

**Projet JavaFX**  
Informatique par Apprentissage  
Année scolaire 2016 – 2017  
3ème Année - Semestre 2



Fiona REZIGA  
Alexandre LEFEBVRE

Frédéric Armetta

Polytech Lyon  
Bâtiment Istil,  
15 Boulevard André Latarget,  
69100 Villeurbanne  
04 72 43 12 24

## Sommaire

Introduction .....	1
Problématique.....	1
Client.....	1
Equipe de développement.....	1
Déroulement du projet .....	2
Choix du sujet.....	2
Outils, langages .....	2
Conception .....	2
Etapas .....	2
Conception UML et développement .....	3
Descriptif de l'application .....	4
Fonctionnalités.....	4
Tetris .....	4
Morpion.....	4
Puissance 4.....	4
Conclusion .....	4
Annexes .....	5
Annexe 1 – Diagramme d'activité du jeu Tetris.....	5
Annexe 2 – Diagramme UML.....	5
Annexe 3 – Architecture du projet.....	6
Annexe 4 – Jeu Tetris.....	7
Annexe 5 – Jeu Morpion .....	8
Annexe 6 – Jeu Puissance 4.....	9



# Introduction

## Problématique

Au cours de ce projet, on souhaite développer une petite bibliothèque de 3 jeux. Ces trois jeux exploitent un plateau pour le placement de pièces. On propose ainsi de réaliser un module de gestion de plateaux, qui sera exploité pour les différentes applications.

## Client

Pour ce projet, c'est le professeur du module qui joue le rôle du client.

## Equipe de développement

L'équipe de développement est composée de 2 étudiants de 3<sup>ème</sup> année de la filière Informatique par Apprentissage de l'école Polytech Lyon (Université Lyon 1).

L'équipe du projet est composée de :

- Fiona Rezig
- Alexandre

Lefebvre

# Déroulement du projet

## Choix du sujet

Le sujet du projet étant de réaliser une bibliothèque de petits jeux, nous avons décidé de développer une application regroupant une simulation des 3 jeux suivants :

- Tetris
- Morpion
- Puissance 4

## Outils, langages

L'application développée doit être de type graphique en langage JAVA, le développement se base donc sur la bibliothèque JavaFX et respecte le modèle MVC (Modèle Vue Contrôleur).

La gestion du code source est réalisée grâce à l'outil GIT. Un dépôt GitHub a été créé au lancement du projet. La gestion du dépôt a ensuite été effectuée grâce au programme GitKraken qui nous a permis d'effectuer facilement des commit et de garder un visuel graphique sur la branche principale de notre projet.

## Conception

### Etapes

L'analyse, le développement et la conception de ce projet ont suivi les étapes suggérées dans le sujet fourni par le professeur.

En amont du développement du projet, nous avons étudié le fonctionnement du projet « calculatrice » et le diagramme d'activité du jeu Tetris (Annexe 1) fournis par le professeur.

Suite à cette phase d'analyse, la conception de ce projet a commencé par le développement de la librairie servant à initialiser un plateau de jeu permettant d'utiliser ce dernier dans n'importe quelle situation de jeu.

Ensuite, nous avons développé le jeu Tetris qui nous a paru comme le plus difficile et le plus complexe à mettre en place.

Une fois que la librairie et le jeu Tetris ont été fonctionnels, nous avons développé les deux autres jeux : Le Morpion et le Puissance 4.

Ainsi, les étapes de conception ont été les suivantes :

- Analyse du projet Calculatrice fourni. (environ 1h)
- Analyse du diagramme d'activité du jeu Tetris. (environ 1h)
- Analyse sur papier avec modélisation Objet du problème (classes, traitements, etc...). (environ 4h)
- Développement de la librairie Modèle – Vue du plateau de jeu de manière à le rendre utilisable par plusieurs applications (ici, les différents Jeux qui seront développés). (environ 6h)
- Développement du jeu Tetris. (environ 4h)
- Développement des deux autres jeux, Morpion et Puissance 4. (environ 4h)

Les différentes phases de construction du projet ont été réalisées durant les heures de cours en période de formation mais aussi en travail personnel.

