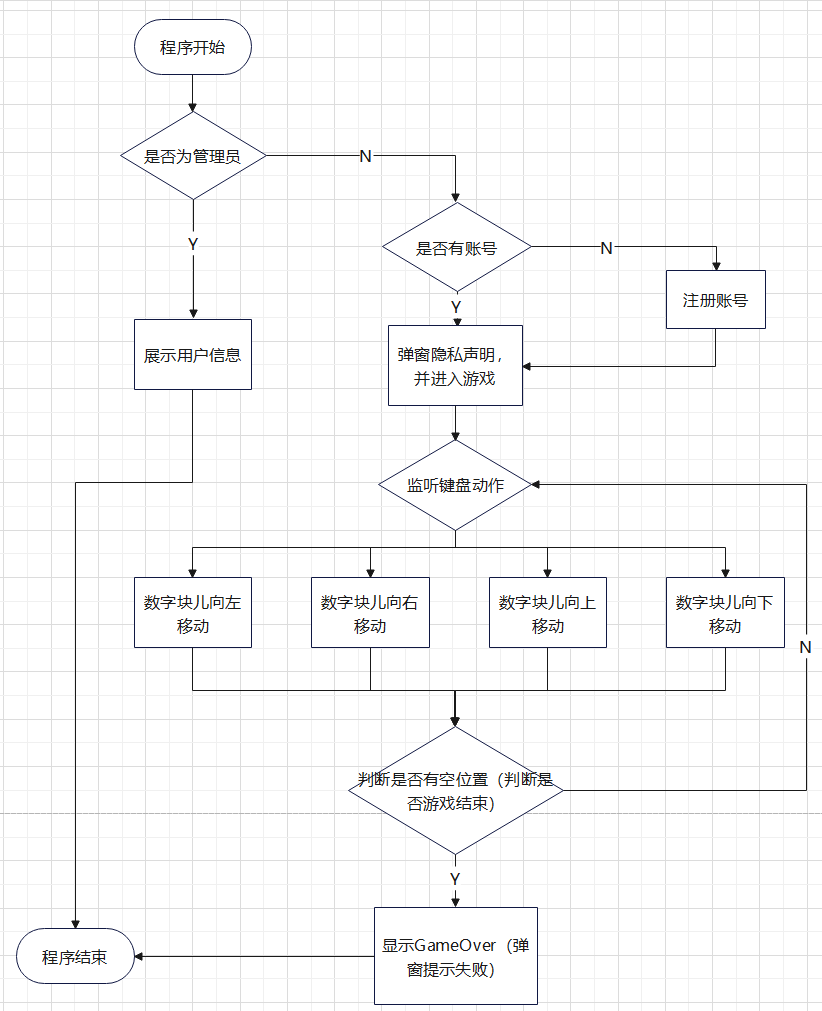


图3.2 系统总体流程图

# 4.详细设计

本游戏通过如下类图，通过使用接口定义功能、类实现接口、使用io文件流存储数据、List的泛型存取信息综合使用本学期Java所学所有知识编写完成，并且大量使用set、get方法进行数据封装，提升程序安全性；多次使用JOptionPane弹窗，提高程序人性化水平。

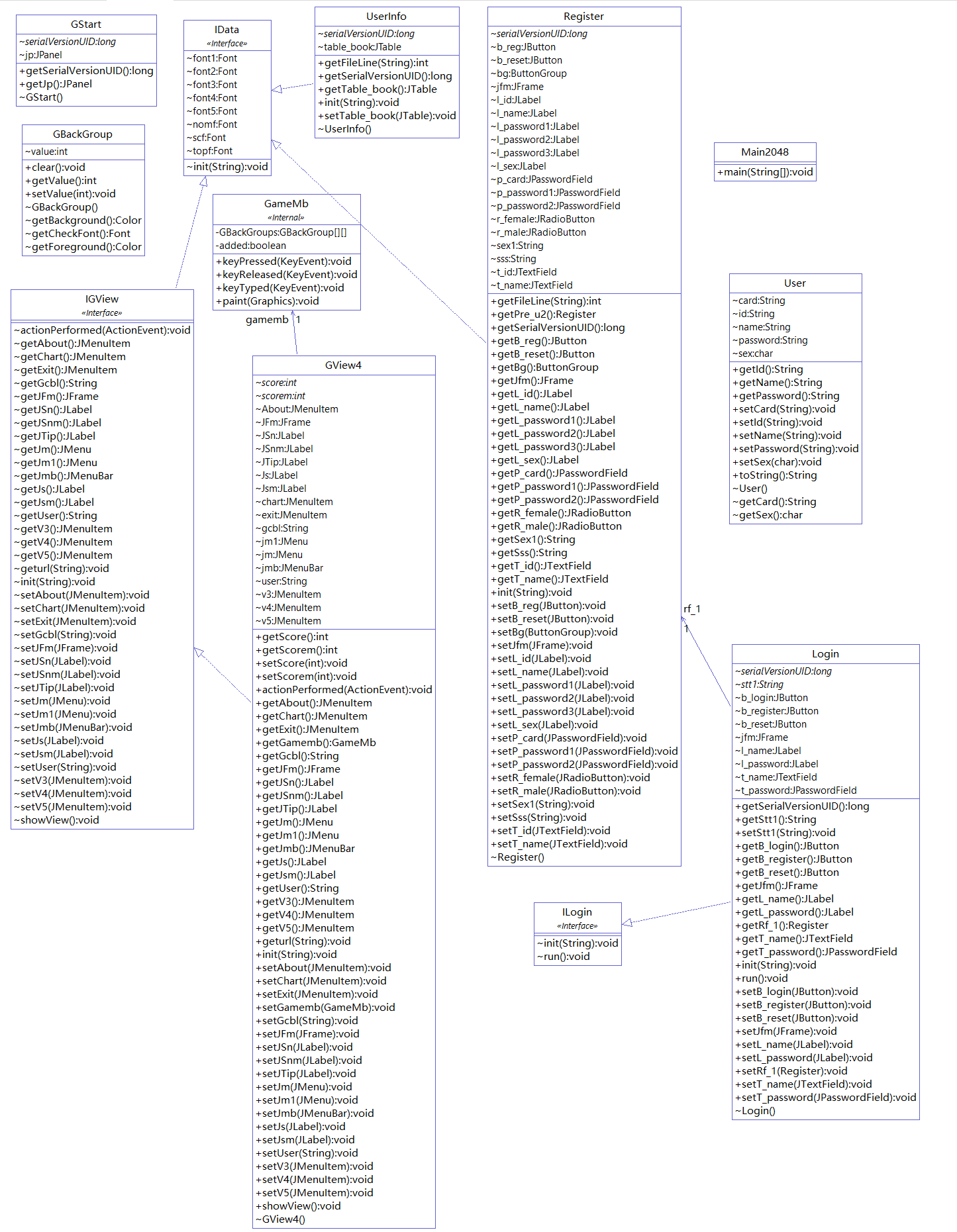


图4.1 系统UML类图

总体上分为数据存储区db和逻辑操作区v2，代码包内分包结构如下：

── g2048.v2  
   ├── db  
    │   ├── 2048gd.txt  
    │   ├── 2048maxcj.txt  
    │   └── user.txt  
    └── v2  
       ├── GBackGroup.java  
       ├── GStart.java  
       ├── IData.java  
       ├── IGView.java  
        ├── GView4.java  
       ├── Login.java  
       ├── Main2048.java  
        ├── Register.java  
       ├── ILogin.java  
       ├── UserInfo.java  
       └── User.java

4.1 GBackGroup.java

本文件为游戏界面数字部分的实现，代码及详细注释如下：  
package g2048.v2;

/\*\*

\* @author Xcl

\* @date 2021/12/1 09:50

\* @package g2048.v2

\*/

import java.awt.\*;

public class GBackGroup {

int value;

/\*\*

\* 调用清空函数

\*/

GBackGroup() {

clear();

}

/\*\*

\* 清空所有位置数组对应值为0

\*/

public void clear() {

setValue(0);

}

/\*\*

\* 设置游戏界面背景

\*/

Color getForeground() {

switch (getValue()) {

case 0:

return new Color(0xcdc1b4);

case 2:

return Color.BLACK;

case 4:

return Color.BLACK;

default:

return Color.WHITE;

}

}

/\*\*

\* 设置游戏中数字2、0、4、8、16、32、64、128、256、512、1024、2048、4096、8192、16384颜色，

\* 已经为日后5\*5模式预留

\*/

Color getBackground() {

switch (getValue()) {

case 0:

return new Color(0xcdc1b4);

case 2:

return new Color(0xeee4da);

case 4:

return new Color(0xede0c8);

case 8:

return new Color(0xf2b179);

case 16:

return new Color(0xf59563);

case 32:

return new Color(0xf67c5f);

case 64:

return new Color(0xf65e3b);

case 128:

return new Color(0xedcf72);

case 256:

return new Color(0xedcc61);

case 512:

return new Color(0xedc850);

case 1024:

return new Color(0xedc53f);

case 2048:

return new Color(0xedc22e);

case 4096:

return new Color(0x65da92);

case 8192:

return new Color(0x5abc65);

case 16384:

return new Color(0x248c51);

default:

return new Color(0x09572A);

}

}

/\*\*

\* 设置游戏界面中数字字体

\*/

Font getCheckFont() {

if (getValue() < 10) return IData.font1;

if (getValue() < 100) return IData.font2;

if (getValue() < 1000) return IData.font3;

if (getValue() < 10000) return IData.font4;

return IData.font5;

}

/\*\*

\* 获取value值的set、get方法

\*/

public int getValue() {

return value;

