

---

# Architecture Logicielle pour l'Interaction

durée : **2 heures**

documents autorisés : **1 feuille A4 reto-verso manuscrite de votre main**

consignes :

Le barème est **indicatif**. La qualité de la **présentation**, de l'**expression**, de l'**orthographe** sera prise en compte dans la notation (**4 points**).

Si le sujet présente des ambiguïtés, précisez vos choix. Il sera tenu compte de vos hypothèses. Lorsqu'il est demandé de produire du code, ne vous focalisez pas sur la correction de la syntaxe.

---

## Généralités (5 points)

---

### Question 1

- Définissez "interface" et "interaction".
- Expliquez la différence entre ces deux termes dans le contexte de l'Interaction Homme-Machine (IHM).

### Question 2

- Comparez l'évolution dans le temps des performances des ordinateurs et celles des humains.
- Situez dans ce cadre la problématique de l'IHM.

### Question 3

Situez dans le temps (donnez la décennie de) :

- l'invention de la souris ;
- l'invention des interfaces graphiques utilisant la métaphore du bureau.

---

## Programmation par événements (6 points)

---

### Question 4

Soient les deux interactions suivantes qui commencent sur l'enfoncement du bouton gauche de la souris ( $\downarrow$ ), se poursuivent lors de son déplacement, bouton enfoncé (D pour *drag*), et se terminent sur le relâchement du bouton ( $\uparrow$ ) :

- la sélection d'objets par une zone rectangulaire (gérée par les fonctions données :  
`start_selection(start_point), update_selection(point), end_selection(end_point)`);
- le déplacement de la zone de travail (petite main gérée par :  
`start_drag(), drag(dx, dy), end_drag()`).

- Dessinez les machines à états réalisant ces interactions.

On souhaite combiner ces deux techniques de la manière suivante : lorsque le bouton gauche de la souris est enfoncé, c'est l'interaction de déplacement qui s'enclenche si la touche espace est enfoncée, sinon c'est l'interaction de sélection.

La touche espace envoie les événements P (*press*) pour l'enfoncement et R (*release*) pour le relâchement.

- Dessinez la machine à état réalisant regroupant les interactions (si besoin, on pourra stocker dans un variable booléenne l'état de la touche espace).

**Question 5**

Réalisez en java la machine à états à laquelle vous êtes parvenu à la question 4b). Pour cela, vous est donnée une classe `Canvas` qui utilisera votre classe `Listener`. C'est cette dernière que vous devez réaliser.

```
package exam.ui;

import javax.swing.JComponent;
import java.awt.Point;

public class Canvas extends JComponent {
    Listener listener;
    public Canvas() {
        listener = new Listener(this);
        addKeyListener(listener);
        addMouseListener(listener);
        addMouseMotionListener(listener);
        ...
    }

    public void start_selection(Point start_point) { ... }
    public void update_selection(Point point) { ... }
    public void end_selection(Point point) { ... }

    public void start_drag() { ... }
    public void drag(int dx, int dy) { ... }
    public void end_drag() { ... }

    ...
}
```

Les interfaces fournies par `java.awt.event` permettant de recevoir les événements sont données ci-dessous, elles disposent toutes d'un adaptateur associé (`XxxxAdapter` pour `XxxxListener`).

Pour écouter la **souris** :

```
public interface MouseListener {
    public void mousePressed(MouseEvent e);
    public void mouseReleased(MouseEvent e);
    ...
}

public interface MouseMotionListener {
    public void mouseDragged(MouseEvent e);
    ...
}
```

L'interface `MouseListener` qui réunit ces deux interfaces est également disponible.

L'interface suivante permet d'écouter le **clavier** :

```
public interface KeyListener {
    public void keyPressed(KeyEvent e);
    public void keyReleased(KeyEvent e);
    public void keyTyped(KeyEvent e);
}
```

Les `MouseEvent` ont les méthodes utiles suivantes :

- `Point` `getPoint()` qui retourne le lieu de l'événement ;
- `int` `getX()` et `int` `getY()` qui retournent la même information.

Les `KeyEvent` ont une méthode `int` `getKeyCode()` dont le résultat peut être comparé à la constante `KeyEvent.VK_SPACE` pour savoir s'il s'agit bien de la touche espace.

