

JAVA

Applications Graphiques

E. ADAM

UVHC - ISTV

Applications graphiques en Java

Java AWT & SWING

Java 2D

Java 3D

Java Swing : un exemple simple

Création d'une fenêtre

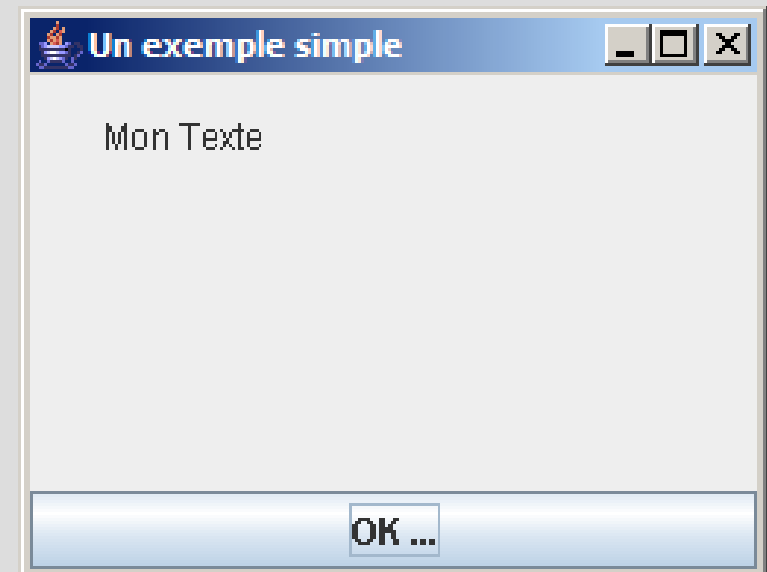
contenant un texte dessiné et un bouton

Taille fenêtre = $\frac{1}{4}$ taille écran

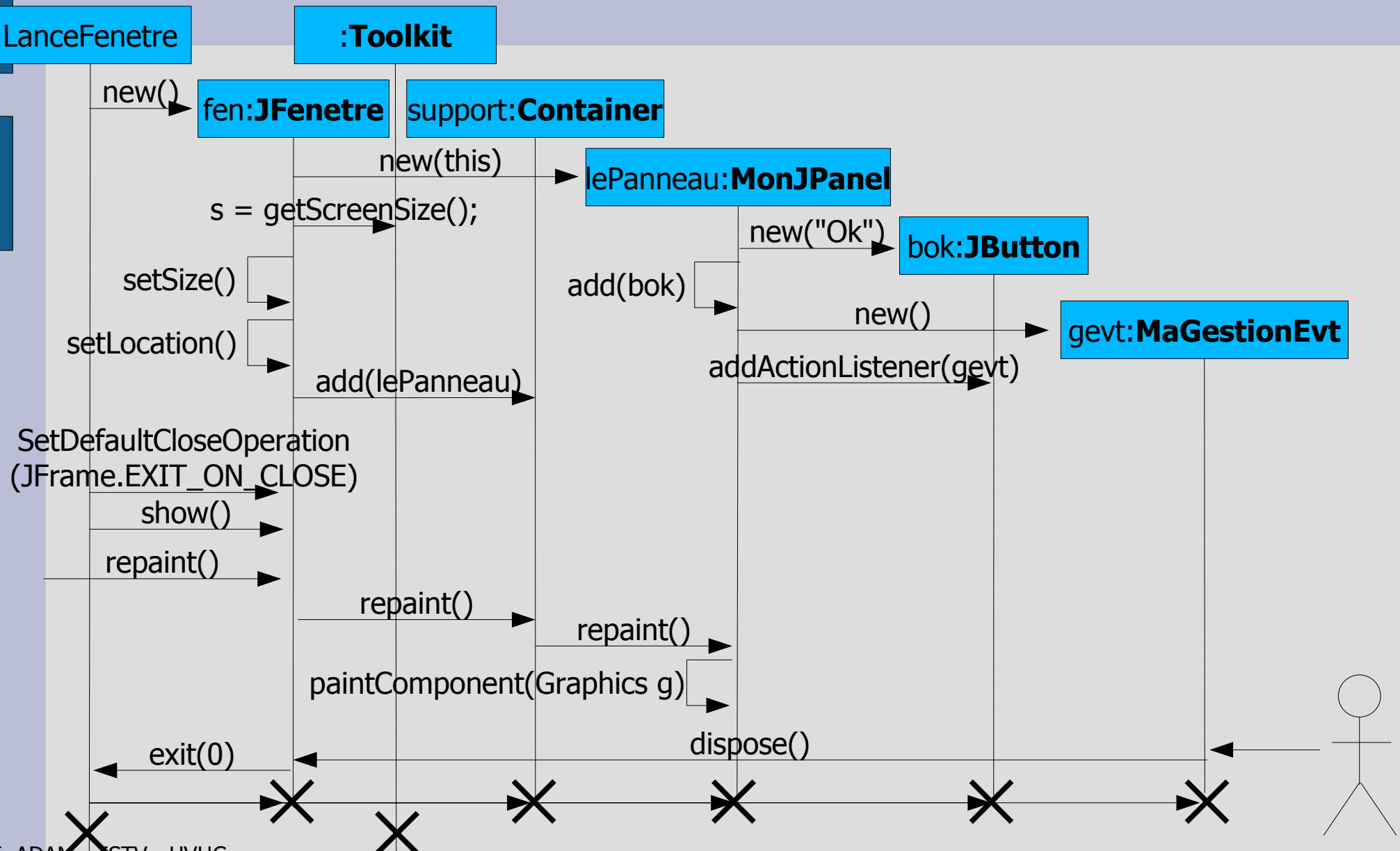
Fenêtre centrée

Utilisation de Java Swing

Nombreuses solutions...



