

```

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.util.ArrayList;
import java.util.List;
import java.util.Random;

Public class DinosaurGame extends JFrame implements KeyListener {

    Private static final int WIDTH = 600;
    Private static final int HEIGHT = 200;
    Private static final int GRAVITY = 1;

    Private int dinoY = HEIGHT - 30; // Ajuste para iniciar en el suelo
    Private int dinoSpeedY = 0;

    Private List<Obstacle> obstacles = new ArrayList<>();
    Private int score = 0;

    Private Timer gameTimer;

    Private JLabel scoreLabel; // Etiqueta para mostrar la puntuación

    Public DinosaurGame() {
        setTitle("Dinosaur Game");
        setSize(WIDTH, HEIGHT);

```

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
setLayout(new BorderLayout());
```

```
scoreLabel = new JLabel("Score: 0", JLabel.CENTER);
```

```
scoreLabel.setFont(new Font("Arial", Font.PLAIN, 18));
```

```
add(scoreLabel, BorderLayout.NORTH);
```

```
addKeyListener(this);
```

```
setFocusable(true);
```

```
gameTimer = new Timer(20, new ActionListener() {
```

```
    @Override
```

```
    Public void actionPerformed(ActionEvent e) {
```

```
        Update();
```

```
        Repaint();
```

```
    }
```

```
});
```

```
gameTimer.start();
```

```
Timer obstacleTimer = new Timer(2000, new ActionListener() {
```

```
    @Override
```

```
    Public void actionPerformed(ActionEvent e) {
```

```
        spawnObstacle();
```

```
    }
```

```
});
```

```
obstacleTimer.start();
```

```
}
```

```

Private void update() {
    dinoSpeedY += GRAVITY;
    dinoY += dinoSpeedY;

    if (dinoY > HEIGHT - 30) {
        dinoY = HEIGHT - 30;
        dinoSpeedY = 0;
    }

    For (Obstacle obstacle : obstacles) {
        Obstacle.move();
        If (obstacle.collidesWithDino()) {
            gameOver();
        }
    }

    removeOffScreenObstacles();
    updateScore();
}

Private void spawnObstacle() {
    Random random = new Random();
    Int obstacleHeight = random.nextInt(40) + 20;
    Obstacles.add(new Obstacle(WIDTH, HEIGHT - obstacleHeight, obstacleHeight));
}

Private void removeOffScreenObstacles() {
    List<Obstacle> toRemove = new ArrayList<>();
    For (Obstacle obstacle : obstacles) {

```

```

        If (obstacle.getX() + obstacle.getWidth() < 0) {
            toRemove.add(obstacle);
        }
    }
    Obstacles.removeAll(toRemove);
}

```

```

Private void updateScore() {
    For (Obstacle obstacle : obstacles) {
        If (obstacle.getX() == 50) { // Dino passes obstacle
            Score++;
            scoreLabel.setText("Score: " + score);
        }
    }
}

```

```

Private void gameOver() {
    gameTimer.stop();
    JOptionPane.showMessageDialog(this, "Game Over! Your score: " + score);
    System.exit(0);
}

```

```

@Override
Public void paint(Graphics g) {
    Super.paint(g);
    drawDinosaur(g);
    drawObstacles(g);
}

```

```
Private void drawDinosaur(Graphics g) {  
    g.setColor(Color.BLACK);  
    g.fillOval(50, dinoY, 30, 30); // Dinosaurio como un óvalo  
}
```

```
Private void drawObstacles(Graphics g) {  
    For (Obstacle obstacle : obstacles) {  
        g.setColor(Color.RED);  
        g.fillRect(obstacle.getX(), obstacle.getY(), obstacle.getWidth(), obstacle.getHeight());  
    }  
}
```

```
@Override  
Public void keyPressed(KeyEvent e) {  
    If (e.getKeyCode() == KeyEvent.VK_SPACE && dinoY == HEIGHT - 30) {  
        dinoSpeedY = -15; // Velocidad de salto  
    }  
}
```

```
@Override  
Public void keyReleased(KeyEvent e) {  
    // Implementar si es necesario  
}
```

```
@Override  
Public void keyTyped(KeyEvent e) {  
    // Implementar si es necesario  
}
```

```
Public static void main(String[] args) {  
    SwingUtilities.invokeLater(new Runnable() {  
        @Override  
        Public void run() {  
            New DinosaurGame().setVisible(true);  
        }  
    });  
}
```

```
Private class Obstacle {
```

```
    Private int x;
```

```
    Private int y;
```

```
    Private int width;
```

```
    Private int height;
```

```
    Public Obstacle(int x, int y, int height) {
```

```
        This.x = x;
```

```
        This.y = y;
```

```
        This.width = 20;
```

```
        This.height = height;
```

```
    }
```

```
    Public int getX() {
```

```
        Return x;
```

```
    }
```

```
    Public int getY() {
```

```
        Return y;
```

```
    }
```

```
Public int getWidth() {  
    Return width;  
}
```

```
Public int getHeight() {  
    Return height;  
}
```

```
Public void move() {  
    X -= 5;  
}
```

```
Public boolean collidesWithDino() {  
    Return x < 80 && x + width > 50 && y + height > dinoY && y < dinoY + 30; // Ajuste de  
colisión  
}  
}  
}
```