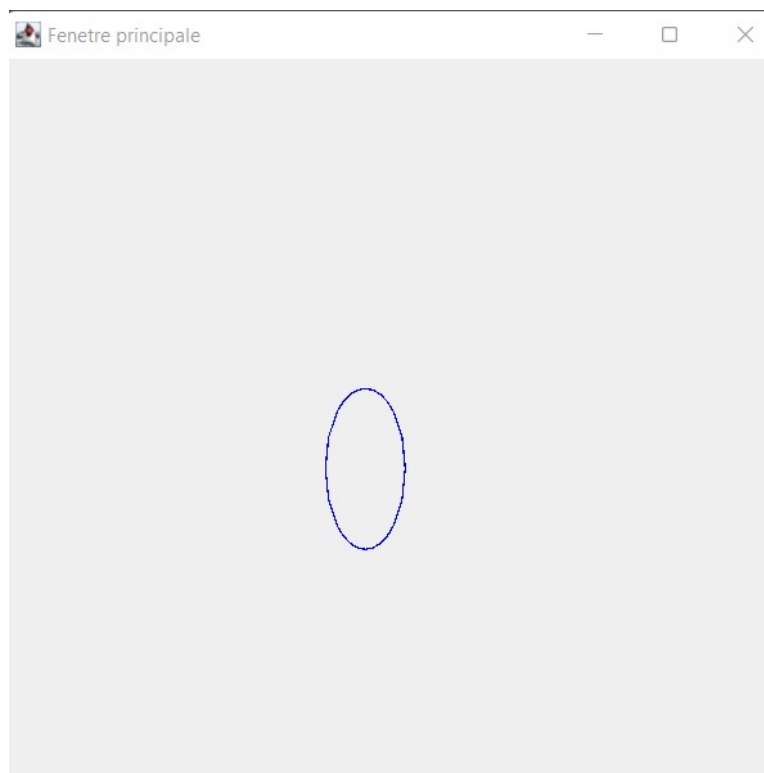


Projet Flappy bird

Tutoriel Séance 1

Le but de ce mini-projet est de réaliser un mini-jeu inspiré de Flappy bird. Ce jeu consistant à faire parcourir un ovale autour d'une ligne brisée sans que celle-ci ne soit touchée. Voilà un exemple de comment pourrait être modélisé notre interface graphique.



Lors de cette 1ère séance, nous avons commencer par poser les bases de notre projet. Nous devons commencer par créer les sous-fonctionnalités suivantes:

- *Création de la fenêtre où serait dessiné notre ovale.
- *Déplacement vers le haut ou vers la bas de l'ovale lorsque l'on clique dans l'interface.

L'implémentation de ces sous-fonctionnalités sont simples à faire mais néanmoins importantes car elles nous permettrons de représenter notre ovale, ainsi que la fenêtre où serait modélisé notre jeu.

Pour le moment, nous avons implémenté une des fonctionnalités principales de ce mini-jeu :