Java Swing : un exemple simple



Java Swing : un exemple simple

Les importations :

```
import java.awt.*; // graphisme de base
import javax.swing.*; // graphisme évolué
import java.awt.event.*; // gestion d'événements de
base
```

• « Lanceur » de fenêtre :

```
public class LanceFenetre
{
    public static void main(String[] args)
    {
        JFenetre fen = new JFenetre();
        fen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        fen.show();
    }
}
```

Java Swing : un exemple simple

La classe JFenetre :

```
class JFenetre extends JFrame
{
    public JFenetre()
    {
        Toolkit tk = Toolkit.getDefaultToolkit();
        Dimension tailleEcran = tk.getScreenSize();
        int largeurEcran = tailleEcran.width;
        int hauteurEcran = tailleEcran.height;

        setTitle("Un exemple simple");
        setSize(largeurEcran/4, hauteurEcran/4);
        setLocation(largeurEcran*3/8, hauteurEcran*3/8);

        MonJPanel lePanneau = new MonJPanel(this);
        Container support = getContentPane();
        support.add(lePanneau);
    }
}
```

Java Swing : un exemple simple

La classe MonJPanel :

```
class LePanneau extends JPanel
{
    private JFenetre fen = null;
    public Contenu(JFenetre _fen)
    {
        fen = _fen;
        setLayout(new BorderLayout());

        JButton bOk = new JButton("OK ...");
        add(bOk, BorderLayout.SOUTH);

        MaGestionEvent myListener = new MaGestionEvent();
        bOk.addActionListener(myListener);
    }
}
```

Java Swing : un exemple simple

La classe **MonJPanel** (suite) :

```
public void paintComponent(Graphics g)
{
    super.paintComponent(g);
    g.drawString("Mon Texte", 25, 25);
}

class MaGestionEvent implements ActionListener
{
    public void actionPerformed(ActionEvent event)
    {
        fen.dispose();
    }
}
}
```


Java Swing : Gestion de clavier

Création d'une fenêtre contenant :

un panneau de label

un panneau de dessin

Une gestion de clavier

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

indications

EssaiTouche

gère le clavier

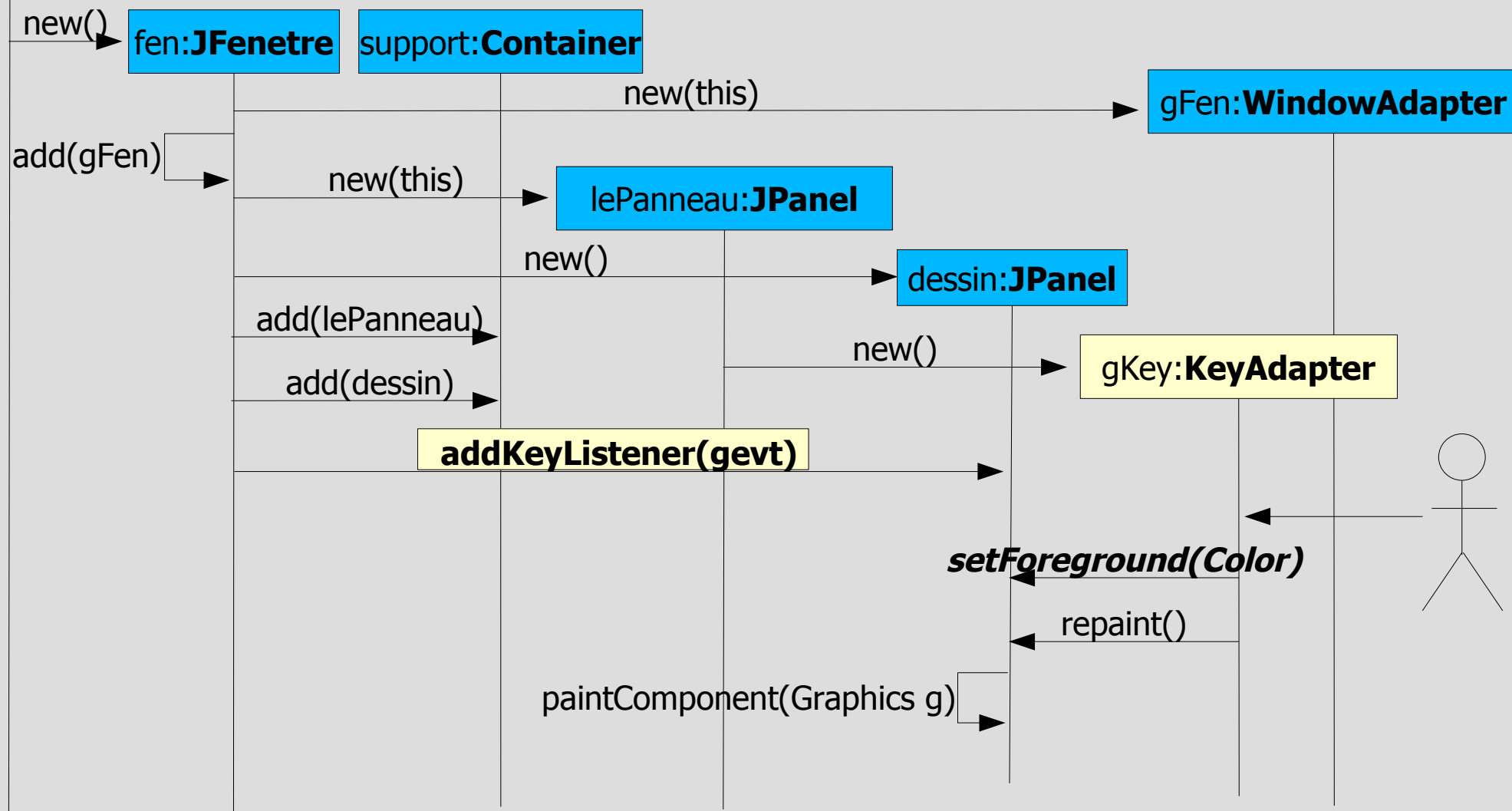
=> Définir somme zone d'attention

Ardoise



Java Swing : Gestion de clavier

LanceFenetre



Java Swing : Gestion de clavier

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

Zone de dessin :

```
class Ardoise extends JPanel
{
    Ardoise()
    {
        setPreferredSize(new Dimension(100, 100));
// dessiner en couleur de fond
        setForeground(getBackground());
    }
// fonction appelée par Java
    public void paintComponent(Graphics g)
    {
        int largeur = getSize().width;
        int hauteur = getSize().height;
// effacer le graphique
        super.paintComponent(g);
//dessiner un ovale
        g.fillOval(10, 10, largeur - 20, hauteur - 20);
    }
}
```

Java Swing : Gestion de clavier

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

Panneau récupérant les accès claviers :

```
public class EssaiTouche extends JPanel
{
    Ardoise ardoise;

    EssaiTouche()
    {
        setLayout(new BorderLayout(5, 5));
        JPanel indications = new JPanel();
        indications.setLayout(new GridLayout(4, 1));
        indications.add(new JLabel(" touche r : disque rouge"));
        indications.add(new JLabel(" touche b : disque bleu"));
        indications.add(new JLabel(" touche v : disque vert"));
        indications.add(new JLabel(" touche e : effacer"));
        ardoise = new Ardoise();
        add(indications, BorderLayout.NORTH);
        add(ardoise, BorderLayout.CENTER);
    }
}
```

...

