A

Project Report

On

**“Ludo Game”**

Submitted to

Amrutvahini Polytechnic, Sangamner

Department of Information Technology

In partial fulfillment of the requirement for the diploma in

Information Technology

Submitted By

64] Sahane Shreya Balasaheb (2100800094)

65]Tajane Sanika Sanjay (2100800095)

**Under The Guidance Of**

**Prof. Navale N.D.**



Amrutvahini Polytechnic, Sangamner

**(Approved by AICTE, NEW DELHI and affiliated to MSBTE)**

**2022-2023**

Amrutvahini Polytechnic, Sangamner

Department of Information Technology

 **CERTIFICATE**

**This is to that the project report entitled,**

**“Ludo Game”**

64] Sahane Shreya Balasaheb (2100800094)

65]Tajane Sanika Sanjay (2100800095)

It is benefited work carrier out by,

In partial fulfillment of the requirement for the diploma in

Information Technology

During the academic year 2022-2023

**Prof. Navale N.D Prof. Chaudhari N.K**

**(Project Guide) (H.O.D) IT**

**ACKNOWLEDGEMENT**

We have taken lots of efforts in this project. However, it would have been possible without the kind support and help of many individuals and organization. We would to kind to extend our sincere thanks to all of them.

First and foremost we want to thanks Prof. Chaudhari N.K H.O.D (Information Technology) Amrutvahini Polytechnic, Sangamner for giving us an opportunity to work on this project.

We are highly indebted to Prof. Navale N.D (Project Guide) for his guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in the project.

We would like to express our gratitude towards our parents and members of Information Technology department for their kind c-operation and encouragement which helps us in completion of this micro project.

Our thanks and appreciations also go to our colleagues in developing the micro-project and people who have willingly helped us with their abilities.

64] Sahane Shreya Balasaheb (2100800094)

65]Tajane Sanika Sanjay (2100800095)

**INDEX**

|  |  |
| --- | --- |
| **Sr.No** | **Contents** |
| 1. | Rationale |
| 2. | Aims and Benefits |
| 3. | Course Outcomes |
| 4. | Literature review |
| 5. | Actual Methodology |
| 6. | Actual resources used |
| 7. | Code |
| 8. | Output of the code |
| 9. | Skill Developed |
| 10. | Applications |

**MICROPROJECT**

**“Ludo Game”**

1. **Rationale:-**

Ludo is a strategy board game for two to four players, in which the players race their four tokens from start to finish according to the rolls of a single die. Like other cross and circle games, Ludo is derived from the Indian game Pachisi, but simpler. The game and its variations

are popular in many countries and under various names.

1. **Aims and benefits:-**

Ludo Game in Java

**Benefits**:-

1) Remain Stress Free.

2) Increase Logical Thinking Capability & Other Cognitive Skills.

3) Ludo online games bring joy and laughter.

1. **Course outcomes:-**

1) Develop program using GUI framework(AWT and Swing).

2) Handle events of AWT and Swing Components.

3) Develop programs to handle events in Java Programming .

4) Develop Java Programs using Networking Concepts.

1. **Literature review:**

**(INFORMATION DESCRIPTION)**

**1. Mouse Listener:-**

The Java MouseListener is notified whenever you change the state of mouse. It is notified against MouseEvent. The MouseListener interface is found in java.awt.event package. It has five methods.

1) mouseClicked()

2) mouseEntered()

3) mouseExited()

4) mousePressed()

5) mouseReleased()

**2. Mouse Motion Listener:-**

The Java MouseMotionListener is notified whenever you move or drag mouse. It is notified against MouseEvent. The MouseMotionListener interface is found in java.awt.event package.

**3. java.awt.event.\*; :-**

The java.awt.event package defines classes and interfaces used for event handling in the AWT and Swing. The members of this package fall into three categories: Events. The classes with names ending in "Event" represent specific types of events, generated by the AWT or by one of the AWT or Swing components.

**4. public void init():-**

A method is a named group of Java statements that can be called. It is similar to a subroutine or function in other programming languages. The term init() is a method name in Java. The name is followed by Java code within {and).

**5. Graph classes (graph):-**

Java implementation of the same is here on my blog. The Graph class represents an undirected graph of vertices named 0 through V-1. It supports the following two primary operations: add an edge to the graph, iterate over all of the vertices adjacent to a vertex.

1. **Actual methodology:**

1) We have decided to do Micro-project on develop a Ludo game in java.

2) We start the program code with the used of import java.awt.event.\* of this java package.

