|  |
| --- |
| **(Autonomous Institute Affiliated to VTU)**  **Department of Information Science and Engineering** |
|  |
| A Project Report on |
| **“WUMPUS WORLD”** |
| *Submitted in partial fulfillment of the CIE for the subject*  **Java and J2EE(IS63)** |
| By |
| **Vybhav Jain** |
| **1MS16IS144** |
|  |

**Table of Contents:**

1. Abstract\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_03
2. Introduction\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_04
3. Implementation \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_05
4. Result\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_28

**1.ABSTRACT:**

Games can provide a variety of sensory experiences for people of all age groups. Additionally, it also actively engages the user’s attention which leads to a better understanding of the matter as compared to the traditional pen and paper approach. Keeping this in mind, the objective of this project is to show a visual representation of the famous “Wumpus World” game which was introduced by Genesereth, and is discussed in Russell-Norvig(2003). The Wumpus World is a simple world (as is the Block World) which is a famous example to represent knowledge and to give clarity involved in logical reasoning approaches. Wumpus world was the basic example to teach an AI system on how to make decision on the basis of mathematical deductions. Here, different methods of the SWING library have been used including but not limited to frames, buttons and panels.

**2.INTRODUCTION:**

The Wumpus world problem is a 4x4 puzzle where the objective of the game is to find the gold. The user must safely navigate their way around bottomless pits of death and evil Wumpus creatures to locate the gold hidden on the board. The goal here is to minimize the number of steps and maximize the points. The project has been implemented in Java and the various components of frames have been used to create a GUI. The various elements in the game are :

1) Wumpus: A deadly monster that lurks in one of the square. This monster emits a pungent odour that can be detected in the adjacent boxes that this monster is present in. When facing a possibility of the presence of this monster, shoot an arrow to kill it or avoid it by taking a different route.

2) Pit: A bottomless hole which will trap you forever. Avoid this at all costs. The pit is so deep, that the breeze that blows because of it can be felt in the adjacent squares, use that as a warning and tread carefully.

3) Gold: The very objective of the game to retrieve this priceless mineral. Identify the square containing the gold by its lustrous glitter which can be seen in the adjacent squares.

4) Actions: There are multiple actions, the most basic being left,right,forward and shoot. Left and right as the names indicate are meant to turn towards their respective directions. Forward will move the player’s position by one square in the direction they are facing.Each forward move costs one point. Lastly, shoot action fires an arrow in the facing direction, and if any wumpus is present, it eliminates it. But, each arrow shot costs 10points.

5) Combo moves: While a single move is good, the goal of the game is to minimise the number of steps. So, here two new actions are introduced: up-right and up-left which as the names indicate, irrespective of the direction, move the player to the diagonal right and diagonal left box respectively at the cost of only 1 point instead of 2.

6) Magic ladder: Present in one of the squares, this ladder will transport the player to a random square. There is no way to find out which square contains the ladder or which square will the ladder lead you to. The only advantage here being the teleportation charge is zero.

**3. IMPLEMENTATION:**

Files present:

1. exp.java (Contains the main program)
2. Actions\_available.java(An interface which defines a method)
3. move\_rightdirection.java(Contains a method to turn towards the right direction)
4. ComboMovement.java(An abstract class which contains an abstract method called move which is later to implemented to move in the diagonal direction)
5. Move\_up\_right.java(Inherits ComboMovement.java and writes a method to move in the up-right direction )
6. Move\_up\_left.java(Inherits ComboMovement.java and writes a method to move in the up-right direction )

**1) exp.java**

package wumpus;

import java.util.Random;

import javax.swing.\*;

import java.awt.GridLayout;

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.Icon.\*;

import javax.swing.border.Border;

public class exp implements Actions\_available {

int index =12;

int num;

int score = 1000;

int ladder;

String Wumpus = "W";

String Smell = "s";

String Gold = "G";

String Glitter = "g";

String Pit = "P";

String Breeze = "b";

Random rand = new Random();

JButton buttons[] = new JButton[30];

JButton buttonsnew[] = new JButton[30];

public exp(){

JFrame frame=new JFrame();

JPanel frame2 = new JPanel();

JPanel frame3 = new JPanel();

frame2.setLayout(new GridLayout(4,4,5,5));

frame3.setLayout(new GridLayout(4,4,5,5));

Border emptyBorder = BorderFactory.createEmptyBorder(20, 20, 10, 10);

Border bored = BorderFactory.createLineBorder(Color.RED);

for (int i = 0; i < 16; i++) {

buttons[i] = new JButton("");

buttons[i].setBorder(emptyBorder);

buttons[i].setBorder(bored);

frame2.add(buttons[i]);

}

for (int i = 0; i < 16; i++) {

buttonsnew[i] = new JButton("??");

buttonsnew[i].setBorder(emptyBorder);

buttonsnew[i].setBorder(bored);

frame3.add(buttonsnew[i]);

}

int pitloc = rand.nextInt(16);

int wumploc = rand.nextInt(16);

int goldloc = rand.nextInt(16);

ladder = rand.nextInt(16);

while(goldloc == wumploc || goldloc == pitloc)

{

goldloc = rand.nextInt(16);

}

while(goldloc == index || index == pitloc || wumploc == index)

{

index = rand.nextInt(16);

}

while(ladder == index)

{

ladder= rand.nextInt(16);

}

buttons[pitloc].setText(Pit);

buttons[wumploc].setText(Wumpus);

buttons[goldloc].setText(Gold);

Icon pit = new ImageIcon("pit.jpg");

Icon breeze = new ImageIcon("breeze.jpg");