Java Swing : Gestion de clavier

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

Panneau récupérant les accès claviers (suite) :

```
...
    addKeyListener (new KeyAdapter()
    {
        public void keyTyped(KeyEvent evt)
        {
            if (evt.getKeyChar() == 'r') ardoise.setForeground(Color.red);

            else if (evt.getKeyChar() == 'b') ardoise.setForeground(Color.blue);

            else if (evt.getKeyChar() == 'v') ardoise.setForeground(Color.green);

            else if (evt.getKeyChar() == 'e') ardoise.setForeground(ardoise.getBackground());

            repaint();
        }
    });
...

```

Java Swing : Gestion de clavier

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

Création de la fenêtre :

```
public static void main(String[] argv)
{
    JFrame monCadre = new JFrame("Touches");

    //Redéfinition du contenu de la fenêtre
    monCadre.setContentPane(new EssaiTouche());

    //Attention du système dirigé vers la fenêtre
    monCadre.addWindowListener(new WindowAdapter()
    {
        public void windowClosing(WindowEvent e) { System.exit(0); }
        public void windowActivated(WindowEvent e)
            { monCadre.getContentPane().requestFocus(); }
    });

    monCadre.pack();
    monCadre.setVisible(true);
}
```

Java Swing : Clavier étendu

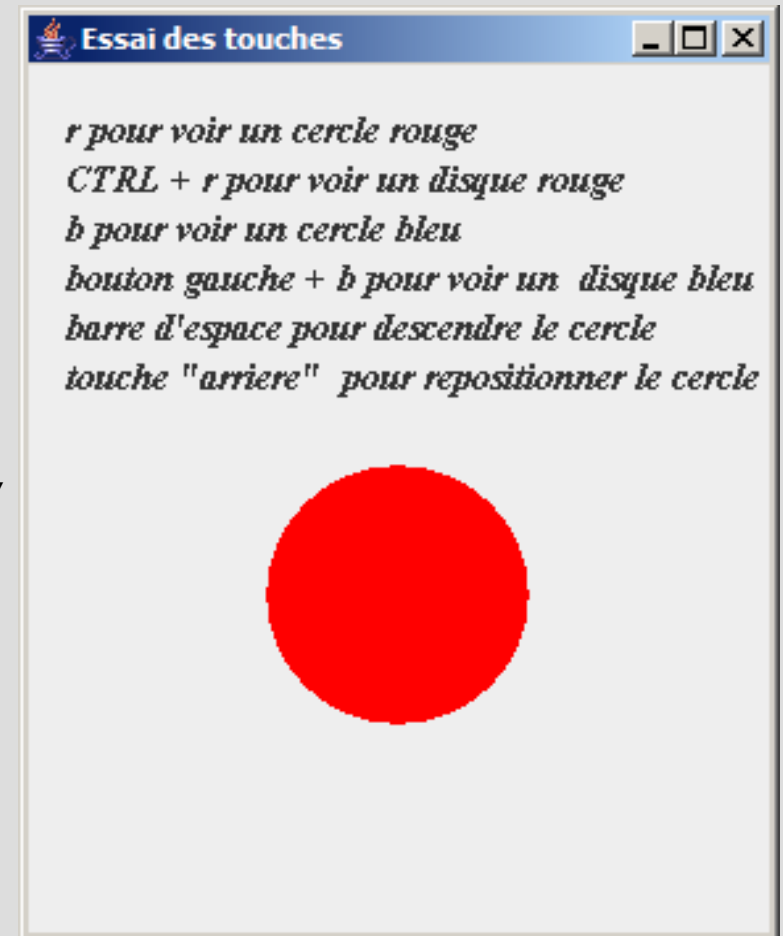
Création d'une fenêtre contenant :

un panneau de dessin de textes

un panneau de dessin

Une gestion de clavier

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>



Java Swing : Clavier étendu

Capture des codes claviers :

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

```
public void keyPressed(KeyEvent evt) {
    int code = evt.getKeyCode();
    if (code == KeyEvent.VK_R) { //touche R
        ardoise.couleur = Color.red;
        if (evt.isControlDown()) ardoise.plein = true;
        else ardoise.plein = false;
    }
    else if (code == KeyEvent.VK_B) { //touche B
        ardoise.couleur = Color.blue;
        //bouton gauche souris en bas
        if ((evt.getModifiersEx() & InputEvent.BUTTON1_DOWN_MASK) != 0)
            ardoise.plein = true;
        else ardoise.plein = false;
    }
    //touche 'espace'
    else if (code == KeyEvent.VK_SPACE) ardoise.hauteur += 20;
    //touche 'retour arrière'
    else if (code == KeyEvent.VK_BACK_SPACE) ardoise.hauteur = 20;

    repaint();
}
```


Java Swing : Clavier étendu

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

Le panneau d'indications :

```
class Indication extends JPanel
{
    int h;
    Indication()
    {
        int hauteur, largeur;
        Font font = new Font("TimesRoman", Font.BOLD|Font.ITALIC, 14);
        FontMetrics ft = getFontMetrics(font);
        h = ft.getHeight();
        largeur = ft.stringWidth ("bouton gauche + b pour voir un disque bleu ");
        setPreferredSize(new Dimension(largeur, 6*h + 20));
        setFont(font);
    }
    ...
}
```

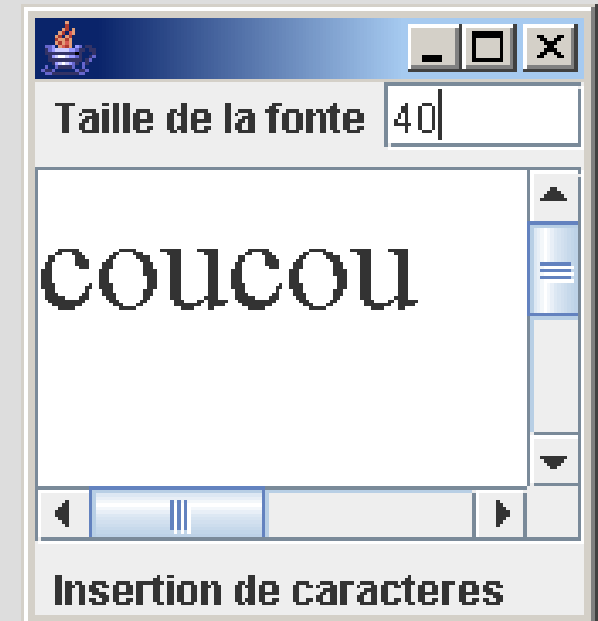
Java Swing : Saisie Texte

Création d'une fenêtre contenant :

