# Rapport de projet

Mahdi RANJBAR

10 janvier 2022

# Table des matières

# Introduction

Ce mini-projet a pour but de mener à bien le développement d'un jeu étant inspiré par le jeu "Flappy Bird" dans lequel un ovale se déplace le long d'une ligne brisée. L'ovale descend constamment vers le bas et en cliquant dans l'interface, ce dernier monte vers le haut de nombre de pixels fixes. Le but du jeu c'est de maintenir l'ovale en cliquant sur l'écran de façon que la ligne soit à l'intérieur de ce dernier. Voici un extrait de l'interface graphique de ce jeu:


FIG. 1: Extrait de GUI du jeu

# Analyse globale

I. La réaction de l’ovale aux clics de l’utilisateur

| Sous-fonctionalité | Difficulté | Priorité |
|---|---|---|
| **Création d’une fenêtre dans laquelle l’ovale est dessiné** | Simple | Priorotaire |
| **Déplacement de l’ovale vers le haut lorsqu’on clique dans l’interface** | Simple | Priorotaire |

II. Le défilement automatique de la ligne brisée

III. L’interface graphique avec l’ovale et la ligne brisée

# Plan de développement

| Liste des tâches | Temps éstimé |
|---|---|
| Analyse du problème | 15 min |
| Conception, développement et test d'un fenêtre avec un ovale | 15 min |
| Conception, développement et test du mécanisme de déplacement de l'ovale | 60 min |
| Acquisition de compétences en Swing | 60 min |
| Documentation du projet | 60 min |

