/\*

\* If Nimbus (introduced in Java SE 6) is not available, stay with the

\* default look and feel. For details see

\* http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(forgetpassword.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(forgetpassword.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(forgetpassword.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(forgetpassword.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\*

\* Create and display the form

\*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

forgetpassword forgetpassword = new forgetpassword();

forgetpassword.setLocationRelativeTo(null);

forgetpassword.setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton Cancel;

private javax.swing.JTextField J1;

private javax.swing.JTextField J3;

private javax.swing.JButton J4;

private javax.swing.JButton J5;

private javax.swing.JButton J6;

private javax.swing.JButton Submit;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JPanel jPanel4;

private javax.swing.JTextField jTYourPN;

private javax.swing.JTextField jTYourYN;

// End of variables declaration

}

**Register Module**

apackage Login\_in;

import java.awt.Color;

import java.awt.HeadlessException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import java.util.regex.Pattern;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author 皮皮

\*/

public class register extends javax.swing.JFrame {

Pattern use = Pattern.compile("^[A-Za-z\_]+[A-Za-z]\\d{0,7}+$");

Pattern passw = Pattern.compile("^(?![0-9]+$)(?![a-zA-Z]+$)[0-9A-Za-z]{8,16}$");

Pattern email = Pattern.compile("^(\\w)+(\\.\\w+)\*@(\\w)+((\\.\\w{2,3}){1,3})$");

Pattern phone = Pattern.compile("^((13[0-9])|(15[^4,\\D])|(18[0,5-9]))\\d{8}$");

Connection con = null;

PreparedStatement pst;

ResultSet rs, rsp, rsu;

public register() {

initComponents();

JL11.setVisible(false);

JL12.setVisible(false);

JL21.setVisible(false);

JL22.setVisible(false);

JL31.setVisible(false);

JL32.setVisible(false);

JL42.setVisible(false);

JL52.setVisible(false);

JT2.setEnabled(false);

JP1.setEnabled(false);

JP2.setEnabled(false);

JT3.setEnabled(false);

setResizable(false);

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel2 = new javax.swing.JPanel();

jPanel1 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

JT2 = new javax.swing.JTextField();

JT1 = new javax.swing.JTextField();

JT3 = new javax.swing.JTextField();

JP1 = new javax.swing.JPasswordField();

JP2 = new javax.swing.JPasswordField();

Cancel = new javax.swing.JButton();

Submit = new javax.swing.JButton();

JL11 = new javax.swing.JLabel();

JL21 = new javax.swing.JLabel();

JL31 = new javax.swing.JLabel();

JL12 = new javax.swing.JLabel();

JL22 = new javax.swing.JLabel();

JL32 = new javax.swing.JLabel();

JL42 = new javax.swing.JLabel();

JL52 = new javax.swing.JLabel();

Cancel1 = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setTitle("Register");

jPanel2.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jPanel1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

jPanel1.setMaximumSize(new java.awt.Dimension(960, 960));

jPanel1.setOpaque(false);

jPanel1.setPreferredSize(new java.awt.Dimension(964, 584));

jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel2.setFont(new java.awt.Font("Viner Hand ITC", 1, 60)); // NOI18N

jLabel2.setText("New User");

jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(470, 50, 320, -1));

jLabel3.setFont(new java.awt.Font("Monotype Corsiva", 1, 48)); // NOI18N

jLabel3.setText("Password ");

jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 222, 250, -1));

jLabel4.setFont(new java.awt.Font("Monotype Corsiva", 1, 48)); // NOI18N

jLabel4.setText("User Name");

jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 140, 250, -1));

jLabel5.setFont(new java.awt.Font("Monotype Corsiva", 1, 48)); // NOI18N

jLabel5.setText("Password Confirm");

jPanel1.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 295, 370, -1));

jLabel6.setFont(new java.awt.Font("Monotype Corsiva", 1, 48)); // NOI18N

jLabel6.setText("Email");

jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 446, 250, -1));

jLabel7.setFont(new java.awt.Font("Monotype Corsiva", 1, 48)); // NOI18N

jLabel7.setText("Phone Number");

jPanel1.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 370, 350, -1));

JT2.setFont(new java.awt.Font("黑体", 0, 24)); // NOI18N

JT2.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

JT2MouseClicked(evt);

}

});

JT2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

JT2ActionPerformed(evt);

}

});

jPanel1.add(JT2, new org.netbeans.lib.awtextra.AbsoluteConstraints(607, 380, 210, 40));

JT1.setFont(new java.awt.Font("黑体", 0, 24)); // NOI18N

JT1.setToolTipText("");

JT1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

JT1MouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

JT1MouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

JT1MouseExited(evt);

}

});

jPanel1.add(JT1, new org.netbeans.lib.awtextra.AbsoluteConstraints(630, 150, 187, -1));

JT3.setFont(new java.awt.Font("黑体", 0, 24)); // NOI18N

JT3.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

JT3MouseClicked(evt);

}

});

jPanel1.add(JT3, new org.netbeans.lib.awtextra.AbsoluteConstraints(607, 450, 210, 40));

JP1.setFont(new java.awt.Font("宋体", 0, 11)); // NOI18N

JP1.setMargin(new java.awt.Insets(2, 0, 2, 6));

JP1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

JP1MouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

JP1MouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

JP1MouseExited(evt);

}

});

jPanel1.add(JP1, new org.netbeans.lib.awtextra.AbsoluteConstraints(630, 230, 187, 40));

JP2.setFont(new java.awt.Font("宋体", 0, 11)); // NOI18N

JP2.setMargin(new java.awt.Insets(2, 0, 2, 6));

JP2.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

JP2MouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

JP2MouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

JP2MouseExited(evt);

}

});

jPanel1.add(JP2, new org.netbeans.lib.awtextra.AbsoluteConstraints(630, 300, 187, 40));

Cancel.setFont(new java.awt.Font("黑体", 0, 18)); // NOI18N

Cancel.setText("Cancel");

Cancel.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

CancelMouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

CancelMouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

CancelMouseExited(evt);

}

public void mousePressed(java.awt.event.MouseEvent evt) {

CancelMousePressed(evt);

}

public void mouseReleased(java.awt.event.MouseEvent evt) {

CancelMouseReleased(evt);

}

});

jPanel1.add(Cancel, new org.netbeans.lib.awtextra.AbsoluteConstraints(600, 560, -1, -1));

Submit.setFont(new java.awt.Font("黑体", 0, 18)); // NOI18N

Submit.setText("Submit");

Submit.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

SubmitMouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

SubmitMouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

SubmitMouseExited(evt);