un panneau de label – champs texte

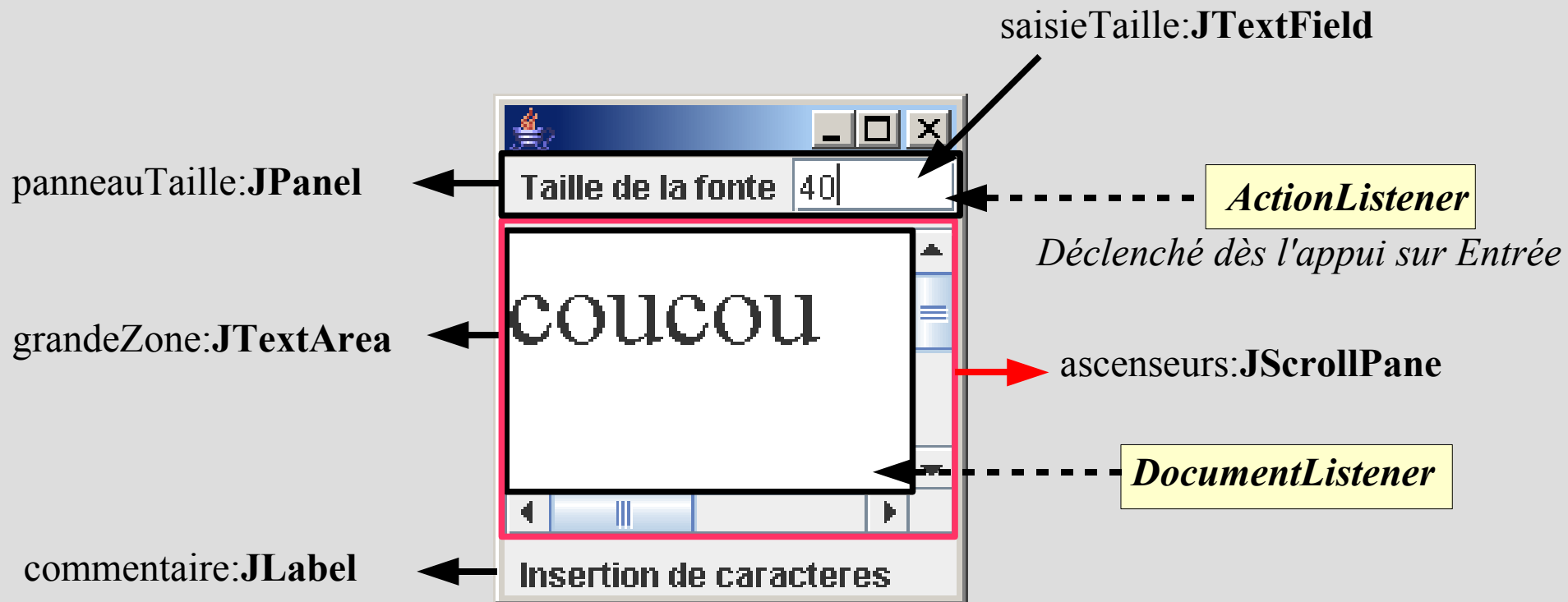
un panneau de zone de texte

Une gestion de clavier



Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

Java Swing : Saisie Texte



Java Swing : Saisie Texte

Le panneau principal :

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

```
...
class EssaiTexte extends JPanel implements ActionListener
{
    JLabel commentaire;
    JTextField saisieTaille;
    JTextArea grandeZone;

    EssaiTexte()
    {
        setLayout(new BorderLayout(5, 5));
        JPanel panneauTaille = new JPanel();
        panneauTaille.setLayout(new BorderLayout(5, 5));
        panneauTaille.add(new JLabel(" Taille de la fonte"), BorderLayout.CENTER);
        saisieTaille = new JTextField("20", 5);
        saisieTaille.addActionListener(this);
        panneauTaille.add(saisieTaille, BorderLayout.EAST);
    }
}
```

...

Java Swing : Saisie Texte

Le panneau principal suite:

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

```
...  
grandeZone = new JTextArea(4, 10);  
grandeZone.setFont(new Font("TimesRoman", Font.PLAIN,20));
```

```
grandeZone.getDocument().addDocumentListener(  
                                new EcouteDocument(commentaire));
```

```
JScrollPane ascenseurs = new JScrollPane(grandeZone);  
add(panneauTaille, BorderLayout.NORTH);  
add(ascenseurs, BorderLayout.CENTER);  
commentaire = new JLabel(" Vous pouvez ecrire ");  
add(commentaire, BorderLayout.SOUTH);  
}
```

```
...
```

Java Swing : Saisie Texte

Le panneau principal suite:

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

```
...
public void actionPerformed(ActionEvent evt)
{
    int taillePolice;

    try
    {
        taillePolice = Integer.parseInt(saisieTaille.getText());
        grandeZone.setFont(new Font("TimesRoman", Font.PLAIN, taillePolice));
    }
    catch(NumberFormatException exc) {
        JOptionPane.showMessageDialog(this, "erreur taille police", "probleme",
            JOptionPane.ERROR_MESSAGE);
    }

    grandeZone.setText(grandeZone.getText());
}
```

Java Swing : Saisie Texte

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

L'écoute d'événement sur la zone de texte :

```
class EcouteDocument implements DocumentListener
{
    JLabel label;
    EcouteDocument(JLabel _label) { label = _label; }

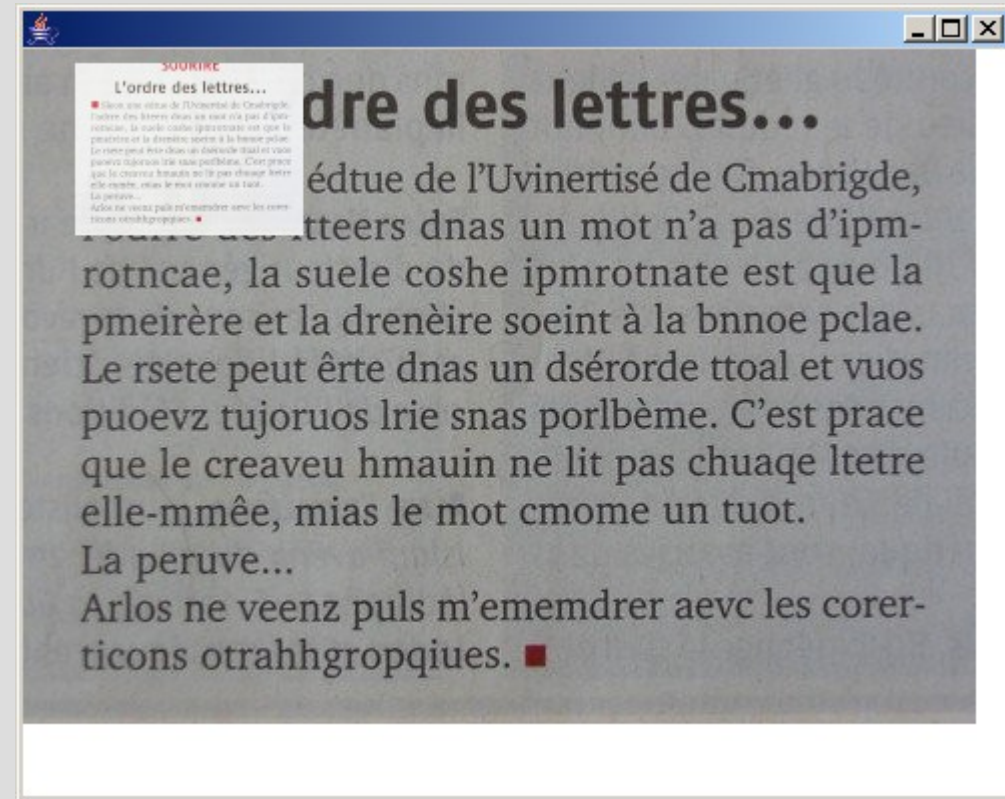
    public void insertUpdate(DocumentEvent e)
        { label.setText(" Insertion de caracteres");}

    public void removeUpdate(DocumentEvent e)
        { label.setText(" Suppression de caracteres"); }

    public void changedUpdate(DocumentEvent e) {}
}
```

