**2017-2018学年第1学期**

**《Java言程序设计》课程大作业报告**

**课题名称：    简易小游戏开发**

**小组成员人数（单选）■1人      □2人      □3人**

# 1. 软件系统分析

**本软件由一些简单的小游戏组成，要求用户先进行注册，登陆后才能开始游戏，用户只需进行用户和密码注册即可开始游戏，还提供了完善信息、修改信息，查询信息，删除信息功能能。**

## （1）登录和注册模块

**要实现的功能分别为用户登录，用户注册。**

## （2）主页面模块

**用户登录后将进入该页面，该页面提供完善信息，修改信息，查询信息，删除信息，还有进入游戏模块的功能。**

## （3）数据库模块

**建立两个数据表分别为，用户表和详细信息表**

**1、用户表中包含用户的ID、姓名、密码三变量，用于储存用户登录和注册信息。**

**2、详细表中包含用户ID、姓名、性别、年龄、电话号码、所在地区。**

**3.两个表中ID设为自增，且两个表之间通过ID进行链接**

## （4）游戏模块

**在游戏模块时，会弹出游戏选择界面，点击其中游戏所代表的按纽就会跳到相应游戏的界面，然后就可以尽情的玩耍啦。**

# 算法分析

## Model

**数据库包一个，游戏包总共有8个，分别是俄罗斯方块，拼图，lianliankan，PlaneWar，SnakeGame，贪吃蛇，数独Tank。因这八个游戏中，有6个游戏是我平时逛论坛所得，所以只将我所做的拼图游戏和数独游戏做详细介绍。**

****拼图****

**九宫格小游戏，总共有九个格子，放有八个方块，方块上分别写有1-8的数字，按照从左至右，从上到下的顺序将格子排好序，游戏通关。从上述九宫格游戏，我进行了改进，将其变成了拼图游戏，代码思路如下：**

**拼图.java**

**首先建立一个num数组填入1-9中的数字，将num[8]即第九个数置为-1,表示该位置为空，即拼图中没有图片的那一块，分别对应九个按纽，将其中值不为-1得八个按纽添加监听事件后，添加到容器中，并且依次在按纽上贴上八张事先准备好的大小合适图片，接下来就是实现点击事件，利用的原理是，当点击的按纽不是第三行的话，就将下一行相邻的空按纽相互交换按纽上的图片，如果不是第一行的话，就与上一行的相邻按纽交换图片，如果不是第一列的话，就与下一列相邻的的空按钮交换图片，如果不是第三列的话就与上一列的空按钮交换图片。**

**最终当所有游戏拼图按顺序拼好后，只需要比较按纽上的数字是否和顺序一样，就可知道是否成功，将步数记录下来，看是否打破纪录。**

**数独**

**数独每一行称为数独的行，每一列称为数独的列，每一个小九宫格称为数独的宫，玩家需要根据9×9盘面上的已知数字，推理出所有剩余空格的数字，并满足每一行、每一列、每一个粗线宫内的数字均含1-9，不重复。 每一道合格的数独谜题都有且仅有唯一答案，推理方法也以此为基础，任何无解或多解的题目都是不合格的。**

**LimitNumDocument.java**

**与JTextArea结合可以限制JTextArea的输入字符数,使即用方法JTextArea.setDocument(new LimitNumDocument(int length))。在该类中重写insertString(int** offs,String str,AttributeSet attr)**)限制JTextArea的输入字符数。**

**MyButton.java**

**该类定义了按纽的样式，利用setBorderPainted去掉按纽矩形边角，将字体前景色设置为天蓝色，用setContentAreaFilled设置透明度，增加鼠标移入让鼠标颜色变换的适配器，移出又将恢复，利用paint()函数，调用Graphics2D类，设置悬浮在按纽上颜色的深度，利用RoundRectangle2.Float 方法绘制了按纽的形状，使其与按纽完全重合，并用GradientPaint类进行颜色渐变的处理。**

**MyTextArea.java**

**该类定义每一个方框的样式，利用setLineWrap设置自动换行，利用setOpaque设置透明度，使用setDocument和LimitNumDocument函数设置方框中个数的限制，并设置前景色为天蓝色，利用paint方法进行方框的绘制，将方框设置为黑色，利用两个drawRect函数进行方框线条粗细的处理，利用getMyAreas()函数进行相应等级数独的绘制。**

**MainFrame.java**

**该类是函数的主框架，将字体都设置为金桥简行楷，设置其大小，利用MyButton类和MyTextArea类进行网格和新游戏，验证游戏结果，显示答案，退出按纽的绘制，使用sudoku类与其进行结合，实现按纽的点击事件和方框空格的focus事件，还有难度等级的单选按纽事件，达到更好地用户体验。**

**Sudoku.java**

**该类是该游戏的逻辑代码实现，首先初始化了9x9的全为0的矩阵a，声明了存储矩阵答案的result数组，对应难度小最小的已知格子数MINFILLED即每行列已知格的底线MINKNOW，以及中间矩阵map和布尔类型HaveResult，这就是该类所具有的所有属性。**

**下面介绍该类所具有的方法：**

**首先是布尔类型的check函数，用于检测当前矩阵是否满足数独每行每列每宫不能有重复值得要求，其次是boolcheck**(**int**[][] data,**int** digit,**int** x,**int** y)**函数用于检测data[x][y]中是否含有digit值，如果没有则返回true，利用setLevel函数设置难度等级，利用无参getMySudoku函数获得一个普通的单解数独默认等级为3，利用有参的getMySudoku函数获得对应等级的单解数独，利用getResultSudoku获得数独的解，利用RandySudokuhanshu 随机选取n个空格放入1-9的随机数，利用LevelSudoku生成一个指定等级的数独，利用EqualChange将终盘等效交换，利用SetLevel进行难度等级的设定，利用FindNext设置对应难度下的挖掉顺序，生成不同难度的数独，利用布尔类型的isOnlyOne判断数独是否具有唯一解，利用MinKnow设置行列已知的格数的底线，利用Solvesudoku判断数独b是否能解，利用put函数在第n个位置放入数字，利用output给出结果，利用test判断填入的数字是否可行。**

## View

**welcome类**

**起始页面，欢迎用户来到娱乐空间，并提供登录按钮。**

**Login类**

**这个类主要实现的是登入窗口的方法，在主窗体里设置了两个文本框用于获得用户名和密码。然后设置两个按钮分别是登陆和注册。关于登陆按钮会先检查二个文本框是否为空，如果是就报错。否则调用数据库查询是否信息正确。关于注册按钮，当用户名密码不为空的时候，检测数据库里没有同名联系人的情况下将新建一个账号。**

**Index类**

**这个类用于完善信息，修改信息，查询信息，删除信息，当点击按钮时，会弹出提示框询问你要进行操作的对象名，当选择取消按钮时会询问你是否取消操作，当你删除时，会询问你是否删除，当输入对象名后，你所要操作的对象信息会出现在提前布局好的图形界面框中，你只要进行相应的处理后点击保存，就会完成相应的操作。**

