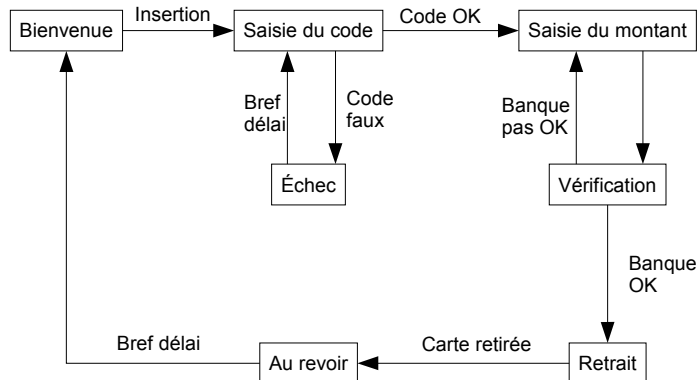


## Corrigé du TD 8 : machine à états

### 1 Faire une pause

Pour faire une pause, il suffit de rajouter un attribut de type boolean à l'applet. Il faut ensuite tenir compte de la valeur de cette variable, à la fois dans la méthode `paint` et dans la méthode `mouseClicked`.

### 2 Distributeur d'argent : machine à états



### 3 États d'un jeu

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

```
public class JeuCorrige extends JApplet implements ActionListener, KeyListener, MouseListener
{
    static final int debut=0;
    static final int enCours=1;
    static final int pause=2;
    static final int fin=3;
    int etat=debut;
    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);
        switch(etat)
        {
            case debut:
                if(e.getKeyCode()==KeyEvent.VK_SPACE){etat=enCours;score=0;}
                break;
            case enCours:
                if(e.getKeyCode()==KeyEvent.VK_SPACE){etat=pause;}
                break;
            case pause:
                if(e.getKeyCode()==KeyEvent.VK_SPACE){etat=enCours;}
                break;
            case fin:
                if(e.getKeyCode()==KeyEvent.VK_SPACE){etat=debut;}
                break;
        }
    }
}

// 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)
{
    switch(etat)
    {
        {
            case debut:break;
            case enCours:
                if(e.getX()>position.x && e.getX()<position.x+taille &&
                    e.getY()>position.y && e.getY()<position.y+taille )
                {
                    score++;
                }
                if(score>=10){etat=fin;}
                break;
            case pause:break;
            case fin:break;
        }
    }
}

```

```

    }
}
public void paint(Graphics g)
{
    g.clearRect(0, 0, getWidth(), getHeight());

    switch(etat)
    {
    case debut:
        g.drawString("appuyer sur espace pour commencer la partie",10,10);
        break;
    case enCours:
        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;
        break;
    case pause:
        g.drawString("en pause",10,10);
        break;
    case fin:
        g.drawString("partie terminée, appuyer sur espace pour recommencer",10,10);
        break;
    }
    requestFocus();
}
}

```