}

public void setValue(int value) {

this.value = value;

}

}

4.2 GStart.java

本文件实现了游戏的启动、初始化登录界面，代码及详细注释如下：

package g2048.v2;

/\*\*

\* @author Xcl

\* @date 2021/12/2 07:58

\* @package g2048.v2

\*/

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

/\*\*

\* 本类继承自JFrame实现swing的方法

\*/

class GStart extends JFrame {

static final long serialVersionUID = -6718310314384392563L;

final JPanel jp = new JPanel(new GridLayout(3, 1, 5, 5));

/\*\*

\* 游戏开始界面通过线程分别执行管理员和普通用户登录

\*

\* @param st

\*/

GStart(String st) {

super(st);

setLayout(new BorderLayout());

JButton[] btn;

btn = new JButton[3];

btn[0] = new JButton("开始游戏");

btn[1] = new JButton("管理员登陆");

btn[2] = new JButton("退出");

getJp().add(btn[0]);

getJp().add(btn[1]);

getJp().add(btn[2]);

getContentPane().add(getJp(), BorderLayout.CENTER);

setSize(450, 450);

setLocationRelativeTo(null);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setResizable(false);

setVisible(true);

Login.setStt1(null);

Login lgf = new Login();

/\*\*

\* 按钮动作事件监听

\*/

btn[0].addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

JOptionPane.showMessageDialog(null, "根据《中华人民共和国防沉迷游戏法规定》，您需要登录/注册" +

"后方可开始游戏。", "提示", JOptionPane.WARNING\_MESSAGE);

Login.setStt1("用户");

Thread t1 = new Thread(lgf);

t1.start();

}

});

btn[1].addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

Login.setStt1("管理员");

Thread t2 = new Thread(lgf);

t2.start();

}

});

btn[2].addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

}

public static long getSerialVersionUID() {

return serialVersionUID;

}

public JPanel getJp() {

return jp;

}

}

4.3 IData.java

本文件为游戏界面接口规划，代码及详细注释如下：

package g2048.v2;

/\*\*

\* @author Xcl

\* @date 2021/12/1 09:48

\* @package g2048.v2

\*/

import java.awt.\*;

/\*\*

\* IDate接口定义

\*/

public interface IData {

Font topf = new Font("", Font.BOLD, 50);

Font scf = new Font("", Font.BOLD, 22);

Font nomf = new Font("", Font.PLAIN, 20);

Font font1 = new Font("", Font.BOLD, 46);

Font font2 = new Font("", Font.BOLD, 40);

Font font3 = new Font("", Font.BOLD, 34);

Font font4 = new Font("", Font.BOLD, 28);

Font font5 = new Font("", Font.BOLD, 22);

void init(String st);

}

4.4 IGView.java

本文件为游戏逻辑层接口规划，代码及详细注释如下：

package g2048.v2;

/\*\*

\* @author Xcl

\* @date 2021/12/4 11:37

\* @package g2048.v3\_mod

\*/

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

/\*\*

\* 自己定义的IGview接口继承自IData, ActionListener，方便后续程序扩展

\*/

public interface IGView extends IData, ActionListener {

void showView();

@Override

void init(String st);

String getUser();

void setUser(String user);

JFrame getJFm();

void setJFm(JFrame JFm);

JLabel getJSnm();

void setJSnm(JLabel JSnm);

JLabel getJSn();

void setJSn(JLabel JSn);

JLabel getJsm();

void setJsm(JLabel jsm);

JLabel getJs();

void setJs(JLabel js);

String getGcbl();

void setGcbl(String gcbl);

JLabel getJTip();

void setJTip(JLabel JTip);

@Override

void actionPerformed(ActionEvent e);

JMenu getJm();

void setJm(JMenu jm);

JMenu getJm1();

void setJm1(JMenu jm1);

JMenuBar getJmb();

void setJmb(JMenuBar jmb);

JMenuItem getV3();

void setV3(JMenuItem v3);

JMenuItem getV4();

void setV4(JMenuItem v4);

JMenuItem getV5();

void setV5(JMenuItem v5);

JMenuItem getAbout();

void setAbout(JMenuItem About);

JMenuItem getExit();

void setExit(JMenuItem exit);

JMenuItem getChart();

void setChart(JMenuItem chart);

void geturl(String url);

}

4.5 GView4.java

本文件为游戏4\*4模式的逻辑层核心代码，代码及详细注释如下：

package g2048.v2;

/\*\*

\* @author Xcl

\* @date 2021/12/1 09:48

\* @package g2048.v2

\*/

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.io.\*;

import java.net.URI;

import java.nio.charset.StandardCharsets;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import java.util.Random;

/\*\*

\* 本类继承自IGView实现接口的所有方法

\*/

public class GView4 implements IGView {

static int score = 0;

static int scorem = 0;

String user = null;

JFrame JFm;

JLabel JSnm;

JLabel JSn;

JLabel Jsm;

JLabel Js;

GameMb gamemb;

String gcbl = null;

JLabel JTip;

JMenu jm;

JMenu jm1;

JMenuBar jmb;

JMenuItem v3;

JMenuItem v4;

JMenuItem v5;

JMenuItem About;

JMenuItem chart;

JMenuItem exit;

/\*\*

\* 本类主显示

\*

\* @param st

\*/

GView4(String st) {

init(st);

setUser(st);

}

/\*\*

\* 大量get、set方法进行封装数据

\*

\* @return

\*/

public static int getScore() {

return score;

}

public static void setScore(int score) {

GView4.score = score;

}

public static int getScorem() {

return scorem;

}

public static void setScorem(int scorem) {

GView4.scorem = scorem;

}

@Override

public void showView() {

getJFm().setVisible(true);

}

/\*\*

\* 游戏进入玩耍的核心逻辑代码

\*

\* @param st

\*/

@Override

public void init(String st) {

JOptionPane.showMessageDialog(null, "您的所有游戏数据将被保存在本地，" +

"继续游戏默认您同意我的隐私协议。", "隐私申明", JOptionPane.INFORMATION\_MESSAGE);

setJFm(new JFrame(st + "的2048小游戏"));

getJFm().setSize(450, 600);

getJFm().setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

getJFm().setLocationRelativeTo(null);

getJFm().setResizable(false);

getJFm().setLayout(null);

setJmb(new JMenuBar());

/\*\*

\* 预留出V3.0版本的所有功能区域

\*/

setJm(new JMenu("功能"));

setV3(new JMenuItem("3\*3"));

getV3().addActionListener(this);

setV4(new JMenuItem("4\*4"));

getV4().addActionListener(this);

setV5(new JMenuItem("5\*5"));

getV5().addActionListener(this);

getJm().add(getV3());

getJm().add(getV4());

getJm().add(getV5());

setJm1(new JMenu("其它"));

setAbout(new JMenuItem("关于"));

getAbout().addActionListener(this);

setChart(new JMenuItem("排行榜"));

getChart().addActionListener(this);

setExit(new JMenuItem("退出"));

getExit().addActionListener(this);

getJm1().add(getAbout());

getJm1().add(getExit());

getJmb().add(getJm());

getJmb().add(getJm1());

getJFm().setJMenuBar(getJmb());

setJSnm(new JLabel("目前最高分", JLabel.CENTER));

getJSnm().setFont(IData.scf);

getJSnm().setForeground(Color.WHITE);

getJSnm().setOpaque(true);

getJSnm().setBackground(Color.GRAY);

getJSnm().setBounds(10, 1, 185, 29);

getJFm().add(getJSnm());

setJsm(new JLabel("0", JLabel.CENTER));

getJsm().setFont(IData.scf);

getJsm().setForeground(Color.WHITE);

getJsm().setOpaque(true);

getJsm().setBackground(Color.GRAY);

getJsm().setBounds(10, 30, 185, 30);

getJFm().add(getJsm());

setJSn(new JLabel(st + "的当前得分", JLabel.CENTER));

getJSn().setFont(IData.scf);

getJSn().setForeground(Color.WHITE);

getJSn().setOpaque(true);

getJSn().setBackground(Color.GRAY);

getJSn().setBounds(225, 1, 195, 29);

getJFm().add(getJSn());

setJs(new JLabel("0", JLabel.CENTER));

getJs().setFont(IData.scf);

getJs().setForeground(Color.WHITE);

getJs().setOpaque(true);

getJs().setBackground(Color.GRAY);

getJs().setBounds(225, 30, 195, 30);

getJFm().add(getJs());

setJTip(new JLabel("操作：↑↓←→/WSAD, 按Esc/R重新开始", JLabel.CENTER));

getJTip().setFont(IData.nomf);

getJTip().setForeground(Color.DARK\_GRAY);

getJTip().setBounds(26, 60, 400, 40);

getJFm().add(getJTip());

setGamemb(new GameMb());

getGamemb().setBounds(0, 100, 450, 450);

getGamemb().setBackground(Color.GRAY);

getGamemb().setFocusable(true);

getGamemb().setLayout(new FlowLayout());

getJFm().add(getGamemb());

}

/\*\*

\* 大量get、set方法进行封装数据

\*

\* @return

\*/

@Override

public String getUser() {

return user;

}

@Override

public void setUser(String user) {

this.user = user;

}

@Override

public JFrame getJFm() {

return JFm;

}

@Override

public void setJFm(JFrame JFm) {

this.JFm = JFm;

}

@Override

public JLabel getJSnm() {

return JSnm;

}

@Override

public void setJSnm(JLabel JSnm) {

this.JSnm = JSnm;

}

@Override

public JLabel getJSn() {

return JSn;

}

@Override

public void setJSn(JLabel JSn) {

this.JSn = JSn;

