

```
1
2 //Name: David Lim
3 //Date: 10/23/22
4 /*
5  * space invader
6  */
7
8 import javax.imageio.ImageIO;
9 import javax.swing.*;
10
11 import java.awt.*;
12 import java.awt.event.ActionListener;
13 import java.awt.event.ActionEvent;
14 import java.awt.image.BufferedImage;
15 import java.io.File;
16 import java.io.IOException;
17
18 import java.awt.event.*;
19
20 public class Main extends JFrame implements KeyListener, ActionListener {
21
22     // declare our componets or fields
23     // a field is a global level variable.
24
25     int intXAmount = 10;
26     int speedX = 10;
27     int speedY = 10;
28     int playerX = (int)(Math.random()*450);
29     int playerY = (int)(Math.random()*450);
30
31     int bulletSpeedX, bulletSpeedY;
32     int bulletX = playerX + 20;
33     int bulletY = playerY + 20;
34     static boolean bulletMove = false;
35
36     boolean hit = false;
37
38     Timer invaderTimer = new Timer(100, this);
39
40     invader invaders[] = new invader[5];
41     boolean timerState = true;
42     private Graphics2D buffer;
43     private Image offscreen;
44     enum direction{
45         UP,
46         DOWN,
47         LEFT,
48         RIGHT
49     }
50
51     direction direction;
52     public Main() {
53
54         super("Mini space invader");
55
56         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
57         setLayout(new FlowLayout());
```

```
58 // pnlInput.setLayout(new GridLayout(0,2));
59 invaderTimer.start();
60 invaderTimer.addActionListener(this);
61 addKeyListener(this);
62 for (int count = 0; count < invaders.length; count++){
63     invaders[count] = new invader();
64 }
65
66 for (int count = 0; count < invaders.length; count++){
67     int xpos, ypos;
68     xpos = (int)(Math.random() * 500) + 500;
69     ypos = (int)(Math.random() * 500) - 500;
70     invaders[count].setInvader(xpos, ypos, 10, 10, "Red");
71 }
72
73 }
74
75 public static void main(String[] args) {
76
77     // Place components on the applet panel
78     final int FRAME_WIDTH = 500;
79     final int FRAME_HEIGHT = 500;
80     Main frame = new Main();
81     frame.setSize(FRAME_WIDTH, FRAME_HEIGHT);
82     frame.setVisible(true);
83
84 }
85
86 // when you push the button it comes this method
87 public void actionPerformed(ActionEvent event) {
88
89     // declare variable to hold which button is called
90     for (int count = 0; count<invaders.length;count++){
91         if (invaders[count].x < playerX){
92             invaders[count].velx =1;
93         }
94         else if (invaders[count].x > playerX){
95             invaders[count].velx = -1;
96         }
97         if (invaders[count].y < playerY){
98             invaders[count].vely =1;
99         }
100         else if (invaders[count].y > playerY){
101             invaders[count].vely = -1;
102         }
103         invaders[count].x += invaders[count].velx;
104         invaders[count].y += invaders[count].vely;
105
106     }
107
108     requestFocus();
109     repaint();
110 }
111
112 // create the paint method to show graphics
113 public void paint(Graphics g) {
114
```

```
115         offscreen = createImage(getSize().width, getSize().height);
116         buffer = (Graphics2D)offscreen.getGraphics();
117         buffer.setColor(Color.red);
118         buffer.fillRect(playerX, playerY, 50, 50);
119
120         buffer.setColor(Color.black);
121         buffer.fillRect(bulletX, bulletY, 10,10);
122         boolean fired = false;
123         if (bulletMove == true){
124             if (direction == direction.UP){
125                 bulletspeedx = 0;
126                 bulletspeedy = -50;
127                 fired = true;
128             }
129             else if (direction == direction.DOWN){
130                 fired = true;
131                 bulletspeedx = 0;
132                 bulletspeedy = 50;
133             }
134             if (direction == direction.LEFT){
135                 fired = true;
136                 bulletspeedx = -50;
137                 bulletspeedy = 0;
138             }
139             else if (direction == direction.RIGHT){
140                 fired = true;
141                 bulletspeedx = 50;
142                 bulletspeedy = 0;
143             }
144         }
145
146         bulletX += bulletspeedx;
147         bulletY += bulletspeedy;
148
149         if (bulletY <= 0 || bulletX <= 0 || bulletY > 500 || bulletX > 500){
150             bulletY = playerY + 20;
151             bulletX = playerX + 20;
152             bulletMove = false;
153             bulletspeedx = 0;
154             bulletspeedy = 0;
155         }
156
157         for (int count = 0; count<invaders.length; count++){
158             buffer.setColor(Color.red);
159             buffer.fillOval(invaders[count].x, invaders[count].y, 30, 30);
160
161             if(bulletX < invaders[count].x + 15 && bulletX > invaders[count].x -15 && bulletY <
invaders[count].y + 15 && bulletY > invaders[count].y -15){
162                 invaders[count].lives -=1;
163                 invaders[count].x = (int)(Math.random() * 500) + 500;
164                 invaders[count].y = (int)(Math.random() * 500) + 500;
165             }
166         }
167
168         g.drawImage(offscreen, 0, 0, this);
169
170
171
```

```
172     }
173
174     public void Update(Graphics gr) {
175         // call the paint method
176         paint(gr);
177     }
178
179     public void keyPressed(KeyEvent e) {
180
181         int key = e.getKeyCode();
182         // nameTextField.setText("" + key);
183         if (key == 37 && playerX >= 25) {
184             playerX -= speedX;
185             if (!bulletMove){
186                 bulletX -= speedX;
187             }
188             direction = direction.LEFT;
189         }
190         if (key == 39 && playerX + 50 <= 475) {
191             playerX += speedX;
192             if (!bulletMove){
193                 bulletX += speedX;
194             }
195             direction = direction.RIGHT;
196         }
197         if (key == 38 && playerY >= 25){
198             playerY -= speedY;
199             if (!bulletMove){
200                 bulletY -= speedY;
201             }
202             direction = direction.UP;
203         }
204         if (key == 40 && playerY + 50 <= 475){
205             playerY += speedY;
206             if (!bulletMove){
207                 bulletY += speedY;
208             }
209             direction = direction.DOWN;
210         }
211         if (key == 32){
212             bulletMove = true;
213         }
214     }
215
216     public void keyReleased(KeyEvent e) {
217         // nameTextField.setText("");
218     }
219
220     public void keyTyped(KeyEvent e) {
221
222     }
223
224     public boolean checkHit(int bulletx, int bullety, int invaderx, int invadery){
225         if (bulletx >= invaderx && bulletx + 10 <= invaderx + 40 && bullety >= invadery &&
bullety + 10 <= invadery + 40 && bulletMove){
226             hit = true;
227         }
```

```
228 |         return hit;
229 |
230 |     }
231 |
232 | }
```