}

public void mousePressed(java.awt.event.MouseEvent evt) {

SubmitMousePressed(evt);

}

public void mouseReleased(java.awt.event.MouseEvent evt) {

SubmitMouseReleased(evt);

}

});

jPanel1.add(Submit, new org.netbeans.lib.awtextra.AbsoluteConstraints(800, 560, -1, -1));

JL11.setForeground(new java.awt.Color(51, 51, 255));

JL11.setText("No more than 8 letters");

jPanel1.add(JL11, new org.netbeans.lib.awtextra.AbsoluteConstraints(630, 190, 145, 14));

JL21.setForeground(new java.awt.Color(51, 51, 255));

JL21.setText("Between 8 and 16 including numbers and letters");

jPanel1.add(JL21, new org.netbeans.lib.awtextra.AbsoluteConstraints(630, 270, 280, 14));

JL31.setForeground(new java.awt.Color(51, 51, 255));

JL31.setText("Should be same with password");

jPanel1.add(JL31, new org.netbeans.lib.awtextra.AbsoluteConstraints(630, 340, -1, 14));

JL12.setFont(new java.awt.Font("黑体", 1, 18)); // NOI18N

JL12.setForeground(new java.awt.Color(255, 51, 51));

JL12.setText("Too long !");

jPanel1.add(JL12, new org.netbeans.lib.awtextra.AbsoluteConstraints(820, 160, -1, -1));

JL22.setFont(new java.awt.Font("黑体", 1, 18)); // NOI18N

JL22.setForeground(new java.awt.Color(255, 51, 51));

JL22.setText("Too short!");

jPanel1.add(JL22, new org.netbeans.lib.awtextra.AbsoluteConstraints(820, 240, -1, -1));

JL32.setFont(new java.awt.Font("黑体", 1, 18)); // NOI18N

JL32.setForeground(new java.awt.Color(255, 51, 51));

JL32.setText("Not same!");

jPanel1.add(JL32, new org.netbeans.lib.awtextra.AbsoluteConstraints(820, 310, -1, -1));

JL42.setFont(new java.awt.Font("黑体", 1, 18)); // NOI18N

JL42.setForeground(new java.awt.Color(255, 51, 51));

JL42.setText("Has been used");

jPanel1.add(JL42, new org.netbeans.lib.awtextra.AbsoluteConstraints(820, 390, -1, -1));

JL52.setFont(new java.awt.Font("黑体", 1, 18)); // NOI18N

JL52.setForeground(new java.awt.Color(255, 51, 51));

JL52.setText("Has been used");

jPanel1.add(JL52, new org.netbeans.lib.awtextra.AbsoluteConstraints(820, 460, -1, -1));

Cancel1.setFont(new java.awt.Font("黑体", 0, 18)); // NOI18N

Cancel1.setText("Login");

Cancel1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

Cancel1MouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

Cancel1MouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

Cancel1MouseExited(evt);

}

public void mousePressed(java.awt.event.MouseEvent evt) {

Cancel1MousePressed(evt);

}

public void mouseReleased(java.awt.event.MouseEvent evt) {

Cancel1MouseReleased(evt);

}

});

jPanel1.add(Cancel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(390, 560, -1, -1));

jPanel2.add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 988, 640));

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/photo/Cache\_-2e8a9f2d6617e3c1..jpg"))); // NOI18N

jPanel2.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 960, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>

private void CancelMouseReleased(java.awt.event.MouseEvent evt) {

Cancel.setForeground(null);

// TODO add your handling code here:

}

private void CancelMousePressed(java.awt.event.MouseEvent evt) {

Cancel.setForeground(Color.red);

// TODO add your handling code here:

}

private void CancelMouseExited(java.awt.event.MouseEvent evt) {

Cancel.setForeground(null);

// TODO add your handling code here:

}

private void CancelMouseEntered(java.awt.event.MouseEvent evt) {

Cancel.setForeground(Color.blue);

// TODO add your handling code here:

}

private void CancelMouseClicked(java.awt.event.MouseEvent evt) {

int ans = JOptionPane.showConfirmDialog(this, "Are you so cruel?", "confirm", JOptionPane.YES\_NO\_CANCEL\_OPTION);

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

this.dispose();

}

}

private void SubmitMouseReleased(java.awt.event.MouseEvent evt) {

Submit.setForeground(null);

// TODO add your handling code here:

}

private void SubmitMousePressed(java.awt.event.MouseEvent evt) {

Submit.setForeground(Color.red);

// TODO add your handling code here:

}

private void SubmitMouseExited(java.awt.event.MouseEvent evt) {

Submit.setForeground(null);

// TODO add your handling code here:

}

private void SubmitMouseEntered(java.awt.event.MouseEvent evt) {

Submit.setForeground(Color.blue);

}

private void SubmitMouseClicked(java.awt.event.MouseEvent evt) {

JL52.setVisible(true);

int a = JT1();

int b = JP1();

int c = JP2();

int d = JT2();

int e = JT3();

if (a == 0) {

} else if (b == 0) {

} else if (c == 0) {

} else if (d == 0) {

} else if (e == 0) {

} else {

try {

con = Connective.getConnection();

String sql = "insert into User\_details values (null,?,?,?,?)";

pst = con.prepareStatement(sql);

pst.setString(1, JT1.getText());

pst.setString(2, JP1.getText());

pst.setString(3, JT2.getText());

pst.setString(4, JT3.getText());

int executeUpdate = pst.executeUpdate();

if (executeUpdate != 0) {

String sq = "call createtable ('" + JT1.getText() + "');";

pst = con.prepareStatement(sq);

int executeUpdate1 = pst.executeUpdate();

pst.close();

con.close();

int showConfirmDialog = JOptionPane.showConfirmDialog(this, "Congratulation ! You have registered successfully.Do you want login now?", "", JOptionPane.YES\_NO\_OPTION);

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

Open.LI(JT1.getText());

dispose();

}

}

} catch (HeadlessException ex) {

System.out.println(ex);

} catch (SQLException ex) {

System.out.println(ex);

} catch (ClassNotFoundException ex) {

System.out.println(ex);

}

}

}

private void JP2MouseExited(java.awt.event.MouseEvent evt) {

JL31.setVisible(false);// TODO add your handling code here:

}

private void JP2MouseEntered(java.awt.event.MouseEvent evt) {

JL31.setVisible(true);// TODO add your handling code here:

}

private void JP2MouseClicked(java.awt.event.MouseEvent evt) {

JL32.setVisible(false);

JP1();

}

private void JP1MouseExited(java.awt.event.MouseEvent evt) {

JL21.setVisible(false);// TODO add your handling code here:

}

private void JP1MouseEntered(java.awt.event.MouseEvent evt) {

JL21.setVisible(true);// TODO add your handling code here:

}

private void JP1MouseClicked(java.awt.event.MouseEvent evt) {

JL22.setVisible(false);

JT1();

JP2.setText("");

}

private void JT3MouseClicked(java.awt.event.MouseEvent evt) {

JL32.setVisible(true);

JL42.setVisible(false);

JT2();

}

private void JT1MouseExited(java.awt.event.MouseEvent evt) {

JL11.setVisible(false);// TODO add your handling code here:

}

private void JT1MouseEntered(java.awt.event.MouseEvent evt) {

JL11.setVisible(true);// TODO add your handling code here:

}

private void JT1MouseClicked(java.awt.event.MouseEvent evt) {

JT2.setEnabled(false);

JP1.setEnabled(false);

JP2.setEnabled(false);

JT3.setEnabled(false);

JL12.setVisible(false);

}

private void JT2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void JT2MouseClicked(java.awt.event.MouseEvent evt) {

JL42.setVisible(false);

JP2();

}

private void Cancel1MouseClicked(java.awt.event.MouseEvent evt) {

Open.LI();

dispose();

}

private void Cancel1MouseEntered(java.awt.event.MouseEvent evt) {

Cancel1.setForeground(Color.blue);

}

private void Cancel1MouseExited(java.awt.event.MouseEvent evt) {

Cancel1.setForeground(null); // TODO add your handling code here:

}

private void Cancel1MousePressed(java.awt.event.MouseEvent evt) {

Cancel1.setForeground(Color.red);

}

private void Cancel1MouseReleased(java.awt.event.MouseEvent evt) {

Cancel1.setForeground(null); // TODO add your handling code here:

}

public static void main(String args[]) {

/\*

\* Set the Nimbus look and feel

\*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\*

\* If Nimbus (introduced in Java SE 6) is not available, stay with the

\* default look and feel. For details see

\* http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\*

\* Create and display the form

\*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

register register = new register();

register.setLocationRelativeTo(null);

register.setVisible(true);

}

});

}

private int JT1() {

JL12.setVisible(true);

Connective connective = new Connective();

try {

con = connective.getConnection();

String sql = "select \* from User\_Details where User\_name=?";

pst = con.prepareStatement(sql);

pst.setString(1, JT1.getText());

rs = pst.executeQuery();

if (JT1.getText().length() == 0) {

JL12.setText("Can't empty");

JL12.setForeground(new java.awt.Color(255, 0, 0));

} else if (JT1.getText().length() > 8) {

JL12.setText("Too long");

JL12.setForeground(new java.awt.Color(255, 0, 0));

} else if (!use.matcher(JT1.getText()).find()) {

JL12.setText("Incorrect type");

JL12.setForeground(new java.awt.Color(255, 0, 0));

} else if (rs.next()) {

JL12.setText("Has been used");

JL12.setForeground(new java.awt.Color(255, 0, 0));

} else {

JL12.setText(" √");

JL12.setForeground(new java.awt.Color(0, 200, 0));

JP1.setEnabled(true);

JT1.setEnabled(false);

JT2.setEnabled(false);

JP2.setEnabled(false);

JT3.setEnabled(false);

rs.close();

pst.close();

con.close();

return 1;

}

rs.close();

pst.close();

con.close();

} catch (ClassNotFoundException ex) {

System.out.println(ex);

} catch (SQLException ex) {

System.out.println(ex);

}

return 0;

}

private int JT2() {

JL42.setVisible(true);

Connective connective = new Connective();

try {

con = connective.getConnection();

String sql = "select \* from User\_Details where User\_phone=?";

pst = con.prepareStatement(sql);

pst.setString(1, JT2.getText());

rs = pst.executeQuery();

if (JP1.getText().length() == 0) {

JL42.setText("Can't empty");

JL42.setForeground(new java.awt.Color(255, 0, 0));

} else if (!phone.matcher(JT2.getText()).find()) {

JL42.setText("Incorrect type");

JL42.setForeground(new java.awt.Color(255, 0, 0));

} else if (rs.next()) {

JL42.setText("Has been used");

JL42.setForeground(new java.awt.Color(255, 0, 0));

} else {

JT3.setEnabled(true);

JT2.setEnabled(false);

JP2.setEnabled(false);

JP1.setEnabled(false);

JT1.setEnabled(false);

JL42.setForeground(new java.awt.Color(0, 200, 0));

JL42.setText(" √");

rs.close();

pst.close();

con.close();

return 1;

}

rs.close();

pst.close();

con.close();

} catch (ClassNotFoundException ex) {

System.out.println(ex);

} catch (SQLException ex) {

System.out.println(ex);

}

return 0;

}

private int JT3() {

JL52.setVisible(true);

Connective connective = new Connective();

try {

con = Connective.getConnection();

String sql = "select \* from User\_Details where User\_email=?";

pst = con.prepareStatement(sql);

pst.setString(1, JT3.getText());

rs = pst.executeQuery();

if (JT3.getText().length() == 0) {

JL52.setForeground(new java.awt.Color(255, 0, 0));

JL52.setText("Can't empty");

} else if (!email.matcher(JT3.getText()).find()) {

JL52.setForeground(new java.awt.Color(255, 0, 0));

JL52.setText("Incorrect type");

} else if (rs.next()) {

JL52.setForeground(new java.awt.Color(255, 0, 0));

JL52.setText("Has been used");

} else {

JL52.setText(" √");

JL52.setForeground(new java.awt.Color(0, 200, 0));

rs.close();

pst.close();

con.close();

return 1;

}

rs.close();

pst.close();

con.close();

} catch (ClassNotFoundException ex) {

System.out.println(ex

);

} catch (SQLException ex) {

System.out.println(ex

);

}

return 0;

}

private int JP1() {

JL22.setVisible(true);

if (JP1.getText().length() == 0) {

JL22.setText("Can't empty");

JL22.setForeground(new java.awt.Color(255, 0, 0));

} else if (JP1.getText().length() < 8) {

JL22.setForeground(new java.awt.Color(255, 0, 0));

JL22.setText("Too short");

} else if (!passw.matcher(JP1.getText()).find()) {

JL22.setText("Incorrect type");

JL22.setForeground(new java.awt.Color(255, 0, 0));

} else {

JL22.setForeground(new java.awt.Color(0, 200, 0));

JP2.setEnabled(true);

JP1.setEnabled(false);

JT2.setEnabled(false);

JT1.setEnabled(false);

JT3.setEnabled(false);

JL22.setText(" √");

return 1;

}

return 0;