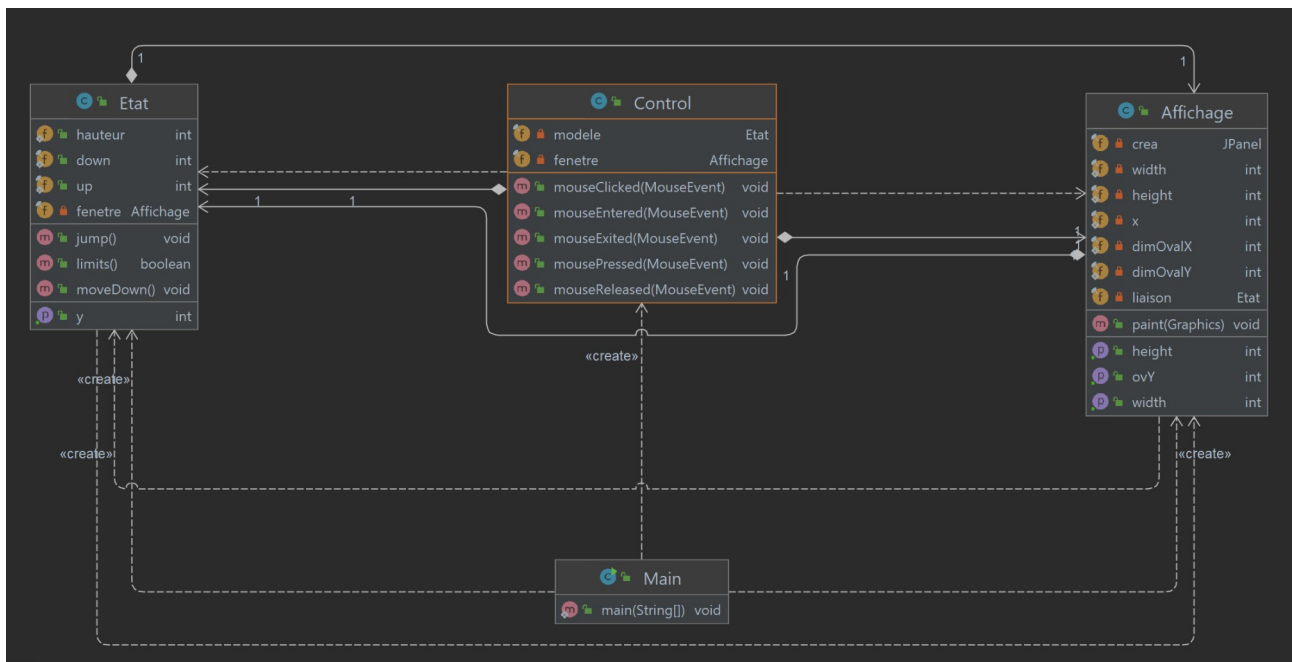
- * la réaction de l'ovale aux clics de l'utilisateur.

Afin de concevoir notre interface graphique, nous avons décidé de développer le motif MVC. En effet, en adoptant ce modèle nous allons pouvoir gérer plus facilement tous les éléments de notre mini-jeu. Pour cela, nous avons divisé notre modèle en 3 classes:

- * Affichage, sert à créer la fenêtre d'affichage de notre mini-jeu
- * Etat, sert à modéliser les données qui caractériseront l'état de notre interface.
- * Control, qui va se charger de modifier l'état de notre ovale en fonction des événements liés à la souris de l'ordinateur.

Lors de la conception de notre fenêtre, nous avons décidé d'utiliser l'API Swing ainsi que la classe JPanel pour le dessin de l'ovale. Nous avons commencé par définir plusieurs constantes essentielles:

- * width et height, les bornes max de notre fenêtre d'affichage
- * x, étant donné que l'ovale ne fera qu'un mouvement vertical, alors la position horizontale n'a pas besoin d'être modifiée pour l'instant.
- * dimOvalX et dimOvalY, les dimensions de l'ovale



Quant au déplacement de notre ovale, nous sommes partis sur l'idée de faire une interaction avec la souris de l'ordinateur. Pour cela, nous allons manier la classe MouseListener. De plus, nous avons aussi défini une constante y, représentant la position verticale de notre ovale.

La 1ère partie de notre mini-jeu a été exportée sous forme de projet à exécuter à l'intérieur d'un IDE. Afin de pouvoir visualiser notre 1ère partie, il faudra importer le projet dans votre IDE, sélectionner la classe Main à la racine du projet puis « Run as Java Application ». Cliquez sur la fenêtre afin de faire monter l'ovale.

Par la suite, nous développerons la ligne brisée ainsi que son déplacement donnant l'illusion que c'est l'ovale qui avance autour de cette ligne.

En conclusion, nous avons réalisé la création de la fenêtre d'affichage du mini-jeu ainsi que l'ovale et son déplacement grâce à l'interaction avec la souris. Sur cette 1ère partie, la partie la plus compliquée était de concevoir la fenêtre d'affichage avec ce nouvel API. Après cela, il n'a pas été extrêmement difficile de fabriquer un motif MVC.