```java

//Name: David Lim
//Date: 10/23/22
/*
* space invader
*/

import javax.imageio.ImageIO;
import javax.swing.*;

import java.awt.*;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import java.awt.event.*;

public class Main extends JFrame implements KeyListener, ActionListener {

  // declare our componets or fields
  // a field is a global level variable.

  int intXAmount = 10;
  int speedX = 10;
  int speedY = 10;
  int playerX = (int)(Math.random()*450);
  int playerY = (int)(Math.random()*450);

  int bulletspeedx, bulletspeedy;
  int bulletX = playerX + 20;
  int bulletY = playerY + 20;
  static boolean bulletMove = false;

  boolean hit = false;

  Timer invaderTimer = new Timer(100, this);

  invader invaders[] = new invader[5];
  boolean timerState = true;
  private Graphics2D buffer;
        private Image offscreen;
  enum direction{
    UP,
    DOWN,
    LEFT,
    RIGHT
  }

  direction direction;
  public Main() {

    super("Mini space invader");

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new FlowLayout());
    // pnlInput.setLayout(new GridLayout(0,2));
    invaderTimer.start();
          invaderTimer.addActionListener(this);
    addKeyListener(this);
    for (int count = 0; count < invaders.length; count++){
      invaders[count] = new invader();
    }

    for (int count = 0; count < invaders.length; count++){
      int xpos, ypos;
      xpos = (int)(Math.random() * 500) + 500;
      ypos = (int)(Math.random() * 500) - 500;
      invaders[count].setInvader(xpos, ypos, 10, 10, "Red");
    }

  }

  public static void main(String[] args) {

    // Place components on the applet panel
    final int FRAME_WIDTH = 500;
    final int FRAME_HEIGHT = 500;
    Main frame = new Main();
    frame.setSize(FRAME_WIDTH, FRAME_HEIGHT);
    frame.setVisible(true);

  }

  // when you push the button it comes this method
  public void actionPerformed(ActionEvent event) {

    // declare variable to hold which button is called
    for (int count = 0; count<invaders.length;count++){
      if (invaders[count].x < playerX){
        invaders[count].velx =1;
      }
      else if (invaders[count].x > playerX){
        invaders[count].velx = -1;
      }
      if (invaders[count].y < playerY){
        invaders[count].vely =1;
      }
      else if (invaders[count].y > playerY){
        invaders[count].vely = -1;
      }
      invaders[count].x += invaders[count].velx;
      invaders[count].y += invaders[count].vely;

    }

    requestFocus();
          repaint();
  }

  // create the paint method to show graphics
  public void paint(Graphics g) {

          offscreen = createImage(getSize().width, getSize().height);
    buffer = (Graphics2D)offscreen.getGraphics();
    buffer.setColor(Color.red);
    buffer.fillRect(playerX, playerY, 50, 50);

    buffer.setColor(Color.black);
    buffer.fillRect(bulletX, bulletY, 10,10);
    boolean fired = false;
    if (bulletMove == true){
      if (direction == direction.UP){
        bulletspeedx = 0;
        bulletspeedy = -50;
        fired = true;
      }
      else if (direction == direction.DOWN){
        fired = true;
        bulletspeedx = 0;
        bulletspeedy = 50;
      }
      if (direction == direction.LEFT){
        fired = true;
        bulletspeedx = -50;
        bulletspeedy = 0;
      }
      else if (direction == direction.RIGHT){
        fired = true;
        bulletspeedx = 50;
        bulletspeedy = 0;
      }

    }
    bulletX += bulletspeedx;
      bulletY += bulletspeedy;

      if (bulletY <= 0 || bulletX <= 0 || bulletY > 500 || bulletX > 500){
        bulletY = playerY + 20;
        bulletX = playerX + 20;
        bulletMove = false;
        bulletspeedx = 0;
        bulletspeedy = 0;
      }

    for (int count = 0; count<invaders.length; count++){
      buffer.setColor(Color.red);
      buffer.fillOval(invaders[count].x, invaders[count].y, 30, 30);

      if(bulletX < invaders[count].x + 15 && bulletX > invaders[count].x -15 && bulletY < invaders[count].y + 15 && bulletY > invaders[count].y -15){
        invaders[count].lives -=1;
        invaders[count].x = (int)(Math.random() * 500) + 500;
        invaders[count].y = (int)(Math.random() * 500) + 500;
      }

    }
    g.drawImage(offscreen, 0, 0, this);



```

```
  }

  public void Update(Graphics gr) {
    // call the paint method
    paint(gr);
  }

  public void keyPressed(KeyEvent e) {

    int key = e.getKeyCode();
    // nameTextField.setText(""+ key);
    if (key == 37 && playerX >= 25) {
      playerX -= speedX;
      if (!bulletMove){
        bulletX -= speedX;
      }
      direction = direction.LEFT;
    }
    if (key == 39 && playerX + 50 <= 475) {
      playerX += speedX;
      if (!bulletMove){
      bulletX += speedX;
      }
      direction = direction.RIGHT;
    }
    if (key == 38 && playerY >= 25){
        playerY -= speedY;
        if (!bulletMove){
          bulletY -= speedY;
        }
        direction = direction.UP;
    }
    if (key == 40 && playerY + 50 <= 475){
        playerY += speedY;
        if (!bulletMove){
          bulletY += speedY;
        }
        direction = direction.DOWN;
    }
    if (key == 32){
      bulletMove = true;
    }
  }

  public void keyReleased(KeyEvent e) {
    // nameTextField.setText("");
  }

  public void keyTyped(KeyEvent e) {

  }

  public boolean checkHit(int bulletx, int bullety, int invaderx, int invadery){
    if (bulletx >= invaderx && bulletx + 10 <= invaderx + 40 && bullety >= invadery && bullety + 10 <= invadery + 40 && bulletMove){
      hit = true;
    }
    return hit;

  }

}
```