3) We get the used from getImage(getCodeBase()) of this function in our Ludo Game program.

4) For some condition expressions we used if else statement in our program. Also we used the Boolean expressions in our Ludo Game program.

5)Taking an output of the code and submit it.

1. **Actual resources used:**

**Software Requirements:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **Name of Resources** | **Specifications** | **Qty** |
| 1) | Operating System | Windows 10 | 1 |
| 2) | Software | JDK 1.8.0 | 1 |

1. **Program Code:-**

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

/\*

<applet code="Ludo" width=800 height=500>

</applet>

\*/

public class Ludo extends Applet implements MouseListener, MouseMotionListener

{

int playerone [][] = new int [45][6];

int playertwo [][] = new int [45][6];

int playerthree [][] = new int [45][6];

int playerfour [][] = new int [45][6];

int roll;

int width = 50;

int height = 50;

int Counter;

int xPos;

int yPos;

int XPos;

int YPos;

int winner;

int picNumber;

boolean playerturn;

boolean oneturn;

boolean twoturn;

boolean gameWon;

boolean counterMoved;

boolean Pressed;

Image img, img1, img2, img3, img4, img5, img6, img7, img8, img9, img10, img11, img12, img13, img14, img15, img16, img17, img18, img19, img20, img21, img22, img23, img24, img25, img26;

public void init()

{

setSize(700,700);

fillarray();

playerone[0][2] = 1;

playertwo[0][2] = 1;

playerthree[0][2] = 1;

playerfour[0][2] = 1;

Counter = 0;

addMouseListener(this);

winner = 0;

counterMoved = true;

Pressed = false;

picNumber = 0;

}

public void paint (Graphics graf)

{

img = getImage(getCodeBase(), "square.JPG");

img1 = getImage(getCodeBase(), "red.JPG");

img2 = getImage(getCodeBase(), "yellow.JPG");

img3 = getImage(getCodeBase(), "green.JPG");

img4 = getImage(getCodeBase(), "blue4.JPG");

img5 = getImage(getCodeBase(), "center.JPG");

img6 = getImage(getCodeBase(), "dice3.JPG");

img7 = getImage(getCodeBase(), "ludo.JPG");

img8 = getImage(getCodeBase(), "roll sign.JPG");

img9 = getImage(getCodeBase(), "one4.JPG");

img10 = getImage(getCodeBase(), "two4.JPG");

img11 = getImage(getCodeBase(), "three4.JPG");

img12 = getImage(getCodeBase(), "four4.JPG");

img13 = getImage(getCodeBase(), "five4.JPG");

img14 = getImage(getCodeBase(), "six4.JPG");

img15 = getImage(getCodeBase(), "gameover4.JPG");

img16 = getImage(getCodeBase(), "onewin4.JPG");

img17 = getImage(getCodeBase(), "twowin4.JPG");

img18 = getImage(getCodeBase(), "threewin.JPG");

img19 = getImage(getCodeBase(), "fourwin.JPG");

img20 = getImage(getCodeBase(), "redstart.JPG");

img21 = getImage(getCodeBase(), "yellowstart.JPG");

img22 = getImage(getCodeBase(), "greenstart.JPG");

img23 = getImage(getCodeBase(), "bluestart.JPG");

img24 = getImage(getCodeBase(), "yellow roll.JPG");

img25 = getImage(getCodeBase(), "green roll.JPG");

img26 = getImage(getCodeBase(), "blue roll.JPG");

for(int i=0; i<=39; i++)

{

//------------------- draw grid -----------------------------

graf.drawImage(img, playerone[i][0], playerone[i][1], this);

}

for(int i=40; i<=43; i++)

{

graf.drawImage(img1, playerone[i][0], playerone[i][1], this);

graf.drawImage(img2, playertwo[i][0], playertwo[i][1], this);

graf.drawImage(img3, playerthree[i][0], playerthree[i][1], this);

graf.drawImage(img4, playerfour[i][0], playerfour[i][1], this);

}

graf.drawImage(img5, 300, 300, this);

graf.drawImage(img7, 100, 100, this);

graf.drawImage(img20, 50, 250, this);

graf.drawImage(img21, 250, 550, this);

graf.drawImage(img22, 550, 350, this);

graf.drawImage(img23, 350, 50, this);

if(gameWon == false)

{

graf.drawImage(img6, 450, 100, this);

}

if(counterMoved == false)

{

graf.drawImage(img8, 400, 400, this);

}

if(picNumber == 1)

{

graf.drawImage(img8, 400, 400, this);