## Conception UML et développement

Nous avons fait le choix de développer une classe pour chaque type de forme. Ainsi pour chaque forme susceptible d'être utilisée par les différents jeux, une classe est développée et va permettre la définition et l'initialisation dès la forme dans le programme du jeu. Cette méthode a permis de rendre le code plus clair et plus compréhensible (nous identifions directement de quelle forme il s'agit) et de faciliter le développement.

La librairie du plateau de jeu est donc composée d'un package model et d'un package View. Le package model regroupe un dossier comprenant toutes les classes des pièces et les classes Board (classe du plateau de jeu) et Grid (classe de la grille du jeu). Le package View est quant à lui composé de la classe GridPane.

Ensuite, chaque jeu développé est inséré dans un package propre au jeu. Chaque package de jeu est composé d'un package Controller, qui va définir la classe Controller du jeu et d'un package model composé lui des classe Board et Grid du jeu. Ces classes vont hériter des classes Board et Grid de la librairie. Enfin, une classe View propre au jeu est défini en héritant de la classe View de la librairie. Toutes les classes model des jeux héritent du model de la librairie et rajoutent les méthodes métiers propre à chaque jeu.

Le jeu Tetris est lui un peu particulier, puisqu'il nécessite un Thread permettant au jeu de se dérouler sans interaction de l'utilisateur. L'ajout du Thread va permettre au jeu d'être autonome et automatique. Ce jeu a aussi une classe en plus, la classe Pièce qui va permettre de définir une liste pièce à partir de la librairie des pièces de manière à en choisir et en générer une aléatoirement.

Une copie d'écran de l'architecture du projet est disponible en annexe (Annexe 3). Le diagramme UML est lui aussi disponible (Annexe 2).

## Descriptif de l'application

Il est donc possible de lancer l'exécution de chaque jeu depuis l'IDE.

L'application regroupant les 3 jeux développés reprend donc les règles bien connues des jeux Tetris, Morpion et Puissance4.

Des captures d'écran de l'exécution sont disponibles en annexe (Annexe 4,5 et 6).

## Fonctionnalités

### Tetris

- Jouer une partie du jeu Tetris
- La vitesse du jeu est proportionnelle au score du joueur
- Indication du score du joueur
- Indication de la prochaine pièce
- Possibilité de lancer une nouvelle partie grâce à un bouton
- Possibilité de lancer une nouvelle partie après la fin du jeu

### Morpion

- Jouer une partie du jeu Morpion à 2 joueurs
- Possibilité de lancer une nouvelle partie après la fin du jeu

### Puissance 4

- Jouer une partie du jeu Puissance 4 à 2 joueurs
- Possibilité de lancer une nouvelle partie après la fin du jeu

