

TP n° 8

Interfaces graphiques 2

Dans ce TP, nous allons implémenter l'interface `MouseListener`. Nous illustrons cela sur un petit jeu : on veut créer une fenêtre avec plusieurs carrés de couleurs aléatoires, cette couleur pouvant changer en fonction des actions faites avec la souris. Le jeu se fini lorsque tous les carrés sont de la même couleur. On rappelle qu'il faut tester systématiquement les méthodes créées.

1. Pour commencer, créer une classe `Cadre` extends `JFrame` ainsi qu'un constructeur sans argument. Ce constructeur affiche simplement une fenêtre de taille 600×600 . De plus, le programme doit s'arrêter lorsque l'on ferme la fenêtre (on utilise pour cela la méthode `setDefaultCloseOperation`).
2. Appeler ce constructeur dans un `main`.
3. Rajouter un conteneur principal dans le constructeur. Celui-ci n'utilisera pas de `LayoutManager` (c'est-à-dire que le `LayoutManager` est initialisé à `null`).
4. Créer une classe interne `Carre` extends `JPanel` avec un constructeur sans argument. Pour le moment, on demande que le constructeur crée un carré de couleur bleue en position (100,200) dont les côtés sont de taille 50. Utiliser pour cela la méthode `setBounds` de la classe `Component`.
5. On demande à présent que la classe `Carre` implémente l'interface `MouseListener` (qui sert à définir les actions faites par la souris). Il y a plusieurs méthodes à redéfinir, pour le moment nous ne demandons pas de leur donner de corps (elles ne font rien). Que faut-il faire pour que (une fois les méthodes redéfinies) les actions voulues aient effectivement lieu ?
6. On souhaite qu'un carré puisse être déplacé à l'aide du click-and-drag (on clique avec le bouton de la souris sur le carré, on déplace le carré en gardant le bouton enfoncé et lorsqu'on le lâche le carré est à la nouvelle position de la souris). On souhaite de plus qu'une fois déplacé le carré soit de couleur rouge. Quelle méthode faut-il implémenter ?
7. Modifier le constructeur de la classe `Carre` pour que la couleur et la position dans la fenêtre soient aléatoires.
8. Modifier le constructeur de la classe `Cadre` de telle sorte que la fenêtre contienne un nombre aléatoire (compris entre 1 et 10) de carrés (créés en faisant appel au constructeur précédent).
9. Écrire une méthode `boolean gagne()` qui teste si tous les carrés sont de la même couleur.
10. Écrire une méthode `void gagne()` qui ferme la fenêtre (et donc arrête le programme) une fois que tous les carrés sont de la même couleur.
11. On souhaite que lorsque le curseur est sur un carré (sans que l'on clique sur la souris) celui-ci devienne bleu. Lorsque l'on clique sur un carré celui doit devenir vert. Écrire les méthodes correspondantes.

12. Modifier la méthode `void gagne()` de telle sorte qu'un message de succès soit affiché sur la fenêtre (on ne demande évidemment plus que la fenêtre se ferme automatiquement). Pour cela on rajoutera dans la classe `Cadre` un attribut `JPanel` `etiquette`.
13. On souhaite que lorsque le jeu est fini, il suffit de cliquer n'importe où sur la fenêtre (y compris sur les carrés) pour que celle-ci se ferme. On souhaite également que la fenêtre se ferme si le curseur de la souris sort du cadre. Écrire les méthodes correspondantes.
14. Si vous avez du temps, vous pouvez réfléchir à une autre façon de modifier la couleur d'un carré. Par exemple, si un carré passe "au-dessus" d'un autre on peut choisir que le carré du dessous récupère la couleur de celui du dessus. Vous pouvez également demander à ce qu'il y ait un message sur la fenêtre décrivant la situation (par exemple "3 carres bleus, 2 carres rouges, un carre vert").