**SelectGame类**

**这个类用于选择游戏。其中包含的游戏有俄罗斯方块，永远点不到的方块，数独，连连看，拼图，贪吃蛇，飞机大战，坦克大战。**

## Control

**Main类**

**函数的运行类，运行从这里开始。**

# 3、软件和代码的实现

## 登陆窗口的实现 Login.java

**package Graphic;**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.sql.\*;**

**import java.util.\*;**

**import com.microsoft.sqlserver.jdbc.\*;**

**import Game.Game1;**

**import javax.swing.JButton;**

**import javax.swing.JOptionPane;**

**public class Login extends Frame implements ActionListener {**

**Label lblUserName = new Label("用户名称：");**

**TextField txtUserName = new TextField();**

**Label lblUserPasswd = new Label("用户密码：");**

**TextField txtUserPasswd = new TextField();**

**Button btnLogin = new Button("登录");**

**Button btnRegister = new Button("注册");**

**private Connection connect() {**

**Connection con = null;**

**try {**

**Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");**

**con = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;DatabaseName=java", "sa", "zwx132");**

**} catch (SQLException e) {**

**System.out.println("连接数据库失败");**

**e.printStackTrace();**

**} catch (ClassNotFoundException e) {**

**e.printStackTrace();**

**}**

**return con;**

**}**

**public Login() {**

**this.setLayout(null); // 手工设置各个组件的位置和大小**

**lblUserName.setBounds(new Rectangle(45, 38, 67, 34));**

**txtUserName.setBounds(new Rectangle(115, 39, 210, 33));**

**lblUserPasswd.setBounds(new Rectangle(43, 86, 66, 26));**

**txtUserPasswd.setBounds(new Rectangle(115, 84, 210, 33));**

**txtUserPasswd.setEchoChar('\*');**

**btnLogin.setBounds(new Rectangle(78, 150, 86, 30));**

**btnRegister.setBounds(new Rectangle(193, 150, 86, 30));**

**this.add(lblUserName); // 把各个组件添加到框架窗口中**

**this.add(txtUserName);**

**this.add(lblUserPasswd);**

**this.add(txtUserPasswd);**

**this.add(btnLogin);**

**this.add(btnRegister);**

**btnLogin.addActionListener(this);**

**btnRegister.addActionListener(this);**

**}**

**public Dimension getPreferredSize()// 设置框架窗口的大小**

**{**

**return new Dimension(355, 204);**

**}**

**public void actionPerformed(ActionEvent e) {**

**Connection con=connect();**

**PreparedStatement ptmt=null;**

**PreparedStatement ptmt2=null;**

**String username=null;**

**String pwd=null;**

**try**

**{**

**username = txtUserName.getText().trim();// 获得用户名**

**pwd = String.valueOf(txtUserPasswd.getText().trim());// 获得密码**

**}**

**catch(Exception exc)**

**{**

**JOptionPane.showMessageDialog(null, "用户名或密码不能为空","哎呀，出错了",JOptionPane.WARNING\_MESSAGE);**

**}**

**try**

**{**

**if(e.getSource()==btnLogin)**

**{**

**String sql="select \* from userlist where name=? and pw=?";**

**ptmt=con.prepareStatement(sql);**

**ptmt.setString(1, username);**

**ptmt.setString(2, pwd);**

**ResultSet rs=ptmt.executeQuery();**

**if (rs.next()) {**

**JOptionPane.showMessageDialog(null, "登入成功！");**

**clear();**

**index frm = new index();**

**frm.addWindowListener(new WindowAdapter() {**

**public void windowClosing(WindowEvent e) {**

**System.exit(0);**

**}**

**});**

**frm.pack();**

**frm.setVisible(true);// 显示框架窗体**

**} else {**

**JOptionPane.showMessageDialog(null, "请输入正确且不为空的用户名和密码","哎呀，出错了",JOptionPane.WARNING\_MESSAGE);**

**clear();**

**}**

**ptmt.close();**

**con.close();**

**}**

**else**

**{**

**String sql = "";**

**if (username.equals(""))**

**{**

**JOptionPane.showMessageDialog(null, "用户名不能为空");**

**clear();**

**}**

**else if (pwd.equals(""))**

**{**

**JOptionPane.showMessageDialog(null, "密码不能为空");**

**clear();**

**}**

**else**

**{**

**sql = "select \* from userlist where name=?";**

**ptmt=con.prepareStatement(sql);**

**ptmt.setString(1,username);**

**ResultSet rs=ptmt.executeQuery();**

**if (rs.next())**

**{// 用户名已经存在**

**JOptionPane.showMessageDialog(null, "对不起，用户名已存在！！！");**

**clear();**

**}**

**else**

**{**

**try**

**{**

**sql = "insert into userlist(name,pw) values(?,?)";**

**String sql2= "insert into xiangxi(name) values(?)";**

**ptmt=con.prepareStatement(sql);**

**ptmt2=con.prepareStatement(sql2);**

**ptmt2.setString(1, username);**

**ptmt.setString(1, username);**

**ptmt.setString(2, pwd);**

**int m1= ptmt2.executeUpdate();**

**int m2=ptmt.executeUpdate();**

**if (m1>0&&m2>0)**

**{// 注册成功**

**clear();**

**JOptionPane.showMessageDialog(null,**

**"恭喜您！！！注册成功，请完善信息");**

**index frm = new index();**

**// 设置框架窗体的事件监听(关闭窗体事件)**

**frm.addWindowListener(new WindowAdapter() {**

**public void windowClosing(WindowEvent e) {**

**System.exit(0);**

**}**

**});**

**frm.pack();**

**frm.setVisible(true);// 显示框架窗体**

**}**

**}**

**catch(SQLException e1)**

**{**

**e1.printStackTrace();**

**}**

**}**

**}**

**}**

**}**

**catch(SQLException exc)**

**{**

**exc.printStackTrace();**

**}**

**}**

**public static void main(String[] args) {**

**Login frmLogin = new Login();**

**// 设置框架窗体的事件监听(关闭窗体事件)**

**frmLogin.addWindowListener(new WindowAdapter() {**

**public void windowClosing(WindowEvent e) {**

**System.exit(0);**

**}**

**});**

**frmLogin.pack();**

**frmLogin.setVisible(true);// 显示框架窗体**

**}**

**private void clear() {**

**// TODO Auto-generated method stub**

**txtUserName.setText("");**

**txtUserPasswd.setText("");**

**}**

**}**

## 主页面图形界面与事务的实现 index.java

package Graphic;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

import java.util.\*;

import com.microsoft.sqlserver.jdbc.\*;

import Game.Game1;

import javax.swing.ButtonGroup;

import javax.swing.JButton;

import javax.swing.JOptionPane;

import javax.swing.JRadioButton;

public class index extends Frame implements ActionListener{

int id;

TextField PHName=new TextField();

ButtonGroup bg=new ButtonGroup();

JRadioButton PHSex1=new JRadioButton("男");

JRadioButton PHSex2=new JRadioButton("女");

TextField PHAge=new TextField();

TextField PHone=new TextField();

TextField PHAddress=new TextField();

TextField PHSearch=new TextField();

TextArea PHInfo=new TextArea();

JButton jbtAdd=new JButton("完善信息");

JButton jbtSearch=new JButton("查询");

JButton jbtUpdate=new JButton("修改");

JButton jbtdelete=new JButton("删除");

JButton jbtSave=new JButton("保存");

JButton jbtBack=new JButton("进入游戏");

Label lblPHName = new Label("用户名称：");

Label lblPHSex = new Label("用户性别：");

Label lblPHAge = new Label("用户年龄：");

Label lblPHone = new Label("用户电话：");

Label lblPHAddress = new Label("用户地址：");

Label lblPHSearch = new Label("搜索：");

Label lblPHInfo=new Label("搜索结果：");

Vector<String> pInfo = new Vector<String>();

private Connection connect() {

Connection con=null;

try

{

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

con=DriverManager.getConnection("jdbc:sqlserver://localhost:1433;DatabaseName=java","sa","zwx132");

}

catch(SQLException e)

{

e.printStackTrace();

}

catch(ClassNotFoundException e)

{

e.printStackTrace();

}

return con;

}

public index()

{

this.setLayout(null); // 手工设置各个组件的位置和大小

lblPHName.setBounds(new Rectangle(45, 38, 67, 34));

PHName.setBounds(new Rectangle(115, 39, 210, 33));

lblPHSex.setBounds(new Rectangle(43, 86, 66, 26));

PHSex1.setBounds(115, 84,54,23);

PHSex2.setBounds(165,84,54,23);

lblPHAge.setBounds(new Rectangle(43, 134, 66, 26));

PHAge.setBounds(new Rectangle(115, 135, 210, 33));

lblPHone.setBounds(new Rectangle(43, 182, 66, 26));

PHone.setBounds(new Rectangle(115, 183, 210, 33));

lblPHAddress.setBounds(new Rectangle(43, 230, 66, 26));

PHAddress.setBounds(new Rectangle(115, 231, 210, 33));

lblPHSearch.setBounds(new Rectangle(43, 278, 66, 26));

PHSearch.setBounds(new Rectangle(115, 278, 210, 33));

jbtAdd.setBounds(new Rectangle(340, 38, 86, 30));

jbtUpdate.setBounds(new Rectangle(340, 86, 86, 30));

jbtSave.setBounds(new Rectangle(340, 134, 86, 30));

jbtSearch.setBounds(new Rectangle(340, 278, 86, 30));

jbtdelete.setBounds(new Rectangle(340, 326, 86, 30));

jbtBack.setBounds(new Rectangle(340, 374, 86, 30));

lblPHInfo.setBounds(43, 326, 66, 26);

PHInfo.setBounds(new Rectangle(115, 326, 210, 86));

this.add(lblPHName); // 把各个组件添加到框架窗口中

this.add(PHName);

this.add(lblPHSex);

bg.add(PHSex1);

bg.add(PHSex2);

this.add(PHSex1);

this.add(PHSex2);

this.add(lblPHAge);

this.add(PHAge);

this.add(lblPHone);

this.add(PHone);

this.add(lblPHAddress);

this.add(PHAddress);

this.add(jbtAdd);

this.add(jbtSearch);

this.add(jbtUpdate);

this.add(jbtdelete);

this.add(lblPHSearch);

this.add(PHSearch);

this.add(lblPHInfo);

this.add(PHInfo);

this.add(jbtSave);

this.add(jbtBack);

jbtAdd.addActionListener(this);

jbtSearch.addActionListener(this);

jbtUpdate.addActionListener(this);

jbtdelete.addActionListener(this);

jbtSave.addActionListener(this);

jbtBack.addActionListener(this);

}

public Vector<String> getInfo()// 从信息面板得到用户输入的信息

{

pInfo.add(PHName.getText().trim());

pInfo.add(PHAge.getText().trim());

String gender = PHSex1.isSelected() ? "男" : "女";

pInfo.add(gender);// 添加性别

pInfo.add(PHone.getText().trim());

pInfo.add(PHAddress.getText().trim());

pInfo.add(PHSearch.getText().trim());

return pInfo;

}

public void actionPerformed(ActionEvent e) {// 按钮注册事件的实现

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

monitorAddButton();

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

monitorSearchButton();

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

monitorUpdateButton();

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

monitorDelButton();

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

monitorSaveButton();

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

monitorBackButton();

}

}

public void clearInfo()// 清空信息面板

{

PHName.setText("");// 清空文本框

PHAge.setText("");

PHone.setText("");

bg.clearSelection();

PHAddress.setText("");

}

private void monitorBackButton() {

// TODO Auto-generated method stub

Game1 G=new Game1();

}

public void monitorAddButton()// 监听新增按钮的方法

{

String str=JOptionPane.showInputDialog("输入你想完善的用户姓名:");

if (str==null) {

int n=JOptionPane.showConfirmDialog(this, "确定取消？", "警告", JOptionPane.YES\_NO\_OPTION);

if(n==1)

{

str=JOptionPane.showInputDialog("输入你想修改的用户姓名:");

try {

Connection con=connect();

String sql1="select \* from xiangxi where name=?";

PreparedStatement ptmt=con.prepareStatement(sql1);

ptmt.setString(1,str);

ResultSet rs=ptmt.executeQuery();

while(rs.next())

{

id=rs.getInt(1);

PHName.setText(rs.getString(2));

if(rs.getString(3)!=null)

{

String sex=rs.getString(3).trim();

if(sex=="男")

PHSex1.setSelected(true);

else

PHSex2.setSelected(true);

PHAge.setText(rs.getString(4));

PHone.setText(rs.getString(5));

PHAddress.setText(rs.getString(6));

}

}

}

catch(SQLException e)

{

e.printStackTrace();