Java Swing : Insertions Images

Création d'une fenêtre contenant :
un panneau d'images



Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

Java Swing : Insertions Images

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

Sous la forme d'une JFrame :

```
class ImageTrack extends JFrame
```

```
{
```

```
Image img1, img2;
```

```
MediaTracker tracker;
```

Gestionnaire d'images

```
ImageTrack()
```

```
{
```

```
tracker = new MediaTracker(this);
```

```
Toolkit tk = getToolkit();
```

Gestionnaire graphique

```
img1 = tk.getImage("Lettre.jpg");
```

```
tracker.addImage(img1,0);
```

```
img2 = tk.getImage("Lettre2.gif");
```

```
tracker.addImage(img2,0);
```

```
try { tracker.waitForID(0); }  
catch (InterruptedException e) {}
```

Attente du chargement des images

```
}
```

Java Swing : Insertions Images

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

Sous la forme d'une JFrame :

```
public void paint(Graphics g)
{
    if (tracker.statusAll(false) != MediaTracker.COMPLETE)
    {
        g.drawString("probleme de chargement d'image", 50, 100);
        return;
    }
    g.drawImage(img1,0,0,480,360,this);
    g.drawImage(img2,30,30,this);
}

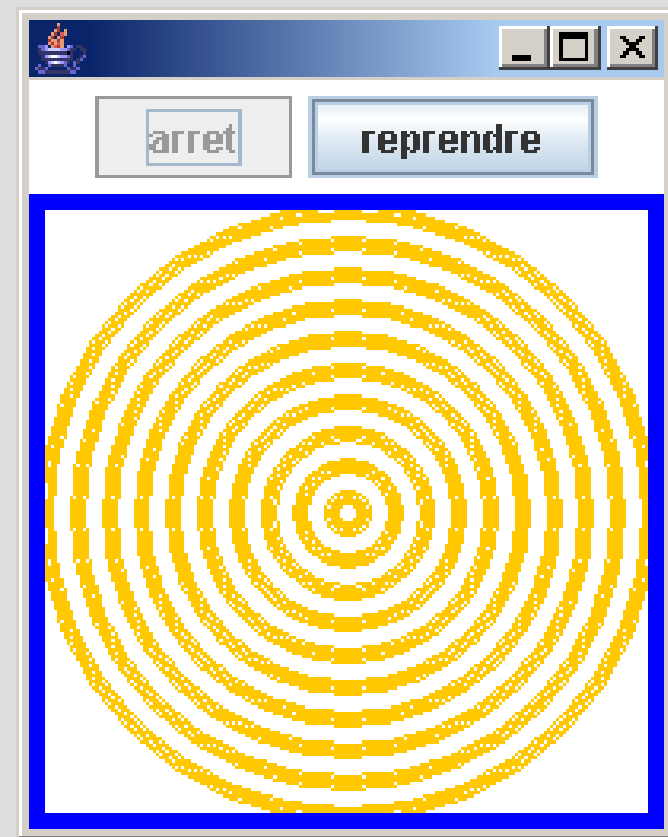
public static void main(String[] argv)
{
    ImageTrack monCadre=new ImageTrack();
    monCadre.setSize(500,400);
    monCadre.setVisible(true);
    monCadre.setDefaultCloseOperation(monCadre.EXIT_ON_CLOSE);
}
```

*Si les images ne sont pas
toutes chargées*

Java Swing : Animation

Création d'une fenêtre contenant :
un panneau animé

Utilisation d'un TIMER !!!



Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

Java Swing : Animation

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

```
class Ardoise extends JPanel implements ActionListener
```

```
{
  int dep = 0;
  Timer timer;
  int largeur = 200, hauteur = 200;
```

Le timer génère des événements

```
Ardoise()
```

```
{
  setPreferredSize(new Dimension(largeur, hauteur));
  setBackground(Color.white);
  timer = new Timer(100, this);
  setForeground(Color.orange);
}
```

Timer de 100 ms

```
public void actionPerformed(ActionEvent e)
```

```
{
  if (e.getSource() == timer)
  {
    repaint();
    dep = (dep - 1) % 10;
  }
  else if (e.getActionCommand().equals("arreter")) timer.stop();
  else if (e.getActionCommand().equals("reprendre")) timer.restart();
}
```

*Si l'événement vient du Timer, définir le déplacement
s'il vient des boutons, l'arrêter ou le redémarrer*

Java Swing : Animation

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

...

```
public void paintComponent(Graphics g)
{
    int i, j;

    super.paintComponent(g);
    for (i = dep - 5; i < largeur/2; i += 10)
        for (j = i; j < i + 5; j++)
            if (j > 0) g.drawOval(j, j, largeur - 2*j, hauteur - 2*j);
}
}
```

*Redessiner les cercles
avec le déplacement actualisé*

Java Swing : Animation

```
class Disque extends JFrame implements ActionListener
```