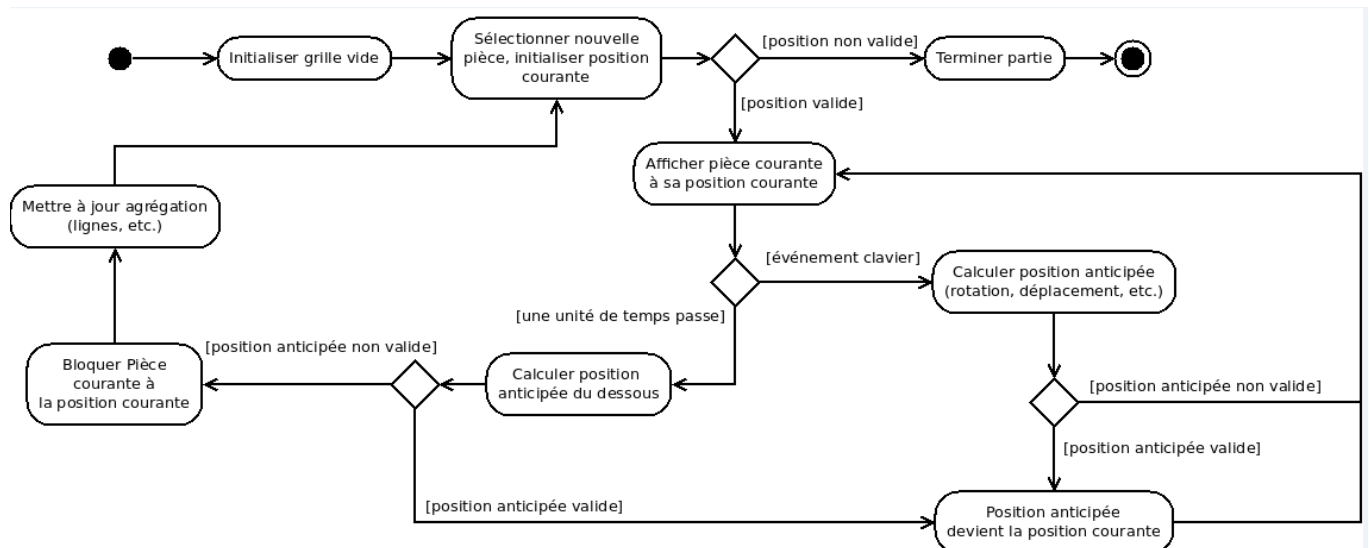
## Conclusion

Ce projet nous a permis de prendre conscience de l'utilité et de la réutilisabilité d'une librairie. Il a aussi été important d'analyser le sujet et de réaliser un travail sur papier avant de se lancer dans le développement.

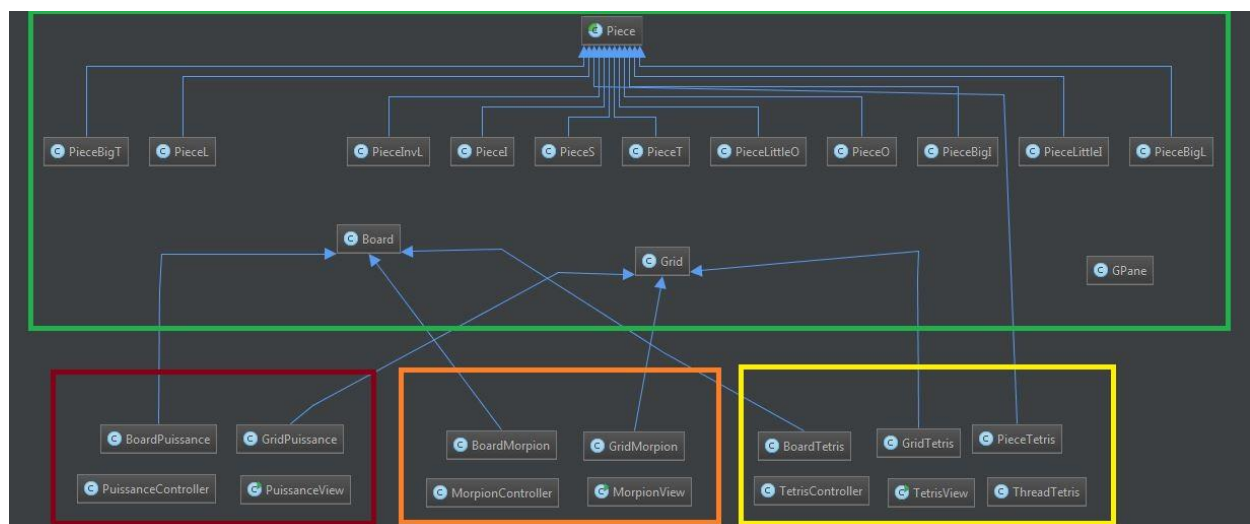
Ce projet nous a aussi permis de prendre connaissance avec la librairie JavaFX.

