

**Travaux Dirigés n° 5 : Interfaces Graphiques**

*Objectifs : Apprendre les mécanismes de la programmation événementielle et des interfaces graphiques en Java. Savoir décomposer une application graphique et manipuler les gestionnaires de disposition (layout managers). Savoir réaliser des dessins basiques.*

## 1 Exercice 1 : gestion d'événements liés à des boutons

1. Écrire une application permettant d'obtenir une interface graphique contenant un bouton. L'appui sur ce bouton doit provoquer l'affichage d'un message sur le terminal de contrôle.
2. En reprenant le programme précédent, ajouter un deuxième bouton qui permet de sortir de l'application. On s'assurera que les deux boutons sont visibles.
3. En reprenant le programme précédent, ajouter un troisième bouton. Au départ le texte affiché sur ce bouton est « 0 », à chaque appui sur ce bouton la valeur affichée est incrémentée de 1.

## 2 Exercice 2 : gestion de la disposition et des événements

Écrire une application permettant d'obtenir l'interface représentée en Figure 1. Chacune des trois bandes horizontales sera d'une couleur différente.



FIG. 1 – Objectif d'interface graphique à réaliser

L'appui sur le bouton copier provoque la recopie du contenu de la zone de texte supérieure vers la zone de texte inférieure. La zone supérieure est alors effacée.

L'appui sur la touche « entrée » dans la zone de texte supérieure provoque le même effet.

Le bouton sortir permet de quitter l'application. On doit pouvoir aussi sortir en cliquant sur le bouton de fermeture de la fenêtre (en haut à droite typiquement).

### 3 Exercice 3 : dessin basique

1. Écrire une application avec une interface du type représenté en Figure 2. Quand on clique sur le bouton Parle, le dessin de la tête est modifié et un message s'affiche. Quand on clique sur Tais-toi, on revient au dessin initial.
2. Modifier l'application précédente en remplaçant les deux boutons par un seul de label clignote, pour que le personnage alterne automatiquement les deux positions toutes les 0.5 secondes. Vous utiliserez pour cela un objet `javax.swing.Timer`.

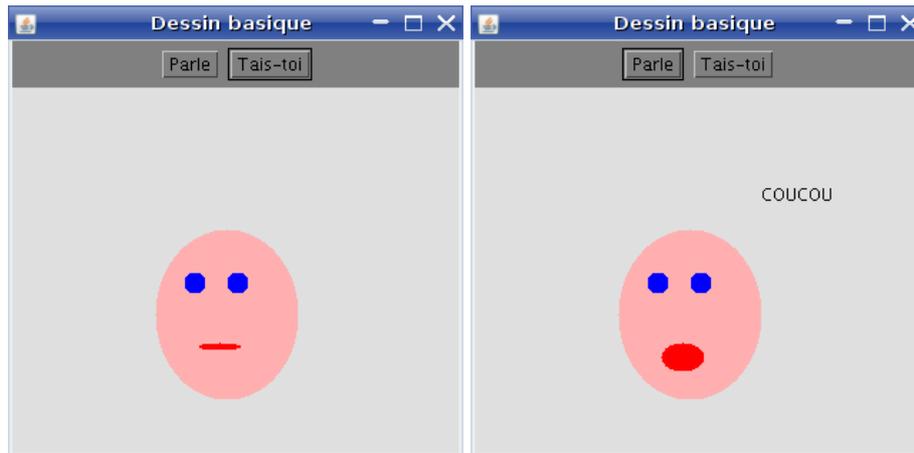


FIG. 2 – Exemple d'interface avec un dessin qui se modifie par action sur un bouton

### 4 Exercice 4 : dessin interactif

Écrire une application qui permet de déplacer une figure ronde à l'aide de la souris. Les captures d'écran en Figure 3 représentent la bille ronde, puis la souris (invisible ici, mais au bout du trait) « tirant » le rond (une extrémité du trait suit le déplacement de la souris), et enfin après avoir relâché la souris, le rond se positionne à l'endroit où on a relâché la souris.

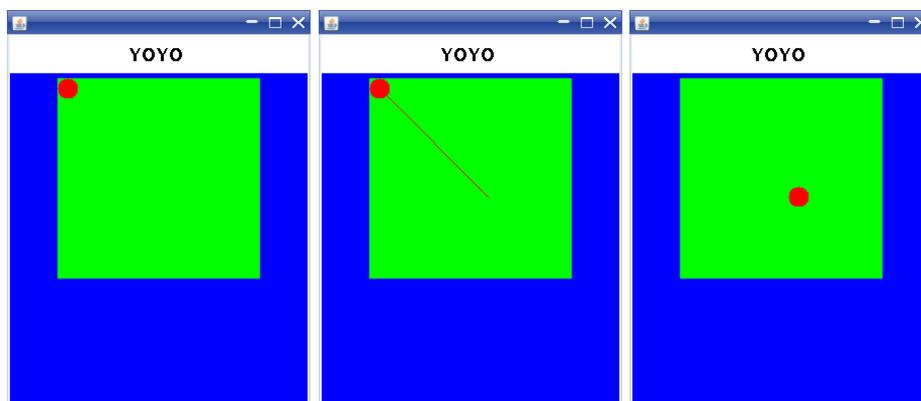


FIG. 3 – Exemple de dessin évoluant en fonction des actions de l'utilisateur à la souris