import java.awt.\*;

import java.awt.event.\*;

import java.awt.image.\*;

import javax.swing.\*;

import java.io.\*;

import javax.imageio.\*;

class JogoBase extends JFrame {

  final int fundo = 0;

  final int parado = 1;

  final int anda0 = 2;

  final int anda1 = 3;

  final int soco0 = 4;

  final int soco1 = 5;

  final int soco2 = 6;

  final int arbusto = 7;

  Image img[] = new Image[8];

  Desenho des = new Desenho();

  int posX = 15;

  int posY = 0;

  int estado = parado;

  Timer t;

  class Desenho extends JPanel {

    Desenho() {

      try {

        setPreferredSize(new Dimension(1000, 600));

        img[0] = ImageIO.read(new File("fundo.jpeg"));

        img[1] = ImageIO.read(new File("parado.gif"));

        img[2] = ImageIO.read(new File("anda0.gif"));

        img[3] = ImageIO.read(new File("anda1.gif"));

        img[4] = ImageIO.read(new File("soco0.gif"));

        img[5] = ImageIO.read(new File("soco1.gif"));

        img[6] = ImageIO.read(new File("soco2.gif"));

        img[7] = ImageIO.read(new File("arbusto.png"));

      } catch (IOException e) {

        JOptionPane.showMessageDialog(this, "A imagem não pode ser carregada!\n" + e, "Erro",

            JOptionPane.ERROR\_MESSAGE);

        System.exit(1);

      }

    }

    public void paintComponent(Graphics g) {

      super.paintComponent(g);

      g.drawImage(img[fundo], 0, 0, getSize().width, getSize().height, this);

      g.drawImage(img[estado], posX, getSize().height - img[parado].getHeight(this) - 15, img[parado].getWidth(this),

          img[parado].getHeight(this), this);

      g.drawImage(img[arbusto], 300, getSize().height - img[arbusto].getHeight(this) - 10, this);

      Toolkit.getDefaultToolkit().sync();

    }

  }

  void anda() {

    if (estado == anda1) {

      estado = anda0;

      posX += 10;

      des.repaint();

    } else if (estado == anda0) {

      estado = anda1;

      posX += 10;

      des.repaint();

    }

  }

  void soco() {

    if (estado == soco0) {

      estado = soco1;

      des.repaint();

    } else if (estado == soco1) {

      estado = soco2;

      des.repaint();

    } else if (estado == soco2) {

      estado = parado;

      des.repaint();

    }

  }

  JogoBase() {

    super("Trabalho");

    setDefaultCloseOperation(EXIT\_ON\_CLOSE);

    add(des);

    pack();

    setVisible(true);

    addKeyListener(new KeyAdapter() {

      public void keyPressed(KeyEvent e) {

        if (e.getKeyCode() == KeyEvent.VK\_RIGHT) {

          estado = anda0;

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

          estado = parado;

          des.repaint();

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

          estado = soco0;

          des.repaint();

        }

      }

    });

    t = new Timer(1000, new ActionListener() {

      public void actionPerformed(ActionEvent ae) {

        anda();

        soco();

      }

    });

    t.start();

  }

  static public void main(String[] args) {

    JogoBase f = new JogoBase();

  }

}