**University of Dhaka**



**Department of Computer Science & Engineering**

**CSE-2112, OOP Lab**

**Lab Project: Table Tennis**

**Submitted To:**

1.Dr. Md. Mustafizur Rahman

Professor,CSE,DU

2.Md. Shiplu Hawlader

Lecturer,CSE,DU

**Submitted By:**

1.Md.Inzamam-Ul-Haque

Roll: 85

2.Mahamudur Rahman

Roll: 83

Date-of-Submission: 26th May, 2016

**package** ponggame;

**import** java.awt.BorderLayout;

**import** java.io.File;

**import** java.io.IOException;

**import** javax.sound.sampled.AudioInputStream;

**import** javax.sound.sampled.AudioSystem;

**import** javax.sound.sampled.Clip;

**import** javax.sound.sampled.LineUnavailableException;

**import** javax.sound.sampled.UnsupportedAudioFileException;

**import** javax.swing.JFrame;

**public** **class** PongMain{

**public** **static** **void** main(String[] args) **throws** UnsupportedAudioFileException, IOException, LineUnavailableException {

JFrame frame = **new** JFrame("Pong");

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

frame.setLayout(**new** BorderLayout());

File soundFile = **new** File("C:\\Users\\TOHA\\Downloads\\image\\bass.wav");

AudioInputStream audioIn = AudioSystem.*getAudioInputStream*(soundFile);

Clip clip = AudioSystem.*getClip*();

clip.open(audioIn);

clip.start();

clip.loop(Clip.***LOOP\_CONTINUOUSLY***);

PongPanel pongPanel = **new** PongPanel();

frame.add(pongPanel, BorderLayout.***CENTER***);

frame.setSize(500,500);

frame.setVisible(**true**);

}

}

**package** ponggame;

**import** java.awt.Color;

**import** java.awt.Font;

**import** java.awt.Graphics;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.KeyEvent;

**import** java.awt.event.KeyListener;

**import** java.awt.image.BufferedImage;

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.LinkedList;

**import** java.util.Queue;

**import** javax.imageio.ImageIO;

**import** javax.sound.sampled.AudioInputStream;

**import** javax.sound.sampled.AudioSystem;

**import** javax.sound.sampled.Clip;

**import** javax.sound.sampled.LineUnavailableException;

**import** javax.sound.sampled.UnsupportedAudioFileException;

**import** javax.swing.JPanel;

**import** javax.swing.Timer;

@SuppressWarnings("serial")

**class** PongPanel **extends** JPanel **implements** ActionListener, KeyListener{

Color rng;

**private** Queue <Color> rong = **new** LinkedList<Color>();

**private** **boolean** showTitleScreen = **true**;

**private** **boolean** playing = **false**;

**private** **boolean** gameOver = **false**;

**private** **boolean** upPressed = **false**;

**private** **boolean** downPressed = **false**;

**private** **boolean** wPressed = **false**;

**private** **boolean** sPressed = **false**;

**private** **int** paddleSpeed = 5;

**private** **int** negPaddleSpeed = -5;

**private** BufferedImage image;

BufferedImage image1;

Ball bl;

Player p1 ;

Player p2 ;

//construct a PongPanel

**public** PongPanel(){

setBackground(Color.***BLACK***);

rong.add(Color.***RED***);

rong.add(Color.***BLUE***);

rong.add(Color.***YELLOW***);

rong.add(Color.***CYAN***);

rong.add(Color.***WHITE***);

rong.add(Color.***GREEN***);

rong.add(Color.***ORANGE***);

rong.add(Color.***MAGENTA***);

**try** {

image1 = ImageIO.*read*(**new** File("C:\\Users\\TOHA\\Downloads\\image\\back2.jpg"));

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

bl = **new** Ball();

p1 = **new** PlayerOne();

p2 = **new** PlayerTwo();

//listen to key presses

setFocusable(**true**);

addKeyListener(**this**);

//call step() 60 fps

Timer timer = **new** Timer(10, **this**);

timer.start();

}

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

**try** {

step();

} **catch** (UnsupportedAudioFileException | IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

} **catch** (LineUnavailableException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

**public** **void** step() **throws** UnsupportedAudioFileException, IOException, LineUnavailableException{

**if**(playing){

//move player 1

**if** (upPressed) {

**if** (p1.getPlayerY()-paddleSpeed > 0) {

p1.updatePlayerY(negPaddleSpeed);

}

}

**if** (downPressed) {

**if** (p1.getPlayerY() + paddleSpeed + p1.getPlayerHeight() < getHeight()) {

p1.updatePlayerY(paddleSpeed);

}

}

//move player 2

**if** (wPressed) {

**if** (p2.getPlayerY()-paddleSpeed > 0) {

p2.updatePlayerY(negPaddleSpeed);

}

}

**if** (sPressed) {

**if** (p2.getPlayerY() + paddleSpeed + p2.getPlayerHeight() < getHeight()) {

p2.updatePlayerY(paddleSpeed);

}

}

//where will the ball be after it moves?

