

TP 8 - AOO - Interfaces graphiques

Olivier Goudet

April 1, 2019

On aura besoin d'éléments de la librairie *javax.swing* dans tout ce TP. Vous pouvez directement importer tous les éléments avec *import javax.swing.**.

Exercice 1

On veut créer un champ de saisie du nom d'un utilisateur comme présenté sur la Figure 1.

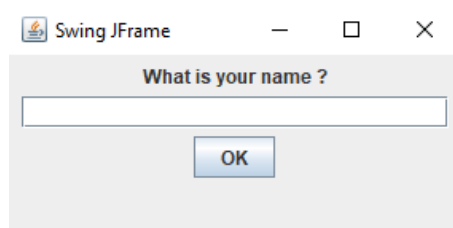


Figure 1: GUI Exo 1

Questions

1. Créer une classe **Test** avec une méthode *main* qui instancie un objet de type **JFrame**. Préciser une taille avec la méthode *setSize*.
2. Ajouter un layout de type **FlowLayout** à cet objet **JFrame** (voir en annexe les différents layouts disponibles).
3. Créer un objet de type **JLabel** avec le texte "What is your name ?"
4. Créer un bouton de type **JButton** avec le texte "OK"
5. Créer un champ de saisie de type **JTextField**. On pourra préciser sa taille en paramètre du constructeur.
6. Ajouter ces différents éléments graphiques à votre objet **JFrame**.
7. Rendez visible la frame avec la méthode *setVisible(true)*.
8. Ajouter un gestionnaire d'évènement de façon à ce qu'une sortie console s'affiche avec le contenu du champ de saisie lorsque l'utilisateur clique sur le bouton "OK". Pour créer un évènement associé à un élément graphique on utilisera un objet de type **ActionListener** (voir annexe du TP). Par exemple, pour ajouter une action au bouton "OK" on pourra faire :

```
bouton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evenement) {  
        //Faire quelque chose...  
    }  
});
```

9. Vérifier que le programme fonctionne correctement.

Exercice 2

On veut créer un jeu de morpion comme présenté sur la Figure 2.

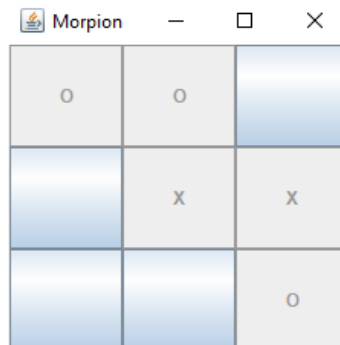


Figure 2: GUI Exo 2

Moteur du jeu

Questions

1. Créer une classe **Game** qui contient un tableau d'entiers à deux entrées de type `int[][]` qui représentera la grille du jeu.
2. Ajouter un attribut de type `int` qui correspond à l'indicateur du joueur qui doit jouer (cet indicateur sera par exemple à 1 si le premier joueur doit jouer ou à 2 sinon).
3. Créer une méthode `jouer(int x, int y)` pour la classe **Game** qui prend en entrée deux positions entières `x` et `y` et effectue l'action sur la grille du jeu en fonction du joueur courant.
Par exemple si le joueur 1 joue en position (0,2), on met la valeur 1 dans la grille du jeu en position (0,2). Ensuite, si le joueur 2 joue en position (1,1), on met un 2 en position (1,1) dans la grille. Les valeurs du tableau d'entiers qui restent à 0 correspondent donc à des cases qui n'ont pas encore été jouées.
Une fois qu'un joueur a joué l'indicateur du joueur courant doit être mis à jour (si il est à 1, il passe à 2 et vice-versa).
4. Créer une classe **Morpion** avec une méthode `main()` qui instancie un objet **Game** et fait appel successivement à la méthode `jouer()` avec différentes positions. Vérifier que tout fonctionne bien en affichant le tableau du jeu à chaque étape.
5. A la fin de la méthode `jouer()`, ajouter un test pour vérifier si le joueur courant a gagné. Si c'est le cas, une sortie console doit être effectuée en affichant l'indicateur du joueur qui a gagné. Par exemple, le joueur 1 gagne si une colonne, une ligne ou une diagonale complète de la grille du jeu a été remplie uniquement avec des 1.
6. Ajouter aussi un test pour vérifier si la partie est nulle (si toutes les cases ont été jouées et qu'il n'y a pas eu de gagnant).
7. Vérifier que le moteur du jeu marche bien que les règles de victoire et de partie nulle ont bien été implémentées.

Interface graphique

Questions

1. Créer une classe **CaseMorpion** avec comme attributs : un bouton de type **JButton**, un objet **Game** et deux entiers `x` et `y`.

2. Ajouter un constructeur qui définira `x` et `y` ainsi que la référence vers l'objet `Game` (en paramètres). Le constructeur doit aussi créer le bouton et ajouter un gestionnaire d'évènement. Si l'utilisateur clique sur le bouton associé à cette case deux choses doivent être réalisées :
 - on appelle la méthode `jouer()` de `Game`.
 - on met à jour l'aspect visuel du bouton avec "X" ou "O" en fonction de l'indicateur du joueur courant et on désactive le bouton.
3. Créer une classe **Morpion** avec une méthode `main()`. Instancier un objet de type **Game**. Créer un objet window de type **JFrame**, définir sa taille et ajouter un layout. Quel layout peut être pratique dans ce cas pour dessiner la grille du jeu ?
4. Créer et ajouter autant d'objets de type **CaseMorpion** que de cases nécessaires pour créer le jeu.
5. Afficher le jeu et tester le.

Bonus

Créer une IA pour le jeu de morpion

1. Implémenter un comportement aléatoire pour le jeu. Tester le jeu en jouant contre lui.
2. Améliorer cette IA de façon à ce que l'ordinateur puisse gagner ou toujours au minimum égaliser.