}

@Override

public JLabel getJsm() {

return Jsm;

}

@Override

public void setJsm(JLabel jsm) {

Jsm = jsm;

}

@Override

public JLabel getJs() {

return Js;

}

@Override

public void setJs(JLabel js) {

Js = js;

}

public GameMb getGamemb() {

return gamemb;

}

public void setGamemb(GameMb gamemb) {

this.gamemb = gamemb;

}

@Override

public String getGcbl() {

return gcbl;

}

@Override

public void setGcbl(String gcbl) {

this.gcbl = gcbl;

}

@Override

public JLabel getJTip() {

return JTip;

}

@Override

public void setJTip(JLabel JTip) {

this.JTip = JTip;

}

@Override

public JMenu getJm() {

return jm;

}

@Override

public void setJm(JMenu jm) {

this.jm = jm;

}

@Override

public JMenu getJm1() {

return jm1;

}

@Override

public void setJm1(JMenu jm1) {

this.jm1 = jm1;

}

@Override

public JMenuBar getJmb() {

return jmb;

}

@Override

public void setJmb(JMenuBar jmb) {

this.jmb = jmb;

}

@Override

public JMenuItem getV3() {

return v3;

}

@Override

public void setV3(JMenuItem v3) {

this.v3 = v3;

}

@Override

public JMenuItem getV4() {

return v4;

}

@Override

public void setV4(JMenuItem v4) {

this.v4 = v4;

}

@Override

public JMenuItem getV5() {

return v5;

}

@Override

public void setV5(JMenuItem v5) {

this.v5 = v5;

}

@Override

public JMenuItem getAbout() {

return About;

}

@Override

public void setAbout(JMenuItem about) {

About = about;

}

@Override

public JMenuItem getChart() {

return chart;

}

@Override

public void setChart(JMenuItem chart) {

this.chart = chart;

}

@Override

public JMenuItem getExit() {

return exit;

}

@Override

public void setExit(JMenuItem exit) {

this.exit = exit;

}

/\*\*

\* 监听选框数据

\*

\* @param e

\*/

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == getV3()) {

JOptionPane.showMessageDialog(null, "下一版本会加入此功能",

"ERROR!", JOptionPane.ERROR\_MESSAGE);

} else if (e.getSource() == getV5()) {

JOptionPane.showMessageDialog(null, "下一版本会加入此功能",

"ERROR!", JOptionPane.ERROR\_MESSAGE);

} else if (e.getSource() == getAbout()) {

JOptionPane.showMessageDialog(null, "这是一个简单的2048小游戏\n版本号: V2.0\n通过'W，A，S，" +

"D'或'↑，↓，←，→'可以轻松控制\n如发现任何Bug,欢迎联系我\n我的联系方式:xcl@xuegao-tzx.top\n我的主页: xuegao-" +

"tzx.top\n我的Github: github.com/xuegao-tzx", "关于", -1);

geturl("https://www.xuegao-tzx.top/g2048h.html");

} else if (e.getSource() == getChart()) {

JOptionPane.showMessageDialog(null, "下一版本会加入此功能",

"ERROR!", JOptionPane.ERROR\_MESSAGE);

} else if (e.getSource() == getExit()) {

getJFm().dispose();

System.exit(0);

}

}

/\*\*

\* 通过给定url，在浏览器中展示帮助界面

\*

\* @param url

\*/

@Override

public void geturl(String url) {

try {

URI uri = URI.create(url);

if (Desktop.getDesktop().isSupported(Desktop.Action.BROWSE)) {

Desktop.getDesktop().browse(uri);

}

} catch (NullPointerException e1) {

e1.printStackTrace();

} catch (IOException e2) {

e2.printStackTrace();

}

}

/\*\*

\* 内部类实现2048运行核心功能

\*/

class GameMb extends JPanel implements KeyListener {

private static final long serialVersionUID = -7237908387425382979L;

private final GBackGroup[][] GBackGroups = new GBackGroup[4][4];

private boolean added = true;

GameMb() {

initGame();

addKeyListener(this);

}

/\*\*

\* 监听键盘输入数据

\*

\* @param e

\*/

@Override

public void keyPressed(KeyEvent e) {

switch (e.getKeyCode()) {

case KeyEvent.VK\_ESCAPE:

initGame();

break;

case KeyEvent.VK\_R:

initGame();

break;

case KeyEvent.VK\_LEFT:

moveLeft();

createCheck();

judgeGameOver();

break;

case KeyEvent.VK\_A:

moveLeft();

createCheck();

judgeGameOver();

break;

case KeyEvent.VK\_RIGHT:

moveRight();

createCheck();

judgeGameOver();

break;

case KeyEvent.VK\_D:

moveRight();

createCheck();

judgeGameOver();

break;

case KeyEvent.VK\_UP:

moveUp();

createCheck();

judgeGameOver();

break;

case KeyEvent.VK\_W:

moveUp();

createCheck();

judgeGameOver();

break;

case KeyEvent.VK\_DOWN:

moveDown();

createCheck();

judgeGameOver();

break;

case KeyEvent.VK\_S:

moveDown();

createCheck();

judgeGameOver();

break;

default:

break;

}

repaint();

}

/\*\*

\* 游戏初始化，并读取本地文件最高分信息

\*/

private void initGame() {

GView4.setScore(0);

getJsm().setText(GView4.getScorem() + "");

try {

File file = new File("src/g2048/db/2048maxcj.txt");

if (file.isFile() && file.exists()) {

InputStreamReader isr = new InputStreamReader(new FileInputStream(file), StandardCharsets.UTF\_8);

BufferedReader br = new BufferedReader(isr);

String data;

data = br.readLine();

GView4.setScorem(Integer.parseInt(data));

br.close();

} else System.out.println("文件不存在！");

} catch (Exception e\_1) {

e\_1.printStackTrace();

}

for (int indexRow = 0; indexRow < 4; indexRow++)

for (int indexCol = 0; indexCol < 4; indexCol++) GBackGroups[indexRow][indexCol] = new GBackGroup();

added = true;

createCheck();

added = true;

createCheck();

}

/\*\*

\* 随机产生新的数字

\*/

private void createCheck() {

List<GBackGroup> list = getEmptyChecks();

if (!list.isEmpty() && added) {

Random random = new Random();

int index = random.nextInt(list.size());

GBackGroup GBackGroup = list.get(index);

// 2, 4，8出现概率6:4:1

int sjj = random.nextInt(13);

if (0 <= sjj && sjj <= 8) GBackGroup.setValue(2);

else if (sjj == 9 || sjj == 10 || sjj == 11 || sjj == 12) GBackGroup.setValue(4);

else if (sjj == 13) GBackGroup.setValue(8);

else System.out.println("不可能执行到这里...");

added = false;

}

}

/\*\*

\* 判断是否有空位

\*

\* @return

\*/

private List<GBackGroup> getEmptyChecks() {

List<GBackGroup> GBackGroupList = new ArrayList<>();

for (int i = 0; i < 4; i++)

for (int j = 0; j < 4; j++)

if (GBackGroups[i][j].getValue() == 0) GBackGroupList.add(GBackGroups[i][j]);

return GBackGroupList;

}

/\*\*

\* 判断是否游戏结束

\*

\* @return

\*/

private boolean judgeGameOver() {

if (GView4.getScorem() < GView4.getScore()) GView4.setScorem(GView4.getScore());

String datime = new SimpleDateFormat("yyyy-MM-dd|HH-mm-ss").format(new Date());

setGcbl(getUser() + "." + GBackGroups[0][0].getValue() + "." + GBackGroups[0][1].getValue() + "." + GBackGroups[0][2].getValue() + "." +

GBackGroups[0][3].getValue() + "." + GBackGroups[1][0].getValue() + "." + GBackGroups[1][1].getValue() + "." + GBackGroups[1][2].getValue() +

"." + GBackGroups[1][3].getValue() + "." + GBackGroups[2][0].getValue() + "." + GBackGroups[2][1].getValue() + "." + GBackGroups[2][2].getValue() +

"." + GBackGroups[2][3].getValue() + "." + GBackGroups[3][0].getValue() + "." + GBackGroups[3][1].getValue() + "." + GBackGroups[3][2].getValue() +

"." + GBackGroups[3][3].getValue() + "." + GView4.getScore() + "." + datime + ":");

try {

File f = new File("src/g2048/db/2048maxcj.txt");

File f1 = new File("src/g2048/db/2048gd.txt");

if (!f.exists()) f.createNewFile();

if (!f1.exists()) f1.createNewFile();

FileWriter fw1 = new FileWriter(f1);

FileWriter fw = new FileWriter(f);

BufferedWriter bw = new BufferedWriter(fw);

BufferedWriter bw1 = new BufferedWriter(fw1);

bw.write(String.valueOf(GView4.getScorem()));

bw1.write(String.valueOf(getGcbl()));

bw.flush();

bw1.flush();

bw.close();

bw1.close();

fw.close();

fw1.close();

} catch (IOException e2) {

e2.printStackTrace();

}

getJs().setText(GView4.getScore() + "");

getJsm().setText(GView4.getScorem() + "");

if (!getEmptyChecks().isEmpty()) return false;

for (int i = 0; i < 3; i++)

for (int j = 0; j < 3; j++)

if (GBackGroups[i][j].getValue() == GBackGroups[i][j + 1].getValue()

|| GBackGroups[i][j].getValue() == GBackGroups[i + 1][j].getValue()) return false;

return true;