# Annexes

## Annexe 1 – Diagramme d'activité du jeu Tetris

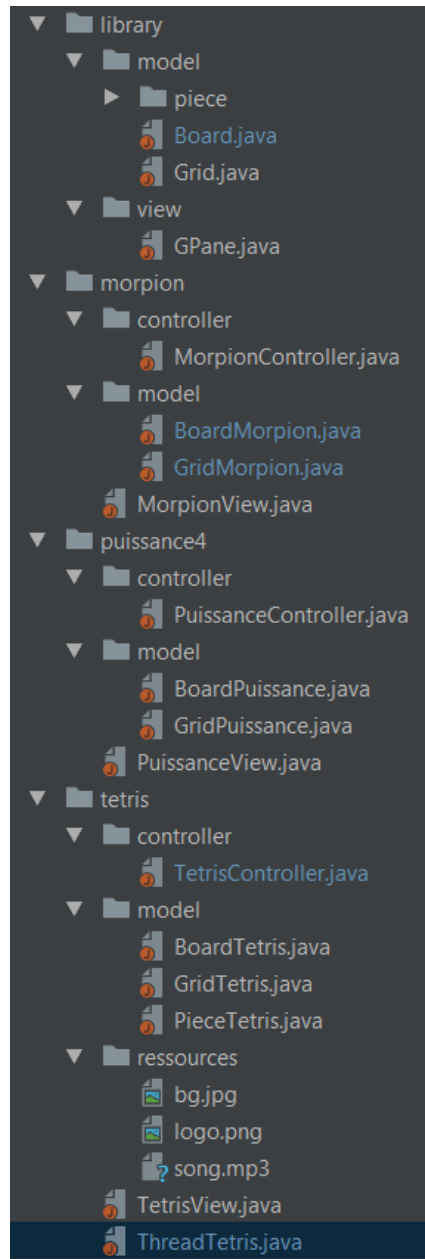


## Annexe 2 – Diagramme UML



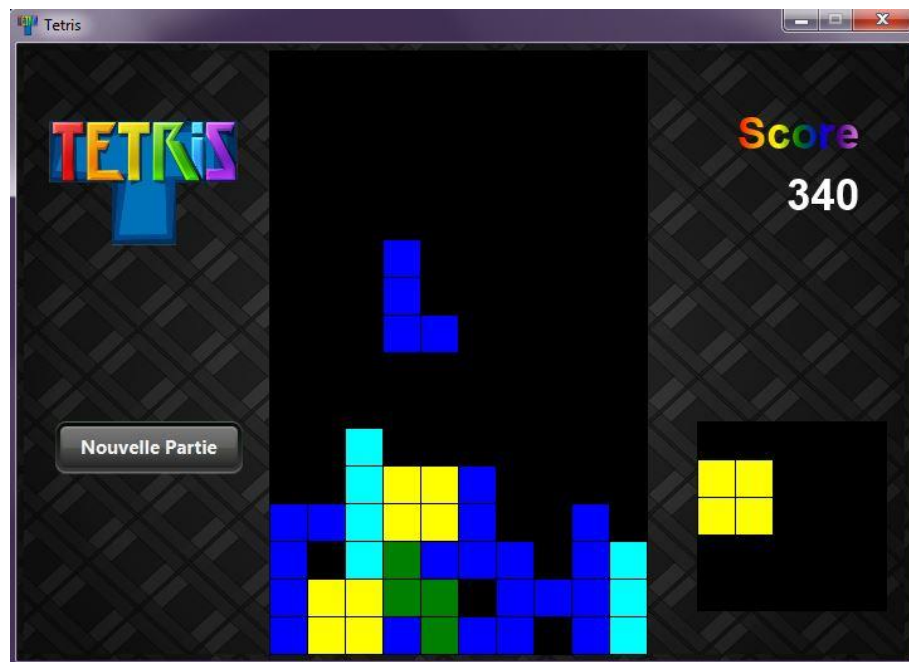
Librairie – Jeu Puissance 4 – Jeu Morpion – **Jeu Tetris**

