

PROJET JAVA

---

Projet 2 : Le jeu du TETRIS

---

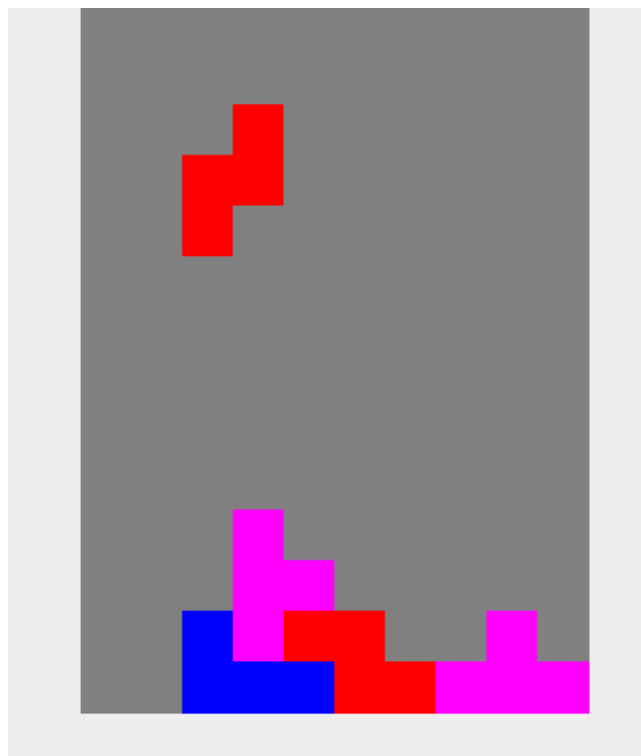


FIGURE 1 – Image issue de notre projet

Ariane DEPONTHIEUX  
Roxane LEDUC  
Anouk ANDRE  
GM3

*Encadré par : M. BOURGAIS*

Avril 2022 — Mai 2022

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation du jeu</b>	<b>3</b>
<b>3</b>	<b>Diagramme de cas d'utilisation</b>	<b>4</b>
<b>4</b>	<b>Diagramme de classe</b>	<b>5</b>
<b>5</b>	<b>Quelques précisions sur les différentes classes</b>	<b>6</b>
5.1	La classe Piece . . . . .	6
5.2	La classe Grille . . . . .	7
5.3	La classe JeuTetris . . . . .	8
5.4	Les classes liées à l'interface graphique . . . . .	8
<b>6</b>	<b>Diagramme Etats-transitions</b>	<b>9</b>
<b>7</b>	<b>Les difficultés rencontrées</b>	<b>10</b>
7.1	Conclusion . . . . .	11

# 1 Introduction

Dans le cadre de notre mini-projet de Java, nous avons décidé d'implémenter un jeu de Tetris. Au début, nous étions parties sur un Mastermind mais nous n'étions pas inspirées du tout et nous avons préféré opter pour un jeu que nous préférons et connaissons mieux. Nous nous sommes finalement décidées à choisir le Tetris, un peu plus compliqué en matière de code mais beaucoup plus intéressant.

## 2 Présentation du jeu

Une partie de Tetris se déroule de la façon suivante : des pièces aux formes multiples descendent le long de l'écran, le but du joueur est de les positionner de façon à former des lignes complètes. Une ligne complète disparaît et rapporte des points. La partie s'arrête lorsqu'une pièce sort du haut de l'écran.

Dans notre jeu, la vitesse augmente au cours du temps.