}

}

}

else

{

if (str.equals("")) {

JOptionPane.showMessageDialog(this, "查找条件不能为空！！！", "哎呀，出错了", JOptionPane.WARNING\_MESSAGE);

}

else

{

try {

Connection con=connect();

String sql1="select \* from xiangxi where name=?";

PreparedStatement ptmt=con.prepareStatement(sql1);

ptmt.setString(1,str);

ResultSet rs=ptmt.executeQuery();

while(rs.next())

{

id=rs.getInt(1);

PHName.setText(rs.getString(2));

if(rs.getString(3)!=null)

{

String sex=rs.getString(3).trim();

if(sex=="男")

PHSex1.setSelected(true);

else

PHSex2.setSelected(true);

PHAge.setText(rs.getString(4));

PHone.setText(rs.getString(5));

PHAddress.setText(rs.getString(6));

}

}

}

catch(SQLException e)

{

e.printStackTrace();

}

}

}

};

public void monitorSearchButton() {// 监听搜索按钮的方法

String search = PHSearch.getText().trim();// 获得搜索框内容

if (search.equals("")) {

JOptionPane.showMessageDialog(this, "查找条件不能为空！！！", "哎呀，出错了", JOptionPane.WARNING\_MESSAGE);

}

else

{

try {

Connection con=connect();

String sql1="select \* from xiangxi where name=?";

PreparedStatement ptmt=con.prepareStatement(sql1);

ptmt.setString(1,search);

ResultSet rs=ptmt.executeQuery();

while(rs.next())

{

String id1=rs.getString(1);

String Name=rs.getString(2);

String Sex=rs.getString(3);

String Age=rs.getString(4);

String Phone=rs.getString(5);

String Address=rs.getString(6);

String str="ID:"+id1+"姓名："+Name+"性别："+Sex+"年龄："+Age+"电话："+Phone+"地址："+Address ;

PHInfo.setText(str);

}

ptmt.close();

/\*con.close();\*/

}

catch(SQLException e)

{

e.printStackTrace();

JOptionPane.showMessageDialog(this, "用户不存在！", "哎呀，出错了", JOptionPane.WARNING\_MESSAGE);

}

}

};

private void monitorUpdateButton() {

// TODO Auto-generated method stub

String str=JOptionPane.showInputDialog("输入你想修改的用户姓名:");

if (str==null) {

int n=JOptionPane.showConfirmDialog(this, "确定取消？", "警告", JOptionPane.YES\_NO\_OPTION);

if(n==1)

{

str=JOptionPane.showInputDialog("输入你想修改的用户姓名:");

try {

Connection con=connect();

String sql1="select \* from xiangxi where name=?";

PreparedStatement ptmt=con.prepareStatement(sql1);

ptmt.setString(1,str);

ResultSet rs=ptmt.executeQuery();

while(rs.next())

{

id=rs.getInt(1);

PHName.setText(rs.getString(2));

if(rs.getString(3)!=null)

{

String sex=rs.getString(3).trim();

if(sex=="男")

PHSex1.setSelected(true);

else

PHSex2.setSelected(true);

PHAge.setText(rs.getString(4));

PHone.setText(rs.getString(5));

PHAddress.setText(rs.getString(6));

}

}

}

catch(SQLException e)

{

e.printStackTrace();

}

}

}

else

{

if (str.equals("")) {

JOptionPane.showMessageDialog(this, "查找条件不能为空！！！", "哎呀，出错了", JOptionPane.WARNING\_MESSAGE);

}

else

{

try {

Connection con=connect();

String sql1="select \* from xiangxi where name=?";

PreparedStatement ptmt=con.prepareStatement(sql1);

ptmt.setString(1,str);

ResultSet rs=ptmt.executeQuery();

while(rs.next())

{

id=rs.getInt(1);

PHName.setText(rs.getString(2));

if(rs.getString(3)!=null)

{

String sex=rs.getString(3).trim();

if(sex=="男")

PHSex1.setSelected(true);

else

PHSex2.setSelected(true);

PHAge.setText(rs.getString(4));

PHone.setText(rs.getString(5));

PHAddress.setText(rs.getString(6));

}

}

}

catch(SQLException e)

{

e.printStackTrace();

}

}

}

}

private void monitorDelButton() {

try

{

Connection con1=connect();

String str=JOptionPane.showInputDialog("请输入你要删除的姓名:");

if (str==null)

{

int n=JOptionPane.showConfirmDialog(this, "确定取消？", "警告", JOptionPane.YES\_NO\_OPTION);

if(n==1)

{

str=JOptionPane.showInputDialog("请输入你要删除的姓名:");

}

int x=JOptionPane.showConfirmDialog(this, "确定删除？", "警告", JOptionPane.YES\_NO\_OPTION);

if(x==0)

{

String sql="delete from xiangxi where name=?";

PreparedStatement ptmt1=con1.prepareStatement(sql);

ptmt1.setString(1, str);

int m= ptmt1.executeUpdate();

if(m>0)

{

JOptionPane.showMessageDialog(this, "删除成功");

}

ptmt1.close();

con1.close();

}

}

else

{

if (str.equals("")) {

JOptionPane.showMessageDialog(this, "查找条件不能为空！！！", "哎呀，出错了", JOptionPane.WARNING\_MESSAGE);

}

else

{

int x=JOptionPane.showConfirmDialog(this, "确定删除？", "警告", JOptionPane.YES\_NO\_OPTION);

if(x==0)

{

String sql="delete from xiangxi where name=?";

String sql2="delete from userlist where name=?";

PreparedStatement ptmt1=con1.prepareStatement(sql);

PreparedStatement ptmt2=con1.prepareStatement(sql2);

ptmt1.setString(1, str);

ptmt2.setString(1, str);

int m1= ptmt1.executeUpdate();

int m2= ptmt2.executeUpdate();

if(m1>0&&m2>0)

{

JOptionPane.showMessageDialog(this, "删除成功");

}

ptmt1.close();

con1.close();

}

}

}

}

catch(SQLException e)

{

e.printStackTrace();

}

}

private void monitorSaveButton() {

// TODO Auto-generated method stub

try {

Connection con=connect();

String name = PHName.getText().trim();

String sex = PHSex1.isSelected() ? "男" : "女";

String age = PHAge.getText().trim();

String phone = PHone.getText().trim();

String address = PHAddress.getText().trim();

String sql="update xiangxi set name=?,sex=?,age=?,phone=?,address=? where id=?";

String sql2="update uselist set name=? where id=?";

PreparedStatement ptmt=con.prepareStatement(sql);

PreparedStatement ptmt2=con.prepareStatement(sql2);

ptmt.setString(1, name);

ptmt.setString(2,sex);

ptmt.setString(3,age);

ptmt.setString(4,phone);

ptmt.setString(5,address);

ptmt.setInt(6,id);

ptmt2.setString(1, name);

ptmt2.setInt(2,id);

int m= ptmt.executeUpdate();

if(m>0)

{

clearInfo();

JOptionPane.showMessageDialog(this, "保存成功！！");

}

ptmt.close();

}

catch(SQLException e)

{

e.printStackTrace();

}

}

public static void main(String[] args) {

index frm = new index();

// 设置框架窗体的事件监听(关闭窗体事件)

frm.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

frm.pack();

frm.setVisible(true);// 显示框架窗体

}

public Dimension getPreferredSize()// 设置框架窗口的大小

{

return new Dimension(450,450);

}

}

## 欢迎页面的实现welcome.java

**package Graphic;**

import java.awt.Button;

import java.awt.Dimension;

import java.awt.Frame;

import java.awt.Label;

import java.awt.Rectangle;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

public class Welcome extends Frame implements ActionListener{

Label lblDengLu = new Label("欢迎来到娱乐空间：");

Button btnLogin = new Button("登录");

public Welcome()

{

this.setLayout(null); // 手工设置各个组件的位置和大小

lblDengLu.setBounds(new Rectangle(43, 86, 150, 26));

btnLogin.setBounds(new Rectangle(220, 84,66 , 33));

this.add(lblDengLu);

this.add(btnLogin);

btnLogin.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

Login frmLogin = new Login();

// 设置框架窗体的事件监听(关闭窗体事件)

frmLogin.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

frmLogin.pack();

frmLogin.setVisible(true);// 显示框架窗体

}

public static void main(String[] args) {

Welcome frmLogin = new Welcome();

// 设置框架窗体的事件监听(关闭窗体事件)

frmLogin.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

frmLogin.pack();

frmLogin.setVisible(true);// 显示框架窗体

}

public Dimension getPreferredSize()// 设置框架窗口的大小

{

return new Dimension(355, 204);

}

}

## 主函数的实现Main.java

package Graphic;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

public class Main {

public static void main(String[] args) {

Welcome frmLogin = new Welcome();

// 设置框架窗体的事件监听(关闭窗体事件)

frmLogin.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

frmLogin.pack();

frmLogin.setVisible(true);// 显示框架窗体

}

}

## （5）九宫格及拼图游戏实现。

拼图.java

package 拼图;

import java.awt.\*;

import java.awt.event.\*;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import javax.swing.\*;

public class 拼图 extends JFrame implements ActionListener,KeyListener{

int number\_path=0; //记录步数

int best\_recond=0; //最佳成绩

int replay=0;

int[] num={1,2,3,4,5,6,7,8,0};

String str="已走步数：";

String str1="最佳成绩：";

static JButton[][] Buttons=new JButton[3][3]; //九宫格

static JButton[][] Buttons\_valid=new JButton[3][3];

public 拼图(){

super("拼图");

this.setBounds(400,100,360,360);

this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

this.setLayout(new GridLayout(3,3));

this.init();

this.setVisible(true);

}

public void init(){ //初始化使用算法使其随机出现，并填加格子以及事件监听和键盘监听

for(int i = 0;i < 8;i++)

{

this.num[i] = i+1;

}

this.num[8] = -1;//不能出现数字9

this.num[5] = -1;//为空的那一格的初始位置。

this.num[8] = 6;//数字9的空位正好补充因填充-1而覆盖的数字。

for(int i = 0;i < 8;i++)