Icon wumpus = new ImageIcon("wumpus.png");

Icon smell = new ImageIcon("smell.jpg");

Icon gold = new ImageIcon("gold.png");

Icon glitter = new ImageIcon("glitter.jpg");

setpos(pitloc,Breeze,pit,breeze);

setpos(wumploc,Smell,wumpus,smell);

setpos(goldloc,Glitter,gold,glitter);

reveal(index);

String appender = buttonsnew[index].getText();

appender = appender + "^";

buttonsnew[index].setText(appender);

frame3.setVisible(true);

frame3.setBounds(450,20,720,720);

frame.add(frame3);

JButton forward = new JButton("Forward");

forward.setBounds(30,10,90,60);

JButton left = new JButton("Left");

left.setBounds(30,80,90,60);

JButton right = new JButton("Right");

right.setBounds(30,150,90,60);

JButton shoot= new JButton("Shoot");

shoot.setBounds(30,220,90,60);

JTextField tf = new JTextField("Current score will be displayed here");

tf.setBounds(30,290,400,60);

JTextField tf1 = new JTextField("On click shoot will be displayed here");

tf1.setBounds(30,360,400,60);

JTextArea tf2 = new JTextArea("WELCOME TO WUMPUS WORLD: \n INSTRUCTIONS: \n 0) Start position is denoted by : '^' \n 1)Objective of the game is to retrive the gold. \n 2)Each step forward costs 1 point, and every arrow shot costs 10points. \n 3)Wumpus can be killed by clicking on shoot while facing towards it. \n 4) you start off with 1000 points \n 5)Avoid pitfalls and wumpus(They will kill you). \n 6)b represents breeze(Pit nearby). \n 7)s represents smell(Wumpus nearby) .\n 8)g represents glitter(Gold nearby). \n 9)Good Luck adventurer!!");

tf2.setBounds(30,430,400,200);

frame.add(tf2);

JButton actions= new JButton("Actions");

actions.setBounds(180,20,250,250);

frame.add(actions);

actions.setEnabled(false);

JButton up\_right= new JButton("Up-Right");

up\_right.setBounds(30,700,90,60);

frame.add(up\_right);

JButton up\_left= new JButton("Up-Left");

up\_left.setBounds(130,700,90,60);

frame.add(up\_left);

up\_right.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

String s1 = buttonsnew[index].getText();

String s3 = "";

num = index;

score-=1;

if (s1.contains("^"))

{

s3 = "^";

}

else if (s1.contains(">"))

{

s3 = ">";

}

else if (s1.contains("<"))

{

s3 = "<";

}

else

{

s3 = "v";

}

move\_up\_right mk = new move\_up\_right();

index = mk.move(index,num,score);

reveal(index);

if((buttonsnew[index].getText()).contains("P") ||(buttons[index].getText()).contains("W") )

{

tf.setText("Game over, you have lost!!!");

forward.setEnabled(false);

left.setEnabled(false);

right.setEnabled(false);

shoot.setEnabled(false);

up\_right.setEnabled(false);

up\_left.setEnabled(false);

String s8 = buttons[index].getText();

buttonsnew[num].setText(s1.replaceAll("[v,>,^,<]",""));

if(s8.contains(s3))

buttonsnew[index].setText(s8);

else

buttonsnew[index].setText(s3+s8);

buttonsnew[index].setBackground(Color.RED);

buttonsnew[index].setForeground(Color.BLACK);

}

else if(index == ladder)

{

buttonsnew[num].setText(s1.replaceAll("[v,>,^,<]",""));

index = rand.nextInt(16);

String s8 = buttons[index].getText();

if(s8.contains(s3))

buttonsnew[index].setText(s8);

else

buttonsnew[index].setText(s3+s8);

tf.setText(String.valueOf(score + "you found a magic ladder!!!"));

buttonsnew[index].setBackground(Color.RED);

buttonsnew[index].setForeground(Color.BLACK);

}

else if((buttonsnew[index].getText()).contains("G"))

{

tf.setText("CONGRATULATIONS , YOU HAVE WON!! Score is :" + String.valueOf(score));

forward.setEnabled(false);

left.setEnabled(false);

right.setEnabled(false);

shoot.setEnabled(false);

up\_right.setEnabled(false);

up\_left.setEnabled(false);

String s8 = buttons[index].getText();

buttonsnew[num].setText(s1.replaceAll("[v,>,^,<]",""));

if(s8.contains(s3))

buttonsnew[index].setText(s8);

else

buttonsnew[index].setText(s3+s8);

buttonsnew[index].setBackground(Color.RED);

buttonsnew[index].setForeground(Color.BLACK);

}

else

{

buttonsnew[num].setText(s1.replaceAll("[v,>,^,<]",""));

String s8 = buttons[index].getText();

if(s8.contains(s3))

buttonsnew[index].setText(s8);

else

buttonsnew[index].setText(s3+s8);

tf.setText(String.valueOf(score));

buttonsnew[index].setBackground(Color.RED);

buttonsnew[index].setForeground(Color.BLACK);

}

}

});

up\_left.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

String s1 = buttonsnew[index].getText();

score-=1;

String s3 = "";

num = index;

if (s1.contains("^"))

{

s3 = "^";

}

else if (s1.contains(">"))

{

s3 = ">";

}

else if (s1.contains("<"))

{

s3 = "<";

}

else

{

s3 = "v";

}

move\_up\_left mk = new move\_up\_left();

index = mk.move(index,num,score);

reveal(index);

if((buttonsnew[index].getText()).contains("P") ||(buttons[index].getText()).contains("W") )

