```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
import java.util.Random;

public class SimpleFlappyBird extends
JPanel implements ActionListener,
KeyListener {

    private int birdY = 300;
    private int velocity = 0;
    private ArrayList<Rectangle> pipes =
new ArrayList<>();
    private int score = 0;
    private boolean gameOver = false;
    private Random random = new
Random();
    private Timer timer;
```

```
    public SimpleFlappyBird() {
        setPreferredSize(new
Dimension(400, 600));
        addKeyListener(this);
        setFocusable(true);
        timer = new Timer(20, this);
        timer.start();
        generatePipe();
    }

    private void generatePipe() {
        int height = random.nextInt(300) +
100;
        pipes.add(new Rectangle(400, 0,
50, height));
        pipes.add(new Rectangle(400,
height + 200, 50, 600 - height - 200));
    }

    @Override
```

```
    public void
actionPerformed(ActionEvent e) {
        if (!gameOver) {
            velocity += 1;
            birdY += velocity;

            for (int i = 0; i < pipes.size(); i++) {
                Rectangle pipe = pipes.get(i);
                pipe.x -= 5;
                if (pipe.x + pipe.width < 0) {
                    pipes.remove(i);
                    i--;
                    if(i % 2 == 1){
                        generatePipe();
                        score++;
                    }
                }
                if (pipe.intersects(100, birdY,
20, 20) || birdY < 0 || birdY > 600) {
                    gameOver = true;
```

```
            }
        }
        repaint();
    }
}

@Override
protected void
paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setColor(Color.CYAN);
    g.fillRect(0, 0, 400, 600);
    g.setColor(Color.YELLOW);
    g.fillOval(100, birdY, 20, 20);
    g.setColor(Color.GREEN);
    for (Rectangle pipe : pipes) {
        g.fillRect(pipe.x, pipe.y,
pipe.width, pipe.height);
    }
    g.setColor(Color.BLACK);
```

```
        g.drawString("Score: " + score, 10,
20);
        if (gameOver) {
            g.drawString("Game Over! Press
Space to Restart", 100, 300);
        }
    }

    @Override
    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() ==
KeyEvent.VK_SPACE) {
            if (gameOver) {
                birdY = 300;
                velocity = 0;
                pipes.clear();
                score = 0;
                gameOver = false;
                generatePipe();
            } else {
```

```
                velocity = -15;
            }
        }
    }

    @Override
    public void keyReleased(KeyEvent e) {}

    @Override
    public void keyTyped(KeyEvent e) {}

    public static void main(String[] args) {
        JFrame frame = new
JFrame("Simple Flappy Bird");
        SimpleFlappyBird game = new
SimpleFlappyBird();
        frame.add(game);
        frame.pack();

frame.setDefaultCloseOperation(JFrame
```

```
.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }
}
```