{

int idx =(int) (Math.random() \* 8)+1;

int tmp = this.num[7];

this.num[7] = this.num[idx];

this.num[idx] = tmp;

}

/\* for(int i = 0;i < 3;i++)

{

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

{

int temp=i\*3+j;

Game2.Buttons\_valid[i][j] = new JButton();

Game2.Buttons\_valid[i][j].setIcon(new ImageIcon("src/拼图/img/boy"+temp+".png"));

}

}\*/

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

{

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

{

if(this.num[i \* 3 + j] != -1)

{

拼图.Buttons[i][j] = new JButton();

拼图.Buttons[i][j].setIcon(new ImageIcon("src/拼图/img/boy"+this.num[i \* 3 + j]+".png"));

拼图.Buttons[i][j].addActionListener(this);

拼图.Buttons[i][j].addKeyListener(this);

this.getContentPane().add(拼图.Buttons[i][j]);

}

else

{

拼图.Buttons[i][j] = new JButton();

拼图.Buttons[i][j].addActionListener(this);

拼图.Buttons[i][j].addKeyListener(this);

this.getContentPane().add(拼图.Buttons[i][j]);

拼图.Buttons[i][j].setEnabled(false);

}

}

}

}

@Override

public void actionPerformed(ActionEvent e) { //鼠标事件监听，交换格子内容步数加1

int i,j;

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

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

if(e.getSource()==Buttons[i][j]){

if(i+1!=3&&Buttons[i+1][j].getIcon()==null){//不是第三行的话

Buttons[i+1][j].setIcon(Buttons[i][j].getIcon());

Buttons[i+1][j].setEnabled(true);

Buttons[i][j].setIcon(null);

Buttons[i][j].setEnabled(false);

number\_path++;

}

if(i-1!=-1&&Buttons[i-1][j].getIcon()==null){//不是第一行的话

Buttons[i-1][j].setIcon(Buttons[i][j].getIcon());

Buttons[i-1][j].setEnabled(true);

Buttons[i][j].setIcon(null);

Buttons[i][j].setEnabled(false);

number\_path++;

}

if(j+1!=3&&Buttons[i][j+1].getIcon()==null){//不是第三列的话

Buttons[i][j+1].setIcon(Buttons[i][j].getIcon());

Buttons[i][j+1].setEnabled(true);

Buttons[i][j].setIcon(null);

Buttons[i][j].setEnabled(false);

number\_path++;

}

if(j-1!=-1&&Buttons[i][j-1].getIcon()==null){//不是第一列的话

Buttons[i][j-1].setIcon(Buttons[i][j].getIcon());

Buttons[i][j-1].setEnabled(true);

Buttons[i][j].setIcon(null);

Buttons[i][j].setEnabled(false);

number\_path++;

}

}

}

}

/\* if(winfail()){ //判断这次操作后是否胜利，读出最佳成绩，并判断是否打破记录

Isrecord();

}\*/

}

public void keyPressed(KeyEvent e){ //键盘事件监听，交换格子内容步数加1

int i,j;

boolean b=false;

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

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

if(Buttons[i][j].getIcon()==null){

if(i+1!=3&&e.getKeyCode()==KeyEvent.VK\_UP){

Buttons[i][j].setEnabled(true);

Buttons[i][j].setIcon(Buttons[i+1][j].getIcon());

Buttons[i+1][j].setIcon(null);

Buttons[i+1][j].setEnabled(false);

number\_path++;

b=true;break;

}

if(i-1!=-1&&e.getKeyCode()==KeyEvent.VK\_DOWN){

Buttons[i][j].setEnabled(true);

Buttons[i][j].setIcon(Buttons[i-1][j].getIcon());

Buttons[i-1][j].setIcon(null);

Buttons[i-1][j].setEnabled(false);

number\_path++;

b=true;break;

}

if(j+1!=3&&e.getKeyCode()==KeyEvent.VK\_LEFT){

Buttons[i][j].setEnabled(true);

Buttons[i][j].setIcon(Buttons[i][j+1].getIcon());

Buttons[i][j+1].setIcon(null);

Buttons[i][j+1].setEnabled(false);

number\_path++;

b=true;break;

}

if(j-1!=-1&&e.getKeyCode()==KeyEvent.VK\_RIGHT){

Buttons[i][j].setEnabled(true);

Buttons[i][j].setIcon(Buttons[i][j-1].getIcon());

Buttons[i][j-1].setIcon(null);

Buttons[i][j-1].setEnabled(false);

number\_path++;

b=true;break;

}

}

}

if(b) break;

}

/\* if(winfail()){ //判断这次操作后是否胜利，读出最佳成绩，并判断是否打破记录

Isrecord();

}\*/

}

public void Isrecord()

{

try {

FileInputStream fin=new FileInputStream("D:\\bestrecond.int");

DataInputStream din=new DataInputStream(fin);

best\_recond=din.readInt();

din.close();

fin.close();

} catch (FileNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

if(number\_path>best\_recond){

JOptionPane.showConfirmDialog(this, "你太厉害了！！这么难都能赢！！！才用了"+Integer.toString(number\_path)+"步！！然而并没有打破记录。历史最佳成绩是："+Integer.toString(best\_recond));

if(replay==JOptionPane.YES\_OPTION){

拼图 G=new 拼图();

}

}

else{

JOptionPane.showConfirmDialog(this, "你太厉害了！！这么难都能赢！！！才用了"+Integer.toString(number\_path)+"步！！打破了历史记录！！历史最佳成绩是："+Integer.toString(best\_recond));

try {

FileOutputStream fin=new FileOutputStream("D:\\bestrecond.int");

DataOutputStream din=new DataOutputStream(fin);

din.writeInt(number\_path);

din.close();

fin.close();

} catch (FileNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

if(replay==JOptionPane.YES\_OPTION){

拼图 G=new 拼图();

}

}

}

/\* public static boolean winfail(){ //判断输赢函数，每个格的字符都符合即可

int i,j,k=1,n=0;

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

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

if(Buttons[i][j].getIcon().equals(Buttons\_valid[i][j].getIcon())){

n++;

}

}

}

if(n>=8) return true;

return false;

}\*/

public static void main(String[] args) {

拼图 G=new 拼图();