}

/\*\*

\* 2048游戏界面向右移动后数据变化

\*/

private void moveRight() {

boolean key = true;

boolean flag = true;

boolean mark = true;

while (key) {

while (mark) {

mark = false;

for (int i = 0; i < 4; i++)

for (int j = 3; j > 0; j--)

if ((GBackGroups[i][j].getValue() == 0) && (GBackGroups[i][j - 1].getValue() != 0)) {

mark = true;

int temp = GBackGroups[i][j].getValue();

GBackGroups[i][j].setValue(GBackGroups[i][j - 1].getValue());

GBackGroups[i][j - 1].setValue(temp);

}

}

key = false;

while (flag) {

flag = false;

for (int i = 0; i < 4; i++)

for (int j = 3; j > 0; j--)

if ((GBackGroups[i][j].getValue() == GBackGroups[i][j - 1].getValue()) && GBackGroups[i][j].getValue() != 0) {

key = true;

mark = true;

GBackGroups[i][j].setValue(GBackGroups[i][j].getValue() \* 2);

GBackGroups[i][j - 1].setValue(0);

GView4.setScore(GView4.getScore() + GBackGroups[i][j].getValue());

}

}

}

added = true;

}

/\*\*

\* 2048游戏界面向左移动后数据变化

\*/

private void moveLeft() {

boolean key = true;

boolean flag = true;

boolean mark = true;

while (key) {

while (mark) {

mark = false;

for (int i = 0; i < 4; i++)

for (int j = 0; j < 3; j++)

if ((GBackGroups[i][j].getValue() == 0) && (GBackGroups[i][j + 1].getValue() != 0)) {

mark = true;

int temp = GBackGroups[i][j].getValue();

GBackGroups[i][j].setValue(GBackGroups[i][j + 1].getValue());

GBackGroups[i][j + 1].setValue(temp);

}

}

key = false;

while (flag) {

flag = false;

for (int i = 0; i < 4; i++)

for (int j = 0; j < 3; j++)

if ((GBackGroups[i][j].getValue() == GBackGroups[i][j + 1].getValue()) && GBackGroups[i][j].getValue() != 0) {

key = true;

mark = true;

GBackGroups[i][j].setValue(GBackGroups[i][j].getValue() \* 2);

GBackGroups[i][j + 1].setValue(0);

GView4.setScore(GView4.getScore() + GBackGroups[i][j].getValue());

}

}

}

added = true;

}

/\*\*

\* 2048游戏界面向上移动后数据变化

\*/

private void moveUp() {

boolean key = true;

boolean flag = true;

boolean mark = true;

while (key) {

while (mark) {

mark = false;

for (int i = 0; i < 3; i++)

for (int j = 0; j < 4; j++)

if ((GBackGroups[i][j].getValue() == 0) && (GBackGroups[i + 1][j].getValue() != 0)) {

mark = true;

int temp = GBackGroups[i][j].getValue();

GBackGroups[i][j].setValue(GBackGroups[i + 1][j].getValue());

GBackGroups[i + 1][j].setValue(temp);

}

}

key = false;

while (flag) {

flag = false;

for (int i = 0; i < 3; i++)

for (int j = 0; j < 4; j++)

if ((GBackGroups[i][j].getValue() == GBackGroups[i + 1][j].getValue()) && GBackGroups[i][j].getValue() != 0) {

key = true;

mark = true;

GBackGroups[i][j].setValue(GBackGroups[i][j].getValue() \* 2);

GBackGroups[i + 1][j].setValue(0);

GView4.setScore(GView4.getScore() + GBackGroups[i][j].getValue());

}

}

}

added = true;

}

/\*\*

\* 2048游戏界面向下移动后数据变化

\*/

private void moveDown() {

boolean key = true;

boolean flag = true;

boolean mark = true;

while (key) {

while (mark) {

mark = false;

for (int i = 3; i > 0; i--)

for (int j = 0; j < 4; j++)

if ((GBackGroups[i][j].getValue() == 0) && (GBackGroups[i - 1][j].getValue() != 0)) {

mark = true;

int temp = GBackGroups[i][j].getValue();

GBackGroups[i][j].setValue(GBackGroups[i - 1][j].getValue());

GBackGroups[i - 1][j].setValue(temp);

}

}

key = false;

while (flag) {

flag = false;

for (int i = 3; i > 0; i--)

for (int j = 0; j < 4; j++)

if ((GBackGroups[i][j].getValue() == GBackGroups[i - 1][j].getValue()) && GBackGroups[i][j].getValue() != 0) {

key = true;

mark = true;

GBackGroups[i][j].setValue(GBackGroups[i][j].getValue() \* 2);

GBackGroups[i - 1][j].setValue(0);

GView4.setScore(GView4.getScore() + GBackGroups[i][j].getValue());

}

}

}

added = true;

}

/\*\*

\* 游戏界面绘画

\*

\* @param g

\*/

@Override

public void paint(Graphics g) {

super.paint(g);

for (int i = 0; i < 4; i++) for (int j = 0; j < 4; j++) drawCheck(g, i, j);

if (judgeGameOver()) {

JOptionPane.showMessageDialog(null, "游戏结束，游戏数据将自动保存", "提示", JOptionPane.ERROR\_MESSAGE);

g.setColor(Color.darkGray);

g.fillRect(0, 0, getWidth(), getHeight());

g.setColor(Color.WHITE);

g.setFont(IData.topf);

FontMetrics fms = getFontMetrics(IData.topf);

final String value = "Game Over!";

g.drawString(value,

(getWidth() - fms.stringWidth(value)) / 2,

getHeight() / 2);

}

}

/\*\*

\* 使用Oracle.Java™ Tutorials-2D Graphics进行绘画2048游戏轮廓

\*

\* @param g

\* @param i

\* @param j

\*/

private void drawCheck(Graphics g, int i, int j) {

Graphics2D gg = (Graphics2D) g;

gg.setRenderingHint(RenderingHints.KEY\_ANTIALIASING,

RenderingHints.VALUE\_ANTIALIAS\_ON);

gg.setRenderingHint(RenderingHints.KEY\_STROKE\_CONTROL,

RenderingHints.VALUE\_STROKE\_NORMALIZE);

GBackGroup GBackGroup = GBackGroups[i][j];

gg.setColor(GBackGroup.getBackground());

gg.fillRoundRect(10 + 105 \* j,

10 + 105 \* i,

95, 95, 25, 25);

gg.setColor(GBackGroup.getForeground());

gg.setFont(GBackGroup.getCheckFont());

FontMetrics fms = getFontMetrics(GBackGroup.getCheckFont());

String value = String.valueOf(GBackGroup.getValue());

gg.drawString(value,

10 + 105 \* j +

(95 - fms.stringWidth(value)) / 2,

10 + 105 \* i +

(95 - fms.getAscent() - fms.getDescent()) / 2

+ fms.getAscent());

}

@Override

public void keyReleased(KeyEvent e) {

}

@Override

public void keyTyped(KeyEvent e) {

}

}

}

4.6 Login.java

本文件为游戏登录界面，代码及详细注释如下：

package g2048.v2;

/\*\*

\* @author Xcl

\* @date 2021/12/1 09:48

\* @package g2048.v2

\*/

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileInputStream;

import java.io.InputStreamReader;

import java.nio.charset.StandardCharsets;

/\*\*

\* 本类继承自JFrame实现swing图形界面，实现Runnable, ILogin接口的方法

\*/

public class Login extends JFrame implements Runnable, ILogin {

static final long serialVersionUID = 3878971623272611814L;

static String stt1 = null;

JFrame jfm = this;

JLabel l\_name;

JLabel l\_password;

JTextField t\_name;

JPasswordField t\_password;

JButton b\_login;

JButton b\_reset;

JButton b\_register;

Register rf\_1 = null;

/\*\*

\* Login类的显示

\*/

Login() {

setTitle(getStt1() + "登录窗体");

setVisible(getStt1() != null);

init(getStt1());

setSize(380, 300);

setLocation(200, 100);

}

public static long getSerialVersionUID() {

return serialVersionUID;

}

public static String getStt1() {

return stt1;

}

public static void setStt1(String stt1) {

Login.stt1 = stt1;

}

/\*\*

\* 实现接口中的init展示登录信息的界面

\*/

@Override

public void init(String st) {

setLayout(null);

setL\_name(new JLabel("用户名", JLabel.CENTER));

setL\_password(new JLabel("用户密码", JLabel.CENTER));

setT\_name(new JTextField());

setT\_password(new JPasswordField());

setB\_login(new JButton("登录"));

getB\_login().addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

JOptionPane.showMessageDialog(null, "登录成功会跳转！", " ", -1);

if (st.equals("用户")) {

try {

File file = new File("src/g2048/db/user.txt");

if (file.isFile() && file.exists()) {

InputStreamReader isr = new InputStreamReader(new FileInputStream(file), StandardCharsets.UTF\_8);

BufferedReader br = new BufferedReader(isr);

String data;

while ((data = br.readLine()) != null) {

String[] split = data.split(":");

String[] split2 = split[0].split("\\.");

if (String.valueOf(getT\_name().getText()).equals(split2[0])) {

if (String.valueOf(getT\_password().getPassword()).equals(split2[2])) {

getJfm().dispose();

new GView4(split2[0]).showView();

}

}

}

br.close();

} else System.out.println("文件不存在！");

} catch (Exception e\_1) {

e\_1.printStackTrace();

}

/\*\*

\* 写死的用户管理员

\*/

} else if (st.equals("管理员")) {

if (String.valueOf(getT\_name().getText()).equals("admin123")) {

if (String.valueOf(getT\_password().getPassword()).equals("111111")) {

getJfm().dispose();

new UserInfo(String.valueOf(getT\_name().getText()));

}

}

} else {

System.out.println("不可能执行到这里...");