Votre classe `Listener` pourra commencer ainsi :

```
package exam.ui;

public class Listener ... {
    protected Canvas canvas;
    public Listener(Canvas canvas) {
        this.canvas = canvas;
        ...
    }
    ...
}
```

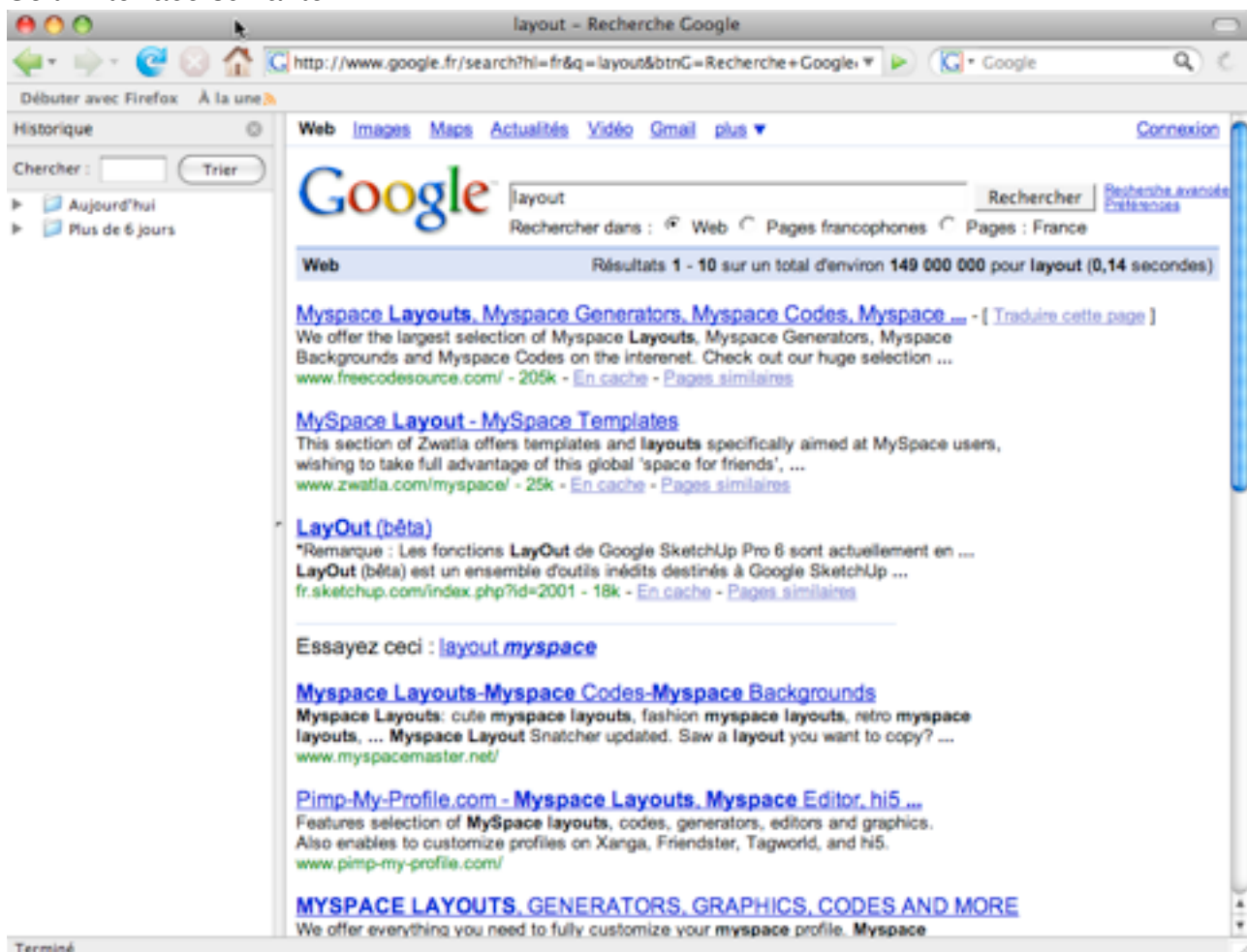
---

## Composants interactifs (5 points)

---

### Question 6

Soit l'interface suivante :



Donnez sous forme graphique l'arbre des composants interactifs constituant cette interface en indiquant les gestionnaires de géométrie utilisés pour les conteneurs.