}

@Override

public void keyReleased(KeyEvent arg0) {

// TODO Auto-generated method stub

}

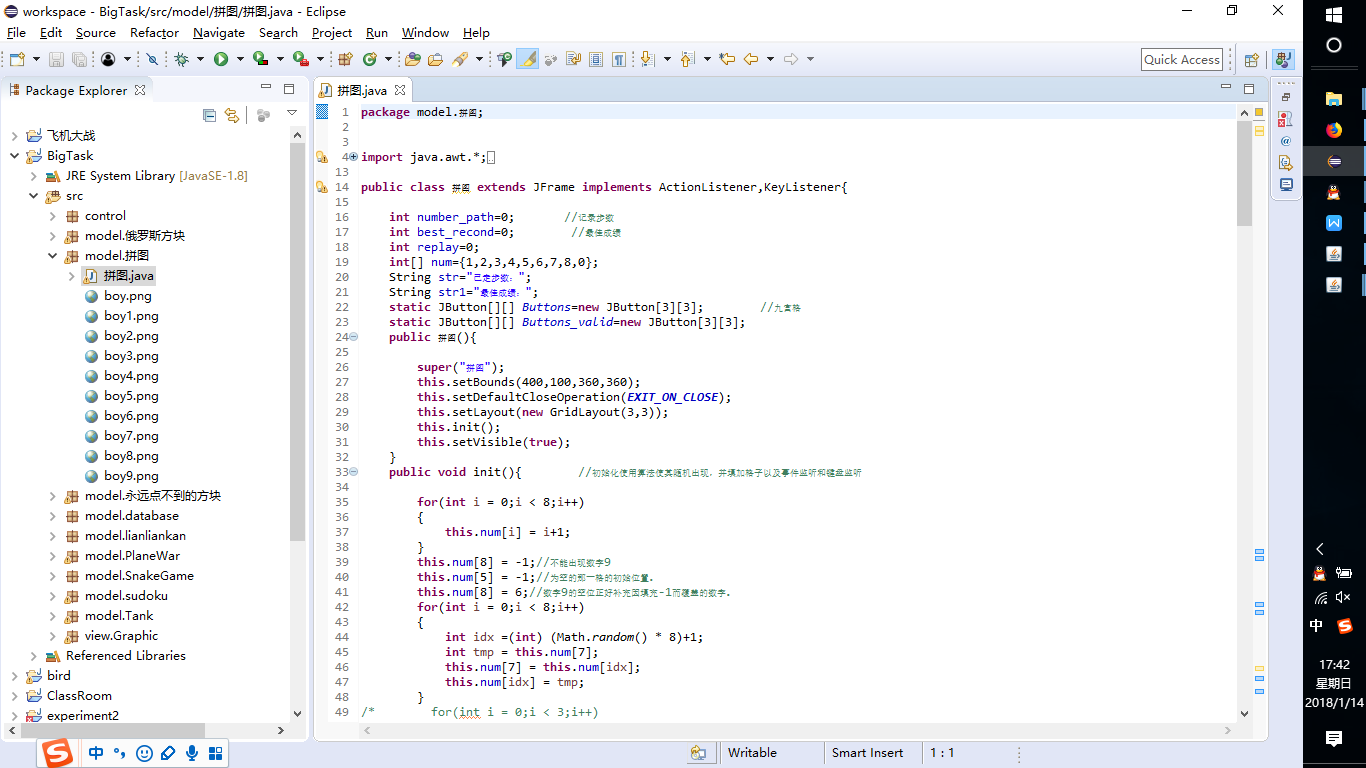
@Override

public void keyTyped(KeyEvent arg0) {

// TODO Auto-generated method stub

}

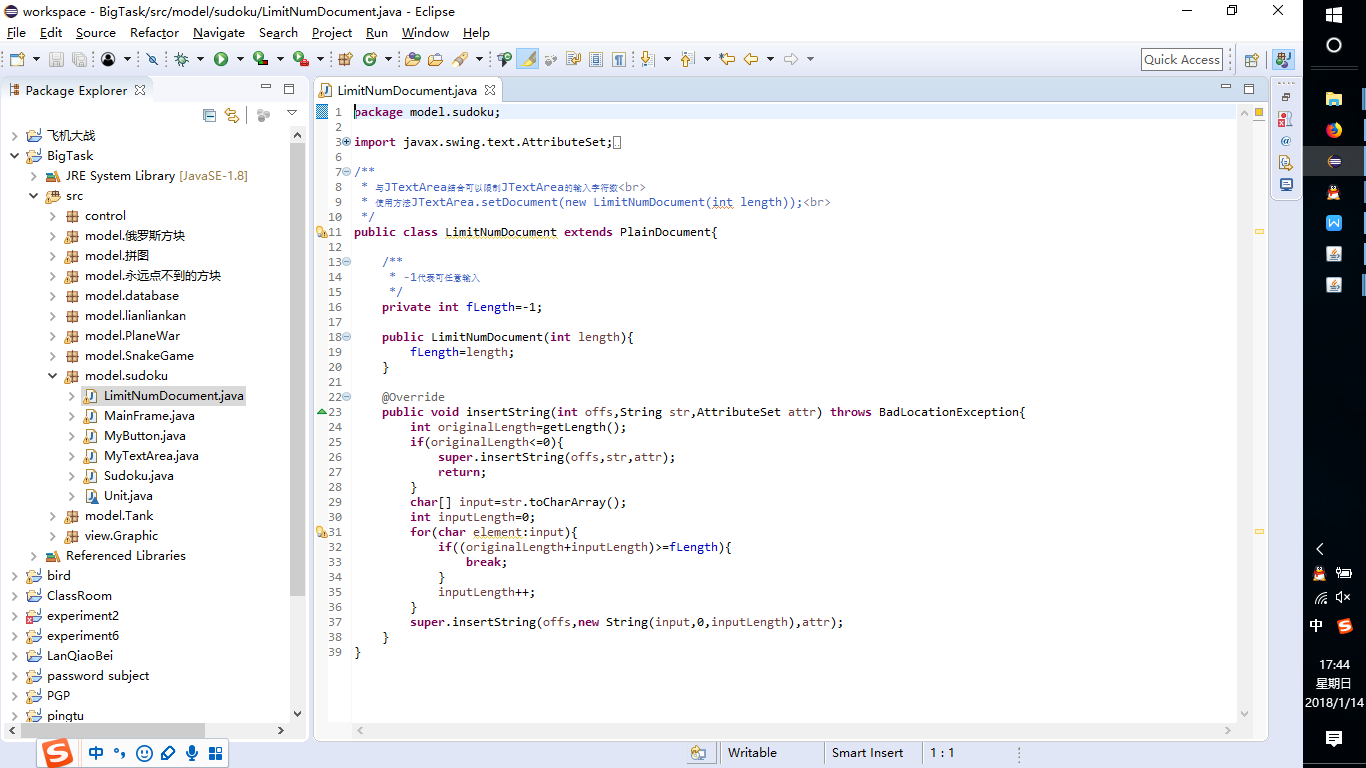
}



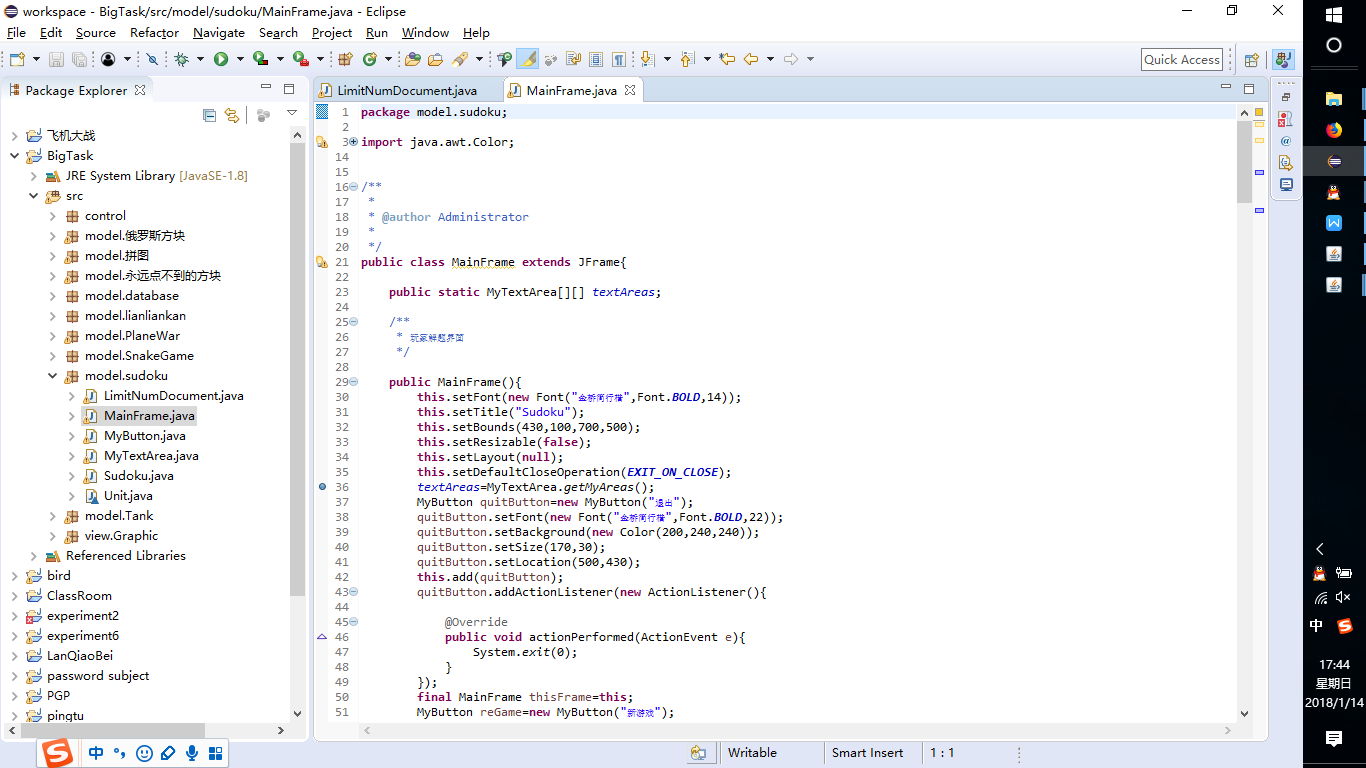
## 数独游戏实现

因代码太长，所以附上每一个java文件的截图：

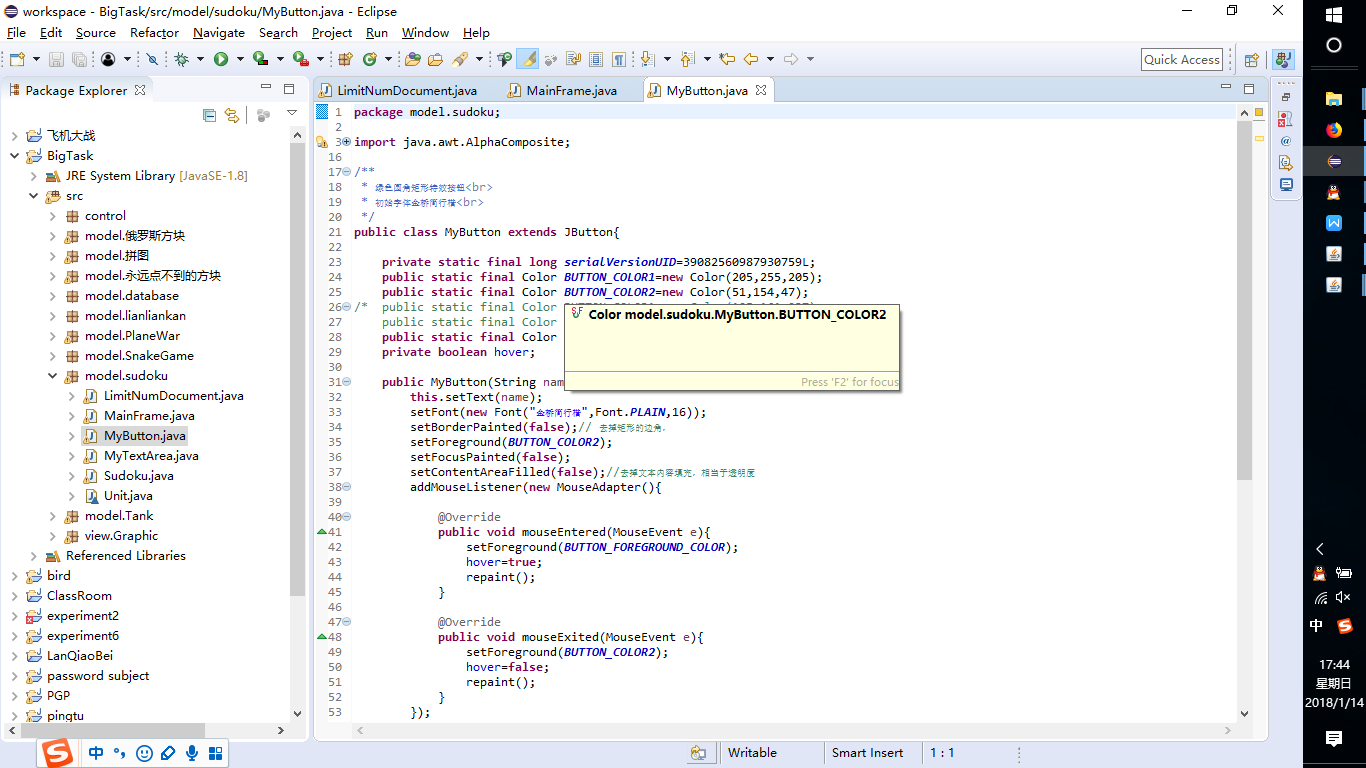
