OOP Final Project: Pong game in Java

Name: Sulthan from L2CC

Student ID: 2502020075

What is pong?

Pong is a game where a player or 2 players control a rectangle-like shape to prevent the ball from going into your goal. It is like tennis.

Java Classes implemented:

1. PongGame.java

```java
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
//executable
public class PongGame {

    Run | Debug
    public static void main(String[] args){

        GameFrame frame = new GameFrame();

    }

}
```

This will be the main initializer, it will create a new window that relates to GameFrame.java.

2. GameFrame.java

```java
// This is for the window
public class GameFrame extends JFrame{

    GamePanel panel;

    GameFrame(){
        panel = new GamePanel();
        this.add(panel);
        this.setTitle(title: "Pong Game");
        this.setResizable(resizable: false);
        this.setBackground(Color.black);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.pack();
        this.setVisible(b: true);
        this.setLocationRelativeTo(c: null);
    }
}
```

This will be the main window for the game. Pretty self-explanatory with the window not resizing, the title, and window visibility.

3. GamePanel.java

This is where the main things happen.

```java
//objects
public void newBall() {
    random = new Random();
    ball = new Ball((GAME_WIDTH/2)-(BALL_DIAMETER/2),random.nextInt(GAME_HEIGHT-BALL_DIAMETER),BALL_DIAMETER,BALL_DIAMETER);
}
public void newPaddles() {
    paddle1 = new Paddle(x: 0,(GAME_HEIGHT/2)-(PADDLE_HEIGHT/2),PADDLE_WIDTH,PADDLE_HEIGHT,id: 1);
    paddle2 = new Paddle(GAME_WIDTH-PADDLE_WIDTH,(GAME_HEIGHT/2)-(PADDLE_HEIGHT/2),PADDLE_WIDTH,PADDLE_HEIGHT,id: 2);
}
```

The image above is for the ball and both player paddles.

```
public void checkCollision() {

    //bounces the ball
    if(ball.y <=0) {
        ball.setYDirection(-ball.yVelocity);
    }
    if(ball.y >= GAME_HEIGHT-BALL_DIAMETER) {
        ball.setYDirection(-ball.yVelocity);
    }
    if(ball.intersects(paddle1)) {
        ball.xVelocity = Math.abs(ball.xVelocity);
        ball.xVelocity++;
        if(ball.yVelocity>0)
            ball.yVelocity++;
        else
            ball.yVelocity--;
        ball.setXDirection(ball.xVelocity);
        ball.setYDirection(ball.yVelocity);
    }
    if(ball.intersects(paddle2)) {
        ball.xVelocity = Math.abs(ball.xVelocity);
        ball.xVelocity++;
        if(ball.yVelocity>0)
            ball.yVelocity++;
        else
            ball.yVelocity--;
        ball.setXDirection(-ball.xVelocity);
        ball.setYDirection(ball.yVelocity);
    }
```

Collision check and bounces the ball.

```
    //Give points and creates a new session
    if(ball.x <=0) {
        score.player2++;
        newPaddles();
        newBall();
        System.out.println("Player 2: "+score.player2);
    }
    if(ball.x >= GAME_WIDTH-BALL_DIAMETER) {
        score.player1++;
        newPaddles();
        newBall();
        System.out.println("Player 1: "+score.player1);
    }
```

Will give points, and create a new round when a player scores.

4. Paddle.java

This class is more focused on the controls for the game.

```
// Controls
public void keyPressed(KeyEvent e) {
    switch(id) {
    case 1:
        if(e.getKeyCode()==KeyEvent.VK_W) {
            setYDirection(-speed);
        }
        if(e.getKeyCode()==KeyEvent.VK_S) {
            setYDirection(speed);
        }
        break;
    case 2:
        if(e.getKeyCode()==KeyEvent.VK_UP) {
            setYDirection(-speed);
        }
        if(e.getKeyCode()==KeyEvent.VK_DOWN) {
            setYDirection(speed);
        }
        break;
    }
}
public void keyReleased(KeyEvent e) {
    switch(id) {
    case 1:
        if(e.getKeyCode()==KeyEvent.VK_W) {
            setYDirection(yDirection: 0);
        }
        if(e.getKeyCode()==KeyEvent.VK_S) {
            setYDirection(yDirection: 0);
        }
        break;
    case 2:
        if(e.getKeyCode()==KeyEvent.VK_UP) {
            setYDirection(yDirection: 0);
```

5. Ball.java

This class is for the direction of the ball.