{

tf.setText("Game over, you have lost!!!");

forward.setEnabled(false);

left.setEnabled(false);

right.setEnabled(false);

shoot.setEnabled(false);

up\_right.setEnabled(false);

up\_left.setEnabled(false);

String s8 = buttons[index].getText();

buttonsnew[num].setText(s1.replaceAll("[v,>,^,<]",""));

if(s8.contains(s3))

buttonsnew[index].setText(s8);

else

buttonsnew[index].setText(s3+s8);

buttonsnew[index].setBackground(Color.RED);

buttonsnew[index].setForeground(Color.BLACK);

}

else if(index == ladder)

{

buttonsnew[num].setText(s1.replaceAll("[v,>,^,<]",""));

index = rand.nextInt(16);

String s8 = buttons[index].getText();

if(s8.contains(s3))

buttonsnew[index].setText(s8);

else

buttonsnew[index].setText(s3+s8);

tf.setText(String.valueOf(score + "you found a magic ladder!!!"));

buttonsnew[index].setBackground(Color.RED);

buttonsnew[index].setForeground(Color.BLACK);

}

else if((buttonsnew[index].getText()).contains("G"))

{

tf.setText("CONGRATULATIONS , YOU HAVE WON!! Score is :" + String.valueOf(score));

forward.setEnabled(false);

left.setEnabled(false);

right.setEnabled(false);

shoot.setEnabled(false);

up\_right.setEnabled(false);

up\_left.setEnabled(false);

String s8 = buttons[index].getText();

buttonsnew[num].setText(s1.replaceAll("[v,>,^,<]",""));

if(s8.contains(s3))

buttonsnew[index].setText(s8);

else

buttonsnew[index].setText(s3+s8);

buttonsnew[index].setBackground(Color.RED);

buttonsnew[index].setForeground(Color.BLACK);

}

else

{

buttonsnew[num].setText(s1.replaceAll("[v,>,^,<]",""));

String s8 = buttons[index].getText();

if(s8.contains(s3))

buttonsnew[index].setText(s8);

else

buttonsnew[index].setText(s3+s8);

tf.setText(String.valueOf(score));

buttonsnew[index].setBackground(Color.RED);

buttonsnew[index].setForeground(Color.BLACK);

}

}

});

forward.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

move\_forward(tf,forward,shoot,left,right);

}

});

left.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

String s1 = buttonsnew[index].getText();

String replaceString = "";

//String s2 = "-";

if(s1.contains("^"))

{

replaceString=s1.replace("^","<");

}

else if(s1.contains(">"))

{

replaceString=s1.replace(">","^");

}

else if(s1.contains("<"))

{

replaceString=s1.replace("<","v");

}

else if(s1.contains("v"))

{

replaceString=s1.replace("v",">");

}

buttonsnew[index].setText(replaceString);

}

});

right.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

turn\_right();

}

});

shoot.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

String s1 = buttonsnew[index].getText();

String s2 = "";

String s3 = "";

int num;

score-=10;

if(index > 3 && s1.contains("^"))

{

num = index-4;

s2 = buttons[num].getText();

if(s2.contains("W"))

{

s3 = s2.replaceAll("W","");

tf1.setText("Congrats,Wumpus shot!!!");

buttons[num].setText(s3);

}

else

tf1.setText("No Wumpus present");

}

if(index % 4 != 0 && s1.contains("<"))

{

num = index-1;

s2 = buttons[num].getText();

if(s2.contains("W"))

{

s3 = s2.replaceAll("W","");

tf1.setText("Congrats,Wumpus shot!!!");

buttons[num].setText(s3);

}

else

tf1.setText("No Wumpus present");

}

if(index % 4 !=3 && s1.contains(">"))

{

num = index+1;

s2 = buttons[num].getText();

if(s2.contains("W"))

{

s3 = s2.replaceAll("W","");

tf1.setText("Congrats,Wumpus shot!!!");

buttons[num].setText(s3);

}

else

tf1.setText("No Wumpus present");

}

if(index <12 && s1.contains("v"))

{

num = index+4;

s2 = buttons[num].getText();

if(s2.contains("W"))

{

s3 = s2.replaceAll("W","");

tf1.setText("Congrats,Wumpus shot!!!");

buttons[num].setText(s3);

}

else

tf1.setText("No Wumpus present");

}}

});

frame.add(forward);

frame.add(left);

frame.add(right);

frame.add(shoot);

frame.add(tf);

frame.add(tf1);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(1300,900);

frame.setLayout(null);

frame.setVisible(true);

}

void reveal(int i) // Function that reveals the content of the square when landed upon

{

Icon t;

String s;

t = buttons[i].getIcon();

s = buttons[i].getText();

buttonsnew[i].setIcon(t);

buttonsnew[i].setText(s);

}

void move\_forward(JTextField tf,JButton forward,JButton shoot,JButton left,JButton right) // Function to move forward in the facing direction

{

String s1 = buttonsnew[index].getText();

String s3 = "";

num = index;

score-=1;

if(s1.contains("^"))

{

if(index<=3)

{

index+=4;

score+=1;

}

index-=4;

s3 = "^";

reveal(index);

}

if(s1.contains(">"))

{

if(index%4 == 3)

{

index-=1;

score+=1;

}

index+=1;

s3 = ">";

reveal(index);

}

if(s1.contains("<"))

{

if(index%4 == 0)

{

index+=1;

score+=1;

}

index-=1;

s3 = "<";

reveal(index);

}

if(s1.contains("v"))

{

if(index >11)

{

index-=4;

score+=1;

