Techniques des Logiciels Interactifs

durée indicative : **2 heures**
documents autorisés : **1 feuille A4 recto-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 de **manière significative**.
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

a) Expliquez en quoi les principes de la manipulation directe (actions physiques sur des représentations d'objets ; et opérations rapides, incrémentales et réversibles) augmentent l'utilisabilité d'une interface.
b) Donnez un exemple et un contre-exemple de manipulation directe dans un système que vous utilisez régulièrement.

Question 2

a) Comment distinguer ce qui fait partie du noyau fonctionnel de ce qui fait partie de l'interface d'une application ?
b) Selon Fekete [1996], le noyau fonctionnel doit fournir les trois services suivants : notification, prévention des erreurs, et annulation. Expliquez pourquoi en donnant un exemple d'interaction impossible ou difficile à réaliser pour chacun de ces services lorsqu'il est absent.

Programmation par événements (8 points)

Question 3

Vous allez spécifier le fonctionnement de l'interacteur « bouton inverseur » à l'aide d'une machine à états.

En partant du modèle simple pour lequel un bouton :
- a deux états (`DISARMED` et `ARMED`) ;
- passe de `DISARMED` à `ARMED` lorsqu'il reçoit un évènement ↓ (enfoncement du bouton gauche de la souris) ; et
- revient au premier état en déclenchant l'action `action()` lorsqu'il reçoit l'évènement ↑ (i.e. relâchement du bouton gauche),

a) ajoutez la prise en compte des évènements `enter` et `leave` qui se produisent lorsque le curseur entre et sort du bouton (dans ce cas, `action()` a lieu si et seulement si ↓ et ↑ ont tous deux lieu dans le bouton, quoiqu'il se passe entretemps).

Un bouton inverseur bascule de plus entre deux comportements à chaque fois que l'action est déclenchée. Par exemple, le bouton qui déclenche la lecture de la musique dans iTunes se transforme en bouton arrêtant la lecture quand il est utilisé (et vice-versa).

b) En repartant de la machine de l'étape précédente, donnez la machine à états de ce bouton inverseur qui déclenche alternativement les action `play()` et `stop()`.

Question 4

Soit le code suivant qui réalise l'interaction de translation lors du déplacement de la souris, bouton gauche enfoncé, et celle de zoom sur le rectangle défini par les points où le bouton droit a été enfoncé puis relâché après déplacement de la souris.

```
package grapher.ui;

...
import java.awt.Point;
import java.awt.Cursor;

import java.awt.event.*;
import javax.swing.event.*;

import javax.swing.JPanel;


public class GraphPresentation extends JPanel {
	...
	protected Listener listener;

	public GraphPresentation() {
		...
		listener = new Listener();
		addMouseListener(listener);
		addMouseMotionListener(listener);
	}
	...

	private class Listener
		extends MouseInputAdapter {

		final Cursor hand_cursor = new Cursor(Cursor.HAND_CURSOR);
		final Cursor default_cursor = Cursor.getDefaultCursor();

		Point mouse, left, right;

		public void mousePressed(MouseEvent e) {
			mouse = e.getPoint();
			switch(e.getButton()) {
			case MouseEvent.BUTTON1:
				left = mouse;
				setCursor(hand_cursor);
				break;
			case MouseEvent.BUTTON3:
				right = mouse;
				break;
			}
		}
```

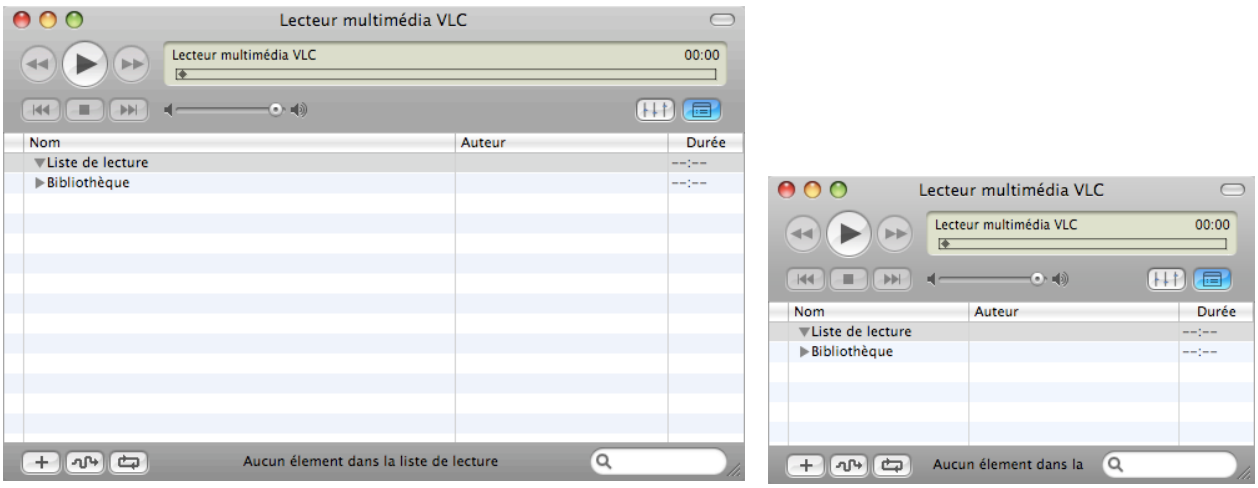
```
            public void mouseReleased(MouseEvent e) {
                  switch(e.getButton()) {
                  case MouseEvent.BUTTON1:
                        left = null;
                        setCursor(default_cursor);
                        break;
                  case MouseEvent.BUTTON3:
                        zoom(right, mouse);
                        right = null;
                        repaint();
                        break;
                  }
            }

            public void mouseDragged(MouseEvent e) {
                  mouse = e.getPoint();
                  if(left != null) {
                        translate(mouse.x - left.x, mouse.y - left.y);
                        left = mouse;
                  }
                  if(right != null) {
                        repaint();
                  }
            }

      }

      protected void translate(int dX, int dY) { ... }
      protected void zoom(Point p0, Point p1)  { ... }
}
```

a) Dessinez la machine à états de l'interaction réalisée.
b) Modifiez la réalisation de la classe Listener pour qu'elle utilise une énumération pour stocker son état courant.

Composants interactifs (5 points)

Question 5

Soit l'interface graphique suivante (avant et après redimensionnement) :



Donnez l'imbrication de conteneurs permettant de réaliser cette interface en Java en vous arrêtant aux conteneurs ayant pour *layout* un `BorderLayout` (pour mémoire, un `BorderLayout` comporte au plus cinq fils, ceux des bords pouvant être absents : un en haut et un en bas de hauteurs fixées et de largeur variant avec celle de la fenêtre, un à gauche et un à droite de largeurs fixées et de hauteur variant avec celle de la fenêtre, et un central occupant la place restante).

Remarque : la liste de lecture est visible dans les deux exemples, il n'est pas demandé de traiter le cas où elle disparaîtrait lors du redimensionnement.