

## TD 8 : machine à états

### 1 Faire une pause

Considérons le programme graphique suivant :

```
import javax.swing.JApplet;
import java.awt.Graphics;
import java.awt.Point;
import javax.swing.Timer;
import java.awt.event.*;

public class Jeu extends JApplet implements ActionListener, KeyListener, MouseListener
{
    int taille=10;
    int score=0;
    Point position=new Point(50,50);
    Timer minuteur;

    public void init()
    {
        addMouseListener(this);
        addKeyListener(this);
    }

    public void start()
    {
        minuteur=new Timer(100,this);
        minuteur.start();
    }

    // callback pour le minuteur
    public void actionPerformed(ActionEvent evt) {repaint();}

    // KeyListener methods
    public void keyReleased(KeyEvent e){}
    public void keyTyped(KeyEvent e) {}
    public void keyPressed(KeyEvent e)
    {
        System.out.println("event:"+e);
    }
    // MouseListener methods
    public void mouseEntered (MouseEvent e) {}
    public void mouseExited (MouseEvent e) {}
    public void mousePressed (MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseClicked (MouseEvent e)
    {
        if(e.getX()>position.x && e.getX()<position.x+taille &&
```

```

        e.getY()>position.y && e.getY()<position.y+taille )
    {
        score++;
    }
}
public void paint(Graphics g)
{
    g.clearRect(0, 0, getWidth(), getHeight());
    g.drawString("score:"+score,10,10);
    g.drawRect(position.x,position.y,taille,taille);
    position.x+=(int)(Math.random()*5)-2;
    position.y+=(int)(Math.random()*5)-2;
    requestFocus();
}
}

```

L'exécution de ce programme est rythmé par les appels réguliers à la méthode `paint`. Ce jeu très simple affiche un rectangle qui se déplace aléatoirement. Chaque fois que le joueur réussit à cliquer à l'intérieur, il gagne un point.

Nous souhaitons pouvoir faire une pause dans ce programme en appuyant sur la barre d'espace. Lorsque le programme sera en pause, il devra afficher uniquement *en pause*. Pour cela nous ne toucherons pas au minuteur : `paint` continuera à être appelée régulièrement. Comment peut-on faire ?

## 2 Distributeur d'argent : machine à états

Supposons que l'on veuille écrire un logiciel pour gérer un distributeur automatique.

- Identifiez les différentes étapes (états) du programme, par exemple les étapes *bienvenue*, *saisie du code*, etc.
- Identifiez les conditions nécessaires pour passer d'un état à un autre.
- Représentez l'ensemble par un graphe où les nœuds seront les états et les arêtes représenteront le passage d'un état à un autre état.

Cette représentation est une machine à états. Elle est souvent utile pour décrire un logiciel en interaction avec un utilisateur.

## 3 États d'un jeu

Nous souhaitons modifier le programme de la première question pour qu'au démarrage rien ne soit affiché à l'écran mise à part *appuyez sur espace pour commencer la partie*. De même, lorsque le score atteint 10, nous souhaitons afficher uniquement *partie terminée, appuyez sur espace pour recommencer*.

- Faites un schéma représentant les différents états du jeu et les transitions entre les états.
- En utilisant une seule variable pour représenter les états possibles, modifiez le programme pour obtenir le déroulement souhaité.