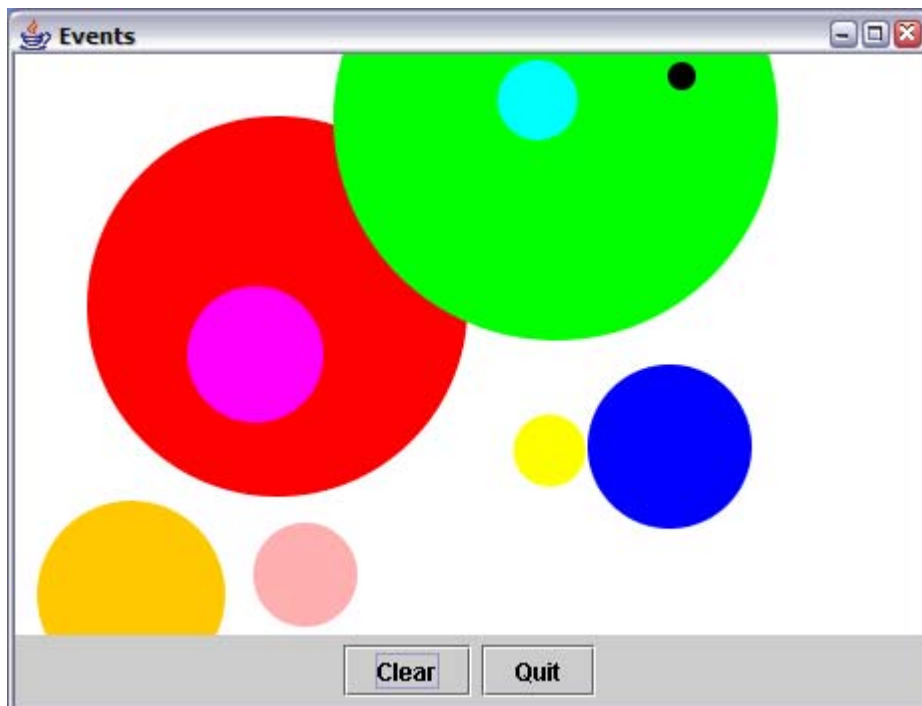


Laboratoire 10: Disques

Durée du laboratoire: 4 périodes.

1. Introduction

Il s'agira de se familiariser avec la gestion de l'interface graphique Java (API Swing, événements et classes internes) en réalisant une application permettant de dessiner des disques.



Un disque est défini par:

- Son centre, lors de la pression du bouton (gauche) de la souris.
- Son rayon, par la distance entre son centre et le point où est relâché le bouton de la souris.
- Son identifiant, permettant de déterminer la couleur à utiliser pour le dessiner dans une liste de couleurs prédéfinies.

De plus:

- Avant que le bouton de la souris ne soit relâché, il doit être possible de visualiser le disque en cours de création.
- Un disque doit pouvoir être effacé en utilisant le bouton droit de la souris (ou de la séquence équivalente sur Macintosh). Si deux ou plusieurs disques sont superposés, effacer celui qui a été ajouté le plus récemment.
- Un disque doit pouvoir être déplacé en appuyant sur la touche SHIFT et par opération glisser-déplacer. Le disque doit être visible au cours de son déplacement.
- Le bouton Clear permet de réinitialiser l'application et le bouton Quit de la quitter.

2. Mise en œuvre

Listes:

- Utiliser la classe `LinkedList`.
- Pour pouvoir parcourir une liste dans les deux sens, utiliser l'interface `ListIterator` (et non pas simplement `Iterator`).

Modules à importer:

- `javax.swing.*`, `java.awt.*` et `java.awt.event.*`.

Composants graphiques:

- Fenêtre principale : `JFrame`
- Zone de dessin (centre): sous-classe de `JPanel`
- Zone de boutons (bas): `JPanel`
- Boutons: `JButton`

Ajout d'un composant fils dans un composant (p.ex. d'un `JButton` dans un `JPanel`):

- Eventuellement définition d'un layout (p.ex. `frame.getContentPane().setLayout(LayoutManager)`).
- Ajout dans une `JFrame`: `frame.getContentPane().add(...)`.
- Ajout dans un `JPanel`: `panel.add(...)`.
- Voir la documentation des classes `JFrame`, `JPanel`, `LayoutManager` et ses sous-classes (notamment `BorderLayout`).

Dimensionner une fenêtre à la taille préférée de ses composants:

- `frame.pack()`.
- Si un des composants est une spécialisation d'un composant existant (p.ex. une extension de `JPanel`), la méthode `public Dimension getPreferredSize()` devra y être redéfinie.

Rendre visible une `JFrame`:

- `frame.setVisible(true)`.

Dessin dans un `JPanel`:

- Définir la couleur de fond d'un composant: `setBackground(Color c)`.
- Méthode de dessin automatiquement appelée par le système sur tous les composants, à éventuellement redéfinir: `public void paintComponent(Graphics g)`.
- Primitives de dessin: voir la classe `Graphics`.

Evènements:

- Boutons, au moyen d'une classe anonyme:

```
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        /* gestion de l'évènement */
    }) ;
```

- Fenêtre de dessin:

- Implémenter les interfaces `MouseListener` et `MouseMotionListener` dans la classe gérant le dessin.
- Enregistrer les objets de cette classe comme pouvant répondre aux évènements de la souris (`addMouseListener(this)` et `addMouseMotionListener(this)` dans le constructeur).
- Détection d'une pression du bouton droit dans `public void mousePressed(MouseEvent e)`: `e.getButton() == MouseEvent.BUTTON3` (ou `e.isPopupTrigger()`).
- Détection de l'appui de la touche `SHIFT` (permettant le déplacement d'un disque): à rechercher dans la documentation Java !