

Programmation Orientée Objets

TD n°7 : Interfaces Graphiques (suite) : les événements

Exercice 1 : Un premier bouton avec écouteur

Ajouter un bouton intitulé « Arc-en-ciel » de dimension 180*60 pixels, centré en haut de votre panneau. L'évènement généré par un clic sur ce bouton est traité par le panneau qui lance un appel à la méthode `changeCouleurs()`. Dans un premier temps, `Panneau` pourra implémenter `ActionListener`. Attention au rafraîchissement de l'affichage (méthode `repaint()`).

Exercice 2 : Un deuxième bouton (et un 2^{ème} écouteur)

Écrire la méthode `ajouterFormeAlea()` de la classe `EnsembleFormeAlea` permettant d'ajouter une forme à l'ensemble de la même manière que dans le constructeur d'`EnsembleFormeAlea` (cf. exercice 3, TD n°4). Si l'ensemble est complet, une forme choisie aléatoirement est remplacée par la nouvelle forme.

Ajouter un bouton intitulé « Nouvelle forme » de dimension 180*60 pixels, à côté de votre premier bouton. Un clic sur le bouton ajoute une nouvelle forme aléatoire sur le panneau. Afin de faciliter la gestion des événements sur les deux boutons, remplacer l'implémentation d'`ActionListener` (... extends `JPanel` `ActionListener`) par deux adaptateurs d'évènements : un écouteur est créé pour chaque bouton sous la forme d'une classe interne anonyme.

Exercice 3 : Dessiner un cercle

Ajouter à votre panneau un écouteur d'évènements permettant à l'utilisateur de créer un cercle à la souris. La pression du bouton gauche de la souris positionne le centre du cercle, et le rayon est déterminé quand l'utilisateur relâche le bouton.

Dans un deuxième temps, modifier l'écouteur afin que l'utilisateur puisse visualiser le cercle en cours de création à partir du moment où il a pressé le bouton gauche de la souris. Attention, ce comportement doit cesser dès que l'utilisateur a terminé le cercle (en relâchant le bouton).

Attention au rafraîchissement : `repaint()` ne suffit plus (`super.paintComponent(g)`).

Exercice 4 : Dessiner un polygone

De la même manière, permettre à l'utilisateur de dessiner des polygones. Pour signaler son intention de dessiner un cercle ou un polygone, l'utilisateur clique sur un bouton prévu à cet effet, intitulé « Nouveau Cercle » ou « Nouveau Polygone ».

Puis, des clics successifs (bouton gauche de la souris) positionnent les sommets du polygone. L'utilisateur termine le polygone par un clic droit (à partir de 3 sommets).

L'utilisateur doit pouvoir visualiser le polygone au fur et à mesure de sa construction : les points déjà positionnés sont reliés par des segments, et un segment supplémentaire s'affiche entre le dernier point positionné et le curseur de la souris. Le plus simple consiste à afficher le polygone incomplet dans la méthode `paintComponent()` de `Panneau`, plutôt que d'utiliser la méthode `seDessiner()` de `Polygone`.

Exercice 5 : Organisation de la fenêtre et barre d'état

Modifier l'organisation de la fenêtre de manière à disposer de 3 zones distinctes : une barre d'outils flottante (`JToolBar`) contenant les différents boutons, initialement positionnée dans la partie supérieure de la fenêtre, l'espace de dessin (`JPanel`) au centre, et une barre d'état (`JLabel`) affichant des informations à propos de l'action en cours : dessin d'un cercle, d'un polygone, etc., ainsi que la position courante de la souris.

Exercice 6 : Sauvegarde

Ajouter deux boutons intitulés « Sauvegarde » et « Restauration », permettant de sauvegarder l'ensemble de formes dans un fichier `ensembleFormes.ser` puis de le restaurer après avoir fermé et relancé l'application. `Forme` et `EnsembleForme` pourront implémenter / hériter de `java.io.Serializable`.

Programmation Orientée Objets

TD n°7 bis (facultatif) : Interfaces Graphiques

Exercice 1 : Remplir une forme

Proposer à l'utilisateur un bouton lui permettant de remplir une forme (de la même couleur que sa bordure). Une fois le bouton cliqué, l'utilisateur clique¹ sur la forme à remplir.

Exercice 2 : Choisir une couleur

Proposer à l'utilisateur de choisir la couleur de remplissage, à l'aide du composant `JColorChooser`.

Exercice 3 : Supprimer une forme

Proposer à l'utilisateur un bouton lui permettant¹ de supprimer une forme.

Exercice 4 : Confirmation

Compléter l'exercice 3 en demandant une confirmation à l'utilisateur, via une boîte de dialogue affichant le message « Vous avez demandé la suppression de la forme n°4 (Cercle) ».

Exercice 5 : Sélectionner une forme

Permettre à l'utilisateur de supprimer plusieurs formes simultanément. Le bouton de l'exercice 3 est utilisé, mais cette fois-ci l'utilisateur sélectionne une zone au lieu d'un point. Toutes les formes entièrement contenues dans cette zone sont supprimées. La boîte de dialogue de confirmation donne la liste des formes concernées.

Exercice 6 : Modifier un polygone

Proposer à l'utilisateur un bouton lui permettant de modifier un polygone. Une fois le bouton cliqué, l'utilisateur peut presser le bouton gauche de la souris sur la zone de dessin. Le point le plus proche sera sélectionné et déplacé par glisser-déposer.

Exercice 7 : Personnalisation : boutons et curseur de la souris

Personnaliser les boutons afin qu'ils représentent l'action sous-jacente : un petit cercle (respectivement un petit quadrilatère) quand l'utilisateur dessine un cercle (respectivement un polygone), un arc-en-ciel pour le coloriage des formes, et une croix rouge (ou une tête de mort) pour la suppression des formes.

Personnaliser également le curseur de la souris selon l'action en cours : dessin d'un cercle, d'un polygone, ou suppression d'une forme.

Exercice 8 : Menu contextuel

Proposer un menu contextuel à l'utilisateur pour agir sur les formes. Un clic¹ droit dans la zone de dessin affiche la liste d'actions qu'il est possible d'effectuer sur la forme : supprimer, remplir, zoomer, changer de couleur, rotation.

Le zoom est réalisé par défaut avec un facteur 2. Proposer également à l'utilisateur de donner un facteur personnalisé de zoom au clavier. La rotation est réalisée par défaut avec un angle de 90° dans le sens des aiguilles d'une montre. Proposer également à l'utilisateur de donner un angle personnalisé au clavier.

¹ C'est-à-dire : l'utilisateur clique sur un point contenu dans la forme visée. Si la position cliquée est contenue dans plusieurs formes, c'est la première forme trouvée qui est remplie.