**int** nextBallLeft = bl.getBallX() + bl.getBallDeltaX();

**int** nextBallRight = bl.getBallX()+bl.getDiameter() +bl.getBallDeltaX();

**int** nextBallTop = bl.getBallY() + bl.getBallDeltaY();

**int** nextBallBottom = bl.getBallY()+bl.getDiameter() +bl.getBallDeltaY();

**int** playerOneRight = p1.getPlayerX() +p1.getPlayerWidth();

**int** playerOneTop = p1.getPlayerY();

**int** playerOneBottom = p1.getPlayerY() + p1.getPlayerHeight();

**int** playerTwoLeft = p2.getPlayerX();

**int** playerTwoTop =p2.getPlayerY();

**int** playerTwoBottom = p2.getPlayerY()+p2.getPlayerHeight();

//ball bounces off top and bottom of screen

**if** (nextBallTop < 0 || nextBallBottom > getHeight()) {

**int** y = -1;

bl.updateBallDeltaY(y);

}

//will the ball go off the left side?

**if** (nextBallLeft < playerOneRight) {

//is it going to miss the paddle?

**if** (nextBallTop > playerOneBottom || nextBallBottom < playerOneTop) {

**int** point = 1;

p2.updatePlayerScore(point);

File soundFile = **new** File("C:\\Users\\TOHA\\Downloads\\image\\miss.wav");

AudioInputStream audioIn = AudioSystem.*getAudioInputStream*(soundFile);

Clip clip = AudioSystem.*getClip*();

clip.open(audioIn);

clip.start();

clip.loop(0);

**if** (p2.getPlayerScore() == 3) {

playing = **false**;

gameOver = **true**;

}

bl.setBallX(250);

bl.setBallY(250);

}

**else** {

File soundFile = **new** File("C:\\Users\\TOHA\\Downloads\\image\\tada1.wav");

AudioInputStream audioIn = AudioSystem.*getAudioInputStream*(soundFile);

Clip clip = AudioSystem.*getClip*();

clip.open(audioIn);

clip.start();

clip.loop(1);

rng=rong.element();

rong.add(rng);

rong.remove();

**int** x = -1;

bl.updateBallDeltaX(x);

}

}

//will the ball go off the right side?

**if** (nextBallRight > playerTwoLeft) {

//is it going to miss the paddle?

**if** (nextBallTop > playerTwoBottom || nextBallBottom < playerTwoTop) {

**int** point = 1;

p1.updatePlayerScore(point);

File soundFile = **new** File("C:\\Users\\TOHA\\Downloads\\image\\miss.wav");

AudioInputStream audioIn = AudioSystem.*getAudioInputStream*(soundFile);

Clip clip = AudioSystem.*getClip*();

clip.open(audioIn);

clip.start();

clip.loop(0);

**if** (p1.getPlayerScore() == 3) {

playing = **false**;

gameOver = **true**;

}

bl.setBallX(250);

bl.setBallY(250);

}

**else** {

File soundFile = **new** File("C:\\Users\\TOHA\\Downloads\\image\\tada1.wav");

AudioInputStream audioIn = AudioSystem.*getAudioInputStream*(soundFile);

Clip clip = AudioSystem.*getClip*();

clip.open(audioIn);

clip.start();

clip.loop(1);

rng=rong.element();

rong.add(rng);

rong.remove();

**int** x = -1;

bl.updateBallDeltaX(x);

}

}

//move the ball

bl.updateBallX(bl.getBallDeltaX());

bl.updateBallY(bl.getBallDeltaY());

}

//stuff has moved, tell this JPanel to repaint itself

repaint();

}

//paint the game screen

**public** **void** paintComponent(Graphics g){

**super**.paintComponent(g);

g.setColor(Color.***WHITE***);

g.drawImage(image1,0,0,500,500,**null**);