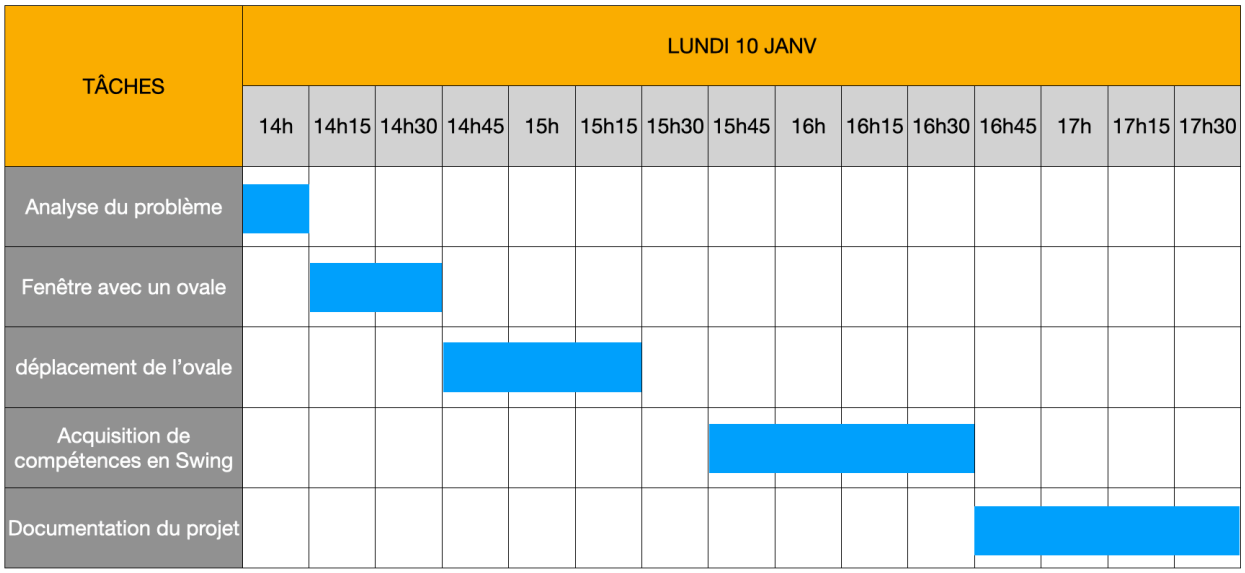
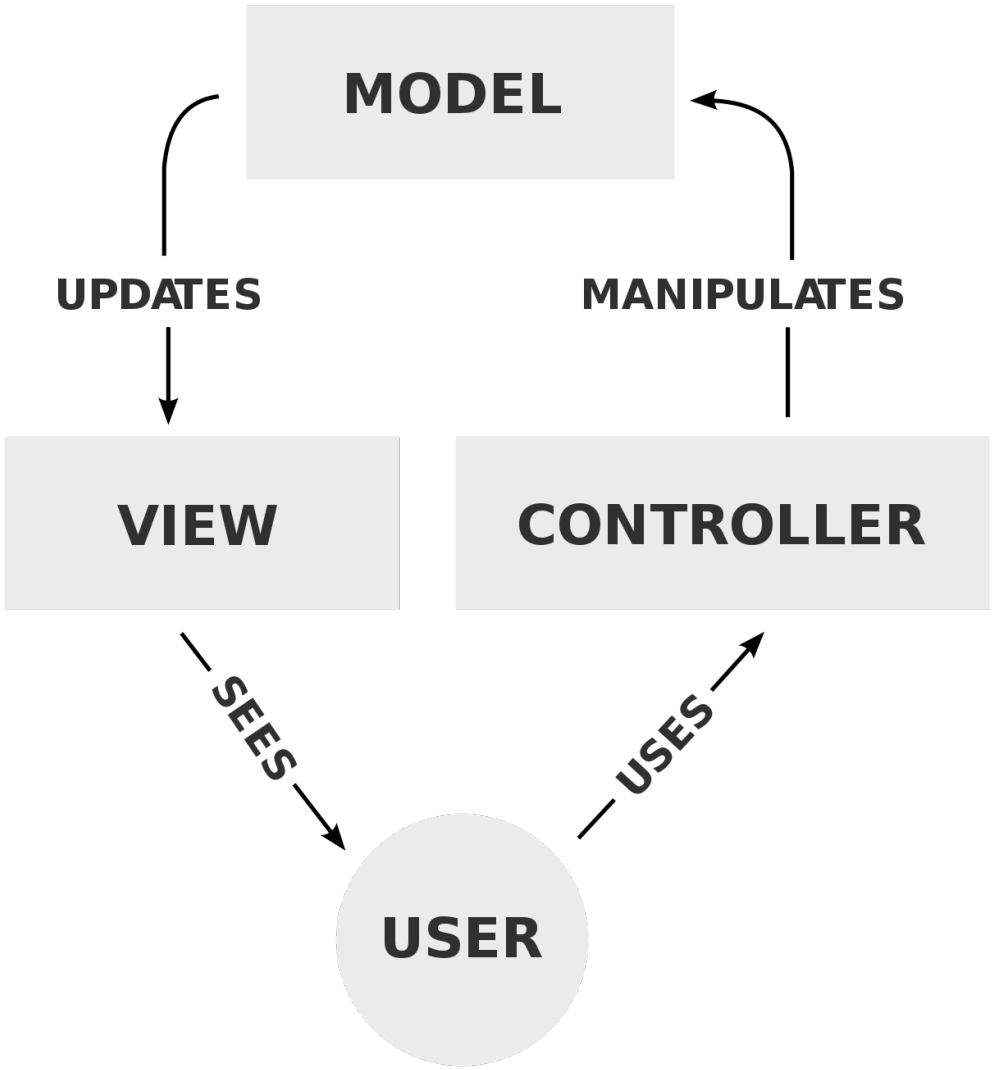