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

```
{
```

```
    Ardoise ardoise = new Ardoise();
    JButton arret = new JButton("arret");
    JButton reprise = new JButton("reprendre");
```

```
    Disque()
```

```
{
```

```
        JPanel p = new JPanel();
        p.setBackground(Color.white);
        arret.setActionCommand("arreter");
        arret.addActionListener(ardoise);
        arret.addActionListener(this);
        reprise.setActionCommand("reprendre");
        reprise.addActionListener(ardoise);
        reprise.addActionListener(this);
        p.add(arret);
        p.add(reprise);
        reprise.setEnabled(false);
        Container interieur = getContentPane();
        interieur.setLayout(new BorderLayout());
        interieur.add(p, BorderLayout.NORTH);
        interieur.add(ardoise, BorderLayout.CENTER);
        interieur.setBackground(Color.white);
```

Identifiant du bouton

*Evenements sur le bouton
gérés par 2 entités !!!*

Java Swing : Animation

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

```
public void actionPerformed(ActionEvent e)
{
    if (e.getActionCommand().equals("arreter"))
    {
        arret.setEnabled(false);
        reprise.setEnabled(true);
    }
    if (e.getActionCommand().equals("reprendre"))
    {
        arret.setEnabled(true);
        reprise.setEnabled(false);
    }
}
```

**Gestion locale des événements :
changer les propriétés des boutons**

```
public static void main(String[] argv)
{
    Disque monCadre = new Disque();
    monCadre.pack();
    monCadre.setVisible(true);
    monCadre.ardoise.timer.start();
}
}
```

Java Swing : Défilement

Création d'une fenêtre contenant :

Une image défilante

Constituée

D'une image (jpg, gif)

Et d'un texte

Création d'une image !!

Utilisation d'un TIMER



Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

Java Swing : Défilement

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

```
class Defile extends JPanel implements ActionListener
```

```
{
  int x, largeurImage, pas = 10, largeur = 300, hauteur = 150;
```

```
  Timer timer;
```

```
  Image image;
```

```
  Defile()
  {
```

```
    setPreferredSize(new Dimension(largeur, hauteur));
```

```
    setBackground(Color.yellow);
```

```
    timer = new Timer(100, this);
```

```
    timer.start();
```

```
  }
```

```
  void construireImage()
  {
```

```
    Image img = getToolkit().getImage("Lettre2.gif");
```

```
    MediaTracker tracker = new MediaTracker(this);
```

```
    tracker.addImage(img, 0);
```

```
    try { tracker.waitForID(0); }
    catch (InterruptedException e) {
```

```
        System.out.println("probleme au moment du chargement de l'image");
```

```
        System.exit(0);
```

```
    }
```

Le timer génère des événements

Timer de 100 ms

Chargement de l'image

Java Swing : Défilement

```
int largeurimg = img.getWidth(this);  
int hauteurimg = img.getHeight(this);
```

```
String chaine= "JE SUIS CONTENT D'ETUDIER A LA FAC";
```

```
Font fonte = new Font("TimesRoman",Font.PLAIN, 25);
```

```
FontMetrics mesure = getFontMetrics(fonte);
```

```
int hauteurChaine = mesure.getHeight();
```

```
int largeurChaine = mesure.stringWidth(chaine);
```

*Récupération en pixel
de la taille de la chaîne*

```
largeurlImage = largeurChaine + largeurimg + 10;
```

```
int hauteurImage = (hauteurimg > hauteurChaine ? hauteurimg : hauteurChaine);
```

```
image = createImage(largeurlImage, hauteurImage);
```

Création d'une zone graphique

```
Graphics g = image.getGraphics();
```

```
g.setColor(Color.yellow);
```

```
g.fillRect(0, 0, largeurlImage, hauteurImage);
```

```
g.drawImage(img, 0, (hauteurImage - hauteurimg)/2, this);
```

```
g.setFont(fonte);
```

```
g.setColor(Color.red);
```

```
int yChaine = (hauteurImage + mesure.getAscent())/2;
```

```
g.drawString(chaine, largeurimg + 10, yChaine);
```

```
}
```

Java Swing : Défilement

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

```
public void actionPerformed(ActionEvent e)
{
    if (e.getSource() == timer) {
        if (image == null) construireImage();
        x = x - pas;
        if (x < -largeurImage) x = getWidth();
        repaint();
    }
}
```

A chaque top on décale de 'pas' pixel à gauche

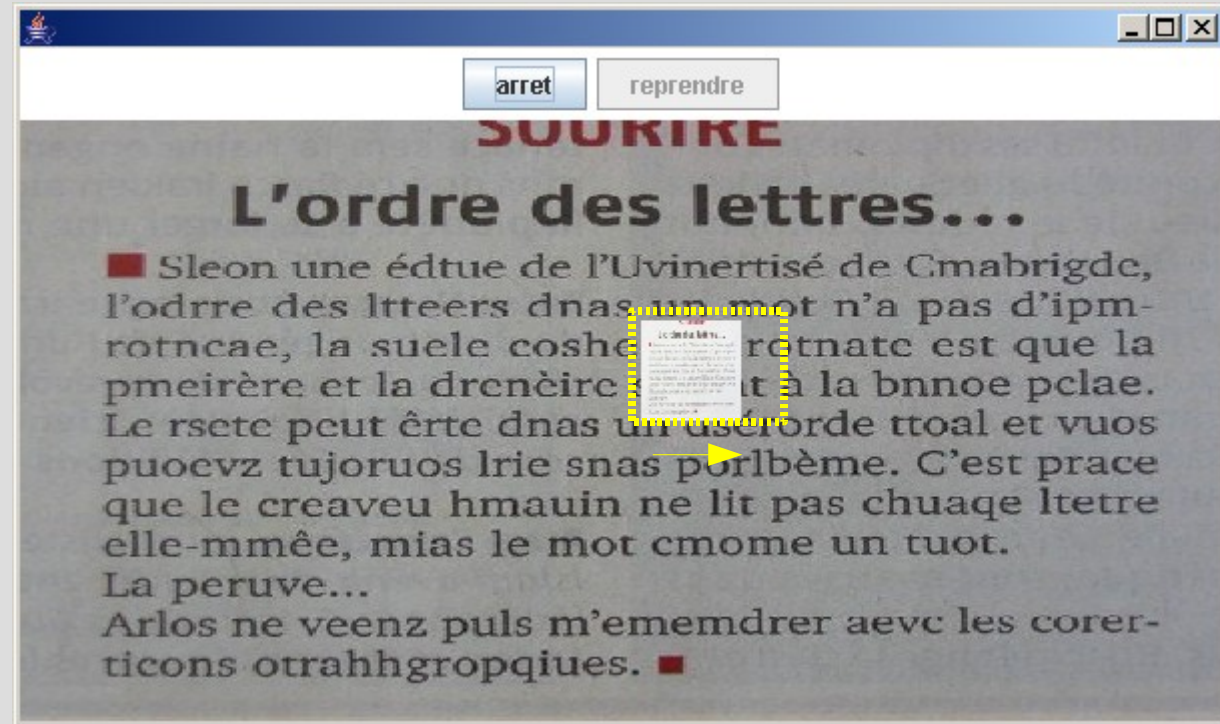
```
public void paintComponent(Graphics g)
{
    super.paintComponent(g);
    if (image != null) g.drawImage(image, x, 40, this);
}
```

```
public static void main(String[] argv)
{
    JFrame monCadre = new JFrame();
    monCadre.setContentPane(new Defile());
    monCadre.setLocation(100, 100);
    monCadre.pack();    monCadre.setVisible(true);
}
```

Java Swing : Déplacement

Création d'une fenêtre contenant :

**Une image
passant sur
une autre image**



Utilisation d'un TIMER

Définition d'une zone à rafraichir !!!