}

}

});

/\*\*

\* 重置按钮事件监听

\*/

setB\_reset(new JButton("重置"));

getB\_reset().addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

getT\_name().setText("");

getT\_password().setText("");

}

});

/\*\*

\* 注册按钮事件监听

\*/

setB\_register(new JButton("点我注册"));

getB\_register().addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (st.equals("管理员"))

JOptionPane.showMessageDialog(null, "注意：管理员模式登录禁止注册！",

"ERROR", JOptionPane.ERROR\_MESSAGE);

else {

if (getRf\_1() != null) getRf\_1().dispose();

Register rf = new Register();

setRf\_1(rf);

rf.setVisible(true);

}

}

});

JPanel p = new JPanel();

p.setLayout(new GridLayout(2, 2, 5, 5));

p.add(getL\_name());

p.add(getT\_name());

p.add(getL\_password());

p.add(getT\_password());

p.setBounds(5, 5, 350, 185);

add(p);

p = new JPanel();

p.setLayout(new GridLayout(1, 3, 5, 5));

p.add(getB\_login());

p.add(getB\_reset());

p.add(getB\_register());

p.setBounds(5, 205, 350, 50);

add(p);

}

/\*\*

\* 所有变量的set、get方法封装

\*/

@Override

public void run() {

new Login();

}

public JFrame getJfm() {

return jfm;

}

public void setJfm(JFrame jfm) {

this.jfm = jfm;

}

public JLabel getL\_name() {

return l\_name;

}

public void setL\_name(JLabel l\_name) {

this.l\_name = l\_name;

}

public JLabel getL\_password() {

return l\_password;

}

public void setL\_password(JLabel l\_password) {

this.l\_password = l\_password;

}

public JTextField getT\_name() {

return t\_name;

}

public void setT\_name(JTextField t\_name) {

this.t\_name = t\_name;

}

public JPasswordField getT\_password() {

return t\_password;

}

public void setT\_password(JPasswordField t\_password) {

this.t\_password = t\_password;

}

public JButton getB\_login() {

return b\_login;

}

public void setB\_login(JButton b\_login) {

this.b\_login = b\_login;

}

public JButton getB\_reset() {

return b\_reset;

}

public void setB\_reset(JButton b\_reset) {

this.b\_reset = b\_reset;

}

public JButton getB\_register() {

return b\_register;

}

public void setB\_register(JButton b\_register) {

this.b\_register = b\_register;

}

public Register getRf\_1() {

return rf\_1;

}

public void setRf\_1(Register rf\_1) {

this.rf\_1 = rf\_1;

}

}

4.7 Main2048.java

本文件为2048小游戏主函数，代码及详细注释如下：

package g2048.v2;

/\*\*

\* @author Xcl

\* @date 2021/12/1 09:50

\* @package g2048.v2

\*/

public class Main2048 {

/\*\*

\* 整个程序总进入入口

\*/

public static void main(String[] args) {

new GStart("2048游戏窗口");

}

}

4.8 Register.java

本文件为2048小游戏注册界面，代码及详细注释如下：

package g2048.v2;

/\*\*

\* @author Xcl

\* @date 2021/12/1 09:48

\* @package g2048.v2

\*/

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.\*;

import java.nio.charset.StandardCharsets;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

/\*\*

\* 本类继承自JFrame实现swing图形界面，实现IData接口的所有方法

\*/

public class Register extends JFrame implements IData {

static final long serialVersionUID = -3035113424256798327L;

static final Register pre\_u2 = null;

String sss = null;

JFrame jfm;

ButtonGroup bg;

JLabel l\_id;

JLabel l\_name;

JLabel l\_password1;

JLabel l\_password2;

JLabel l\_sex;

JLabel l\_password3;

JTextField t\_id;

JTextField t\_name;

JPasswordField p\_password1;

JPasswordField p\_password2;

JPasswordField p\_card;

JRadioButton r\_male;

JRadioButton r\_female;

JButton b\_reg;

JButton b\_reset;

String sex1;

/\*\*

\* Register类的显示

\*/

Register() {

init(getSss());

setTitle("2048游戏注册");

setSize(300, 450);

setLocation(100, 150);

}

/\*\*

\* 获取文件中行数并返回

\*

\* @param filepath

\* @return

\*/

public static int getFileLine(String filepath) {

try (LineNumberReader lnr = new LineNumberReader(new FileReader(filepath))) {

lnr.skip(Long.MAX\_VALUE);

int lnm1 = lnr.getLineNumber();

return lnm1 + 1;

} catch (Exception e) {

e.printStackTrace();

return -1;

}

}

public static long getSerialVersionUID() {

return serialVersionUID;

}

public static Register getPre\_u2() {

return pre\_u2;

}

/\*\*

\* 实现接口中的init展示注册信息的界面

\*/

@Override

public void init(String sss) {

setLayout(new GridLayout(7, 2, 5, 5));

setL\_id(new JLabel("用户名", JLabel.CENTER));

setL\_name(new JLabel("真实姓名", JLabel.CENTER));

setL\_password1(new JLabel("密码", JLabel.CENTER));

setL\_password2(new JLabel("确认密码", JLabel.CENTER));

setL\_password3(new JLabel("身份证号", JLabel.CENTER));

setL\_sex(new JLabel("性别", JLabel.CENTER));

setT\_id(new JTextField());

setT\_name(new JTextField());

setP\_password1(new JPasswordField());

setP\_password2(new JPasswordField());

setP\_card(new JPasswordField());

JPanel p = new JPanel();

setBg(new ButtonGroup());

setR\_male(new JRadioButton("男"));

setR\_female(new JRadioButton("女"));

getBg().add(getR\_male());

getBg().add(getR\_female());

setB\_reg(new JButton("注册"));

getB\_reg().addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String panduancf = null;

JOptionPane.showMessageDialog(null, "务必牢记自己的用户信息，一旦注册成功后只有管理员可以更改密码。", "提示", JOptionPane.CANCEL\_OPTION);

if (Register.getPre\_u2() != null) Register.getPre\_u2().dispose();

if (getR\_female().isSelected()) setSex1("女");

else if (getR\_male().isSelected()) setSex1("男");

int line = getFileLine("src/g2048/db/user.txt");//获取行数

List<String> list1 = new ArrayList<>();//List的泛型应用

/\*\*

\* 用户个人信息的文件读写

\*/

try {

File file = new File("src/g2048/db/user.txt");

if (file.isFile() && file.exists()) {

InputStreamReader isr = new InputStreamReader(new FileInputStream(file), StandardCharsets.UTF\_8);

BufferedReader br = new BufferedReader(isr);

String data;

while ((data = br.readLine()) != null) {

String[] split = data.split(":");

list1.add(split[0].split("\\.")[0]);

}

br.close();

} else System.out.println("文件不存在！");

} catch (Exception e\_1) {

e\_1.printStackTrace();

}

/\*\*

\* 判断用户名是否重复

\*/

for (int b = 0; b < line; b++) {

if (String.valueOf(getT\_name().getText()).equals(list1.get(b))) {

panduancf = "CF";

} else {

panduancf = "wg";

}

}

if (panduancf.equals("CF")) {

JOptionPane.showMessageDialog(null, "用户名重复，请重新注册！",

"ERROR", JOptionPane.ERROR\_MESSAGE);

getT\_id().setText("");

getT\_name().setText("");

getP\_password1().setText("");

getP\_password2().setText("");

getP\_card().setText("");

} else {

if (String.valueOf(getP\_password1().getPassword()).equals(String.valueOf((getP\_password2().getPassword())))) {

User u = new User(String.valueOf(getT\_id().getText()), String.valueOf(getT\_name().getText()),

String.valueOf(getP\_password1().getPassword()), String.valueOf(getSex1()).charAt(0),

String.valueOf(getP\_card().getPassword()));

String datime = new SimpleDateFormat("yyyy-MM-dd|HH-mm-ss").format(new Date());

JOptionPane.showMessageDialog(null, "注册成功！", " ", -1);

List<String> listuser = new ArrayList<>();

listuser.add(u.getId());

listuser.add(u.getName());

listuser.add(u.getPassword());

listuser.add(u.getCard());

listuser.add(String.valueOf(u.getSex()));

listuser.add(datime);

String data = listuser.get(0) + "." + listuser.get(1) + "." + listuser.get(2) + "." + listuser.get(3)

+ "." + listuser.get(4) + "." + listuser.get(5) + ":";

/\*\*

\* 用户个人信息的文件读写

\*/

try {

File f = new File("src/g2048/db/user.txt");

if (!f.exists()) f.createNewFile();

FileOutputStream fw = new FileOutputStream(f, true);

OutputStreamWriter ow = new OutputStreamWriter(fw);

BufferedWriter bw = new BufferedWriter(ow);

bw.newLine();

bw.write(data);

bw.flush();

bw.close();

ow.close();

fw.close();

} catch (FileNotFoundException e1) {

e1.printStackTrace();

} catch (IOException e2) {

e2.printStackTrace();

}

new GView4(listuser.get(0)).showView();

} else {

getP\_password1().setText("");

getP\_password2().setText("");

}

}

}

});

/\*\*

\* 重置按钮事件监听

\*/

setB\_reset(new JButton("重置"));

getB\_reset().addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

getT\_id().setText("");

getT\_name().setText("");

getP\_password1().setText("");

getP\_password2().setText("");

getP\_card().setText("");

JOptionPane.showMessageDialog(null, "重置成功！", " ", 1);