### 3 Diagramme de cas d'utilisation

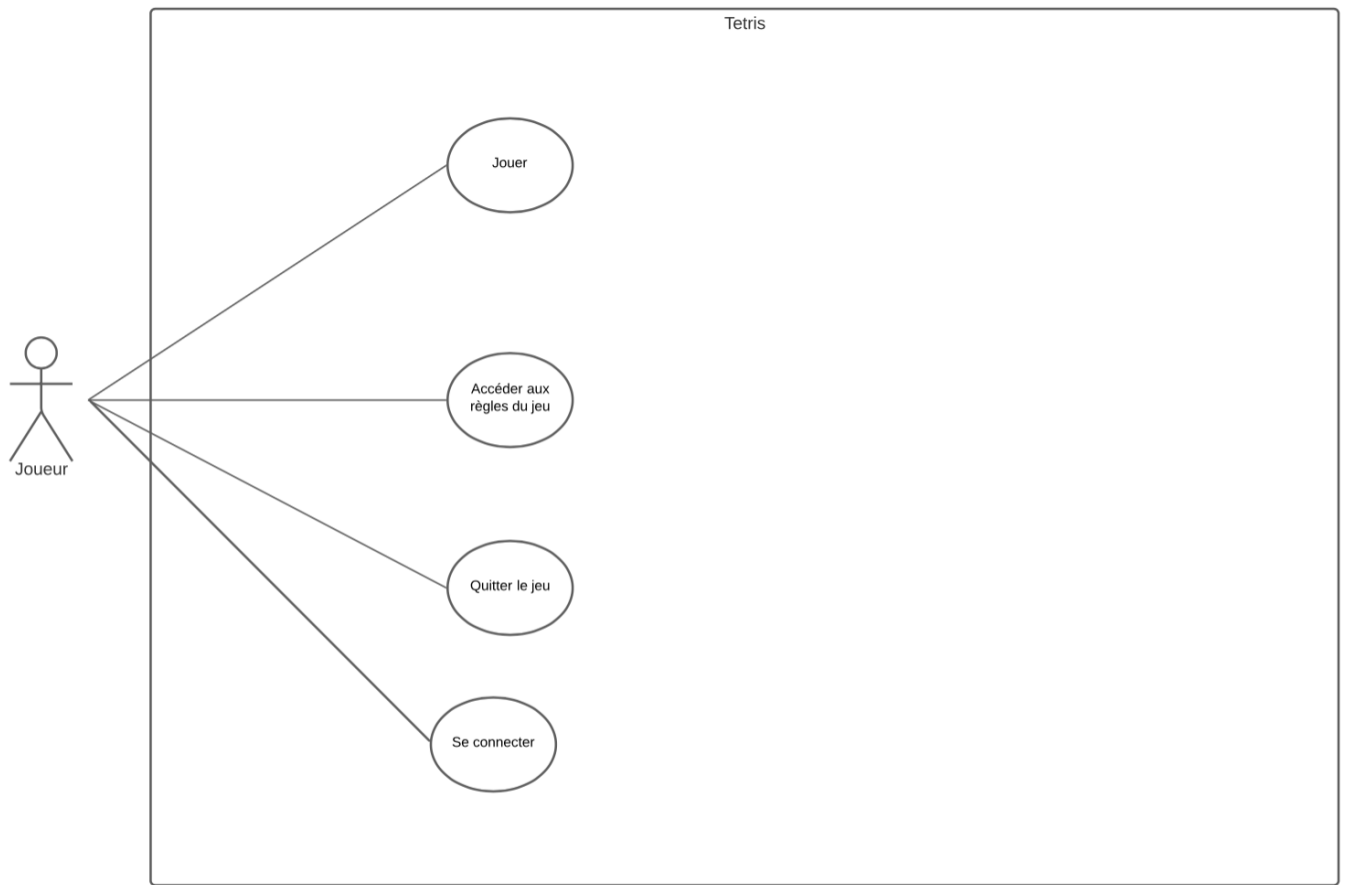


FIGURE 2 – Diagramme cas d'utilisations

## 4 Diagramme de classe

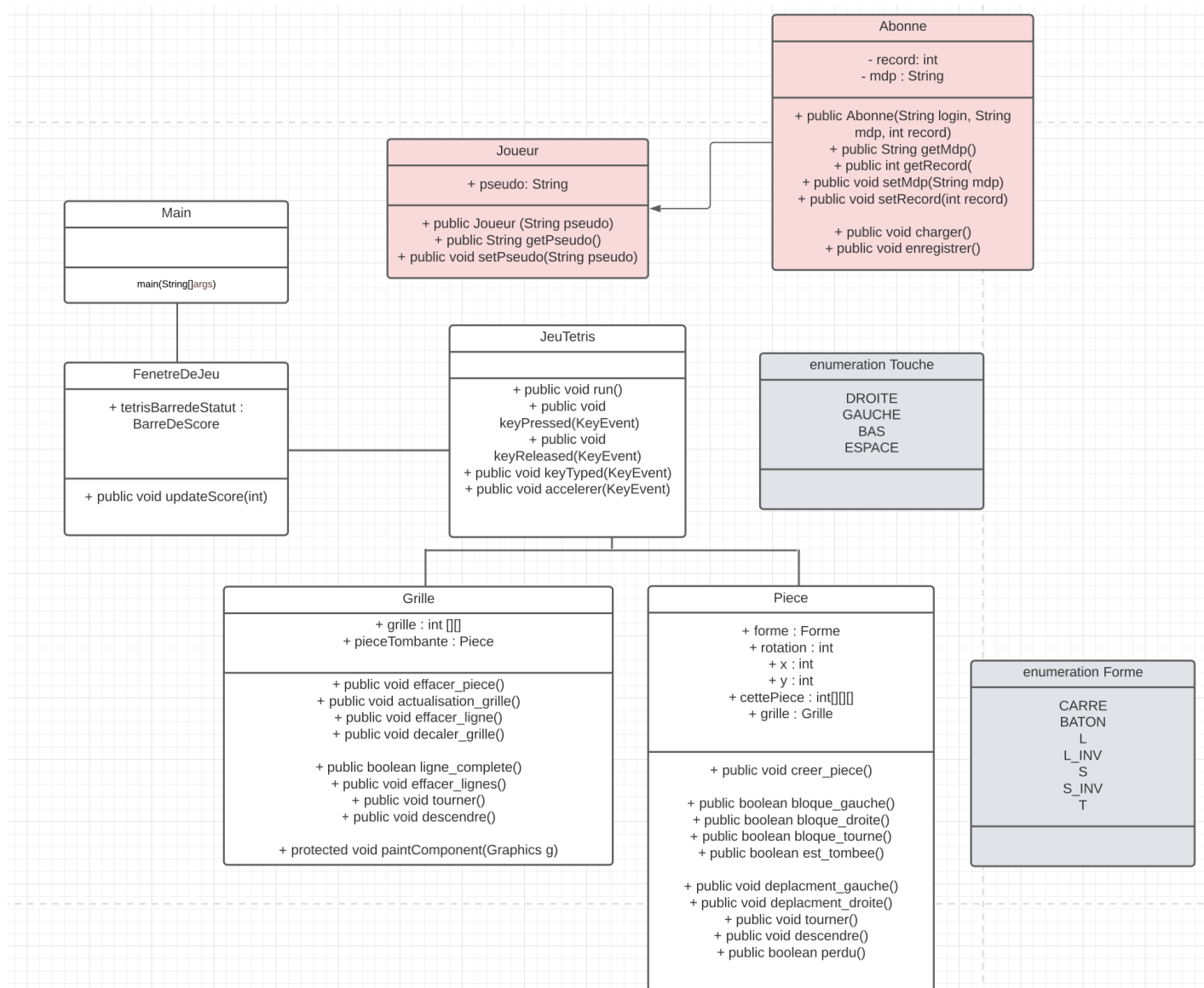


FIGURE 3 – Diagramme de classe

Nous avons décidé de ne pas faire apparaître les classes uniquement liées à l'interface graphique pour plus de lisibilité. Tous les éléments en rose sont des éléments qui figurent dans notre code mais que nous n'avons pas eu le temps d'intégrer au programme principal.

## 5 Quelques précisions sur les différentes classes

### 5.1 La classe Piece

Le jeu Tetris comporte sept pièces différentes. Les voici :