}

if(picNumber == 2)

{

graf.drawImage(img24, 400, 400, this);

}

if(picNumber == 3)

{

graf.drawImage(img25, 400, 400, this);

}

if(picNumber == 4)

{

graf.drawImage(img26, 400, 400, this);

}

//------------------- draw counters -----------------------------

for (int i=0; i<=43; i++)

{

if(playerone[i][3] == 1 & counterMoved == false)

{

graf.setColor(Color.red);

graf.fillOval(playerone[i][0] + 10, playerone[i][1] + 10,30,30);

}

if(playerone[i][2] == 1 & counterMoved == true)

{

graf.setColor(Color.red);

graf.fillOval(playerone[i][0] + 10, playerone[i][1] + 10,30,30);

}

if(playertwo[i][2] == 1)

{

graf.setColor(Color.yellow);

graf.fillOval(playertwo[i][0] + 10, playertwo[i][1] + 10,30,30);

}

if(playerthree[i][2] == 1)

{

graf.setColor(Color.green);

graf.fillOval(playerthree[i][0] + 10, playerthree[i][1] + 10,30,30);

}

if(playerfour[i][2] == 1)

{

graf.setColor(Color.blue);

graf.fillOval(playerfour[i][0] + 10, playerfour[i][1] + 10,30,30);

}

}

if(playerone[44][2] == 1 & counterMoved == true)

{

graf.setColor(Color.red);

graf.fillOval(298, 298,30,30);

}

if(playertwo[44][2] == 1)

{

graf.setColor(Color.yellow);

graf.fillOval(298, 322,30,30);

}

if(playerthree[44][2] == 1)

{

graf.setColor(Color.green);

graf.fillOval(322, 322,30,30);

}

if(playerfour[44][2] == 1)

{

graf.setColor(Color.blue);

graf.fillOval(322, 298,30,30);

}

//---------------------- Draw dice roll --------------------------

if(roll == 1)

{

graf.drawImage(img9, 450, 455, this);

}

if(roll == 2)

{

graf.drawImage(img10, 450, 455, this);

}

if(roll == 3)

{

graf.drawImage(img11, 450, 455, this);

}

if(roll == 4)

{

graf.drawImage(img12, 450, 455, this);

}

if(roll == 5)

{

graf.drawImage(img13, 450, 455, this);

}

if(roll == 6)

{

graf.drawImage(img14, 450, 455, this);

}

//---------------------- display winner --------------------------

if(winner == 1 & counterMoved == true)

{

graf.drawImage(img16, 50, 425, this);

}

if(winner == 2)

{

graf.drawImage(img17, 50, 425, this);

}

if(winner == 3)

{

graf.drawImage(img18, 50, 425, this);

}

if(winner == 4)

{

graf.drawImage(img19, 50, 425, this);

}

if(gameWon == true & counterMoved == true)

{

graf.drawImage(img15, 450, 100, this);

graf.setColor(Color.white);

graf.fillRect(400,400,400,400);

picNumber = 0;

}

}

//---------------------- Create grid --------------------------

public void fillarray()

{

int xCoord = 50;

int yCoord = 250;

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

{

playerone[i][0] = xCoord;

playerone[i][1] = yCoord;

playertwo[i+10][0] = xCoord;

playertwo[i+10][1] = yCoord;

playerthree[i+20][0] = xCoord;

playerthree[i+20][1] = yCoord;

playerfour[i+30][0] = xCoord;

playerfour[i+30][1] = yCoord;

xCoord = xCoord + 50;

}

yCoord = yCoord - 50;

xCoord = xCoord - 50;

for(int i=5; i<=8; i++)

{

playerone[i][0] = xCoord;

playerone[i][1] = yCoord;

playertwo[i+10][0] = xCoord;

playertwo[i+10][1] = yCoord;

playerthree[i+20][0] = xCoord;

playerthree[i+20][1] = yCoord;

playerfour[i+30][0] = xCoord;

playerfour[i+30][1] = yCoord;

yCoord = yCoord - 50;

}

playerone[9][0] = 300;

playertwo[19][0] = 300;

playerthree[29][0] = 300;

playerfour[39][0] = 300;

playerone[9][1] = 50;

playertwo[19][1] = 50;

playerthree[29][1] = 50;

playerfour[39][1] = 50;

yCoord = 50;

xCoord = 350;

for(int i=10; i<=14; i++)