}

});

add(getL\_id());

add(getT\_id());

add(getL\_name());

add(getT\_name());

add(getL\_password1());

add(getP\_password1());

add(getL\_password2());

add(getP\_password2());

add(getL\_password3());

add(getP\_card());

add(getL\_sex());

p.add(getR\_male());

p.add(getR\_female());

add(p);

add(getB\_reg());

add(getB\_reset());

}

/\*\*

\* 所有变量的set、get方法封装

\*/

public String getSss() {

return sss;

}

public void setSss(String sss) {

this.sss = sss;

}

public JFrame getJfm() {

return jfm;

}

public void setJfm(JFrame jfm) {

this.jfm = jfm;

}

public ButtonGroup getBg() {

return bg;

}

public void setBg(ButtonGroup bg) {

this.bg = bg;

}

public JLabel getL\_id() {

return l\_id;

}

public void setL\_id(JLabel l\_id) {

this.l\_id = l\_id;

}

public JLabel getL\_name() {

return l\_name;

}

public void setL\_name(JLabel l\_name) {

this.l\_name = l\_name;

}

public JLabel getL\_password1() {

return l\_password1;

}

public void setL\_password1(JLabel l\_password1) {

this.l\_password1 = l\_password1;

}

public JLabel getL\_password2() {

return l\_password2;

}

public void setL\_password2(JLabel l\_password2) {

this.l\_password2 = l\_password2;

}

public JLabel getL\_sex() {

return l\_sex;

}

public void setL\_sex(JLabel l\_sex) {

this.l\_sex = l\_sex;

}

public JLabel getL\_password3() {

return l\_password3;

}

public void setL\_password3(JLabel l\_password3) {

this.l\_password3 = l\_password3;

}

public JTextField getT\_id() {

return t\_id;

}

public void setT\_id(JTextField t\_id) {

this.t\_id = t\_id;

}

public JTextField getT\_name() {

return t\_name;

}

public void setT\_name(JTextField t\_name) {

this.t\_name = t\_name;

}

public JPasswordField getP\_password1() {

return p\_password1;

}

public void setP\_password1(JPasswordField p\_password1) {

this.p\_password1 = p\_password1;

}

public JPasswordField getP\_password2() {

return p\_password2;

}

public void setP\_password2(JPasswordField p\_password2) {

this.p\_password2 = p\_password2;

}

public JPasswordField getP\_card() {

return p\_card;

}

public void setP\_card(JPasswordField p\_card) {

this.p\_card = p\_card;

}

public JRadioButton getR\_male() {

return r\_male;

}

public void setR\_male(JRadioButton r\_male) {

this.r\_male = r\_male;

}

public JRadioButton getR\_female() {

return r\_female;

}

public void setR\_female(JRadioButton r\_female) {

this.r\_female = r\_female;

}

public JButton getB\_reg() {

return b\_reg;

}

public void setB\_reg(JButton b\_reg) {

this.b\_reg = b\_reg;

}

public JButton getB\_reset() {

return b\_reset;

}

public void setB\_reset(JButton b\_reset) {

this.b\_reset = b\_reset;

}

public String getSex1() {

return sex1;

}

public void setSex1(String sex1) {

this.sex1 = sex1;

}

}

4.9 ILogin.java

本文件为注册界面的接口规划，代码及详细注释如下：

package g2048.v2;

/\*\*

\* @author Xcl

\* @date 2021/12/3 21:54

\* @package g2048.v3

\*/

public interface ILogin {

/\*\*

\* ILogin接口定义

\*

\* @param st

\*/

void init(String st);

void run();

}

4.10 UserInfo.java

本文件为管理员登陆后显示用户详细信息的界面，代码及详细注释如下：

package g2048.v2;

/\*\*

\* @author Xcl

\* @date 2021/12/1 09:48

\* @package g2048.v2

\*/

import javax.swing.\*;

import java.io.\*;

import java.nio.charset.StandardCharsets;

import java.util.ArrayList;

import java.util.List;

/\*\*

\* 本类继承自JFrame实现swing图形界面，实现IData接口的所有方法

\*/

public class UserInfo extends JFrame implements IData {

static final long serialVersionUID = -4598519373915560300L;

JTable table\_book;

UserInfo(String st) {

init(st);

setVisible(true);

setSize(800, 600);

setTitle("管理员" + st + "的用户信息展示区");

}

/\*\*

\* 获取文件中行数并返回

\*

\* @param filepath

\* @return

\*/

public static int getFileLine(String filepath) {

try (LineNumberReader lnr = new LineNumberReader(new FileReader(filepath))) {

lnr.skip(Long.MAX\_VALUE);

int lnm1 = lnr.getLineNumber();

return lnm1 + 1;

} catch (Exception e) {

e.printStackTrace();

return -1;

}

}

public static long getSerialVersionUID() {

return serialVersionUID;

}

/\*\*

\* 整个管理员展示用户信息界面的展示

\*

\* @param st

\*/

@Override

public void init(String st) {

setLayout(null);

Object[] title = {"账号", "密码", "姓名", "性别", "身份证号", "注册时间"};

int line = getFileLine("src/g2048/db/user.txt");

Object[][] bookList = new String[line][6];

List<String> list1 = new ArrayList<>();

try {

File file = new File("src/g2048/db/user.txt");

if (file.isFile() && file.exists()) {

InputStreamReader isr = new InputStreamReader(new FileInputStream(file), StandardCharsets.UTF\_8);

BufferedReader br = new BufferedReader(isr);

String data;

while ((data = br.readLine()) != null) {

String[] split = data.split(":");

list1.add(split[0].split("\\.")[0]);

list1.add(split[0].split("\\.")[2]);

list1.add(split[0].split("\\.")[1]);

list1.add(split[0].split("\\.")[4]);

list1.add(split[0].split("\\.")[3]);

list1.add(split[0].split("\\.")[5]);

}

br.close();

} else System.out.println("文件不存在！");

} catch (Exception e\_1) {

e\_1.printStackTrace();

}

for (int j = 0; j < 6; j++)

for (int b = 0, k = 0; b < line; b++, k += 6)

switch (j) {

case 0:

bookList[b][j] = list1.get(k);

break;

case 1:

bookList[b][j] = list1.get(k + 1);

break;

case 2:

bookList[b][j] = list1.get(k + 2);

break;

case 3:

bookList[b][j] = list1.get(k + 3);

break;

case 4:

bookList[b][j] = list1.get(k + 4);

break;

case 5:

bookList[b][j] = list1.get(k + 5);

break;

}

setTable\_book(new JTable(bookList, title));

JScrollPane pane = new JScrollPane();

pane.setBounds(20, 20, 750, 500);

pane.getViewport().add(getTable\_book());

getContentPane().add(pane);

}

public JTable getTable\_book() {

return table\_book;

}

public void setTable\_book(JTable table\_book) {

this.table\_book = table\_book;

}

}

4.11 User.java

本文件为用户信息实体类的实现方式类，代码及详细注释如下：

package g2048.v2;

/\*\*

\* @author Xcl

\* @date 2021/12/1 09:48

\* @package g2048.v2

\*/

public class User {

String id;

String name;

String password;

String card;

char sex;

User(String id, String name, String password, char sex, String card) {

super();

this.setId(id);

this.setName(name);

this.setPassword(password);

this.setSex(sex);

this.setCard(card);

}

/\*\*

\* user实体类中所有的get、set方法

\*/

@Override

public String toString() {

return "User{" +

"id='" + id + '\'' +

", name='" + name + '\'' +

", password='" + password + '\'' +

", card='" + card + '\'' +

", sex=" + sex +

'}';

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

char getSex() {

return sex;

}

public void setSex(char sex) {

this.sex = sex;

}

String getCard() {

return card;

}

public void setCard(String card) {

this.card = card;

