



## INSTITUT SUPERIEUR POLYTECHNIQUE DE MADAGASCAR

### TP JAVA HS01 – Interactivité swing : Animation et clavier

- 1) Implémentez les codes sources suivantes, dans un projet JAVA, placer le package convenablement :

#### SimpleKeyboardAnimation.java

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5  import java.awt.event.KeyEvent;
6  import java.awt.event.KeyListener;
7
8  public class SimpleKeyboardAnimation extends JPanel implements KeyListener, ActionListener {
9      // Propriétés du rectangle joueur
10     private int playerX = 370; // Commence au milieu de l'écran
11     private int playerWidth = 60;
12     private int playerHeight = 40;
13     private int playerSpeed = 10;
14
15     // Drapeaux de mouvement
16     private boolean leftPressed = false;
17     private boolean rightPressed = false;
18
19     private Timer timer;
20
21     public SimpleKeyboardAnimation() {
22         setBackground(Color.BLACK);
23         setFocusable(true);
24         addKeyListener(this);
25
26         // Timer pour une animation fluide (60 FPS)
27         timer = new Timer(16, this);
28         timer.start();
29     }
30
31     @Override
32     protected void paintComponent(Graphics g) {
33         super.paintComponent(g);
34
35         // Calculer la position du joueur (toujours en bas)
36         int playerY = getHeight() - playerHeight - 10;
37
38         // Dessiner le rectangle joueur en bas
39         g.setColor(Color.CYAN);
40         g.fillRect(playerX, playerY, playerWidth, playerHeight);
41         g.setColor(Color.WHITE);
42         g.drawRect(playerX, playerY, playerWidth, playerHeight);
43
44         // Dessiner les instructions
45         g.setColor(Color.WHITE);
46         g.setFont(new Font("Arial", Font.PLAIN, 16));
47         g.drawString("Flèches Gauche/Droite: Bouger", 10, 30);
48         g.drawString("R: Remettre Position", 10, 50);
49     }
50
51     private void updateGame() {
52         // Mettre à jour la position du joueur (mouvement horizontal seulement)
53         if (leftPressed && playerX > 0) {
54             playerX -= playerSpeed;
55         }
56         if (rightPressed && playerX < getWidth() - playerWidth) {
57             playerX += playerSpeed;
58         }
59
60         // Garder le joueur dans les limites
61         playerX = Math.max(0, Math.min(playerX, getWidth() - playerWidth));
```

```

62     }
63
64     @Override
65     public void actionPerformed(ActionEvent e) {
66         updateGame();
67         repaint();
68     }
69
70     @Override
71     public void keyPressed(KeyEvent e) {
72         int keyCode = e.getKeyCode();
73
74         switch (keyCode) {
75             case KeyEvent.VK_LEFT:
76                 rightPressed = false;
77                 leftPressed = true;
78                 break;
79             case KeyEvent.VK_RIGHT:
80                 leftPressed = false;
81                 rightPressed = true;
82                 break;
83             case KeyEvent.VK_R:
84                 // Remettre la position du joueur au milieu de l'écran
85                 playerX = getWidth() / 2 - playerWidth / 2;
86                 leftPressed = false;
87                 rightPressed = false;
88                 break;
89         }
90     }
91
92     @Override
93     public void keyReleased(KeyEvent e) {
94         int keyCode = e.getKeyCode();
95
96         switch (keyCode) {
97             case KeyEvent.VK_LEFT:
98                 //leftPressed = false;
99                 break;
100             case KeyEvent.VK_RIGHT:
101                 //rightPressed = false;
102                 break;
103         }
104     }
105
106     @Override
107     public void keyTyped(KeyEvent e) {
108         // Non utilisé mais requis par l'interface KeyListener
109     }
110
111     public static void main(String[] args) {
112         JFrame frame = new JFrame("Animation de Mouvement Simple");
113         SimpleKeyboardAnimation animation = new SimpleKeyboardAnimation();
114
115         frame.add(animation);
116         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
117         frame.setSize(800, 600);
118         frame.setLocationRelativeTo(null);
119         frame.setResizable(true);
120         frame.setVisible(true);
121
122         // Request focus for keyboard input
123         animation.requestFocusInWindow();
124     }
125 }

```

- 2) Décommenter les lignes 98 et 101. Quels sont les changements que vous remarquez ?
- 3) Quelles sont les méthodes abstraites contenues dans l'interface KeyListener ? Essayez d'expliquer chacune de ces méthodes.
- 4) Expliquez ce qui se passe entre les lignes 106..109.
- 5) Ajouter un fonction à la touche [ESPACE] (laser, tir de balle, ...).