}

index+=4;

s3 = "v";

reveal(index);

}

if((buttonsnew[index].getText()).contains("P")||(buttons[index].getText()) .contains("W"))

{

tf.setText("Game over, you have lost!!!");

forward.setEnabled(false);

left.setEnabled(false);

right.setEnabled(false);

shoot.setEnabled(false);

String s8 = buttons[index].getText();

buttonsnew[num].setText(s1.replaceAll("[v,>,^,<]",""));

if(s8.contains(s3))

buttonsnew[index].setText(s8);

else

buttonsnew[index].setText(s3+s8);

buttonsnew[index].setBackground(Color.RED);

buttonsnew[index].setForeground(Color.BLACK);

}

else if(index == ladder)

{

buttonsnew[num].setText(s1.replaceAll("[v,>,^,<]",""));

index = rand.nextInt(16);

String s8 = buttons[index].getText();

if(s8.contains(s3))

buttonsnew[index].setText(s8);

else

buttonsnew[index].setText(s3+s8);

tf.setText(String.valueOf(score + "you found a magic ladder!!!"));

buttonsnew[index].setBackground(Color.RED);

buttonsnew[index].setForeground(Color.BLACK);

}

else if((buttonsnew[index].getText()).contains("G"))

{

tf.setText("CONGRATULATIONS , YOU HAVE WON!! Score is :" + String.valueOf(score));

forward.setEnabled(false);

left.setEnabled(false);

right.setEnabled(false);

shoot.setEnabled(false);

String s8 = buttons[index].getText();

buttonsnew[num].setText(s1.replaceAll("[v,>,^,<]",""));

if(s8.contains(s3))

buttonsnew[index].setText(s8);

else

buttonsnew[index].setText(s3+s8);

buttonsnew[index].setBackground(Color.RED);

buttonsnew[index].setForeground(Color.BLACK);

}

else

{

buttonsnew[num].setText(s1.replaceAll("[v,>,^,<]",""));

String s8 = buttons[index].getText();

if(s8.contains(s3))

buttonsnew[index].setText(s8);

else

buttonsnew[index].setText(s3+s8);

tf.setText(String.valueOf(score));

buttonsnew[index].setBackground(Color.RED);

buttonsnew[index].setForeground(Color.BLACK);

}

}

public void turn\_right() // inherited abstract method from actions\_available to turn right

{

move\_rightdirection mrd = new move\_rightdirection();

mrd.move(index,buttonsnew);

}

void setpos(int i,String s,Icon x,Icon y) //x is main image, y is effect of the image x

{

String s1,s2;

if(i < 12)

{

s2 = buttons[i+4].getText();

s1 = s2 + s;

buttons[i+4].setText(s1);

buttons[i+4].setIcon(y);

}

if(i % 4 != 3)

{

s2 = buttons[i+1].getText();

s1 = s2 + s;

buttons[i+1].setText(s1);

buttons[i+1].setIcon(y);

}

if(i % 4 != 0)

{

s2 = buttons[i-1].getText();

s1 = s2 + s;

buttons[i-1].setText(s1);

buttons[i-1].setIcon(y);

}

if(i > 3)

{

s2 = buttons[i-4].getText();

s1 = s2 + s;

buttons[i-4].setIcon(y);

buttons[i-4].setText(s1);

}

buttons[i].setIcon(x);

}

public static void main(String[] args) {

exp t1 = new exp(); //Creates object of exp

}

}

**2) Actions\_available.java**

package wumpus;

public interface Actions\_available {

public void turn\_right();

}

**3)move\_rightdirection.java**

package wumpus;

import javax.swing.JButton;

public class move\_rightdirection {

public move\_rightdirection()

{

}

public void move(int index,JButton buttonsnew[])

{

String s1 = buttonsnew[index].getText();

String replaceString = "";

if(s1.contains("^"))

{

replaceString=s1.replace("^",">");

}

else if(s1.contains(">"))

{

replaceString=s1.replace(">","v");

}

else if(s1.contains("<"))

{

replaceString=s1.replace("<","^");

}

else if(s1.contains("v"))

{

replaceString=s1.replace("v","<");

}

buttonsnew[index].setText(replaceString);

}

}

**4)ComboMovement.java**

package wumpus;

import java.util.Random;

public abstract class ComboMovement {

Random rand = new Random();

public abstract int move(int index,int num,int score);

}

**5) Move\_up\_right.java**

package wumpus;

public class move\_up\_right extends ComboMovement {

public int move(int index,int num,int score)

{

num = index;

if(index<=3)

{

index+=4;

}

index-=4;

if(index%4 == 3)

{

index-=1;

}

index+=1;

score-=1;

return index;

}

}

**6) Move\_up\_left.java**

package wumpus;

public class move\_up\_left extends ComboMovement {

public int move(int index,int num,int score) {

num = index;

if(index<=3)

{

index+=4;

}

index-=4;

if(index%4 == 0)

{

index+=1;

}

index-=1;

score-=1;

return index;