## Annexe 3 – Architecture du projet

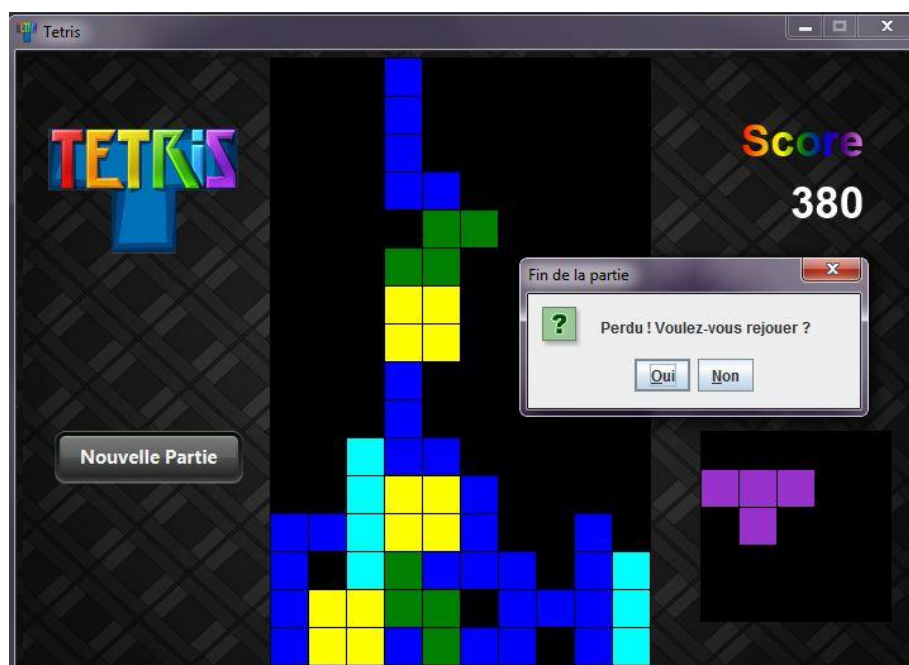




## Annexe 4 – Jeu Tetris

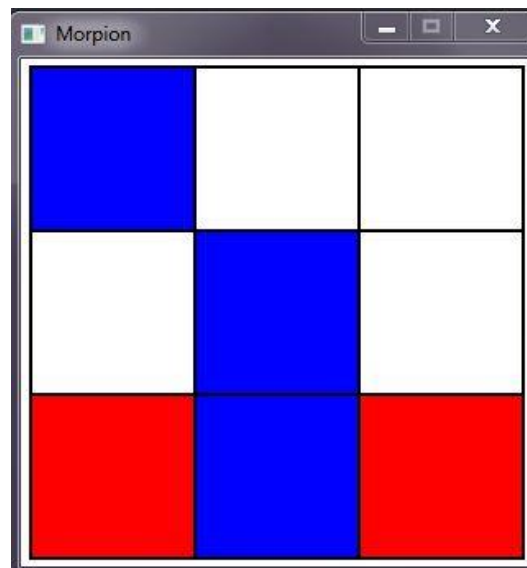


Déroulement d'une partie

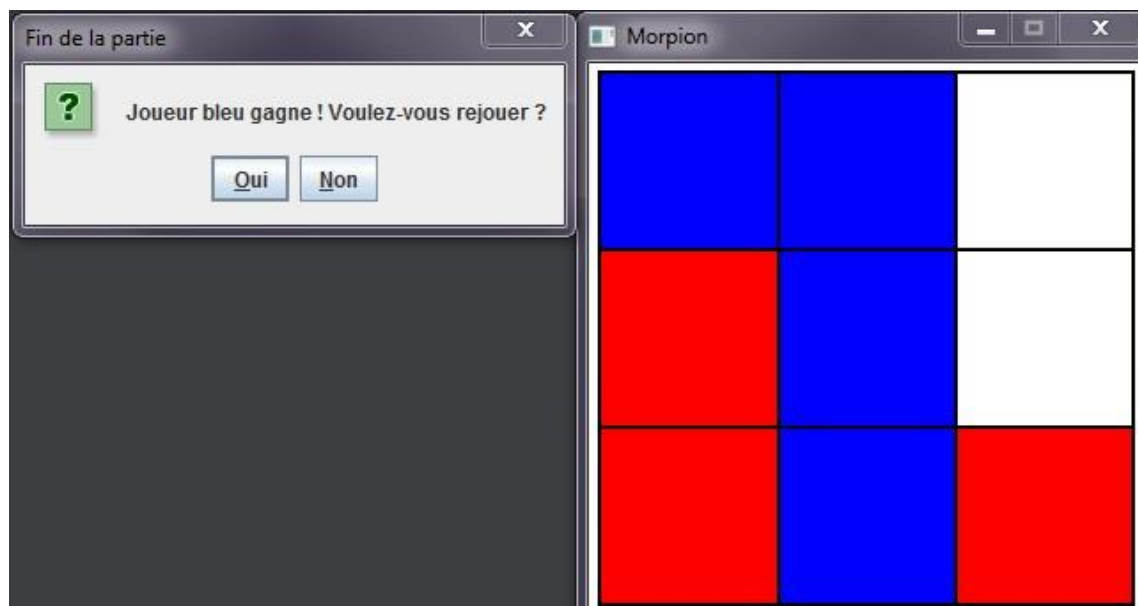