Inspiré de <http://www.infres.enst.fr/~charon/coursJava/>

Java Swing : Défilement

```
class Ardoise extends JPanel implements ActionListener
{
    javax.swing.Timer timer;
    Image imgMobile, imgFixe;
    int largeur = 600, hauteur = 300, largeurImage = 50, hauteurImage = 50;
    int x, y = 100;
    Rectangle vieuxRectangle;
    MediaTracker tracker;
    Ardoise()
    {
        setPreferredSize(new Dimension(largeur, hauteur));
        setBackground(Color.blue);
        x = -largeurImage - 5;
        tracker = new MediaTracker(this);
        imgMobile = getToolkit().getImage("Lettre2.gif"); tracker.addImage(imgMobile,0);
        imgFixe = getToolkit().getImage("Lettre.jpg"); tracker.addImage(imgFixe,0);
        try { tracker.waitForID(0); }
        catch (InterruptedException e) {
            System.out.println("probleme au moment du chargement de l'image");
            System.exit(0); }
        vieuxRectangle = new Rectangle(0, 0, 0, 0);
        timer = new javax.swing.Timer(100, this);
    }
}
```

*Chargement
de 2 images*

Timer sur 100 ms

Java Swing : Défilement

```
public void actionPerformed(ActionEvent e)
```

```
{
```

```
    if (e.getSource() == timer)
```

```
    {
```

```
        largeur = getWidth();
```

```
        x += 10;
```

```
        if (x > largeur) x = -largeurImage;
```

```
        y = getHeight()/3;
```

```
        Rectangle nouveauRectangle = new Rectangle(x, y, largeurImage, hauteurImage);
```

```
        Rectangle rect = nouveauRectangle.union(vieuxRectangle);
```

```
        vieuxRectangle = nouveauRectangle;
```

```
        repaint(rect.x, rect.y, rect.width, rect.height);
```

```
    }
```

```
    else if (e.getActionCommand().equals("arreter")) timer.stop();
```

```
    else if (e.getActionCommand().equals("reprendre")) timer.restart();
```

```
}
```

```
public void paintComponent(Graphics g)
```

```
{
```

```
    super.paintComponent(g);
```

```
    largeur = getWidth(); hauteur = getHeight();
```

```
    if (!tracker.checkID(0)) return;
```

```
    g.drawImage(imgFixe, 0, 0, largeur, hauteur, this);
```

```
    g.drawImage(imgMobile, x, y, largeurImage, hauteurImage, this);
```

A chaque top

Déplacement x

Rectangle où doit s'afficher l'image

Repeindre juste la zone

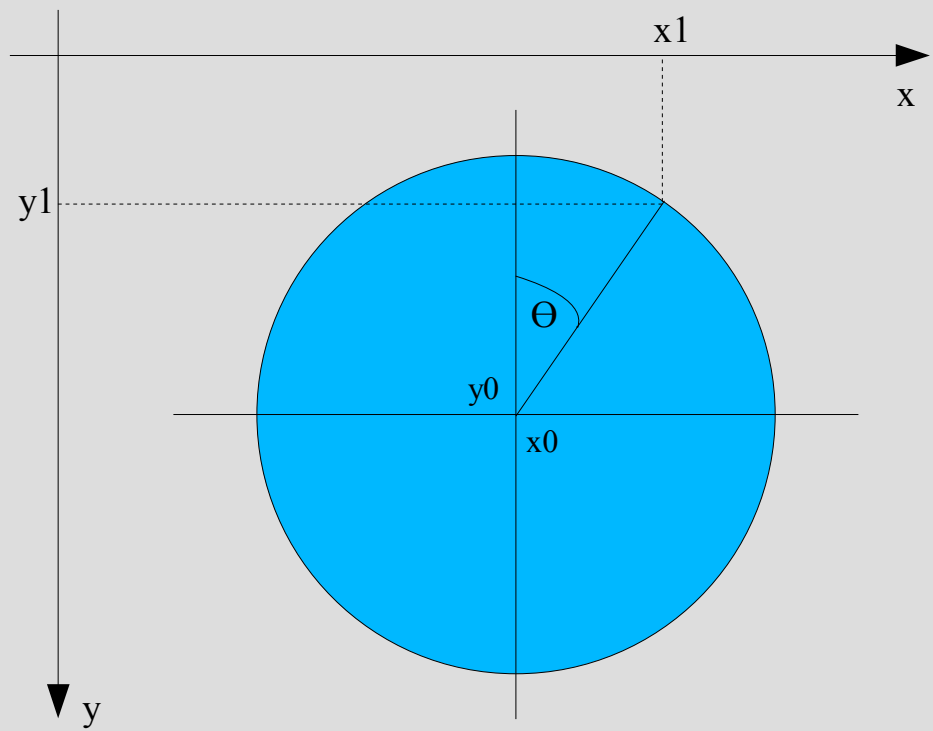
Zone de mise à jour

Peindre les images

Java Swing : Défilement

```
public class ImagePassante extends JFrame implements ActionListener
{
    JButton arret = new JButton("arret"); JButton reprise = new JButton("reprendre");
    Ardoise ardoise = new Ardoise();
    ImagePassante() {
        JPanel p = new JPanel(); p.setBackground(Color.white);
        arret.setActionCommand("arreter"); arret.addActionListener(ardoise); arret.addActionListener(this);
        reprise.setActionCommand("reprendre"); reprise.addActionListener(ardoise);
        reprise.addActionListener(this); reprise.setEnabled(false);
        p.add(arret); p.add(reprise);
        Container interieur = getContentPane();
        interieur.setLayout(new BorderLayout());
        interieur.add(p, BorderLayout.NORTH); interieur.add(ardoise, BorderLayout.CENTER);
        interieur.setBackground(Color.white);
    }
    public void actionPerformed(ActionEvent e) {
        if (e.getActionCommand().equals("arreter")) arret.setEnabled(false); reprise.setEnabled(true); }
        if (e.getActionCommand().equals("reprendre")) { arret.setEnabled(true); reprise.setEnabled(false); }
    }

    public static void main(String[] argv) {
        ImagePassante monCadre = new ImagePassante ();
        monCadre.pack(); monCadre.setVisible(true);
        monCadre.ardoise.timer.start(); } }
}
```