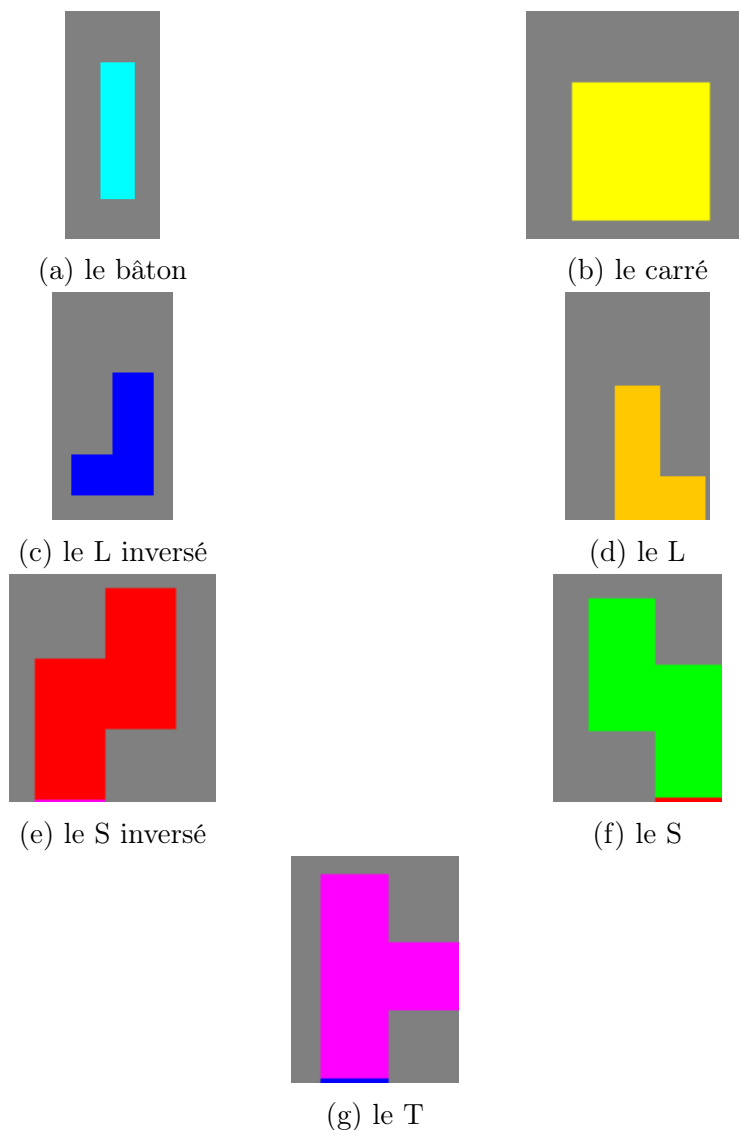


FIGURE 4 – Captures de nos 7 pièces

Comme on peut le voir, chaque forme de pièce a sa couleur associée. Tous les carrés sont jaunes par exemple. De plus, Chaque pièce peut être tournée lors de sa descente, et on a donc pour chaque pièce 4 rotations possibles.

Nous les avons manipuler à l'aide d'une sous-grille. Voici les 4 rotations de nos 7 pièces :