**LimitNumDocument.java**



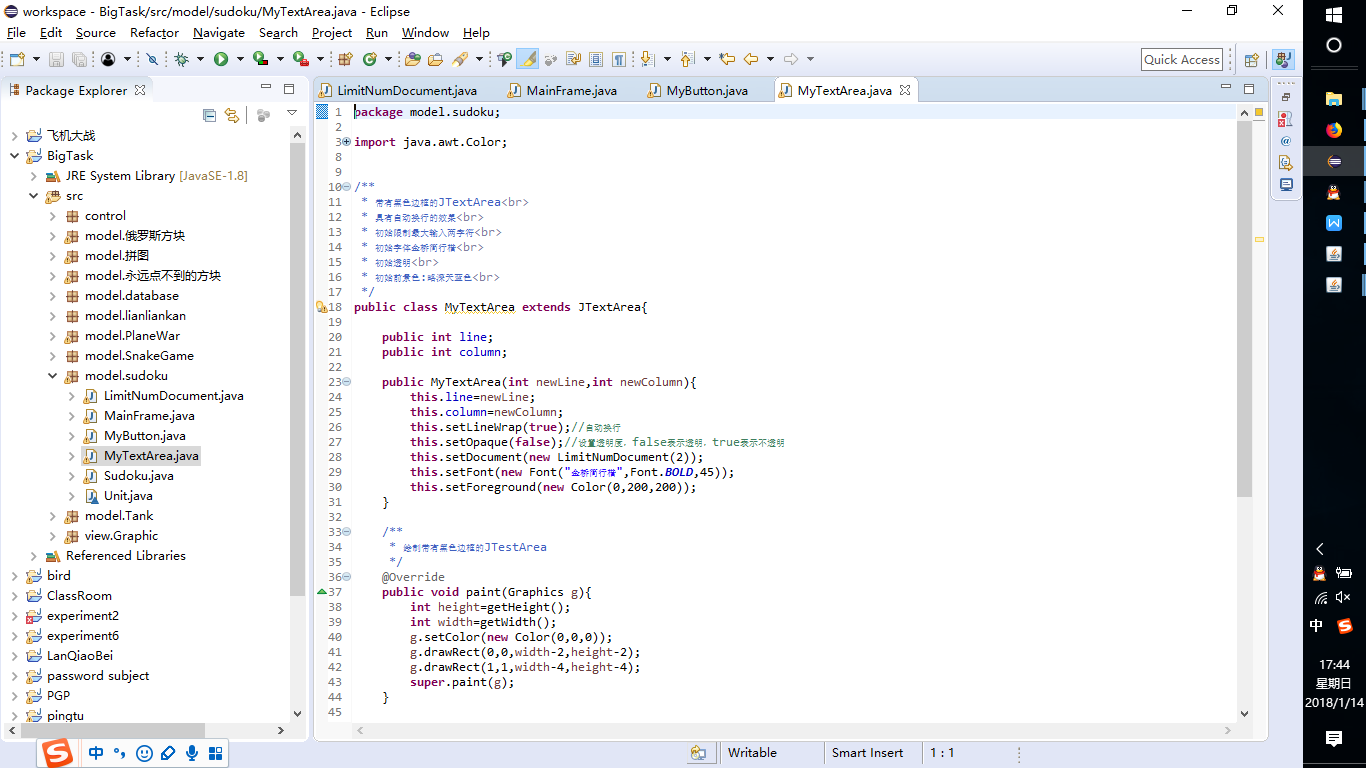
**MainFrame.java**



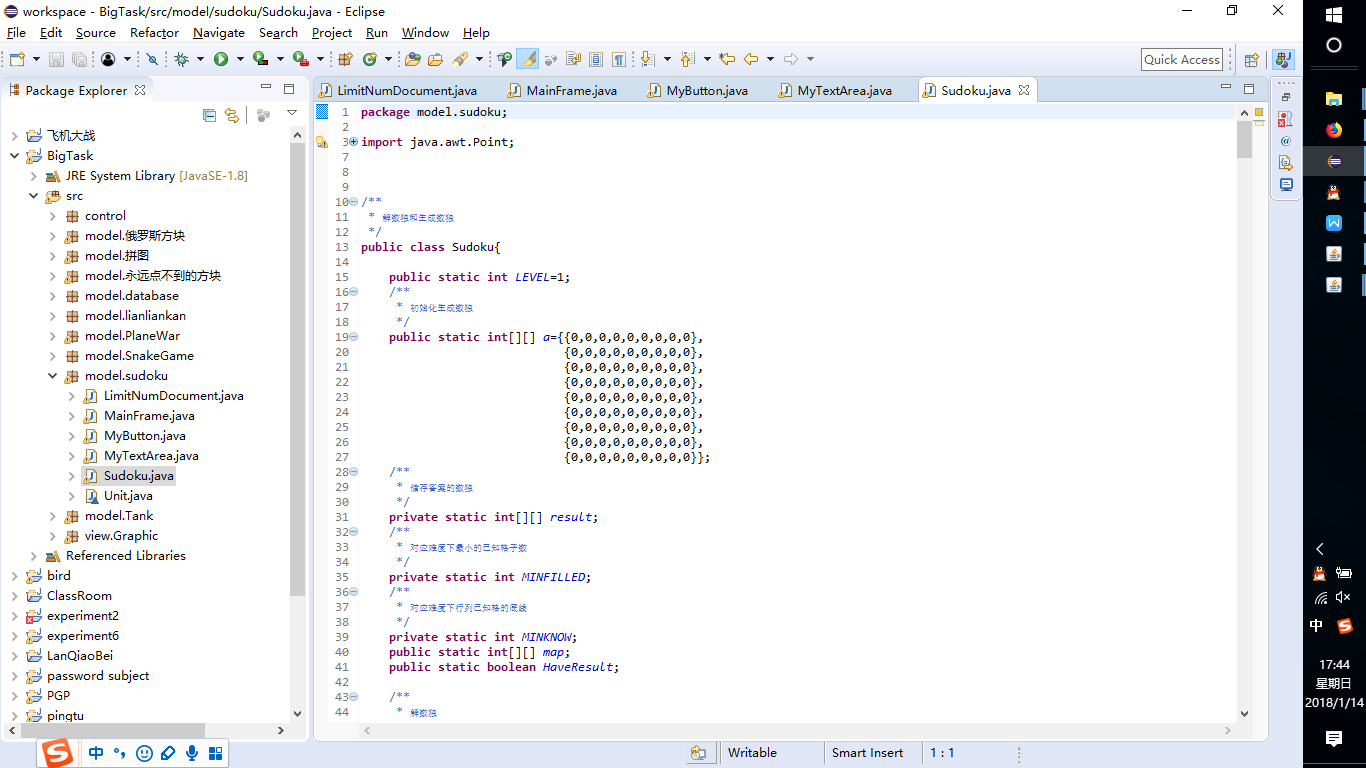
**MyButton.java**



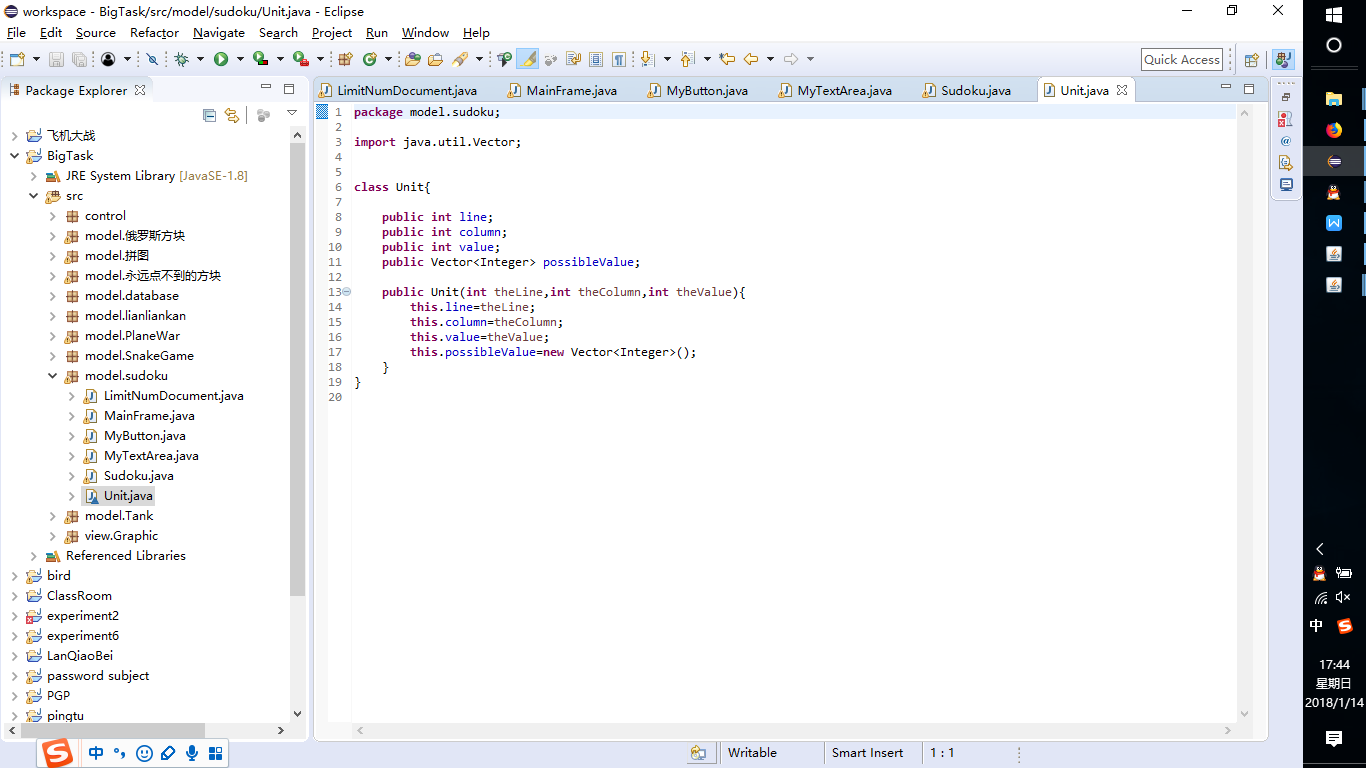
**MyTextArea.java**



**Sudoku.java**

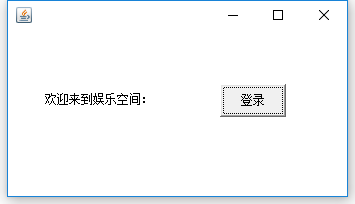


**Unit.java**



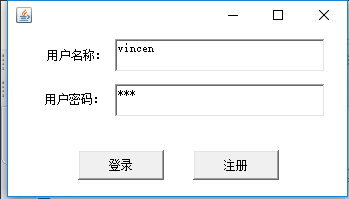
# 软件测试

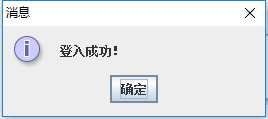
## 欢迎页面的实现

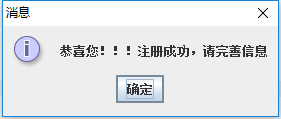