Fin d'une partie

## Annexe 5 – Jeu Morpion

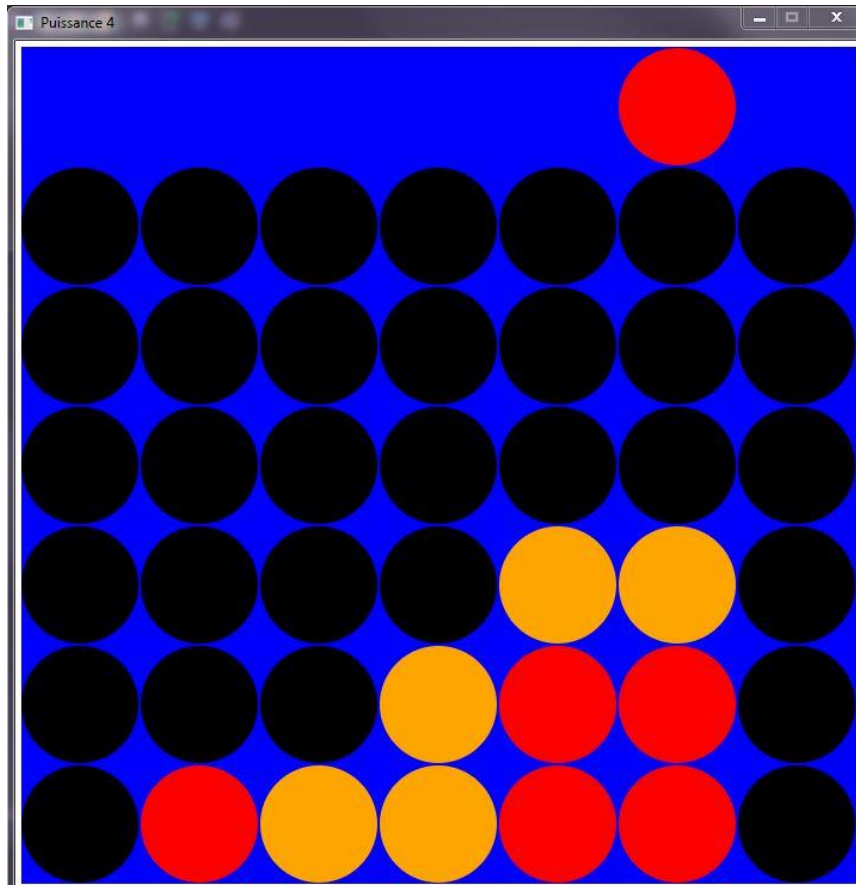


Déroulement d'une partie



Fin d'une partie

## Annexe 6 – Jeu Puissance 4



Déroulement d'une partie