FIG. 2: Diagramme de Gantt 1ière séance

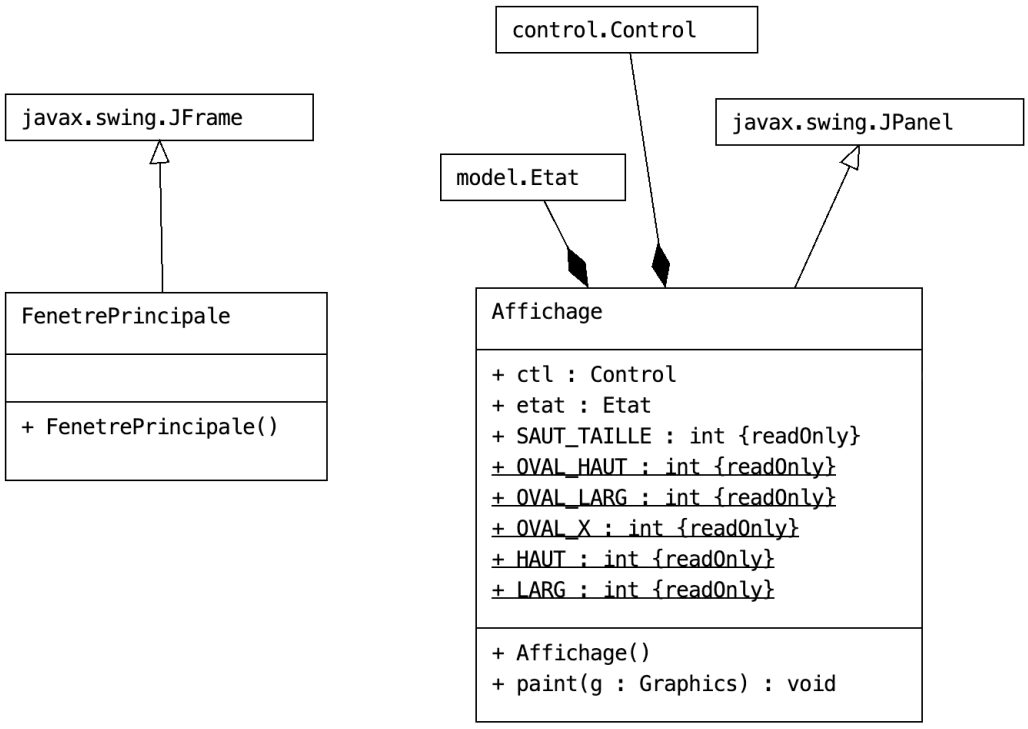
## Conception générale

Nous avons adopté le motif MVC pour le développement de notre interface graphique. Création de la fenêtre dans laquelle est dessiné l'ovale est associé au bloc View parce que c'est la représentation (affichage) de GUI du jeu. Ensuite, pour le déplacement de l'ovale vers le haut lorsqu'on clique dans l'interface, sachant qu'il s'agit de gestion de l'événement clique et d'effectuation des changements dans le bloc Model et réaffichage, est donc associé au bloc Controller.