Graphique 2D

- **L'objet Graphics2D de Java Swing permet :**
 - **la transformation affine**
 - (rotation, basculement, étirement, ...)
 - **la gestion de couleurs**
 - (contraste, luminosité, netteté, saturation...)
 - **La gestion du rendu**
 - (texture, dégradé de couleurs, ...)
 - **L'édition de texte améliorée**

Transformations

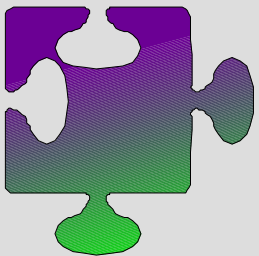
- **G2D permet :**
 - la rotation (rotate), le redimensionnement (scale), le basculement (shear), la translation (translate) de graphiques
 - Il est possible de combiner les transformations :
 - Utilisation de la classe de Transformation Affine :
 - *AffineTransform*

Transformations

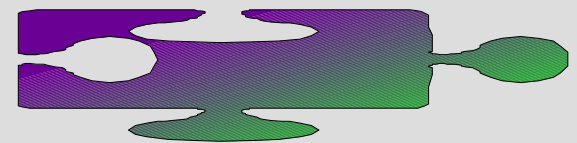
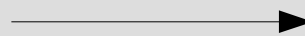
- **Affine Transform** : appliquer une transformation affine sous forme de matrice à chaque point (x, y) d'un graphique :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} a & c & tx \\ b & d & ty \end{bmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a.x + c.y + tx \\ b.x + d.y + ty \end{pmatrix}$$

Redimensionnement

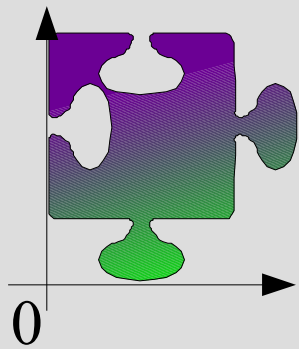


$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix}$$

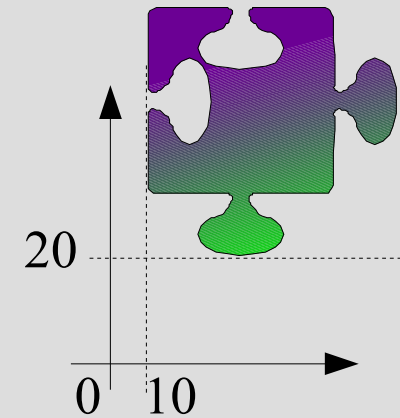
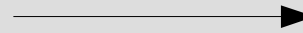


Transformations

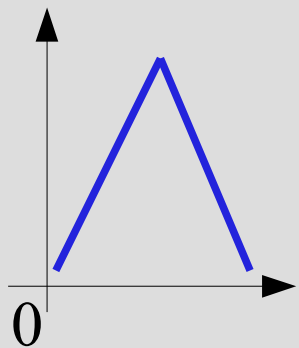
Translation



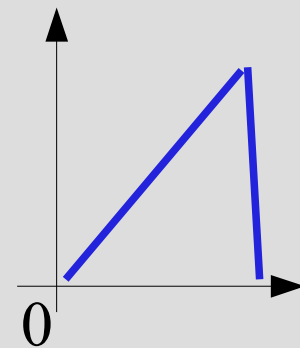
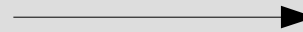
$$\begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 20 \end{bmatrix}$$



Basculement

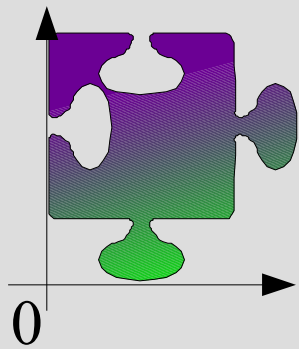


$$\begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

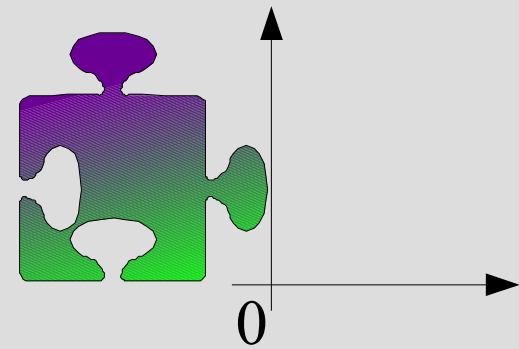
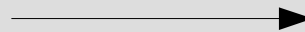


Transformations

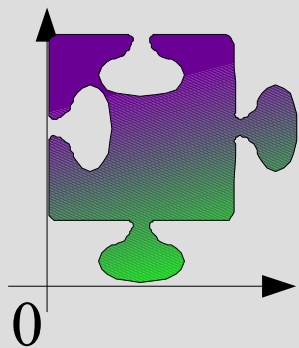
Rotation $\pi/2$



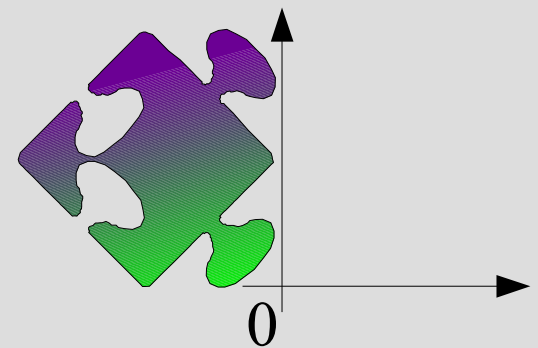
$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$



Rotation θ



$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \end{bmatrix}$$



Transformations

- Utilisation de TransformAffine :
 - Dans la méthode paint ou paintComponent :

```
public void paintComponent(Graphics g)
```

```
{
```

```
    super.paintComponent(g);
```

```
    if (!tracker.checkID(0)) return;
```

```
        Graphics2D g2d = (Graphics2D) g;
```

```
        AffineTransform atOld = g2d.getTransform();
```

```
        AffineTransform atT = new AffineTransform();
```

```
        atT.rotate(Math.PI/6);
```

```
        g2d.drawImage(imgFixe, atT, this);
```

```
}
```