{

playerone[i][0] = xCoord;

playerone[i][1] = yCoord;

playertwo[i+10][0] = xCoord;

playertwo[i+10][1] = yCoord;

playerthree[i+20][0] = xCoord;

playerthree[i+20][1] = yCoord;

playerfour[i-10][0] = xCoord;

playerfour[i-10][1] = yCoord;

yCoord = yCoord + 50;

}

yCoord = 250;

xCoord = 400;

for(int i=15; i<=18; i++)

{

playerone[i][0] = xCoord;

playerone[i][1] = yCoord;

playertwo[i+10][0] = xCoord;

playertwo[i+10][1] = yCoord;

playerthree[i+20][0] = xCoord;

playerthree[i+20][1] = yCoord;

playerfour[i-10][0] = xCoord;

playerfour[i-10][1] = yCoord;

xCoord = xCoord + 50;

}

playerone[19][0] = 550;

playertwo[29][0] = 550;

playerthree[39][0] = 550;

playerfour[9][0] = 550;

playerone[19][1] = 300;

playertwo[29][1] = 300;

playerthree[39][1] = 300;

playerfour[9][1] = 300;

xCoord = 550;

yCoord = 350;

for(int i=20; i<=24; i++)

{

playerone[i][0] = xCoord;

playerone[i][1] = yCoord;

playertwo[i+10][0] = xCoord;

playertwo[i+10][1] = yCoord;

playerthree[i-20][0] = xCoord;

playerthree[i-20][1] = yCoord;

playerfour[i-10][0] = xCoord;

playerfour[i-10][1] = yCoord;

xCoord = xCoord - 50;

}

xCoord = 350;

yCoord = 400;

for(int i=25; i<=28; i++)

{

playerone[i][0] = xCoord;

playerone[i][1] = yCoord;

playertwo[i+10][0] = xCoord;

playertwo[i+10][1] = yCoord;

playerthree[i-20][0] = xCoord;

playerthree[i-20][1] = yCoord;

playerfour[i-10][0] = xCoord;

playerfour[i-10][1] = yCoord;

yCoord = yCoord + 50;

}

playerone[29][0] = 300;

playertwo[39][0] = 300;

playerthree[9][0] = 300;

playerfour[19][0] = 300;

playerone[29][1] = 550;

playertwo[39][1] = 550;

playerthree[9][1] = 550;

playerfour[19][1] = 550;

xCoord = 250;

yCoord = 550;

for(int i=30; i<=34; i++)

{

playerone[i][0] = xCoord;

playerone[i][1] = yCoord;

playertwo[i-30][0] = xCoord;

playertwo[i-30][1] = yCoord;

playerthree[i-20][0] = xCoord;

playerthree[i-20][1] = yCoord;

playerfour[i-10][0] = xCoord;

playerfour[i-10][1] = yCoord;

yCoord = yCoord - 50;

}

xCoord = 200;

yCoord = 350;

for(int i=35; i<=38; i++)

{

playerone[i][0] = xCoord;

playerone[i][1] = yCoord;

playertwo[i-30][0] = xCoord;

playertwo[i-30][1] = yCoord;

playerthree[i-20][0] = xCoord;

playerthree[i-20][1] = yCoord;

playerfour[i-10][0] = xCoord;

playerfour[i-10][1] = yCoord;

xCoord = xCoord - 50;

}

playerone[39][0] = 50;

playertwo[9][0] = 50;

playerthree[19][0] = 50;

playerfour[29][0] = 50;

playerone[39][1] = 300;

playertwo[9][1] = 300;

playerthree[19][1] = 300;

playerfour[29][1] = 300;

xCoord = 100;

yCoord = 300;

for(int i=40; i<=44; i++)

{

playerone[i][0] = xCoord;

playerone[i][1] = yCoord;

xCoord = xCoord + 50;

}

xCoord = 300;

yCoord = 500;

for(int i=40; i<=44; i++)

{

playertwo[i][0] = xCoord;

playertwo[i][1] = yCoord;

yCoord = yCoord - 50;

}

xCoord = 500;

yCoord = 300;

for(int i=40; i<=44; i++)

{

playerthree[i][0] = xCoord;

playerthree[i][1] = yCoord;

xCoord = xCoord - 50;

}

xCoord = 300;

yCoord = 100;

for(int i=40; i<=44; i++)

{

playerfour[i][0] = xCoord;

playerfour[i][1] = yCoord;

yCoord = yCoord + 50;

}

}

private void setValues (int x, int y)

{

xPos = x;

yPos = y;

}

private void setvalues (int x, int y)

{

XPos = x;

YPos = y;

rollDice();

repaint();

}

public void mouseClicked(MouseEvent e)