}

private int JP2() {

JL32.setVisible(true);

if (JP1.getText().length() == 0) {

JL32.setText("Can't empty");

JL32.setForeground(new java.awt.Color(255, 0, 0));

} else if (!JP1.getText().equals(JP2.getText())) {

JL32.setText("Not same");

JL32.setForeground(new java.awt.Color(255, 0, 0));

} else {

JT2.setEnabled(true);

JP2.setEnabled(false);

JP1.setEnabled(false);

JT1.setEnabled(false);

JT3.setEnabled(false);

JL32.setForeground(new java.awt.Color(0, 200, 0));

JL32.setText(" √");

return 1;

}

return 0;

}

// Variables declaration - do not modify

private javax.swing.JButton Cancel;

private javax.swing.JButton Cancel1;

private javax.swing.JLabel JL11;

private javax.swing.JLabel JL12;

private javax.swing.JLabel JL21;

private javax.swing.JLabel JL22;

private javax.swing.JLabel JL31;

private javax.swing.JLabel JL32;

private javax.swing.JLabel JL42;

private javax.swing.JLabel JL52;

private javax.swing.JPasswordField JP1;

private javax.swing.JPasswordField JP2;

private javax.swing.JTextField JT1;

private javax.swing.JTextField JT2;

private javax.swing.JTextField JT3;

private javax.swing.JButton Submit;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

// End of variables declaration

}

Welcome.java

package Login\_in;

import frame.MainFrame;

import java.awt.Color;

import static model.Level.setlive;

/\*\*

\*

\* @author 皮皮

\*/

public class welcome extends javax.swing.JFrame {

public static String User="";

/\*\*

\* Creates new form welcome

\*/

public welcome(String abc) {

initComponents();

User = abc;

String pri = " Hello " + User + " , welcome to NMD";

jL.setText(pri);

setResizable(false);

}

public welcome() {

initComponents();

setResizable(false);

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jL = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setTitle("Welcome");

getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jL.setFont(new java.awt.Font("Segoe UI Emoji", 3, 36)); // NOI18N

getContentPane().add(jL, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 100, 770, 70));

jLabel2.setFont(new java.awt.Font("华文隶书", 2, 36)); // NOI18N

jLabel2.setText("Continue");

jLabel2.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jLabel2MouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

jLabel2MouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

jLabel2MouseExited(evt);

}

public void mousePressed(java.awt.event.MouseEvent evt) {

jLabel2MousePressed(evt);

}

public void mouseReleased(java.awt.event.MouseEvent evt) {

jLabel2MouseReleased(evt);

}

});

getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(410, 380, 200, -1));

jLabel4.setFont(new java.awt.Font("华文隶书", 2, 36)); // NOI18N

jLabel4.setText("New Game");

jLabel4.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jLabel4MouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

jLabel4MouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

jLabel4MouseExited(evt);

}

public void mousePressed(java.awt.event.MouseEvent evt) {

jLabel4MousePressed(evt);

}

public void mouseReleased(java.awt.event.MouseEvent evt) {

jLabel4MouseReleased(evt);

}

});

jLabel4.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

jLabel4KeyTyped(evt);

}

});

getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(410, 310, 200, -1));

jLabel5.setFont(new java.awt.Font("华文隶书", 2, 36)); // NOI18N

jLabel5.setText("Logout");

jLabel5.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jLabel5MouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

jLabel5MouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

jLabel5MouseExited(evt);

}

public void mousePressed(java.awt.event.MouseEvent evt) {

jLabel5MousePressed(evt);

}

public void mouseReleased(java.awt.event.MouseEvent evt) {

jLabel5MouseReleased(evt);

}

});

getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(410, 460, 200, -1));

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/photo/Cache\_26de8fed954b334d..jpg"))); // NOI18N

getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

pack();

}// </editor-fold>

private void jLabel4MouseEntered(java.awt.event.MouseEvent evt) {

jLabel4.setForeground(Color.red);

jLabel4.setFont(new java.awt.Font("华文隶书", 2, 40)); // NOI18N

}

private void jLabel4MouseExited(java.awt.event.MouseEvent evt) {

jLabel4.setForeground(null);

jLabel4.setFont(new java.awt.Font("华文隶书", 2, 36)); // NOI18N

}

private void jLabel2MouseEntered(java.awt.event.MouseEvent evt) {

jLabel2.setForeground(Color.red);

jLabel2.setFont(new java.awt.Font("华文隶书", 2, 40)); // TODO add your handling code here:

}

private void jLabel2MouseExited(java.awt.event.MouseEvent evt) {

jLabel2.setForeground(null);

jLabel2.setFont(new java.awt.Font("华文隶书", 2, 36)); // TODO add your handling code here:

}

private void jLabel4MousePressed(java.awt.event.MouseEvent evt) {

jLabel4.setFont(new java.awt.Font("华文隶书", 3, 40)); // NOI18N

}

private void jLabel2MousePressed(java.awt.event.MouseEvent evt) {

jLabel2.setFont(new java.awt.Font("华文隶书", 3, 40)); // TODO add your handling code here:

}

private void jLabel4KeyTyped(java.awt.event.KeyEvent evt) {

}

private void jLabel4MouseReleased(java.awt.event.MouseEvent evt) {

jLabel4.setForeground(null);

jLabel4.setFont(new java.awt.Font("华文隶书", 2, 36));

}

private void jLabel2MouseReleased(java.awt.event.MouseEvent evt) {

jLabel2.setForeground(null);

jLabel2.setFont(new java.awt.Font("华文隶书", 2, 36));

}

private void jLabel5MouseClicked(java.awt.event.MouseEvent evt) {

Open.LI(User);

dispose();

}

private void jLabel5MouseEntered(java.awt.event.MouseEvent evt) {

jLabel5.setForeground(Color.red);

jLabel5.setFont(new java.awt.Font("华文隶书", 2, 40));

}

private void jLabel5MouseExited(java.awt.event.MouseEvent evt) {

jLabel5.setForeground(null);

jLabel5.setFont(new java.awt.Font("华文隶书", 2, 36));

}

private void jLabel5MousePressed(java.awt.event.MouseEvent evt) {

jLabel5.setFont(new java.awt.Font("华文隶书", 3, 40));// TODO add your handling code here:

}

private void jLabel5MouseReleased(java.awt.event.MouseEvent evt) {

jLabel5.setForeground(null);

jLabel5.setFont(new java.awt.Font("华文隶书", 2, 36));// TODO add your handling code here:

}

private void jLabel4MouseClicked(java.awt.event.MouseEvent evt) {

setlive(1);