}

}

**5.2048小游戏程序使用说明**

首先通过运行Main2048.java启动程序：



图5.1 游戏进入界面

然后可以点击开始游戏，根据中国相关法律法规，需要用户注册、登录后才可以开始游戏，如图：



图5.2 点开始游戏后弹窗提示



图5.3 点确定后用户登录界面

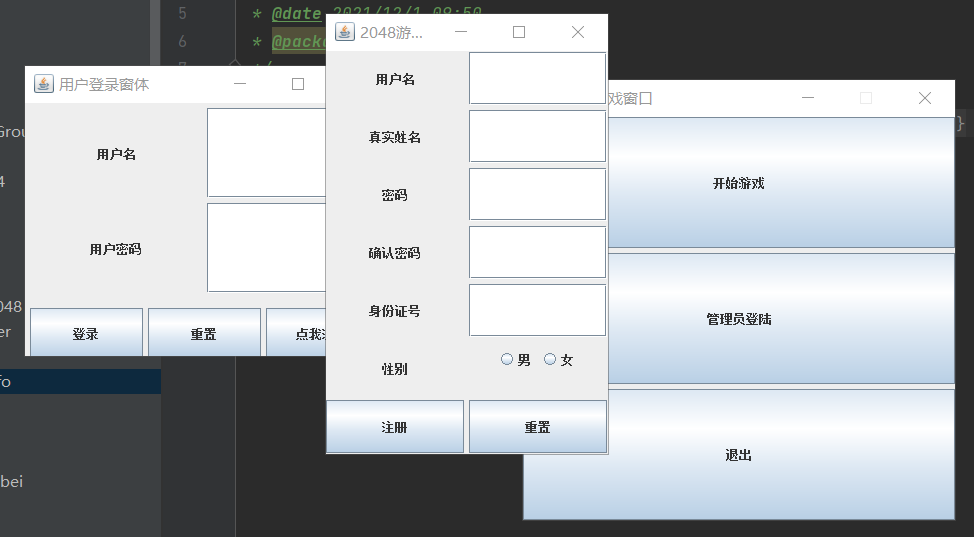


图5.4 点注册后注册界面

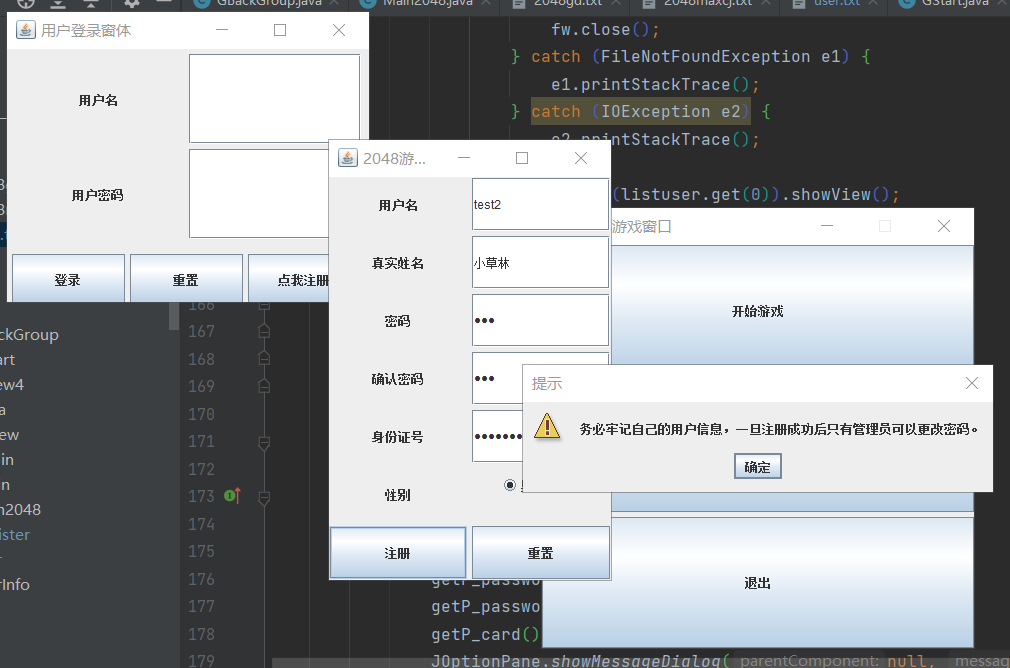


图5.5 输入个人信息后点注册的界面

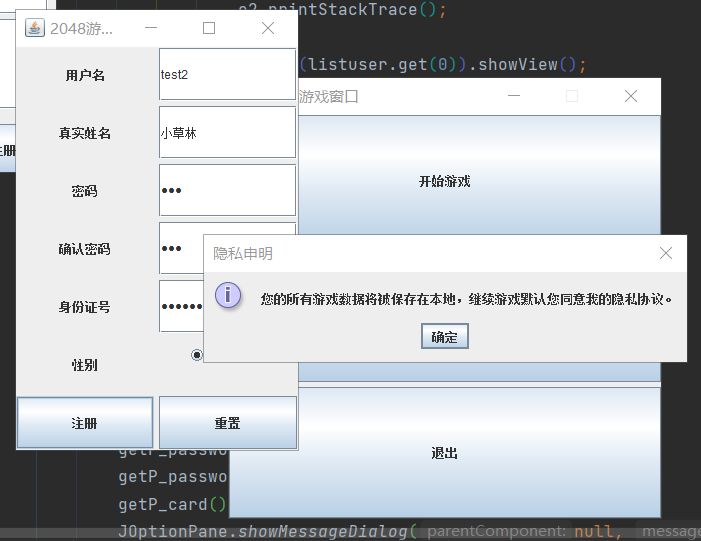


图5.6 进入游戏的隐私声明弹窗

注册登录成功后即可进入如下游戏界面：

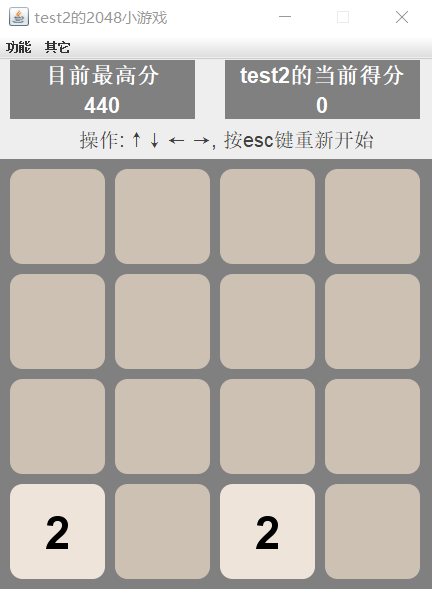


图5.7 进入游戏后界面

点开顶端功能菜单可以选择3\*3和4\*4（默认）和5\*5的2048游戏（3\*3和5\*5功能将在V3.0版本实现），点开其他菜单可以选择退出游戏、查看排行榜（将在V3.0版本中实现）、关于界面，关于界面如图：

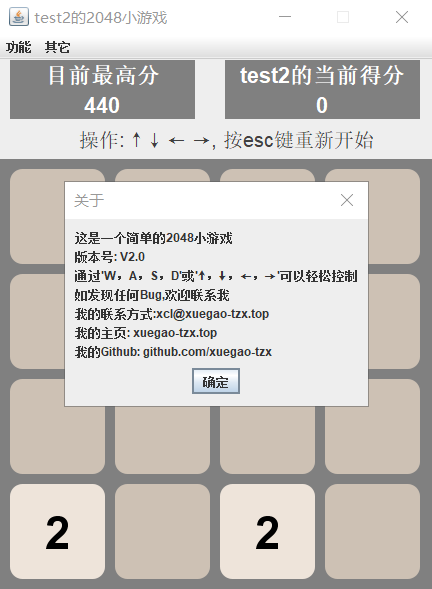


图5.8 关于界面

游戏如果游玩失败，会有如下提示：



图5.9 游戏失败界面1



图5.10 游戏失败界面2

也可以点击管理员登录（账号：admin123密码：111111）查看修改已有用户信息：

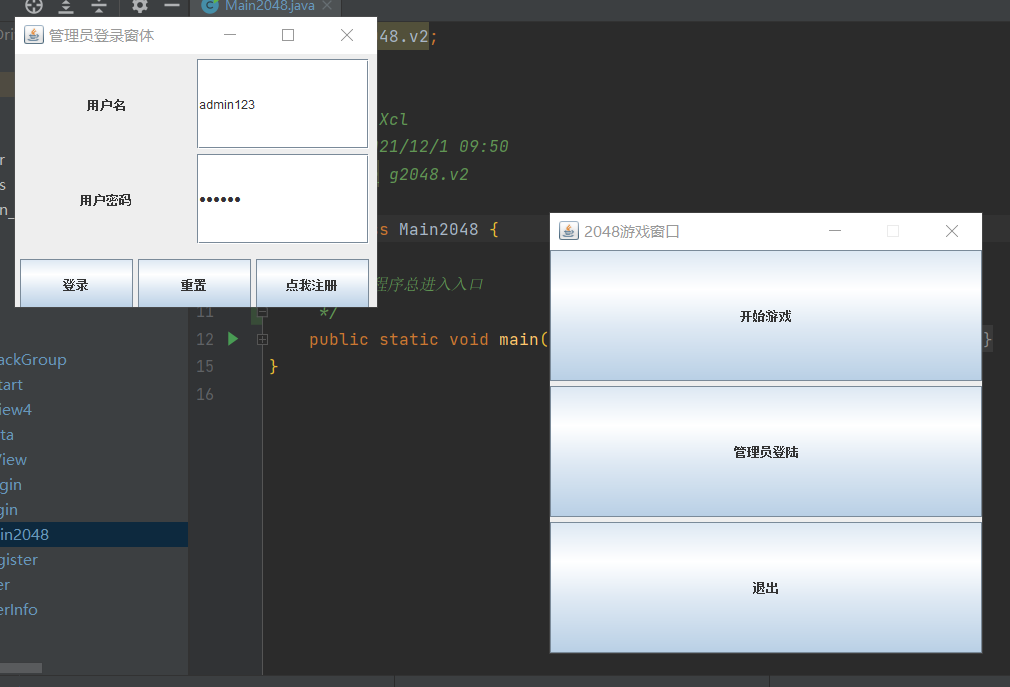


图5.11 点管理员登录后显示

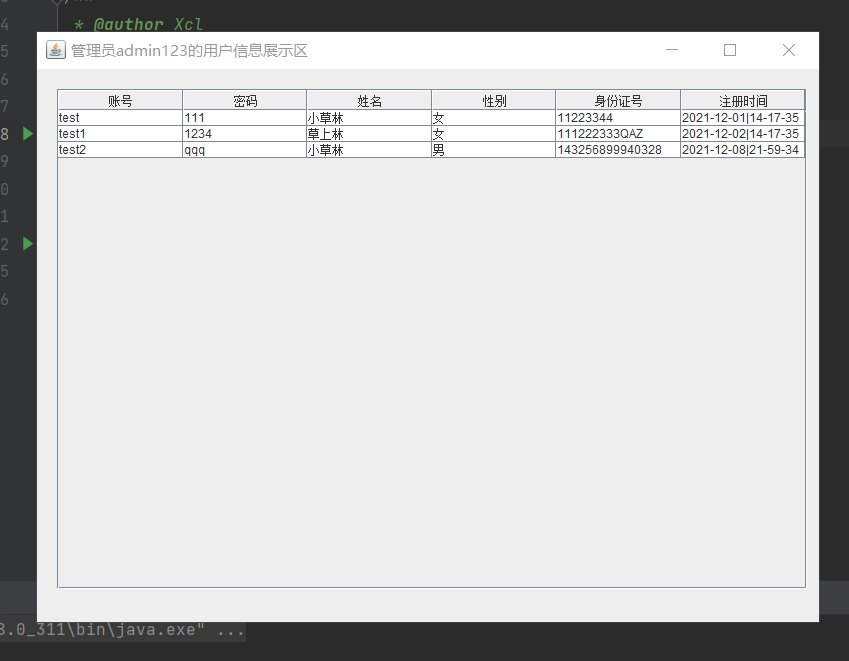


图5.12 管理员登录后用户信息显示界面

# 6.心得体会

本次创作的实验大作业，我使用了《数据结构》、《面向对象程序设计》和《软件工程导论》的知识，我写了这个2048小游戏，理由如下：

（1）2048小游戏本质上是经典的数组问题，使用Java实现相关算法可以夯实算法基础。

1. 系统的界面、相关事件处理等都是在Java课上所学，通过实战检验做学知识并提高了编码能力。
2. 依据曾经编写鸿蒙Js手表版本的2048小游戏的经验可以较好入手