{

}

public void mousePressed(MouseEvent e)

{

setValues(e.getX(), e.getY() );

}

public void mouseReleased(MouseEvent e)

{

setvalues(e.getX(), e.getY() );

}

public void mouseEntered(MouseEvent e)

{

}

public void mouseExited(MouseEvent e)

{

}

public void mouseDragged(MouseEvent e)

{

}

public void mouseMoved(MouseEvent e)

{

}

public void rollDice()

{

if (xPos >= 450 & xPos <= 550 & yPos >= 100 & yPos <= 200)

{

if(Counter == 0)

{

for(int i=0; i<playerone.length; i++)

{

playerone[i][3] = 0;

if(playerone[i][2] == 1)

{

playerone[i][3] = 1;

}

}

//-------------------- Making the move -------------------------

roll = 1 + (int)(Math.random() \* 6);

picNumber = 1;

oneturn = true;

for(int i=39; i<45; i++)

{

if (playerone[39][2] == 1)

{

if (roll<6)

{

playerone[39][2] = 0;

playerone[39+roll][2] = 1;

if(gameWon == false)

{

counterMoved = false;

}

}

oneturn = false;

}

else

if (playerone[40][2] == 1 & oneturn == true)

{

if (roll<5)

{

playerone[40][2] = 0;

playerone[40+roll][2] = 1;

if(gameWon == false)

{

counterMoved = false;

}

}

oneturn = false;

}

else

if (playerone[41][2] == 1 & oneturn == true)

{

if (roll<4)

{

playerone[41][2] = 0;

playerone[41+roll][2] = 1;

if(gameWon == false)

{

counterMoved = false;

}

}

oneturn = false;

}

else

if (playerone[42][2] == 1 & oneturn == true)

{

if (roll<3)

{

playerone[42][2] = 0;

playerone[42+roll][2] = 1;

if(gameWon == false)

{

counterMoved = false;

}

}

oneturn = false;

}

else

if (playerone[43][2] == 1 & oneturn == true)

{

if (roll<2)

{

playerone[43][2] = 0;

playerone[43+roll][2] = 1;

if(gameWon == false)

{

counterMoved = false;

}

}

oneturn = false;

}

else

if (playerone[44][2] == 1 & oneturn == true)

{

oneturn = false;

}

}

if(oneturn == true)

{

for(int i=43; i>=0; i--)

{

if (playerone[i][2] == 1)

{

playerone[i][2] = 0;

playerone[i+roll][2] = 1;

if(gameWon == false)

{

counterMoved = false;

}

}

}

}

if(playertwo[44][2] == 1 & counterMoved == true)

{

Counter = 2;

}

if(playertwo[44][2] == 1 & playerthree[44][2] == 1 & counterMoved == true)

{

Counter = 3;

}

if(playertwo[44][2] == 1 & playerthree[44][2] == 1 & playerfour[44][2] == 1 & counterMoved == true)

{

Counter = 0;

}

if(counterMoved == false)

{

Counter = 5;

}

if(counterMoved == true & playertwo[44][2] != 1)

{

Counter = 1;

}

}

else

if(Counter == 1)

{

roll = 1 + (int)(Math.random() \* 6);

picNumber = 2;

oneturn = true;

for(int i=39; i<45; i++)

{

if (playertwo[39][2] == 1)

{

if (roll<6)

{

playertwo[39][2] = 0;

playertwo[39+roll][2] = 1;

}

oneturn = false;

}

else

if (playertwo[40][2] == 1 & oneturn == true)

{

if (roll<5)

{

playertwo[40][2] = 0;

playertwo[40+roll][2] = 1;

}

oneturn = false;

}

else

if (playertwo[41][2] == 1 & oneturn == true)

{

if (roll<4)

{

playertwo[41][2] = 0;

playertwo[41+roll][2] = 1;

}

oneturn = false;

}

else

if (playertwo[42][2] == 1 & oneturn == true)

{

if (roll<3)

{

playertwo[42][2] = 0;

playertwo[42+roll][2] = 1;

}

oneturn = false;

}

else

if (playertwo[43][2] == 1 & oneturn == true)

{

if (roll<2)

{

playertwo[43][2] = 0;

playertwo[43+roll][2] = 1;

}

oneturn = false;

}

else

if (playertwo[44][2] == 1 & oneturn == true)

{

oneturn = false;

}

}

if(oneturn == true)

{

for(int i=43; i>=0; i--)

{

if (playertwo[i][2] == 1)