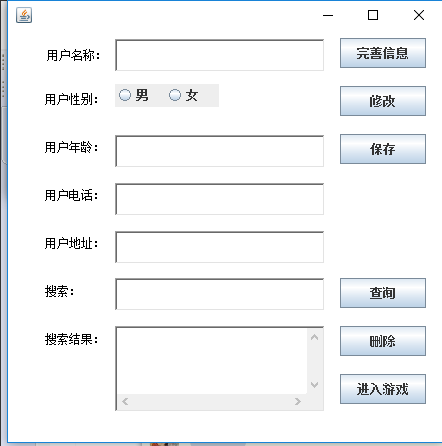
## 注册和登陆功能的实现

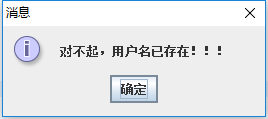
**登录和注册**

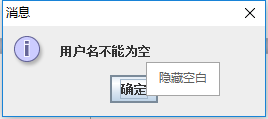




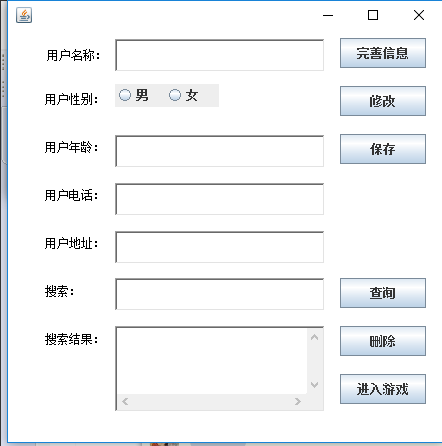






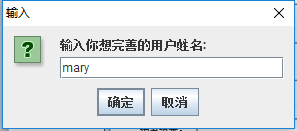


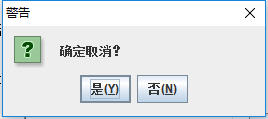
**登录成功**

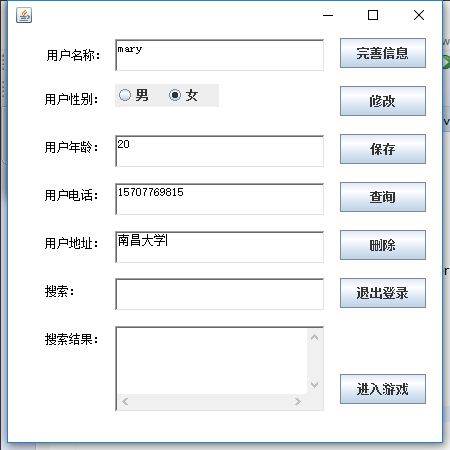
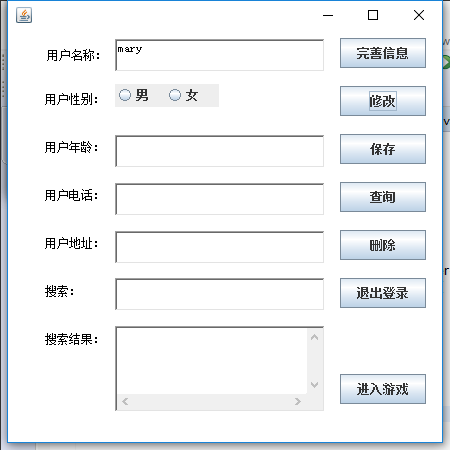