}

}

**4.RESULTS:**

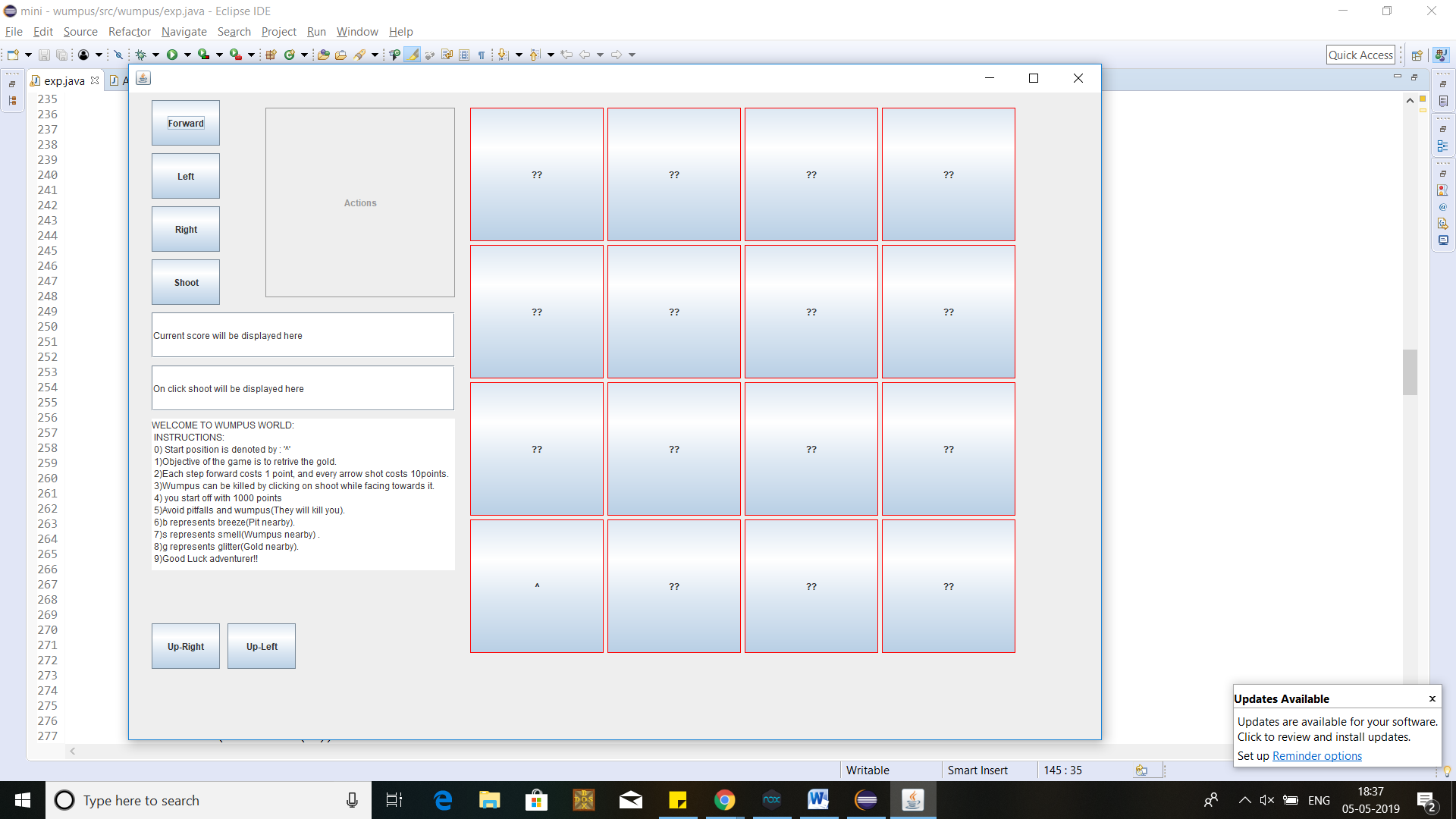
The game and its various elements were successfully implemented using JAVA frames and its components. Below are various snapshots of the game during different instances showing the various scenarios of the game.