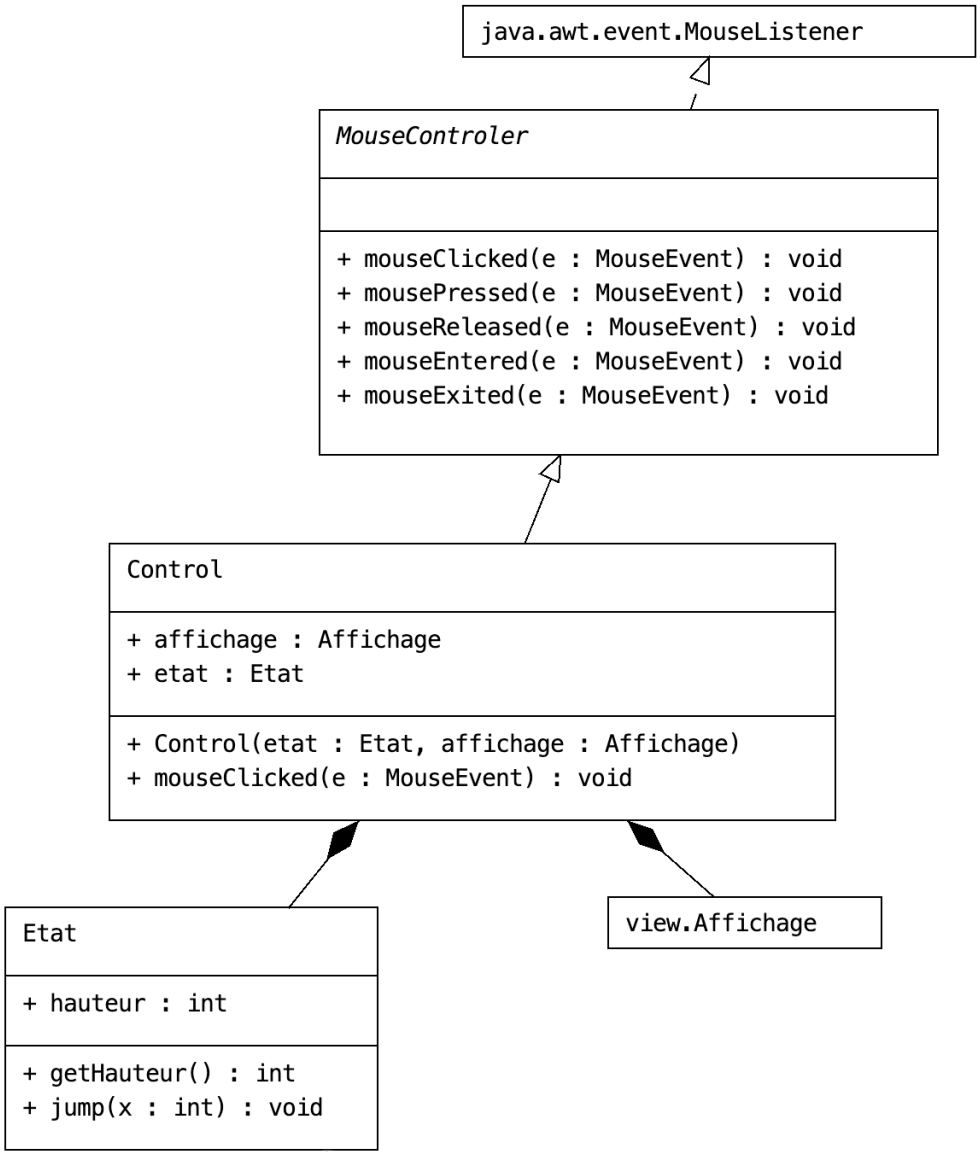


# Conception détaillée

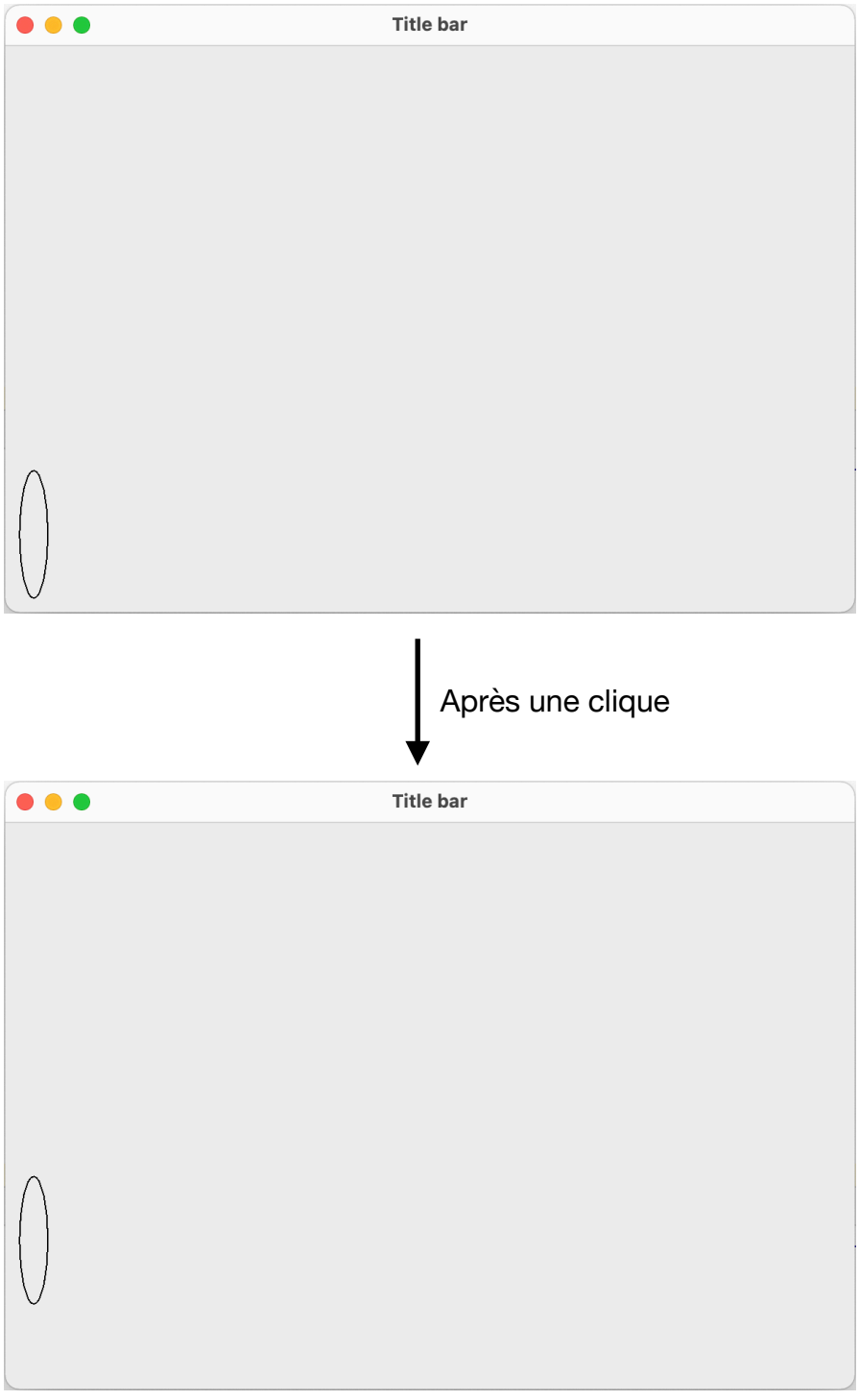
- Tout d'abord on crée la classe FenetrePrincipale qui hérite de la classe JFrame et va nous servir à avoir notre fenêtre principale en appelant son constructeur avec le paramètre titre de la fenêtre. Ensuite on crée la classe Affichage qui hérite la classe JPanel et qui va nous servir à représentation de GUI du jeu. Cette classe contient l'attribut etat qui permet d'avoir accès à la valeur de hauteur et l'attribut control qui permet d'activer l'évènement clique de souris. Ainsi on définit les constantes pour les dimension de notre panel et ovale et la taille du saut .