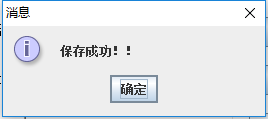
## 主页面及数据库页面的实现

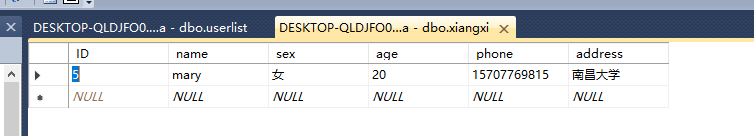
**完善信息**



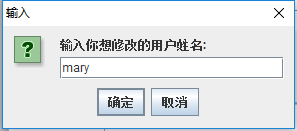


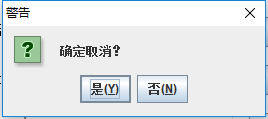


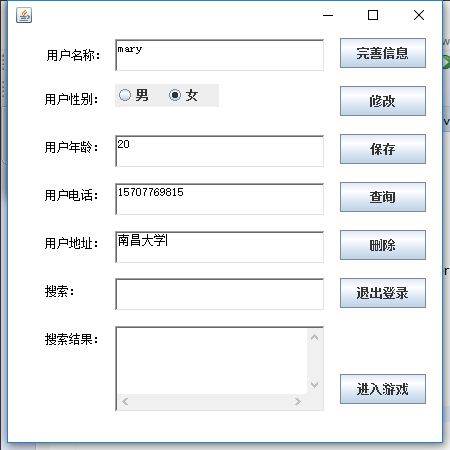


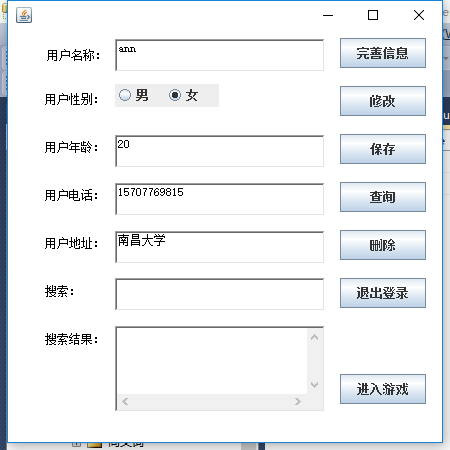


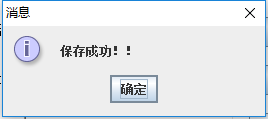
**修改信息**

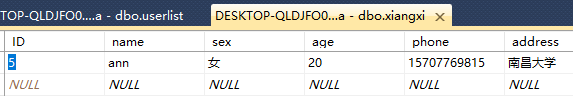


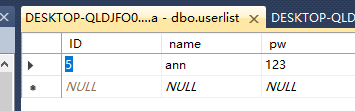




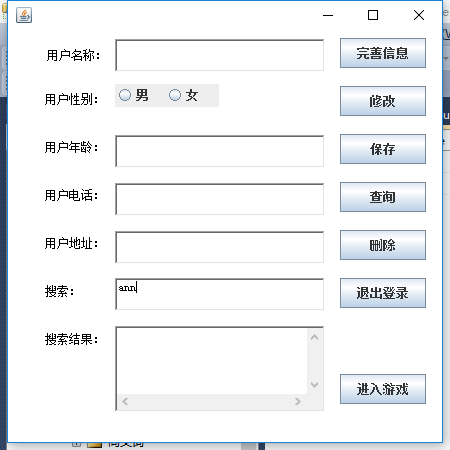


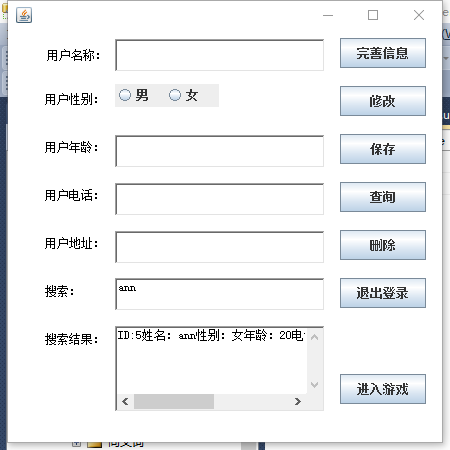




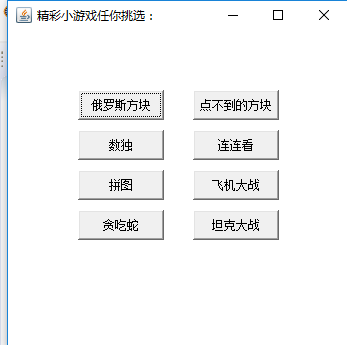


**查询信息**

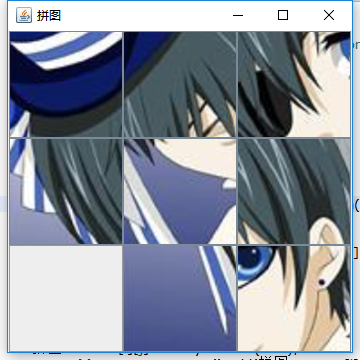




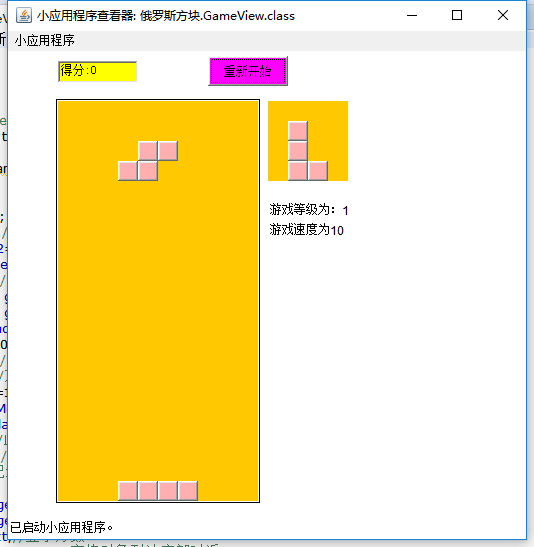
## 游戏选择页面的实现



## 拼图页面



## 俄罗斯方块页面



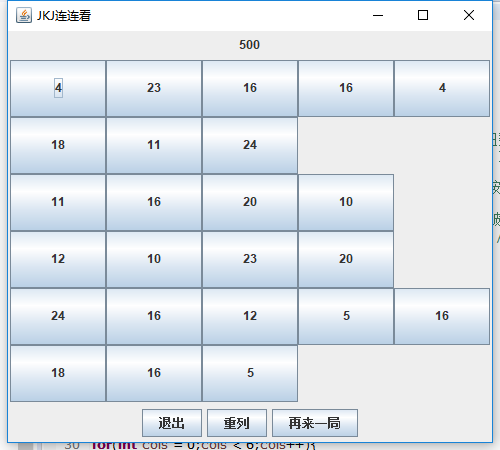
## 数独页面



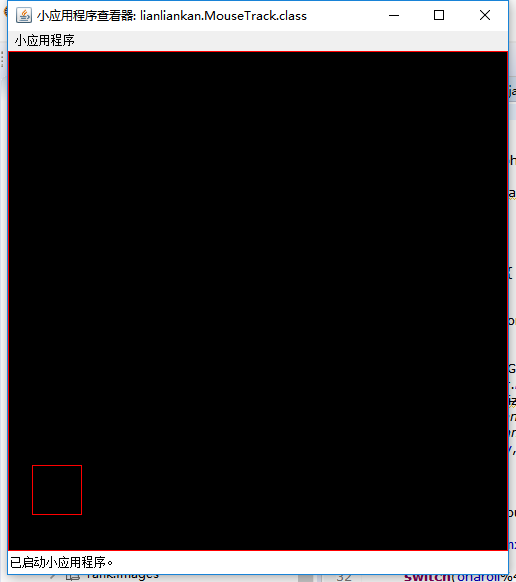




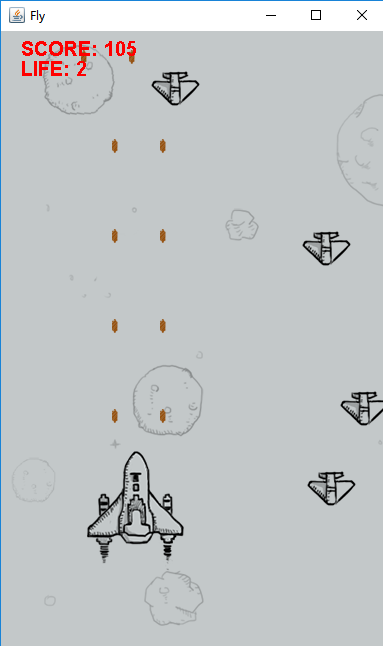
## 连连看页面



## 永远点不到的方框页面



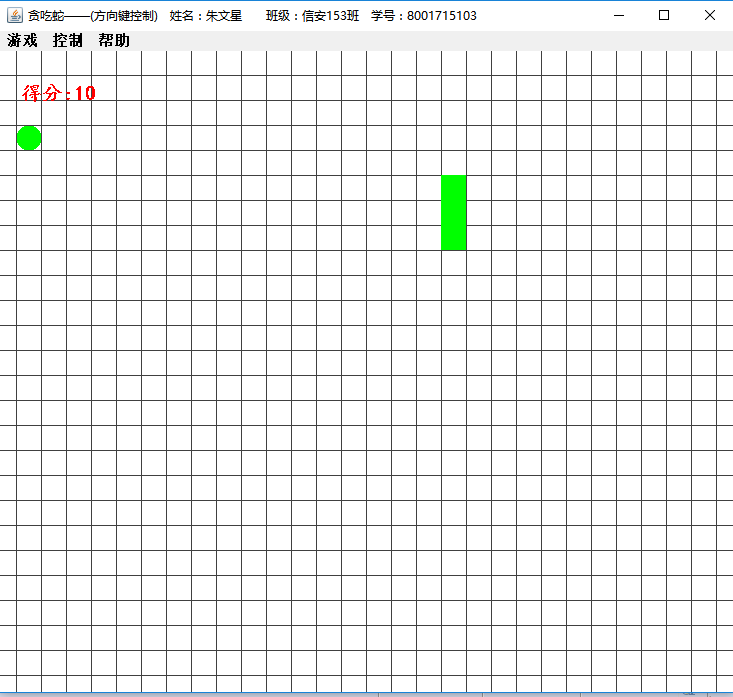
## 飞机大战页面

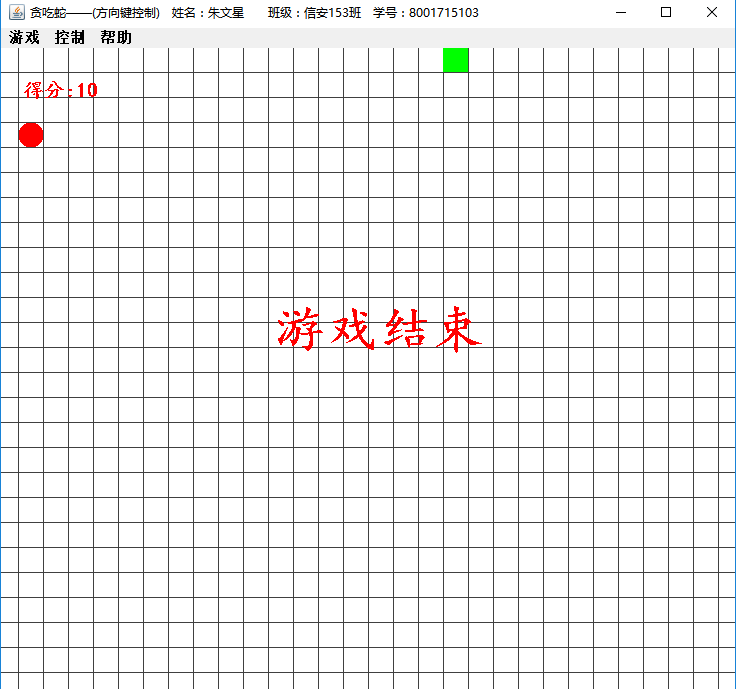






## 贪吃蛇页面





## 坦克大战页面



# 总结与体会、意见和建议

**在这次大作业过程中，可以说我的体会是很多的，刚开始的时候我只是想简单的实现一个数据库的增删改查功能，可能是因为当时还没学到数据库知识，就觉得自己做的挺厉害了，但是慢慢的我开始不满足于自己的作品，我开始追求更多的用户体验，追求完美，所以我就在想着改进自己的代码，有时候一觉醒来就会想着该如何能拿到高分，如何让自己满意，所以我的这个简易小游戏模型就开始出来，我保留了刚开始的用户登录和用户注册功能，加上了另外一张详细信息表，注册时可以只注册用户名和密码，但是注册完成之后可以进行详细信息填写，这更加的人性化，也更符合我们现实生活中手机和网页注册，并且实现了两张表之间id的关联，当修改信息删除信息时，同时会影响两张表，而不是一张，这让我感到颇为有成就感，随后我就想着做几个小游戏，我记得我平时逛论坛的时候总会有一些小游戏，然后我喜欢把他们记录下来，这也是我做这个简易小游戏开发的灵感之一，但是说句实话，有的游戏实在太难懂，而且每个作者的思维方式又千奇百怪，有些游戏不是很了解，这个时候我就想着自己能够做出一个比较小的游戏出来，所以拼图游戏应运而生，我之前就做过一个九宫格方块移动的小游戏，而网上却散布着大量的拼图游戏编程，而且还涉及到了图片的切割函数，当我第一眼看到他们如此复杂的游戏代码时，我就感觉太过于繁琐，所以我就想着图片切割函数为什么就不能用软件实现呢，再将图片贴到按钮上，岂不是更简单，所以我就花费了一个上午进行了大胆的尝试，我用photoScape这个软件将图片切割成九份，舍去第九张，留下八张组成拼图，终于让我成功了，虽然还没有拼成功过，但是还是很有成就感的，随后我将其他几个游戏用图形化按纽的方式，进行了事件处理，其中的俄罗斯方块和点不到的方块游戏是用java Applet写的，所以我将其进行了改进，让其具有了main方法，是的其也能在java application下运行，通过此次学习，我想这让我为以后的职业有了更广泛的选择，也让我对java产生了更加浓厚的兴趣，所以当别人报名参加蓝桥杯的c语言组的时候，我依然决然的报了java组。我想这就是我最大的体会和收获吧。**

**追加：第一次答辩时老师说我虽然功能都有一点，但是都比较浅显，没有深入，所以第二次答辩时，我专注于游戏的实现及更好地用户体验，所以我做出来数独这个游戏，增强了用户体验，也使结构更加完整，使用户能体会到游戏的乐趣，望老师查看。**