{

playertwo[i][2] = 0;

playertwo[i+roll][2] = 1;

}

}

for(int i=0; i<44; i++)

{

if(playertwo[i][2] == 1)

{

for(int j=0; j<playertwo.length; j++)

{

if(playerone[j][2] == 1)

{

if(playertwo[i][0] == playerone[j][0] & playertwo[i][1] == playerone[j][1])

{

playerone[j][2] = 0;

playerone[0][2] = 1;

}

}

}

for(int j=0; j<playerthree.length; j++)

{

if(playerthree[j][2] == 1)

{

if(playertwo[i][0] == playerthree[j][0] & playertwo[i][1] == playerthree[j][1])

{

playerthree[j][2] = 0;

playerthree[0][2] = 1;

}

}

}

for(int j=0; j<playerfour.length; j++)

{

if(playerfour[j][2] == 1)

{

if(playertwo[i][0] == playerfour[j][0] & playertwo[i][1] == playerfour[j][1])

{

playerfour[j][2] = 0;

playerfour[0][2] = 1;

}

}

}

}

}

}

Counter = 2;

if(playerthree[44][2] == 1)

{

Counter = 3;

}

if(playerthree[44][2] == 1 & playerfour[44][2] == 1)

{

Counter = 0;

}

if(playerthree[44][2] == 1 & playerfour[44][2] == 1 & playerone[44][2] == 1)

{

Counter = 1;

if(playertwo[44][2] == 1)

{

if(playerthree[44][2] != 1)

{

Counter = 2;

}

else

if(playerfour[44][2] != 1)

{

Counter = 3;

}

else

Counter = 0;

}

}

}

else

if(Counter == 2)

{

roll = 1 + (int)(Math.random() \* 6);

picNumber = 3;

oneturn = true;

for(int i=39; i<45; i++)

{

if (playerthree[39][2] == 1 & oneturn == true)

{

if (roll<6)

{

playerthree[39][2] = 0;

playerthree[39+roll][2] = 1;

}

oneturn = false;

}

else

if (playerthree[40][2] == 1 & oneturn == true)

{

if (roll<5)

{

playerthree[40][2] = 0;

playerthree[40+roll][2] = 1;

}

oneturn = false;

}

else

if (playerthree[41][2] == 1 & oneturn == true)

{

if (roll<4)

{

playerthree[41][2] = 0;

playerthree[41+roll][2] = 1;

}

oneturn = false;

}

else

if (playerthree[42][2] == 1 & oneturn == true)

{

if (roll<3)

{

playerthree[42][2] = 0;

playerthree[42+roll][2] = 1;

}

oneturn = false;

}

else

if (playerthree[43][2] == 1 & oneturn == true)

{

if (roll<2)

{

playerthree[43][2] = 0;

playerthree[43+roll][2] = 1;

}

oneturn = false;

}

else

if (playerthree[44][2] == 1 & oneturn == true)

{

oneturn = false;

}

}

if(oneturn == true)

{

for(int i=43; i>=0; i--)

{

if (playerthree[i][2] == 1)

{

playerthree[i][2] = 0;

playerthree[i+roll][2] = 1;

}

}

for(int i=0; i<44; i++)

{

if(playerthree[i][2] == 1)

{

for(int j=0; j<playertwo.length; j++)

{

if(playertwo[j][2] == 1)

{

if(playerthree[i][0] == playertwo[j][0] & playerthree[i][1] == playertwo[j][1])

{

playertwo[j][2] = 0;

playertwo[0][2] = 1;

}

}

}

for(int j=0; j<playerone.length; j++)

{

if(playerone[j][2] == 1)

{

if(playerthree[i][0] == playerone[j][0] & playerthree[i][1] == playerone[j][1])

{

playerone[j][2] = 0;

playerone[0][2] = 1;

}

}

}

for(int j=0; j<playerfour.length; j++)

{

if(playerfour[j][2] == 1)

{

if(playerthree[i][0] == playerfour[j][0] & playerthree[i][1] == playerfour[j][1])

{

playerfour[j][2] = 0;

playerfour[0][2] = 1;

}

}

}

}

}

}

Counter = 3;

if(playerfour[44][2] == 1)

{

Counter = 0;

}

if(playerfour[44][2] == 1 & playerone[44][2] == 1)

{

Counter = 1;

}

if(playerfour[44][2] == 1 & playerone[44][2] == 1 & playertwo[44][2] == 1)

{

Counter = 2;

}

}

else

if(Counter == 3)