MainFrame frame = new MainFrame();

frame.setVisible(true);

dispose();

}

private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {

Open.KF();

dispose();

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\*

\* Set the Nimbus look and feel

\*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\*

\* If Nimbus (introduced in Java SE 6) is not available, stay with the

\* default look and feel. For details see

\* http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(welcome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(welcome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(welcome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(welcome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\*

\* Create and display the form

\*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

welcome welcome = new welcome();

welcome.setLocationRelativeTo(null);

welcome.setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JLabel jL;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

// End of variables declaration

}

**Game module**

Mysql

create database game ;

use game;

create table user\_details(

User\_id int not null auto\_increment primary key,

User\_name varchar(30),

User\_password varchar(30),

User\_phone varchar(30),

User\_Email varchar(50)

)engine = memory auto\_increment=1;

insert into user\_details values(null,'root','123456','13453537634','1050289148@qq.com');

DELIMITER //

CREATE PROCEDURE createtable(IN tname varchar(20))

BEGIN

SET @s = CONCAT('CREATE TABLE ', tname, '(ID int not null auto\_increment primary key, Utime Datetime , Ulevel int)engine = memory auto\_increment=1;');

PREPARE stm FROM @s;

EXECUTE stm;

END //

DELIMITER ;

call createtable('root');

insert into root values (null,'2019-5-16 6:20:00',1);

insert into root values (null,'2019-5-16 6:20:00',2);

insert into root values (null,'2019-5-16 6:20:00',3);

insert into root values (null,'2019-5-16 6:20:00',4);

insert into root values (null,'2019-5-16 6:20:00',5);

insert into root values (null,'2019-5-16 6:20:00',6);

insert into root values (null,'2019-5-16 6:20:00',7);

insert into root values (null,'2019-5-16 6:20:00',8);

insert into root values (null,'2019-5-16 6:20:00',9);

insert into root values (null,'2019-5-16 6:20:00',10);

insert into root values (null,'2019-5-16 6:20:00',11);

insert into root values (null,'2019-5-16 6:20:00',12);

insert into root values (null,'2019-5-16 6:20:00',13);

**Direction**

**package Type;**

public enum Direction {

UP,

RIGHT,

DOWN,

LEFT,

}

**TankType**

**package Type;**

public enum TankType {

player,

bot, // 电脑

}

**WallType**

**package Type;**

public enum WallType {

base ,

brick,

grass,

river,

iron ,

}

GamePanel

package frame;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.awt.image.BufferedImage;

import java.util.ArrayList;

import java.util.List;

import java.util.Random;

import java.util.Vector;

import java.awt.\*;

import javax.swing.JPanel;

import Type.TankType;

import model.wall.Wall;

import util.ImageUtil;

import model.\*;

public class GamePanel extends JPanel implements KeyListener {

public static final int FRESH = 20;

private BufferedImage image;

private Tank play;

private boolean space\_key, s\_key, w\_key, a\_key, d\_key;

private int level;

private List<Bullet> bullets;

private volatile List<Tank> allTanks;

private List<Tank> botTanks;

private int botMaxInMap = 6;

private final int botCount = 20;

private int botReadyCount = botCount;

private int botSurplusCount = botCount;

private int botx[] = { 10, 367, 754 };

private List<Tank> palyerTanks;

private volatile boolean finish = false;

private Base base;

private Graphics2D g2;

private MainFrame frame;

private List<Wall> walls;

private List<Boom> boomImage;

private Random r = new Random();

private int createBotTimer = 0;

private Tank survivor;

public GamePanel(MainFrame frame, int level) {

this.frame = frame;

this.level = level;

setBackground(Color.WHITE);

init();

Thread t = new FreshThead();

t.start();

addListener();

}

private void init() {

bullets = new ArrayList<Bullet>();

allTanks = new ArrayList<Tank>();

walls = new ArrayList<Wall>();

boomImage = new ArrayList<Boom>();

image = new BufferedImage(794, 572, BufferedImage.TYPE\_INT\_BGR);

g2 = image.createGraphics();

palyerTanks = new ArrayList<Tank>();

play = new Tank(278, 537, ImageUtil.PYAYER\_UP\_IMAGE\_URL, this, TankType.player);

palyerTanks.add(play);

/\*添加初始位置\*/

botTanks = new Vector<Tank>();

botTanks.add(new Bot(botx[0], 1, this, TankType.bot));

botTanks.add(new Bot(botx[1], 1, this, TankType.bot));

botTanks.add(new Bot(botx[2], 1, this, TankType.bot));

botReadyCount -= 3;

allTanks.addAll(palyerTanks);

allTanks.addAll(botTanks);

base = new Base(367, 532);

initWalls();

}

private void addListener() {

frame.addKeyListener(this);

}

private void initWalls() {

Map map = Map.getMap(level);

walls.addAll(map.getWalls());

walls.add(base);

}

@Override

public void paint(Graphics g) {

paintTankActoin();

CreateBot();

paintImage();

g.drawImage(image, 0, 0, this);

}

private void paintImage() {

g2.setColor(Color.WHITE);

g2.fillRect(0, 0, image.getWidth(), image.getHeight());//

panitBoom();

paintBotCount();

panitBotTanks();

panitPlayerTanks();

allTanks.addAll(palyerTanks);

allTanks.addAll(botTanks);

panitWalls();

panitBullets();

if (botSurplusCount == 0) {

stopThread();

paintBotCount();

g2.setFont(new Font("楷体", Font.BOLD, 100));

g2.setColor(Color.green);

g2.drawString("victory !", 250, 400);

gotoNextLevel();

}

if (!play.isAlive()) {

stopThread();

boomImage.add(new Boom(play.x, play.y));

panitBoom();

paintGameOver();

keepfile(level);

gotoPrevisousLevel();

}

if (!base.isAlive()) {

stopThread();

paintGameOver();

base.setImage(ImageUtil.BREAK\_BASE\_IMAGE\_URL);

gotoPrevisousLevel();

}

g2.drawImage(base.getImage(), base.x, base.y, this);

}

private void paintBotCount() {

g2.setColor(Color.BLUE);

g2.drawString("How many more enemies are there：" + botSurplusCount, 337, 15);

}

private void paintGameOver() {

g2.setFont(new Font("楷体", Font.BOLD, 100));

g2.setColor(Color.RED);

g2.drawString("Game Over !", 250, 400);

}

private void panitBoom() {

for (int i = 0; i < boomImage.size(); i++) {

Boom boom = boomImage.get(i);

if (boom.isAlive()) {

boom.show(g2);

} else {

boomImage.remove(i);

i--;