Figure 1: Denotes the starting appearance of the game

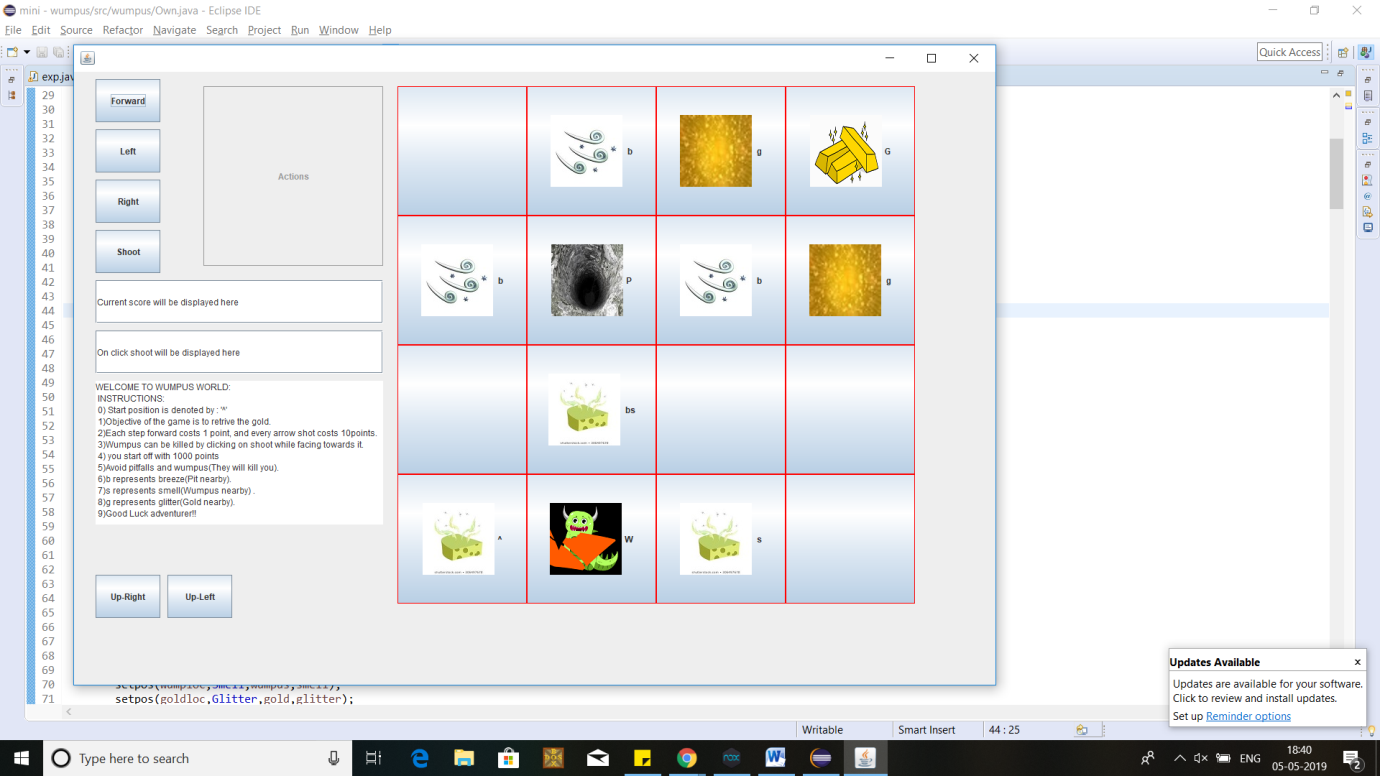


Figure 2: Denotes the board with each of the squares contents revealed

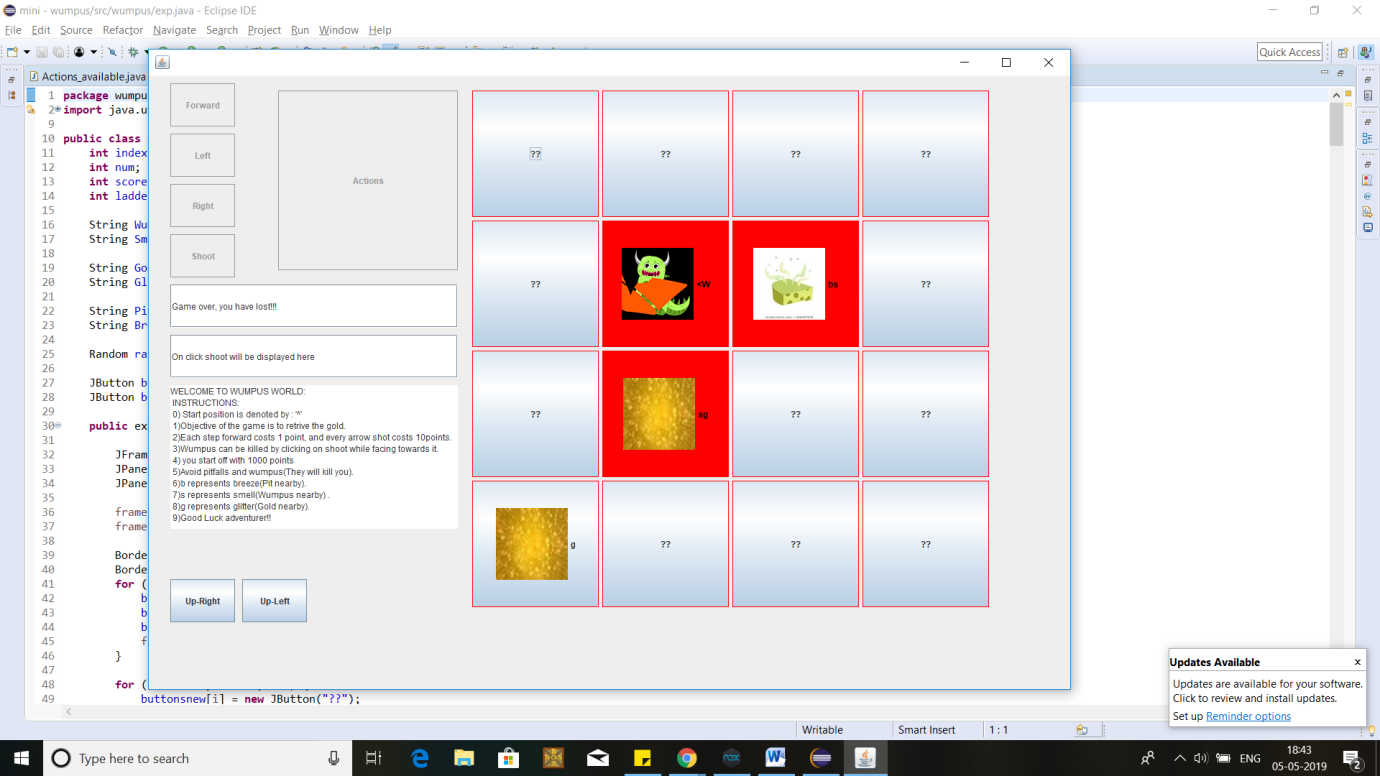


Figure 3: The game over message is displayed below the actions upon interaction with the Wumpus monster.