{

roll = 1 + (int)(Math.random() \* 6);

picNumber = 4;

oneturn = true;

for(int i=39; i<45; i++)

{

if (playerfour[39][2] == 1 & oneturn == true)

{

if (roll<6)

{

playerfour[39][2] = 0;

playerfour[39+roll][2] = 1;

}

oneturn = false;

}

else

if (playerfour[40][2] == 1 & oneturn == true)

{

if (roll<5)

{

playerfour[40][2] = 0;

playerfour[40+roll][2] = 1;

}

oneturn = false;

}

else

if (playerfour[41][2] == 1 & oneturn == true)

{

if (roll<4)

{

playerfour[41][2] = 0;

playerfour[41+roll][2] = 1;

}

oneturn = false;

}

else

if (playerfour[42][2] == 1 & oneturn == true)

{

if (roll<3)

{

playerfour[42][2] = 0;

playerfour[42+roll][2] = 1;

}

oneturn = false;

}

else

if (playerfour[43][2] == 1 & oneturn == true)

{

if (roll<2)

{

playerfour[43][2] = 0;

playerfour[43+roll][2] = 1;

}

oneturn = false;

}

else

if (playerfour[44][2] == 1 & oneturn == true)

{

oneturn = false;

}

}

if(oneturn == true)

{

for(int i=43; i>=0; i--)

{

if (playerfour[i][2] == 1)

{

playerfour[i][2] = 0;

playerfour[i+roll][2] = 1;

}

}

for(int i=0; i<44; i++)

{

if(playerfour[i][2] == 1)

{

for(int j=0; j<playertwo.length; j++)

{

if(playertwo[j][2] == 1)

{

if(playerfour[i][0] == playertwo[j][0] & playerfour[i][1] == playertwo[j][1])

{

playertwo[j][2] = 0;

playertwo[0][2] = 1;

}

}

}

for(int j=0; j<playerthree.length; j++)

{

if(playerthree[j][2] == 1)

{

if(playerfour[i][0] == playerthree[j][0] & playerfour[i][1] == playerthree[j][1])

{

playerthree[j][2] = 0;

playerthree[0][2] = 1;

}

}

}

for(int j=0; j<playerone.length; j++)

{

if(playerone[j][2] == 1)

{

if(playerfour[i][0] == playerone[j][0] & playerfour[i][1] == playerone[j][1])

{

playerone[j][2] = 0;

playerone[0][2] = 1;

}

}

}

}

}

}

Counter = 0;

if(playerone[44][2] == 1)

{

Counter = 1;

}

if(playerone[44][2] == 1 & playertwo[44][2] == 1)

{

Counter = 2;

}

if(playerone[44][2] == 1 & playertwo[44][2] == 1 & playerthree[44][2] == 1)

{

Counter = 3;

}

}

}

if(Counter == 5)

{

for(int i=0; i<playerone.length; i++)

{

if(playerone[i][3] == 1)

{

if(xPos >= playerone[i][0] & xPos <= playerone[i][0]+50 & yPos >= playerone[i][1] & yPos <= playerone[i][1]+50)

{

Pressed = true;

}

}

if(playerone[i][2] == 1)

{

if(XPos >= playerone[i][0] & XPos <= playerone[i][0]+50 & YPos >= playerone[i][1] & YPos <= playerone[i][1]+50)

{

counterMoved = true;

Pressed = false;

Counter = 1;

}

if(counterMoved == true) //If ontop send bottom counter back to start

{

for(int j=0; j<44; j++)

{

if(playertwo[j][2] == 1)

{

if(playerone[i][0] == playertwo[j][0] & playerone[i][1] == playertwo[j][1])

{

playertwo[j][2] = 0;

playertwo[0][2] = 1;

}

}

}

for(int j=0; j<44; j++)

{

if(playerthree[j][2] == 1)

{

if(playerone[i][0] == playerthree[j][0] & playerone[i][1] == playerthree[j][1])

{

playerthree[j][2] = 0;

playerthree[0][2] = 1;

}

}

}

for(int j=0; j<44; j++)

{

if(playerfour[j][2] == 1)

{

if(playerone[i][0] == playerfour[j][0] & playerone[i][1] == playerfour[j][1])

{

playerfour[j][2] = 0;

playerfour[0][2] = 1;

}

}

}

}

}

}

}

if(playerone[44][2] == 1 & winner == 0)

{

winner = 1;

}

if(playertwo[44][2] == 1 & winner == 0)

{

winner = 2;

}

if(playerthree[44][2] == 1 & winner == 0)

{

winner = 3;