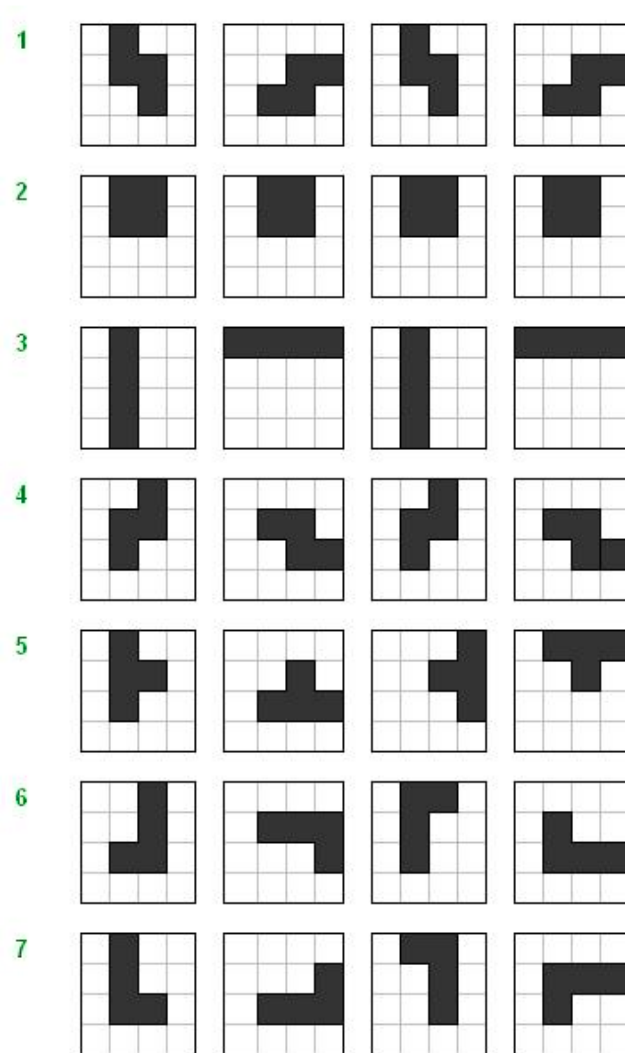


FIGURE 5 – Image des 4 positions des 7 pièces

## 5.2 La classe Grille

Notre classe Grille est principalement composée d'un tableau d'entiers. C'est ce tableau qui correspond à la grille du Tetris, chaque case du tableau est soit zéro (case vide) soit un entier entre 1 et 7 qui correspond à une couleur en particulier. Ce tableau comporte une partie visible que l'on affiche au joueur et une partie invisible qui nous sert à gérer certains points techniques (comme faire apparaître une pièce petit à petit...).



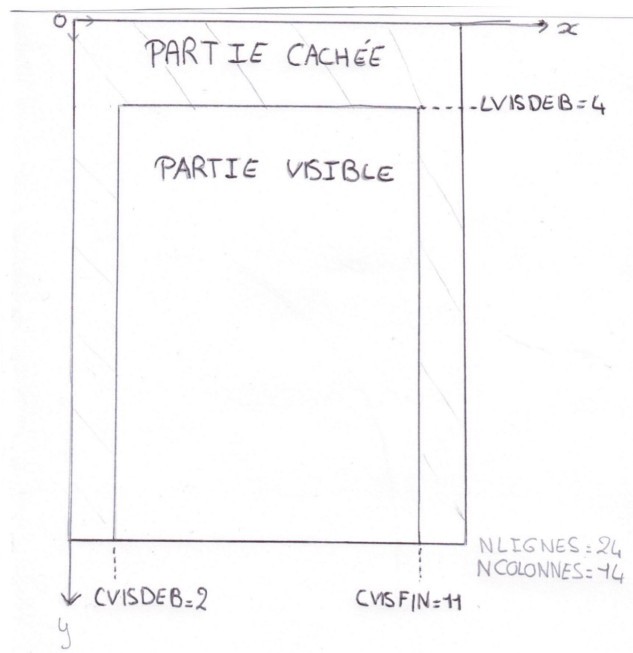


FIGURE 6 – Grille

Elle contient également les constantes qui définissent la taille de la grille ainsi que l'attribut `pieceTombante` qui permet de gérer la pièce qui est entrain de tomber.

### 5.3 La classe `JeuTetris`

La classe `JeuTetris` est un `Thread` qui actualise en continu la grille en fonction de si la pièce est tombée, si une ligne est complète ou de si le joueur la déplace. Nous avons créé un attribut `vitesseDeChute` afin d'augmenter la difficulté au cours du temps. Dès que le score augmente de 100, la vitesse de chute accélère (l'augmentation est de moins en moins rapide sinon la partie devient trop rapidement injouable).

### 5.4 Les classes liées à l'interface graphique

On compte 5 classes liées à l'interface graphique. Depuis la classe `FenetreMenu`, on peut avoir accès aux 3 autres classes : `FenetreRegles`, `FenetreConnexion`, et `FenetreDeJeu`. La classe `FenetreScore` est appelée depuis le `JeuTetris`.