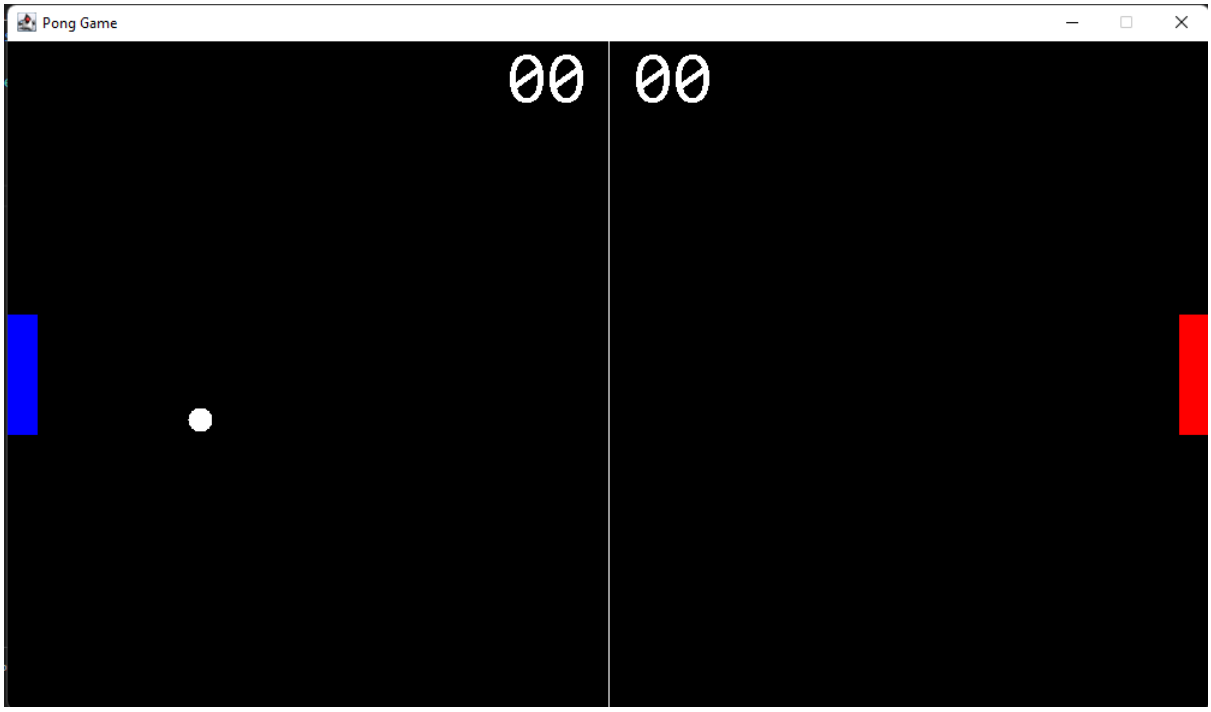
```java
public class Ball extends Rectangle{

    Random random;
    int xVelocity;
    int yVelocity;
    int initialSpeed = 2;

    Ball(int x, int y, int width, int height){
        super(x,y,width,height);
        random = new Random();
        int randomXDirection = random.nextInt(bound: 2);
        if(randomXDirection == 0)
            randomXDirection--;
        setXDirection(randomXDirection*initialSpeed);

        int randomYDirection = random.nextInt(bound: 2);
        if(randomYDirection == 0)
            randomYDirection--;
        setYDirection(randomYDirection*initialSpeed);

    }

    public void setXDirection(int randomXDirection) {
        xVelocity = randomXDirection;
    }
    public void setYDirection(int randomYDirection) {
        yVelocity = randomYDirection;
    }
    public void move() {
        x += xVelocity;
        y += yVelocity;
    }
    public void draw(Graphics g) {
        g.setColor(Color.white);
        g.fillOval(x, y, height, width);
    }
```

6. Score.java

```java
public class Score extends Rectangle{

    static int GAME_WIDTH;
    static int GAME_HEIGHT;
    int player1;
    int player2;

    Score(int GAME_WIDTH, int GAME_HEIGHT){
        Score.GAME_WIDTH = GAME_WIDTH;
        Score.GAME_HEIGHT = GAME_HEIGHT;
    }
    public void draw(Graphics g) {
        g.setColor(Color.white);
        g.setFont(new Font(name: "Consolas",Font.PLAIN,size: 60));

        g.drawLine(GAME_WIDTH/2, y1: 0, GAME_WIDTH/2, GAME_HEIGHT);

        g.drawString(String.valueOf(player1/10)+String.valueOf(player1%10), (GAME_WIDTH/2)-85, y: 50);
        g.drawString(String.valueOf(player2/10)+String.valueOf(player2%10), (GAME_WIDTH/2)+20, y: 50);
    }
}
```

This is for the scoring system.

Running the game



Credits:

Credit to Code Bro for the tutorial.