- On crée la classe Etat qui a comme l'attribut hauteur et les méthodes getHauteur pour retourner la hauteur et jump pour soustraire la taille du saut de la hauteur tout en vérifiant la borne du panel. Ainsi, on définit la classe Control qui hérite la classe MouseControler implémentant l'interface MouseListener pour faire monter l'ovale lorsqu'on clique dans l'interface. Ceci est réalisé à travers de la méthode mouseClicked, en appelant d'abord la méthode jump et ensuite repaint pour redessiner l'ovale. On optimise la méthode repaint en redessinant seulement l'espace de GUI qui change à chaque fois.



# Résultat



# Documentation utilisateur

- **Prérequis :** Java avec un IDE (ou Java tout seul si vous avez fait un export en `.jar` exécutable)
- **Mode d'emploi (cas IDE) :** Importez le projet dans votre IDE, sélectionnez la classe Main à la racine du projet puis « Run as Java Application ». Cliquez sur la fenêtre pour faire monter l'ovale.
- **Mode d'emploi (cas `.jar` exécutable) :** double-cliquez sur l'icône du fichier `.jar`. Cliquez sur la fenêtre pour faire monter l'ovale.

# Documentation développeur

Les fonctionnalités qui ne sont pas encore implémentées, c'est dans un premier temps la création d'une ligne brisée et puis la mise en œuvre de son déplacement automatique pour donner l'impression que l'ovale avance le long de cette ligne.

# Conclusion et perspectives

Jusqu'à présent, on a réussi à faire apparaître une fenêtre contenant un ovale et à faire monter cet ovale en cliquant dans l'interface. La difficulté rencontrée c'était l'utilisation correcte et efficace des nouvelles méthodes de la bibliothèque Swing, ce qui a été résolu en lisant les différentes documentations et ressources. Prochainement, on envisage de créer la ligne brisée avec son déplacement automatique ainsi que la descente automatique d'ovale.