}

}

}

private void panitWalls() {

for (int i = 0; i < walls.size(); i++) {

Wall w = walls.get(i);

if (w.isAlive()) {

g2.drawImage(w.getImage(), w.x, w.y, this);

} else {

walls.remove(i);

i--;

}

}

}

private void panitBullets() {

for (int i = 0; i < bullets.size(); i++) {

Bullet b = bullets.get(i);

if (b.isAlive()) {

b.move();

b.hitBase();

b.hitWall();

b.hitTank();

g2.drawImage(b.getImage(), b.x, b.y, this);

} else {// 如果子弹无效

bullets.remove(i);

i--;

}

}

}

/\*\*

\* 绘制电脑坦克

\*/

private void panitBotTanks() {

for (int i = 0; i < botTanks.size(); i++) {

Bot t = (Bot) botTanks.get(i);

if (t.isAlive()) {

t.go();

g2.drawImage(t.getImage(), t.x, t.y, this);

} else {

botTanks.remove(i);

i--;

boomImage.add(new Boom(t.x, t.y));

decreaseBot();

}

}

}

/\*\*

\* 绘制玩家坦克

\*/

private void panitPlayerTanks() {

for (int i = 0; i < palyerTanks.size(); i++) {

Tank t = palyerTanks.get(i);

if (t.isAlive()) {

g2.drawImage(t.getImage(), t.x, t.y, this);

} else {

palyerTanks.remove(i);

i--;

boomImage.add(new Boom(t.x, t.y));

}

}

}

/\*\*

\* 结束游戏帧刷新

\*/

private synchronized void stopThread() {

frame.removeKeyListener(this);

finish = true;

}

private void keepfile( int level) {

}

/\*\*

\* 游戏帧刷新线程

\*/

private class FreshThead extends Thread {

@Override

public void run() {

while (!finish) {

repaint();

try {

Thread.sleep(FRESH);

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

}

private void CreateBot() {

createBotTimer += FRESH;

if (botTanks.size() < botMaxInMap && botReadyCount > 0 && createBotTimer >= 4000) {

int index = r.nextInt(3);

Rectangle bornRect = new Rectangle(botx[index], 1, 35, 35);

for (int i = 0, lengh = allTanks.size(); i < lengh; i++) {

Tank t = allTanks.get(i);

if (t.isAlive() && t.hit(bornRect)) {

return;

}

}

botTanks.add(new Bot(botx[index], 1, GamePanel.this, TankType.bot));

botReadyCount--;

createBotTimer = 0;

}

}

/\*\*\* 进入下一关卡\*/

private void gotoNextLevel() {

Thread jump = new JumpPageThead(Level.nextLevel());

jump.start();

}

private void gotoPrevisousLevel() {

Thread jump = new JumpPageThead(Level.previsousLevel());

jump.start();

}

public void decreaseBot() {

botSurplusCount--;

}

public void keyPressed(KeyEvent e) {

switch (e.getKeyCode()) {

case KeyEvent.VK\_SPACE:

space\_key = true;

break;

case KeyEvent.VK\_W:

w\_key = true;

a\_key = false;

s\_key = false;

d\_key = false;

break;

case KeyEvent.VK\_A:

w\_key = false;

a\_key = true;

s\_key = false;

d\_key = false;

break;

case KeyEvent.VK\_S:

w\_key = false;

a\_key = false;

s\_key = true;

d\_key = false;

break;

case KeyEvent.VK\_D:

w\_key = false;

a\_key = false;

s\_key = false;

d\_key = true;

break;

}

}

private void paintTankActoin() {

if (space\_key) {

play.attack();

}

if (w\_key) {

play.upward();

}

if (d\_key) {

play.rightward();

}

if (a\_key) {

play.leftward();

}

if (s\_key) {

play.downward();

}

}

public void keyReleased(KeyEvent e) {

switch (e.getKeyCode()) {

case KeyEvent.VK\_SPACE:

space\_key = false;

break;

case KeyEvent.VK\_W:

w\_key = false;

break;

case KeyEvent.VK\_A:

a\_key = false;

break;

case KeyEvent.VK\_S:

s\_key = false;

break;

case KeyEvent.VK\_D:

d\_key = false;

break;

}

}

public void addBullet(Bullet b) {

bullets.add(b);

}

public List<Wall> getWalls() {

return walls;

}

public Base getBase() {

return base;

}

public List<Tank> getTanks() {

return allTanks;

}

private class JumpPageThead extends Thread {

int level;

public JumpPageThead(int level) {

this.level = level;

}

public void run() {

try {

Thread.sleep(1000);

frame.setPanel(new LevelPanel(level, frame));

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

public void keyTyped(KeyEvent e) {

}

}

LevelPanel

package frame;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import javax.swing.JPanel;

public class LevelPanel extends JPanel{

private int level;

private MainFrame frame;

private String levelStr;

private String ready = "";

public LevelPanel(int level, MainFrame frame) {

this.frame = frame;

this.level = level;

levelStr = "Level " + level;

Thread t = new LevelPanelThread();

t.start();

}

public void paint(Graphics g) {

g.setColor(Color.WHITE);

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

g.setFont(new Font("Consolas", Font.BOLD, 50));

g.setColor(Color.BLACK);

g.drawString(levelStr, 260, 300);

g.setColor(Color.RED);

g.drawString(ready, 270, 400);

}

private void gotoGamePanel() {

frame.setPanel(new GamePanel(frame, level));

}

private class LevelPanelThread extends Thread {

public void run() {

for (int i = 0; i < 6; i++) {

if (i % 2 == 0) {

levelStr = "Level " + level;

} else {

levelStr = "";

}

if (i == 4) {

ready = "Ready? !";

}

repaint();

try {

Thread.sleep(500);

} catch (InterruptedException e) {

System.out.println(e);

}

}

gotoGamePanel();

}

}

}

LoginPanel

package frame;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import javax.swing.JPanel;

import static model.Level.getprevisousLevel;

public class LoginPanel extends JPanel implements KeyListener{

private MainFrame frame;// 主窗体

private int y1 = 370, y2 = 430;

private int tankY = y1;

public LoginPanel(MainFrame frame) {

this.frame = frame;

addListener();

}

@Override

public void paint(Graphics g) {

Font font = new Font("黑体", Font.BOLD, 35);

g.setFont(font);

g.setColor(Color.BLUE);

g.drawString("PLEASE PRESS SPACE....", 200, 300);

}

private void gotoLevelPanel() {

frame.removeKeyListener((KeyListener) this);

frame.setPanel(new LevelPanel(getprevisousLevel(), frame));

}

private void addListener() {

frame.addKeyListener(this);

