import java.awt.\*;

import java.awt.event.\*;

/\*\*

\* This class holds the major part of application

\*/

public class TicTacToe implements MouseListener, ActionListener, MouseMotionListener

{

Frame f, sub;

int n, m, i = 0; // n- Ycoordinate m- Xcoordinate flag- Player turn

char ch[] = new char[9];

Graphics g;

Graphics2D g2;

int flag = 2;

/\*\*

\* Constructor that initialize the Frame object and add

\* different listeners to this Frame object

\*/

public TicTacToe()

{

f = new Frame("Tic Tac Toe");

f.setLayout(null);

f.setVisible(true);

f.setSize(600, 600);

f.setBackground(Color.black);

g = f.getGraphics();

g.setFont(new Font("default", Font.BOLD, 20));

//g2 = (Graphics2D)g;

for(i=0; i<9; i++)

ch[i] = 'B';

f.addMouseMotionListener(this);

f.addMouseListener(this);

f.addWindowListener(new WHandler(f));

}

/\*\*

\* This method is executed for every click on the mouse.<br>

\* It is used to draw the Oval and 'X' mark on the frame with respone to user click

\* @param e to hold the object of MouseEvent

\*/

public void mouseClicked(MouseEvent e)

{

m = -1;

n = -1;

flag--;

int x = e.getX();

int y = e.getY();

if(flag == 1)

{

if(x < 200 && y < 200 && ch[0] == 'B')

{

m = 0;

n = 0;

ch[0] ='1';

}

else if((x > 200 && x < 400) && (y < 200) && ch[1] == 'B')

{

m=200;

n=0;

ch[1]='1';

}

else if((x > 400 && x < 600) && (y < 200) && ch[2] == 'B')

{

m = 400;

n = 0;

ch[2] = '1';

}

else if(x < 200 && (y > 200 && y < 400) && ch[3] == 'B')

{

m = 0;

n = 200;

ch[3] = '1';

}

else if((x > 200 && x < 400) && (y > 200 && y < 400) && ch[4] == 'B')

{

m = 200;

n = 200;

ch[4] = '1';

}

else if((x > 400 && x < 600) && (y > 200 && y < 400) && ch[5] == 'B')

{

m = 400;

n = 200;

ch[5] = '1';

}

else if(x < 200 && (y > 400 && y < 600) && ch[6] == 'B')

{

m = 0;

n = 400;

ch[6] = '1';

}

else if((x > 200 && x < 400) && (y > 400 && y < 600) && ch[7] == 'B')

{

m = 200;

n = 400;

ch[7] = '1';

}

else if((x > 400 && x < 600) && (y > 400 && y < 600) && ch[8] == 'B')

{

m = 400;

n = 400;

ch[8] = '1';

}

else

{

m = -1;

n = -1;

flag = 2;

}

if(m != -1 && n != -1)

{

g.setColor(Color.red);

g.drawLine(m, n, m + 199, n + 199);

g.drawLine(m + 199, n, m, n + 199);

}

}

if(flag == 0)

{

if(x < 200 && y < 200 && ch[0] == 'B')

{

m = 0;

n = 20;

ch[0] = '2';

}

else if((x > 200 && x < 400) && (y < 200) && ch[1] == 'B')

{

m = 200;

n = 20;

ch[1] = '2';

}

else if((x > 400 && x < 600) && (y < 200) && ch[2] == 'B')

{

m = 400;

n = 20;

ch[2] = '2';

}

else if(x < 200 && (y > 200 && y < 400) && ch[3] == 'B')

{

m = 0;

n = 200;

ch[3] = '2';

}

else if((x > 200 && x < 400) && (y > 200 && y < 400) && ch[4] == 'B')

{

m = 200;

n = 200;

ch[4] = '2';

}

else if((x > 400 && x < 600) && (y > 200 && y < 400) && ch[5] == 'B')

{

m = 400;

n = 200;

ch[5] = '2';

}

else if(x < 200 && (y > 400 && y < 600) && ch[6] == 'B')

{

m = 0;

n = 400;

ch[6] = '2';

}

else if(( x > 200 && x < 400) && (y > 400 && y < 600) && ch[7] == 'B')

{

m = 200;

n = 400;

ch[7] = '2';

}

else if((x > 400 && x < 600) && (y > 400 && y < 600) && ch[8] == 'B')

{

m = 400;

n = 400;

ch[8] = '2';

}

else

{

m = -1;

n = -1;

flag = 1;

}

if(m != -1 && n != -1)

