

TP10 – Puissance 4 !

NB : pour ce TP nous allons utiliser les codes produits aux exercices 1 et 6 du TP8. Vous devez donc avoir au moins fait ces exercices afin de pouvoir copier les classes produites dans le dossier du projet que vous allez créer ici.

Partie 1: La classe DrawComponent

Exercice 1: On souhaite disposer d'un composant d'interface graphique, simple d'usage, qui permette d'afficher à l'écran des instances d'objet graphique produit à l'aide des classes de la librairie que nous avons réalisées au TP8. Écrivez donc pour cela le code d'une classe **DrawComponent** qui doit au minima présenter les fonctionnalités suivantes :

- Cette classe doit être un composant d'interface graphique (elle doit donc hériter de la classe **JComponent**).
- Un attribut de nom **dessins** (`Vector<ObjetGraphique> dessins;`) qui va contenir la liste des objets graphiques qui devront être dessinés par le composant.
- Masquez la méthode **paint(Graphics)** afin de réaliser les opérations de dessins nécessaire (c'est-à-dire déclencher **draw(...)** sur chaque objet graphique visible du vecteur **dessins**).
- Une méthode **addDessin(ObjetGraphique o)** qui doit permettre d'ajouter un objet graphique dans la liste **dessins**. A la fin de cette méthode déclenchez l'appel de la méthode **repaint()**, donc en ajoutant l'instruction : **this.repaint()**

Exercice 2: Testez le bon fonctionnement de cette classe en créant un programme de nom **TestDC** qui affiche une fenêtre dans laquelle vous aurez placé un composant **DrawComponent**. Ce programme devra enregistrer des objets graphiques Cercle et Rectangle sur l'instance **DrawComponent** et vous devriez pouvoir constater le fait que ces objets graphiques s'affichent correctement à l'écran.

Exercice 3: On souhaite doter la classe **DrawComponent** de la capacité de gérer les événements de clic de souris. Pour cela :

- commencez par définir une interface de nom **ClickMouseListener** qui déclare l'unique méthode : **void mouseClicked(int x, int y)**.
- puis ajoutez dans la classe **DrawComponent** les méthodes d'enregistrement, et de désenregistrement, des écouteurs :
 - **addClickMouseListener(ClickMouseListener l)**
 - et: **removeClickMouseListener(ClickMouseListener l)**
- enfin, complétez le code de la classe **DrawComponent** pour que l'ensemble des écouteurs soient effectivement notifiés de la position de chaque clic de souris.

Exercice 4 : Testez le bon fonctionnement de l'évènement « ClickMouse » à l'aide de votre programme **TestDC** (exercice 2) en enregistrant sur le composant **DrawComponent** un écouteur qui affiche dans la console les coordonnées x et y de chaque clic de souris.

Partie 2: Le jeu Puissance 4

2.1 La base du jeu

Exercice 5: Créer un programme de nom **Puissance4** qui affiche une fenêtre ayant pour titre "Puissance 4" et une taille 700x600. Placez un composant **DrawComponent** dans cette fenêtre et ajoutez dans celui-ci un objet Rectangle de dimension 560x480 à la position {10,10}, compilez et testez.

Exercice 6: Ajoutez à votre classe **Puissance4** un attribut de nom **colonnes** et de type tableau de **Rectangle**. Initialisez cet attribut dans le constructeur en le remplissant avec 7 rectangles de dimension 80x480 et de position $x=10+i*80$ et $y=10$, avec i le numéro du rectangle (de 0 à 6).

Notez que ces rectangles ne doivent pas être ajoutés au composant **DrawComponent**, vous pouvez cependant le faire temporairement pour vos tests (vous mettrez ensuite ce code d'ajout en commentaire).

Exercice 7: Faites en sorte que votre programme **Puissance4** puisse écouter les clics de souris produits sur le composant **DrawComponent**, et dans le code d'écoute examinez si la position du clique s'est produit sur un des rectangles de l'attribut **colonnes**. Si c'est le cas, votre code doit afficher le numéro de la colonne dans laquelle le clique s'est produit. Nous compléterons plus tard le code qui se trouve à ce niveau.

Exercice 8: Ajoutez à votre classe **Puissance4** un attribut de nom **cases** et de type tableau à deux dimensions contenant des objets **Disc**. Initialisez cet attribut dans le constructeur en le dimensionnant sur 7 colonnes de 6 lignes. Nous utiliserons la convention suivante : le premier indice du tableau (de valeur 0 à 6) représentera chacune des 7 colonnes, et le second indice (de valeur 0 à 5) représentera chacune des 6 lignes.

Ensuite, remplissez ce tableau avec des objets disques de diamètre 35 et de couleur de fond blanc. Positionnez ces disques sur les coordonnées : $x=50+i*80$ et $y=50+j*80$ avec i le numéro de colonne du disque (1^{er} indice) et j son numéro de ligne (2^{ième} indice). Ajoutez également chacun de ces disques au composant **DrawComponent**.

Exercice 9: Ajoutez à votre classe **Puissance4** un attribut de nom **curPlayer** et de type **Disc**. Cet attribut doit être initialisé avec un disque la couleur jaune, de diamètre 35 et de position : $x=630$ et $y=80$. Ajoutez également ce disque au composant **DrawComponent**.

Exercice 10: Complétez le code d'écoute de l'évènement clic de souris (exercice 7) afin de réaliser le traitement suivant si une colonne est concernée par le clique :

Vous devez rechercher dans **cases** le disque de cette colonne qui possède une couleur de fond blanche et dont l'indice de ligne est le plus élevé. Si un tel disque existe :

- (1) vous devez changer sa couleur pour lui affecter celle du disque **curPlayer**
- (2) vous devez ensuite intervertir la couleur de **curPlayer** entre rouge et jaune (il doit devenir rouge s'il est jaune, et jaune s'il est rouge)
- (3) Vous terminerez ce traitement en invoquant la méthode **repaint()** sur la fenêtre

Exercice 11: Ajoutez un bouton de non « reset » en bas de votre fenêtre. L'activation de ce bouton doit conduire simplement à affecter la couleur blanche à tous les disques du tableau **cases**, puis à invoquer la méthode **repaint()** sur la fenêtre.

Exercice 12: Remplacez le rectangle de l'exercice 5 par un objet Box de couleur bleu.

2.2 Améliorations bonus (partie facultative)

Exercice 13 (difficile): Ajoutez une méthode **boolean detectWin(Color c)** à votre programme. Cette méthode doit analyser le tableau **cases** afin de déterminer s'il existe dans le tableau **cases** un alignement de 4 couleurs identiques à c . Si c'est le cas elle doit renvoyer **true**, sinon elle doit retourner **false**.

Ajoutez un appel à cette méthode juste après le code (1) de l'exercice 10, en lui passant la couleur du disque **curPlayer**. Et si cet appel renvoie **true** vous devrez alors afficher un message indiquant que le joueur vient de gagner.

Exercice 14 (très difficile): Ajoutez à votre programme la possibilité de jouer contre l'ordinateur ;)