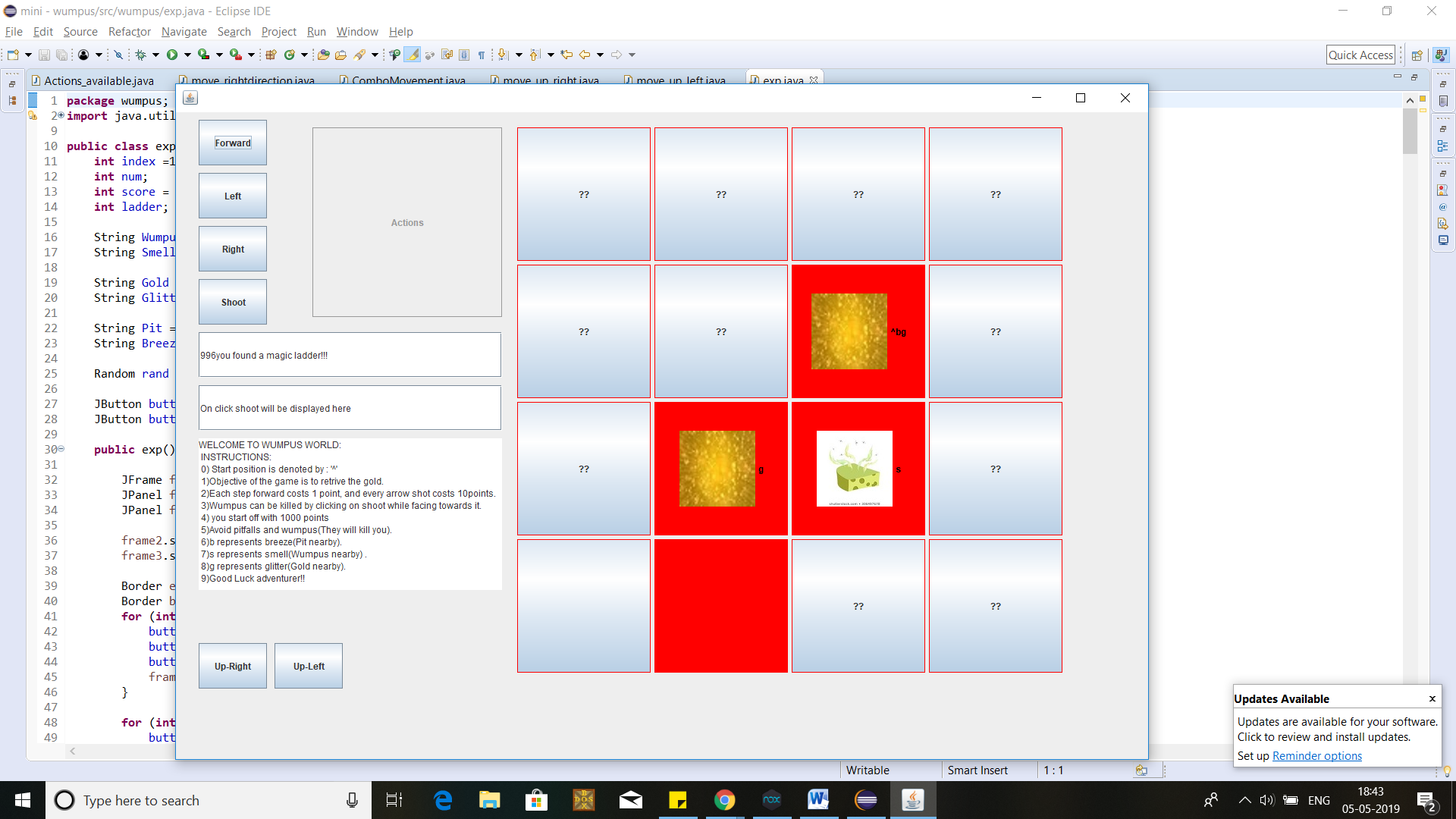


Figure 4: When a magic ladder is encountered, it is notified next to the score.