{

g.setColor(Color.green);

g.drawOval(m+10, n+10, 169, 169);

//g.drawLine(m, n, m+189, n+189);

//g.drawLine(m+199, n, m, n+199);

flag = flag + 2;

}

}

for(i=0; i<9; i++) // for draw

{

if(ch[i] != 'B')

{

if(i == 8)

draw();

}

else

break;

}

for(i=0; i<3; i++) //for vertical

{

if(ch[i] != 'B')

{

if((ch[i+3] == ch[i]) && (ch[i+6] == ch[i]))

win(ch[i]);

}

}

for(i=0; i<7; i++) //for horizontal

{

if(ch[i] != 'B')

{

if((ch[i] == ch[i+1]) && (ch[i] == ch[i+2]))

win(ch[i]);

}

i = i+2;

}

if(ch[4] != 'B') //for diagonals

{

if(((ch[0] == ch[4]) && (ch[4] == ch[8])) || ((ch[2] == ch[4]) && (ch[4] == ch[6])))

win(ch[4]);

}

}

/\*\*

\* Displays a child Frame to show the Victory message

\* @param x denotes the player who won the game

\* @return Returns the frame object

\*/

public Frame win(char x)

{

sub = new Frame("Victory !!!");

Label l = new Label();

Button b1 = new Button("New Game");

Button b2 = new Button("Exit");

if(x == '2')

l.setText("Player-2 Won");

else

l.setText("Player-1 won");

sub.setLayout(null);

sub.add(b1);

sub.add(b2);

sub.add(l);

l.setBounds(100, 40, 100, 50);

sub.setVisible(true);

sub.setSize(300, 300);

b1.setBounds(20, 100, 100, 50);

b2.setBounds(150, 100, 100, 50);

flag = 2;

b1.addActionListener(this);

b2.addActionListener(this);

sub.addWindowListener(new WHandler(sub));

return sub;

}

/\*\*

\* Displays a child Frame to show that the match is a Stalemate

\* @return returns the Frame object

\*/

public Frame draw()

{

sub = new Frame("Result");

Button b1 = new Button("New Game");

Button b2 = new Button("Exit");

Label l1 = new Label("Match Stalemate");

sub.setLayout(null);

sub.add(l1);

sub.add(b1);

sub.add(b2);

l1.setBounds(100, 40, 100, 50);

sub.setVisible(true);

sub.setSize(300, 300);

b1.setBounds(20, 100, 100, 50);

b2.setBounds(150, 100, 100, 50);

flag = 2;

b1.addActionListener(this);

b2.addActionListener(this);

sub.addWindowListener(new WHandler(sub));

return sub;

}

/\*\*

\* Performs two Different operations based on the button pressed in the child Frame

\* @param e Holds the object of ActionEvent

\*/

public void actionPerformed(ActionEvent e)

{

if(e.getActionCommand() == "Exit")

{

sub.dispose();

f.dispose();

}

else if(e.getActionCommand() == "New Game")

{

for(i=0; i<9; i++)

ch[i] = 'B';

g.setColor(Color.black);

g.fillRect(0, 0, 600, 600);

g.setColor(Color.blue);

g.drawLine(200, 0, 200, 600);

g.drawLine(400, 0, 400, 600);

g.drawLine(0, 200, 600, 200);

g.drawLine(0, 400, 600, 400);

sub.dispose();

}

}

/\*\*

\* Maintains the background of the Main Frame window

\* @param e Holds the MouseEvent e

\*/

public void mouseMoved(MouseEvent e)

{

g.setColor(Color.blue);

g.drawLine(200, 0, 200, 600);

g.drawLine(400, 0, 400, 600);

g.drawLine(0, 200, 600, 200);

g.drawLine(0, 400, 600, 400);

}

/\*\*

\* Method to create object to start the game

\* @param args not used

\*/

public static void main(String args [])

{

new TicTacToe();

}

/\*\*

\* method not used

\*/

public void mousePressed(MouseEvent e)

{

// TODO Auto-generated method stub

}

/\*\*

\* method not used

\*/

public void mouseReleased(MouseEvent e)

{

// TODO Auto-generated method stub

}

/\*\*

\* method not used

\*/

public void mouseEntered(MouseEvent e)

{

// TODO Auto-generated method stub

}

/\*\*

\* method not used

\*/

public void mouseExited(MouseEvent e)

{

// TODO Auto-generated method stub

}

/\*\*

\* method not used

\*/

public void mouseDragged(MouseEvent e)

{

// TODO Auto-generated method stub

}

}