}

@Override

public void keyPressed(KeyEvent e) {

int code = e.getKeyCode();

switch (code) {

case KeyEvent.VK\_SPACE:

gotoLevelPanel();

break;

}

}

@Override

public void keyReleased(KeyEvent e) {

}

@Override

public void keyTyped(KeyEvent e) {

}

}

Main Frame

package frame;

import java.util.Date;

import java.awt.Container;

import java.awt.Dimension;

import java.awt.Toolkit;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.sql.Connection;

import java.sql.PreparedStatement;

import Login\_in.Connective;

import Login\_in.Login\_in;

import Login\_in.Open;

import Login\_in.welcome;

import java.sql.SQLException;

import java.text.SimpleDateFormat;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import model.Level;

public class MainFrame extends JFrame {

Connection con = null;

PreparedStatement pst;

public MainFrame() {

setTitle("NationalMD");

setSize(800, 600);

setResizable(false);

Toolkit tool = Toolkit.getDefaultToolkit();

Dimension d = tool.getScreenSize();

setLocation((d.width - getWidth()) / 2, (d.height - getHeight()) / 2);

setDefaultCloseOperation(DO\_NOTHING\_ON\_CLOSE);

addListener();

setPanel(new LoginPanel(this));

}

private void addListener() {

addWindowListener(new WindowAdapter() {

@Override

public void windowClosing(WindowEvent e) {

int closeCode = JOptionPane.showConfirmDialog(MainFrame.this, "Do you want exit","Warning！",JOptionPane.YES\_NO\_OPTION);// 弹出选择对话框，并记录用户选择

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

dispose();

keepdata();

Open.LI(welcome.User);

}

}

});

}

public void setPanel(JPanel panel) {

Container c = getContentPane();

c.removeAll();

c.add(panel);

c.validate();

}

private void keepdata(){

try {

con = Connective.getConnection();

Date date = new Date();

SimpleDateFormat simpleDateFormat = new SimpleDateFormat("YYYY-MM-dd HH:mm:ss");

String time=simpleDateFormat.format(date);

String sql = "insert into " + welcome.User + " values(null,?,?);";

pst = con.prepareStatement(sql);

pst.setString(1, time);

pst.setInt(2, Level.getprevisousLevel());

pst.executeUpdate();

} catch (ClassNotFoundException e) {

System.out.println(e);

} catch (SQLException e) {

System.out.println(e);

}

}

}

package model;

import java.awt.Graphics2D;

import frame.GamePanel;

import util.ImageUtil;

public class Boom extends VisibleImage {

private int timer = 0;

private int fresh = GamePanel.FRESH ;

private boolean alive = true;

public Boom(int x,int y){

super(x,y,ImageUtil.BOOM\_IMAGE\_URL);

}

public void show(Graphics2D g2) {

if (timer >= 500) {

alive = false;

} else {

g2.drawImage(getImage(), x, y, null);

timer += fresh;

}

}

public boolean isAlive() {

return alive;

}

public void setAlive(boolean alive) {

this.alive = alive;

}

}

Bot

package model;

import java.awt.Dimension;

import java.util.List;

import java.awt.Rectangle;

import java.util.Locale;

import java.util.Random;

import Type.Direction;

import Type.TankType;

import frame.GamePanel;

import util.ImageUtil;

public class Bot extends Tank {

private Random random = new Random(); //随机坦克

private Direction dir; // 方向

private int fresh = GamePanel.FRESH;

private int MoveTimer = 0; //移动计时器

public Bot(int x,int y, GamePanel gamePanel ,TankType type){

super(x,y,ImageUtil.BOT\_DOWN\_IMAGE\_URL,gamePanel,type);

dir= Direction.DOWN;

setAttackCoolDowntime(1000);

}

//移动

private Direction randomDirection(){ //随即位置

int rum = random.nextInt(4);

switch(rum){

case 0:

return Direction.UP;

case 1:

return Direction.RIGHT;

case 2:

return Direction.LEFT;

default:

return Direction.DOWN;

}

}

// 碰撞

boolean hitTank(int x, int y){

Rectangle next = new Rectangle(x,y,width,height);

List<Tank> tanks = gamePanel.getTanks();

for(int i = 0, lengh = tanks.size();i<lengh;i++){

Tank t = tanks.get(i);

if(!this.equals(t)){

if(t.alive &&t.hit(next) ){ //存疑？

if(t instanceof Bot ){

dir = randomDirection();

}

return true;

}

}

}

return false;

}

// 攻击

public void attack(){

int rum = random.nextInt(100);

if(rum <4){

super.attack();

}

}

//串联起来

public void go(){

if(isAttackCoolDown()){

attack();

}

if(MoveTimer >= 3000){

dir =randomDirection();

MoveTimer+=fresh;

}else{

MoveTimer +=fresh;

}

switch(dir){

case UP:

upward();

break;

case DOWN:

downward();

break;

case RIGHT:

rightward();

break;

case LEFT:

leftward();

break;

}

}

}

Bullet

package model;

import java.awt.Color;

import java.awt.Graphics;

import java.util.List;

import Type.Direction;

import Type.TankType;

import frame.GamePanel;

import model.wall.BrickWall;

import model.wall.IronWall;

import model.wall.Wall;

//子弹

public class Bullet extends VisibleImage{

Direction direction;

static final int LENGTH =8;

private GamePanel gamePanel ;

private int speed = 7 ;

private boolean alive = true ;

Color color = Color.ORANGE ;

TankType owner ;

private void init(){

Graphics g = image.getGraphics();

g.setColor(Color.WHITE );

g.fillRect(0, 0, LENGTH, LENGTH);

g.setColor(color);

g.fillOval(0, 0, LENGTH, LENGTH);

g.setColor(Color.BLACK);

g.drawOval(0, 0, LENGTH-1, LENGTH-1);

}

public Bullet(int x,int y,Direction direction ,GamePanel gamePanel ,TankType owner){

super(x,y,LENGTH,LENGTH);

this.direction =direction;

this.gamePanel = gamePanel;

this.owner = owner;

init();

}

//移动

private void moveToBorder() {

if(x<0){

dispose();

}else if(x> gamePanel.getWidth()-width){

dispose();

}

if(y<0){

dispose();

}else if(y>gamePanel.getHeight() -height){

dispose();

}

}

private void leftward(){

x -=speed;

moveToBorder();

}

private void rightward(){

x +=speed;

moveToBorder();

}

private void upward(){

y -=speed;

moveToBorder();

}

private void downward(){

y +=speed;

moveToBorder();

}

public void move(){

switch(direction){

case UP:

upward();

break;

case DOWN:

downward();

break;

case RIGHT:

rightward();

break;

case LEFT:{

leftward();

break;