}

if(playerfour[44][2] == 1 & winner == 0)

{

winner = 4;

}

if(playerone[44][2] == 1 & playertwo[44][2] == 1 & playerthree[44][2] == 1 & playerfour[44][2] == 1)

{

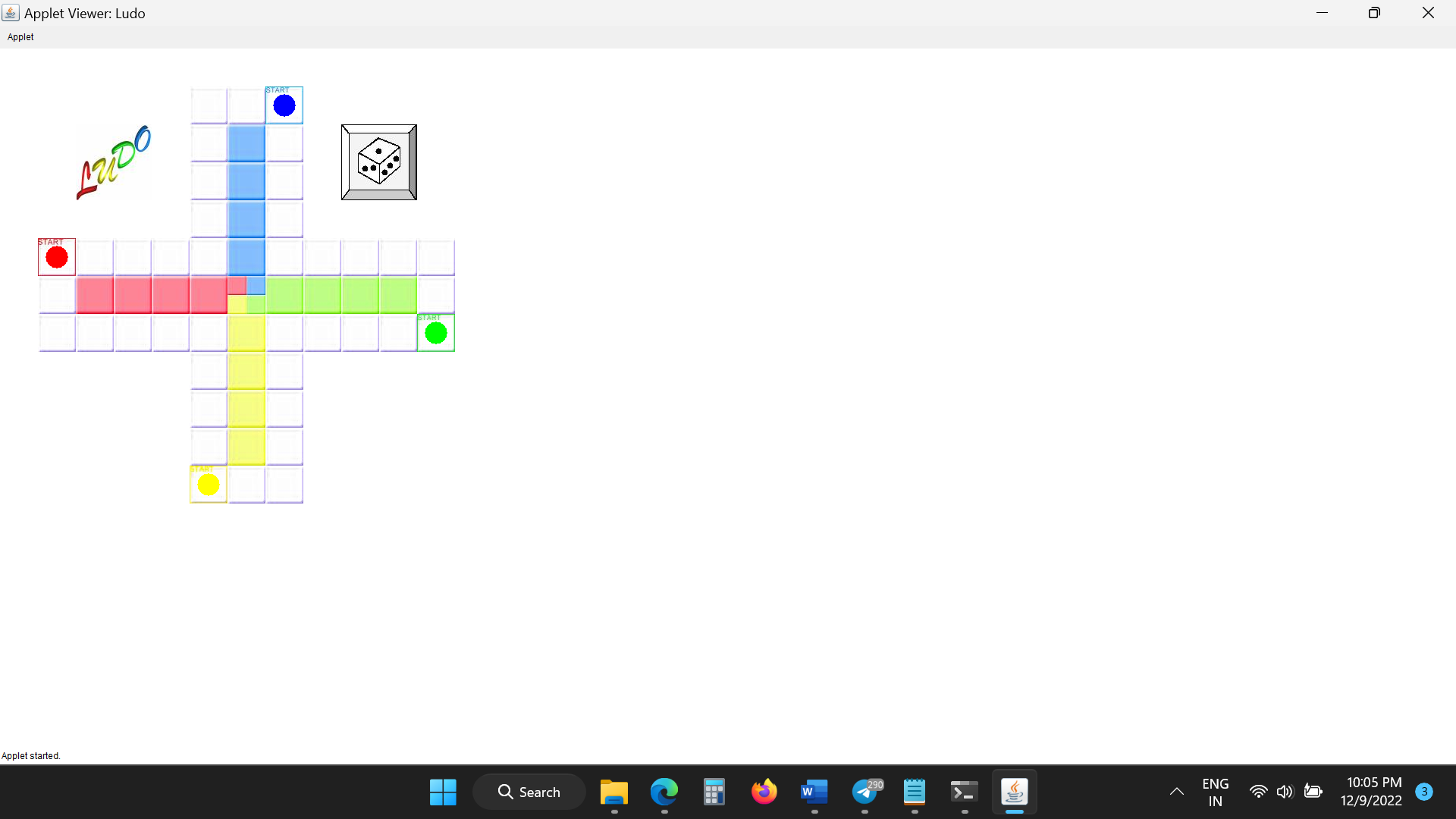
gameWon = true;

}

}

}

1. **Output of the Code:**

****

1. **Skill Developed:**
2. From this project we learn the Mouse Listener, Mouse Motion Listener
3. From this project we learn the Mouse Listener, Mouse Motion Listener.
4. The demo is made more interactive with a mouse interaction module in the program.
5. **Application :**

It is used for customer entertainment purpose.