**if** (showTitleScreen) {

//g.setFont(new Font(Font.DIALOG, Font.BOLD, 36));

g.setColor(Color.***CYAN***);

g.setFont(**new** Font(Font.***SERIF***, Font.***BOLD***, 36));

g.drawString("Table Tennis", 145,70);

g.setFont(**new** Font(Font.***SERIF***, Font.***BOLD***, 18));

g.drawString("Press 'P' to Play.", 135, 430);

g.setFont(**new** Font(Font.***SERIF***, Font.***ITALIC***, 15));

g.drawString("@Toha & Sajib", 365, 450);

**try** {

image = ImageIO.*read*(**new** File("C:\\Users\\TOHA\\Downloads\\image\\ping.jpg"));

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

g.drawImage(image,110,80,250,320,**null**);

}

**else** **if** (playing) {

**int** playerOneRight = p1.getPlayerX() + p1.getPlayerWidth();

**int** playerTwoLeft = p2.getPlayerX();

//draw dashed line down center

**for** (**int** lineY = 0; lineY < getHeight(); lineY += 50) {

g.drawLine(250, lineY, 250, lineY+25);

}

//draw "goal lines" on each side

g.drawLine(playerOneRight, 0, playerOneRight, getHeight());

g.drawLine(playerTwoLeft, 0, playerTwoLeft, getHeight());

//draw the scores

g.setColor(Color.***PINK***);

g.setFont(**new** Font(Font.***SERIF***, Font.***BOLD***, 36));

g.drawString(String.*valueOf*(p1.getPlayerScore()), 100, 100);

g.drawString(String.*valueOf*(p2.getPlayerScore()), 400, 100);

g.setColor(rng);

//draw the ball

g.fillOval(bl.getBallX(), bl.getBallY(), bl.getDiameter(), bl.getDiameter());

g.setColor(Color.***WHITE***);

//draw the paddles

g.fillRect(p1.getPlayerX(),p1.getPlayerY(),p1.getPlayerWidth(),p1.getPlayerHeight());

g.fillRect(p2.getPlayerX(),p2.getPlayerY(),p2.getPlayerWidth(),p2.getPlayerHeight());

}

**else** **if** (gameOver) {

/\*try {

image1 = ImageIO.read(new File("C:\\Users\\TOHA\\Downloads\\image\\back2.jpg"));

} catch (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

g.drawImage(image1,0,0,500,500,null);\*/

**try** {

image = ImageIO.*read*(**new** File("C:\\Users\\TOHA\\Downloads\\image\\gameover.png"));

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

g.drawImage(image,110,80,250,320,**null**);

g.setColor(Color.***CYAN***);

g.setFont(**new** Font(Font.***SERIF***, Font.***BOLD***, 30));

g.drawString(String.*valueOf*(p1.getPlayerScore()), 190, 70);

g.drawString(String.*valueOf*(p2.getPlayerScore()), 260, 70);

g.setFont(**new** Font(Font.***SERIF***, Font.***BOLD***, 36));

**if** (p1.getPlayerScore() > p2.getPlayerScore()) {

g.drawString("Player 1 Wins!", 135, 100);

}

**else** {

g.drawString("Player 2 Wins!", 135, 120);

}

g.setFont(**new** Font(Font.***SERIF***, Font.***BOLD***, 18));

g.drawString("Press Space to Restart.", 150, 400);

}

}

**public** **void** keyTyped(KeyEvent e) {}

**public** **void** keyPressed(KeyEvent e) {

**if** (showTitleScreen) {

**if** (e.getKeyCode() == KeyEvent.***VK\_P***) {

showTitleScreen = **false**;

playing = **true**;

}

}

**else** **if**(playing){

**if** (e.getKeyCode() == KeyEvent.***VK\_UP***) {

upPressed = **true**;

}

**else** **if** (e.getKeyCode() == KeyEvent.***VK\_DOWN***) {

downPressed = **true**;

}

**else** **if** (e.getKeyCode() == KeyEvent.***VK\_W***) {

wPressed = **true**;

}

**else** **if** (e.getKeyCode() == KeyEvent.***VK\_S***) {

sPressed = **true**;

}

}

**else** **if** (gameOver) {

**if** (e.getKeyCode() == KeyEvent.***VK\_SPACE***) {

gameOver = **false**;

showTitleScreen = **true**;

bl = **new** Ball();

p1 = **new** PlayerOne();

p2 = **new** PlayerTwo();

}

}

}

**public** **void** keyReleased(KeyEvent e) {

**if** (playing) {

**if** (e.getKeyCode() == KeyEvent.***VK\_UP***) {

upPressed = **false**;