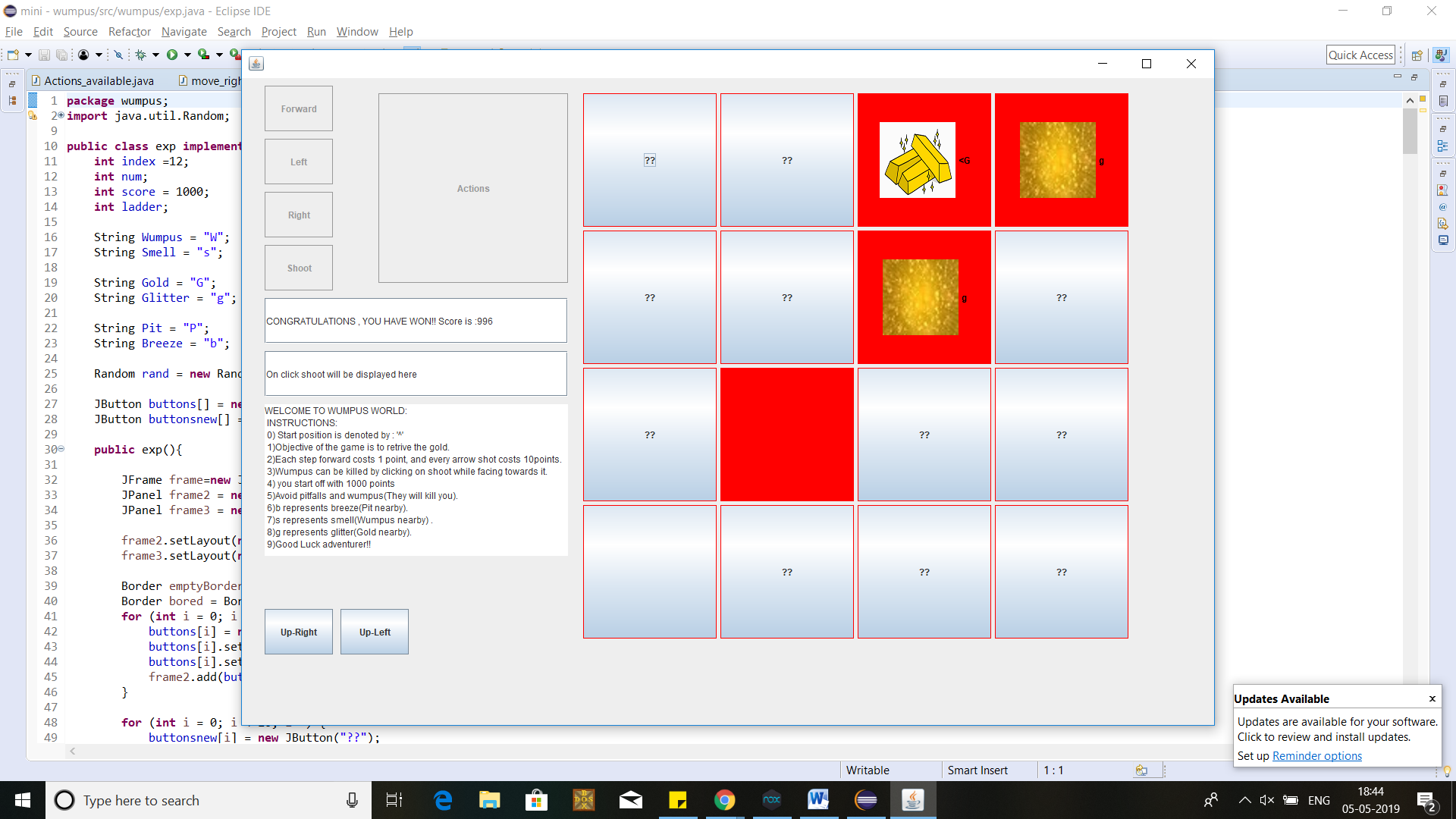


Figure 5: When you land on the gold tile, game is won and message along with final score is displayed.

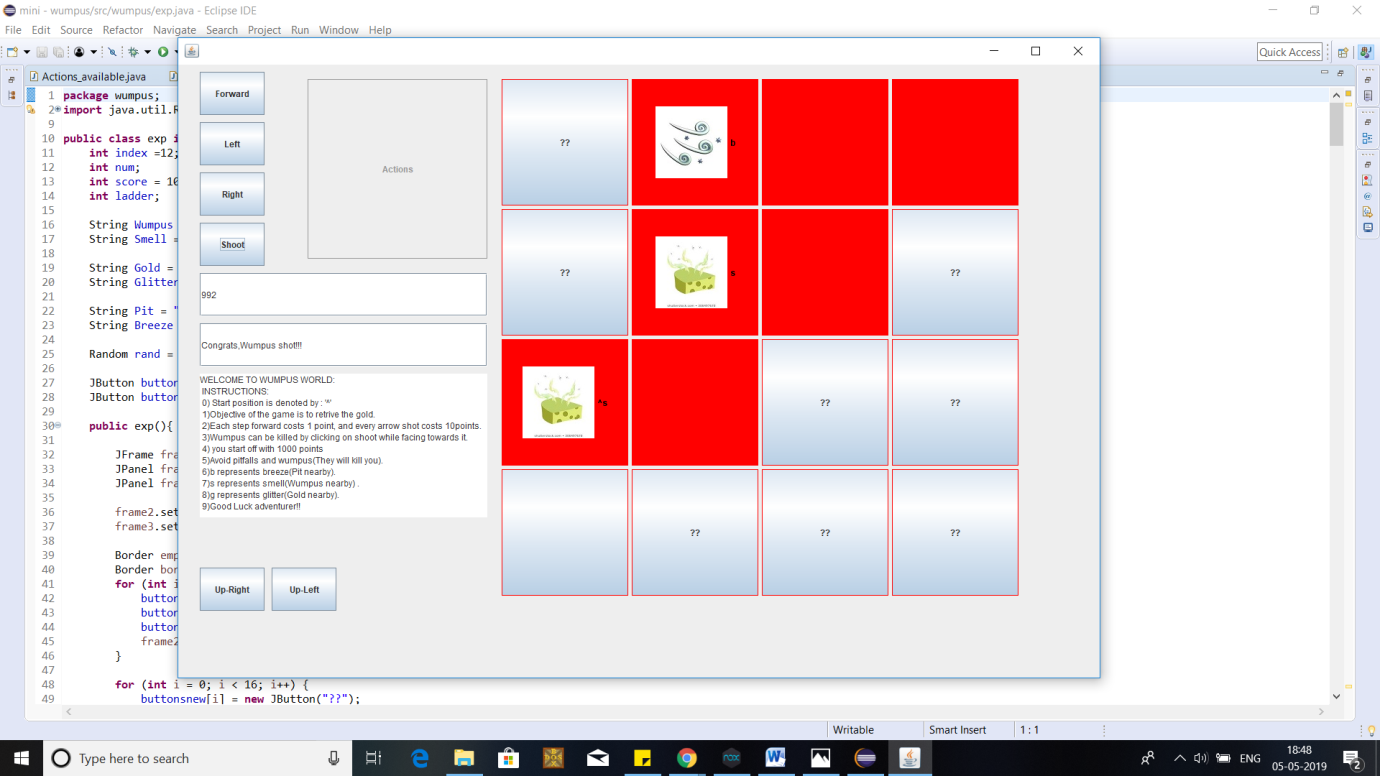


Figure 6: When you click on the shoot action and Wumpus is present, successfully shot message is displayed.