（4）因为系统繁杂，工作量大，不免要使用《软件工程导论》的知识设计系统，划分功能模块，理清各类的职责，使用设计模式以降低模块间耦合。

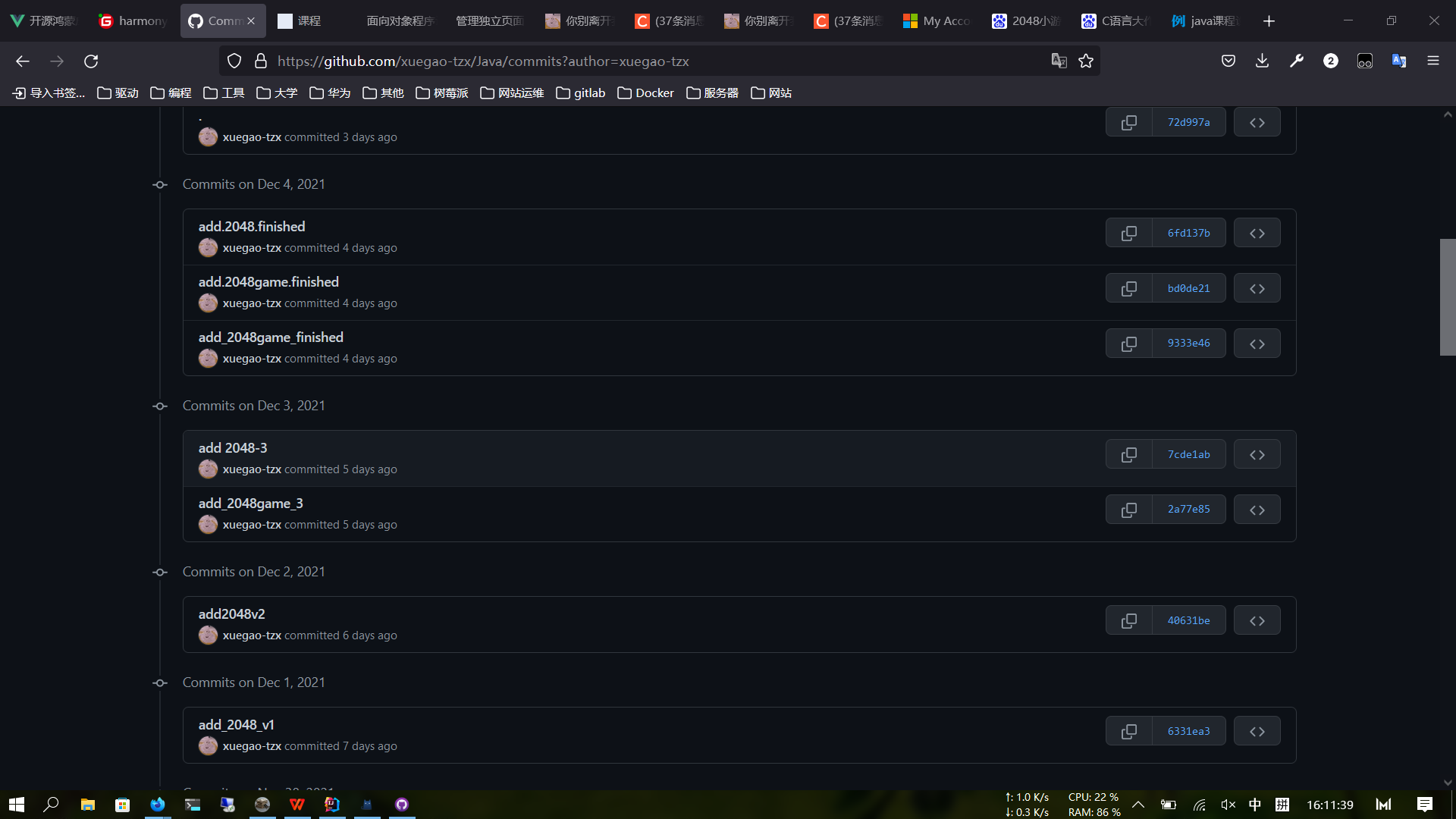


图6.1 Github commit历史1

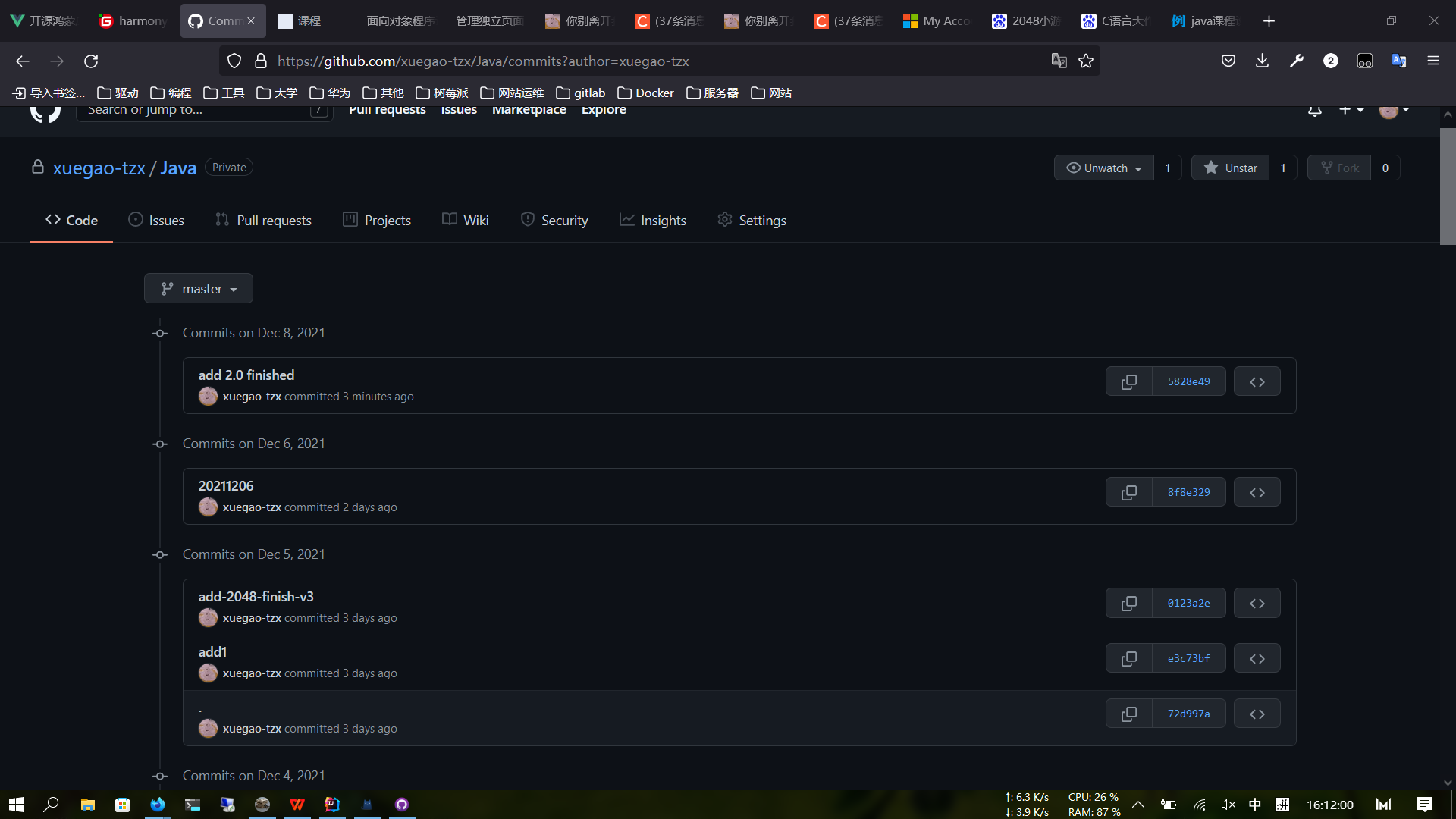


图6.2 Github commit历史2

本项目历时不足一周时间，从第一个预览版本到现在的2.0版，以及将来的3.0版本在版本迭代的过程中基本达到了自己选择这个课题的初心，同时在运用知识解决问题的过程中确实遇到了一些棘手的问题，例如说游戏操作的实现、线程安全问题、界面绘制效率问题等等，通过查阅相关文档在学习的同时也提升了自己能力。

# 7.参考文献

[1]Oracle.Java™ Tutorials-2D Graphics.<https://docs.oracle.com/javase/tutorial/2d/index.html>