}

**else** **if** (e.getKeyCode() == KeyEvent.***VK\_DOWN***) {

downPressed = **false**;

}

**else** **if** (e.getKeyCode() == KeyEvent.***VK\_W***) {

wPressed = **false**;

}

**else** **if** (e.getKeyCode() == KeyEvent.***VK\_S***) {

sPressed = **false**;

}

}

}

}

**package** ponggame;

**class** Ball {

**private** **int** ballX ;

**private** **int** ballY;

**private** **int** diameter;

**private** **int** ballDeltaX;

**private** **int** ballDeltaY;

**public** Ball() {

// **TODO** Auto-generated constructor stub

ballX = 250;

ballY = 250;

diameter = 20;

ballDeltaX = -1;

ballDeltaY = 3;

}

**public** **int** getBallX()

{

**return** ballX;

}

**public** **int** getBallY()

{

**return** ballY;

}

**public** **int** getDiameter()

{

**return** diameter;

}

**public** **int** getBallDeltaX(){

**return** ballDeltaX;

}

**public** **int** getBallDeltaY(){

**return** ballDeltaY;

}

**public** **void** updateBallDeltaX(**int** x){

ballDeltaX \*= x;

}

**public** **void** updateBallDeltaY(**int** y){

ballDeltaY \*= y;

}

**public** **void** updateBallX(**int** x){

ballX+=x;

}

**public** **void** updateBallY(**int** y){

ballY+=y;

}

**public** **void** setBallX(**int** x){

ballX = x ;

}

**public** **void** setBallY(**int** y){

ballY = y;

}

}

**package** ponggame;

**abstract** **class** Player {

**public** **abstract** **int** getPlayerX();

**public** **abstract** **int** getPlayerY();

**public** **abstract** **int** getPlayerWidth();

**public** **abstract** **int** getPlayerHeight();

**public** **abstract** **int** getPlayerScore();

**public** **abstract** **void** updatePlayerX(**int** paddleSpeed);

**public** **abstract** **void** updatePlayerY(**int** paddleSpeed);

**public** **abstract** **void** updatePlayerScore(**int** point);

}

**class** PlayerOne **extends** Player{

**private** **int** playerOneX ;

**private** **int** playerOneY ;

**private** **int** playerOneWidth ;

**private** **int** playerOneHeight ;

**private** **int** playerOneScore;

PlayerOne() {

// **TODO** Auto-generated constructor stub

playerOneX = 25;

playerOneY = 250;

playerOneWidth = 10;

playerOneHeight = 50;

playerOneScore=0;

}

**public** **int** getPlayerX(){

**return** playerOneX;

}

**public** **int** getPlayerY(){

**return** playerOneY;

}

**public** **int** getPlayerWidth(){

**return** playerOneWidth;

}

**public** **int** getPlayerHeight(){

**return** playerOneHeight;

}

**public** **int** getPlayerScore(){

**return** playerOneScore;

}

**public** **void** updatePlayerX(**int** paddleSpeed){

playerOneX+=paddleSpeed;

}

**public** **void** updatePlayerY(**int** paddleSpeed){

playerOneY+=paddleSpeed;

}

**public** **void** updatePlayerScore(**int** point){

playerOneScore+=point;

}

}

**class** PlayerTwo **extends** Player{

**private** **int** playerTwoX ;

**private** **int** playerTwoY ;

**private** **int** playerTwoWidth ;

**private** **int** playerTwoHeight ;

**private** **int** playerTwoScore ;

PlayerTwo() {

// **TODO** Auto-generated constructor stub

playerTwoX = 465;

playerTwoY = 250;

playerTwoWidth = 10;

playerTwoHeight = 50;

playerTwoScore=0;

}

**public** **int** getPlayerX(){

**return** playerTwoX;

}

**public** **int** getPlayerY(){

**return** playerTwoY;

}

**public** **int** getPlayerWidth(){

**return** playerTwoWidth;

}

**public** **int** getPlayerHeight(){

**return** playerTwoHeight;

}

**public** **int** getPlayerScore(){

**return** playerTwoScore;

}

**public** **void** updatePlayerX(**int** paddleSpeed){

playerTwoX+=paddleSpeed;

}

**public** **void** updatePlayerY(**int** paddleSpeed){

playerTwoY+=paddleSpeed;

}

**public** **void** updatePlayerScore(**int** point){

playerTwoScore+=point;

}

}