## 6 Diagramme Etats-transitions

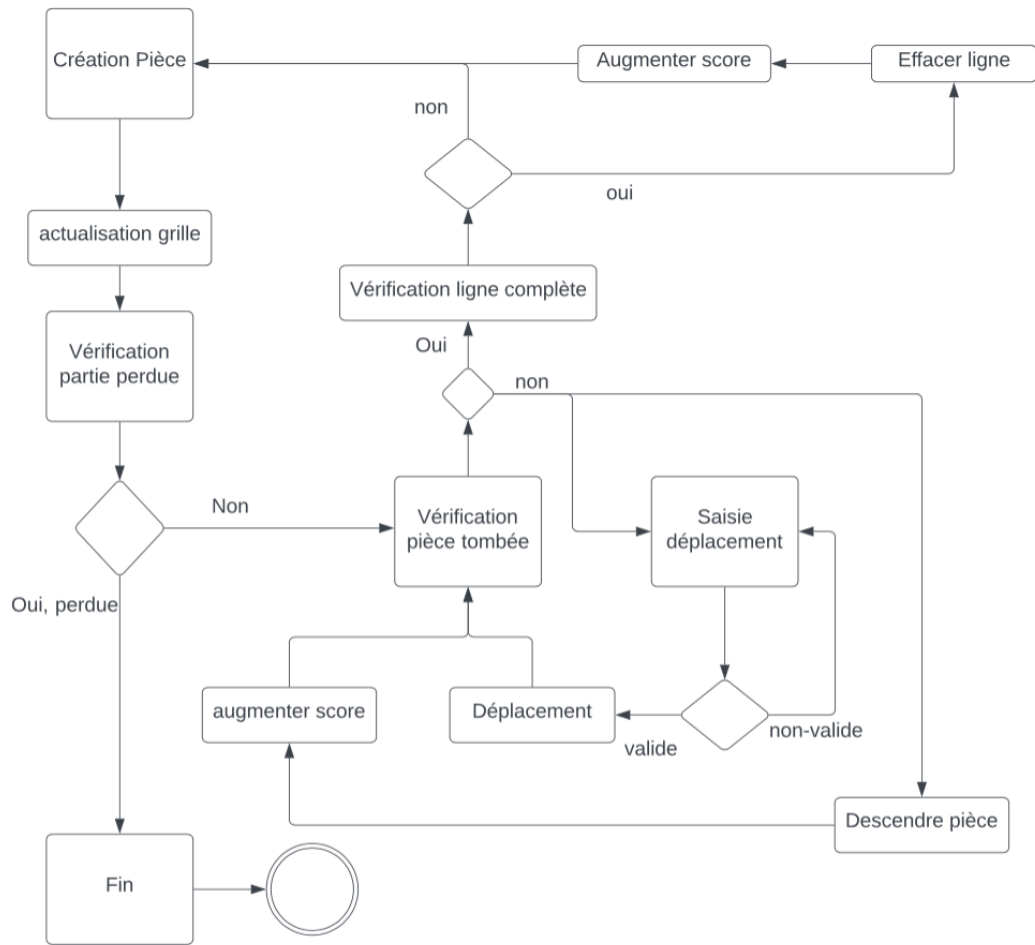


FIGURE 7 – Diagramme d'états-transitions

## 7 Les difficultés rencontrées

Avant de commencer ce projet, nous appréhendions un peu l'utilisation d'un nouveau langage objet. Cela n'a finalement pas été un problème.

Nous avons principalement rencontré quelques soucis au niveau de l'affichage graphique, effectivement nous avons utilisé les classes déjà écrites (JFrame, JPanel ...) et certains raccourcis sont un peu difficile à comprendre et à utiliser. Nous avons quand même réussi à faire une interface graphique correcte mais l'appel de toutes nos fenêtres n'est pas très joli et nous n'avons pas réussi à tout manipuler depuis le main.

Ensuite, nous avions prévu un certain nombre de fonctionnalités à implémenter. Notamment, nous souhaitions faire une distinction entre un joueur normal et un abonné qui aurait eu la possibilité d'enregistrer son meilleur score. Malheureusement, avec les examens nous avons un peu mis le projet de côté et n'avons ainsi pas eu le temps de faire tout ce que nous souhaitions.

## 7.1 Conclusion

Ce projet nous a permis de faire nos premiers pas dans le monde de la programmation orientée objet. Nous avons pu mettre en application ce que nous avons appris en cours et ainsi de vraiment comprendre la manière de raisonner de ce type de programmation. Nous avons également pu apprendre par nous mêmes à nous servir des bibliothèques swing et awt.

Nous n'en sommes encore qu'au début de notre apprentissage de ce langage, mais ce projet nous a permis d'être un peu plus à l'aise et il est toujours très satisfaisant d'arriver à faire marcher un jeu et d'y jouer par la suite. Le fait d'utiliser eclipse a aussi été un vrai plus, car cet éditeur présente de nombreux avantages et nous permet d'être plus rapide (auto-complétion).

Pour ce qui est de la gestion du travail au sein du groupe, nous nous sommes bien réparti les tâches et, étant en relativement bonne entente nous avons réussi à bien communiquer et à bien nous expliquer ce que nous faisions. Nous avons donc toutes bien compris l'ensemble du code.