}

}

}private synchronized void dispose() {

alive = false;

}public boolean isAlive() {

return alive;

}public void hitBase() {

Base b = gamePanel.getBase();

if (this.hit(b)) {

b.setAlive(false);

}

}public void hitTank() {

List<Tank> tanks = gamePanel.getTanks();

for (int i = 0, lengh = tanks.size(); i < lengh; i++) {

Tank t = tanks.get(i);

if (t.isAlive() && this.hit(t)) {

switch (owner) {

case player:// 如果是玩家

if (t instanceof Bot) {// 如果击中的坦克是电脑

alive = false;

t.setAlive(false);

}

break;

case bot:

if (t instanceof Bot) {

alive = false;

} else if (t instanceof Tank) {// 如果击中的是玩家

alive = false;

t.setAlive(false);

}

break;

default:

alive = false;

t.setAlive(false);

}

}

}

}

public void hitWall() {

List<Wall> walls = gamePanel.getWalls();

for (int i = 0, lengh = walls.size(); i < lengh; i++) {

Wall w = walls.get(i);

if (this.hit(w)) {

if (w instanceof BrickWall) {

alive = false;

w.setAlive(false);

}

if (w instanceof IronWall) {

alive = false;

}

}

}

}

}

Level

import java.io.File;

import java.io.FileNotFoundException;

import util.MapIO;

public class Level {

private static int nextLevel = 1;// 下一关记录

private static int previsousLevel = 1;// 上一关记录

private static int count;// 关卡总数

static {

try {

File f = new File(MapIO.DATA\_PATH);

if (!f.exists()) {

throw new FileNotFoundException("地图文件缺失！");

}

File fs[] = f.listFiles();//

count = fs.length;

if (count == 0) {

throw new FileNotFoundException("地图文件缺失！");

}

} catch (FileNotFoundException e) {

e.printStackTrace();

}

}

public static int nextLevel() {

nextLevel++;

previsousLevel = nextLevel;

if (nextLevel > count) {// 如果关卡数大于关卡总数

nextLevel = 1;// 从第一关开始

}

return nextLevel;// 返回下一关的值

}

public static int previsousLevel() {

return previsousLevel;// 返回上一关的值

}

public static int getprevisousLevel(){

return previsousLevel;

}

public static void setlive(int a){

previsousLevel=a;

nextLevel=a;

}

}

Map

package model;

import java.util.ArrayList;

import model.wall.BrickWall;

import model.wall.Wall;

import java.util.List;

import util.MapIO;

public class Map {

private static List<Wall> walls = new ArrayList<Wall>();

private Map() {

}

public static Map getMap(String level) {

walls.clear();

walls.addAll(MapIO.readMap(level));

// 基地砖墙

for (int a = 347; a <= 407; a += 20) {

for (int b = 512; b <= 572; b += 20) {

if (a >= 367 && a <= 387 && b >= 532) {

continue;

} else {

walls.add(new BrickWall(a, b));

}

}

}

return new Map();

}

public static Map getMap(int level) {

return getMap(String.valueOf(level));

}

public List<Wall> getWalls() {

return walls;

}

}

Tank

package model;

import java.util.ArrayList;

import model.wall.BrickWall;

import model.wall.Wall;

import java.util.List;

import util.MapIO;

public class Map {

private static List<Wall> walls = new ArrayList<Wall>();

private Map() {

}

public static Map getMap(String level) {

walls.clear();

walls.addAll(MapIO.readMap(level));

// 基地砖墙

for (int a = 347; a <= 407; a += 20) {

for (int b = 512; b <= 572; b += 20) {

if (a >= 367 && a <= 387 && b >= 532) {

continue;

} else {

walls.add(new BrickWall(a, b));

}

}

}

return new Map();

}

public static Map getMap(int level) {

return getMap(String.valueOf(level));

}

public List<Wall> getWalls() {

return walls;

}

}

package model;

import java.awt.Rectangle;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import javax.imageio.ImageIO;

public abstract class VisibleImage {

public int x;

public int y;

int width ;

int height ;

BufferedImage image;

public VisibleImage(int x,int y ,int width ,int height){

this.x = x;

this.y = y;

this.width =width;

this.height = height;

image = new BufferedImage(width,height,BufferedImage.TYPE\_INT\_RGB);

}

public VisibleImage(int x ,int y ,String url){

this.x = x;

this.y = y;

try{

image = ImageIO.read(new File(url));

this.width = image.getWidth();

this.height = image.getHeight();

}catch(IOException e ){

System.out.println(e);

}

}

public BufferedImage getImage() {

return image;

}

public void setImage(String url){

try{

this.image = ImageIO.read(new File(url));

}catch(IOException e){

System.out.println(e);

}

}

public boolean hit(VisibleImage v) {

return hit(v.getBounds());

}

public boolean hit(Rectangle r) {

if (r == null) {

return false;

}

return getBounds().intersects(r);

}

public Rectangle getBounds() {

return new Rectangle(x, y, width, height);

}

public int getWidth() {

return width;

}

public void setWidth(int width) {

this.width = width;

}

public int getHeight() {

return height;

}

public void setHeight(int height) {

this.height = height;

}

@Override

public String toString() {

return "Visiblemage [x=" + x + ", y=" + y + ", width=" + width + ", height=" + height + "]";

}

}

package model.wall;

import util.ImageUtil;

/\*\*

\*

\* @author 皮皮

\*/

public class BrickWall extends Wall{

/\*\*

\*

\* @param x

\* @param y

\*/

public BrickWall(int x,int y){

super(x,y,ImageUtil.BRICKWALL\_IMAGE\_URL); //

}

}

Grass wll

package model.wall;

import util.ImageUtil;

//草地

public class GrassWall extends Wall {

public GrassWall(int x,int y){

super(x,y,ImageUtil.GRASSWALL\_IMAGE\_URL);

}

}

*Rival wall*

*package model.wall;*

*import util.ImageUtil;*

*//河流*

*/\*\**

*\**

*\* @author 皮皮*

*\*/*

*public class RiverWall extends Wall {*

*public RiverWall(int x, int y){*

*super(x,y,ImageUtil.RIVERWALL\_IMAGE\_URL);*

*}*

*}*

*Wall*

*package model.wall;*

*import model.VisibleImage;*

*public abstract class Wall extends VisibleImage {*

*private boolean alive = true;*

*public Wall(int x, int y, String url) {*

*super(x, y, url);*

*}*

*public boolean isAlive() {*

*return alive;*

*}*

*public void setAlive(boolean alive) {*

*this.alive = alive;*

*}*