

Mathieu  
Guillaume

## Projet – POO en java

L'objectif de ce projet est de réaliser un jeu d'echec dans le modèle MVC. Ce rapport aura pour but de présenter les différentes fonctionnalités de ce jeu ainsi que les extensions proposées.

Tout d'abord, je vais présenter dans quel ordre et en quelles proportions de temps mon projet a évolué.

La première étape a été de créer un modèle vue controller correct :

- La vue gérant à elle seule les affichages en fonction de ce qu'elle trouvait dans le model;
- Le model, avec les données métiers du problème;
- Le controller, qui prend en paramètre la vue et le model pour faire le lien entre les deux.
- 

Je n'avais jamais utilisé ce design pattern donc il m'a fallu toute la première séance de TP soit 4h pour arriver à un modèle MVC satisfaisant.

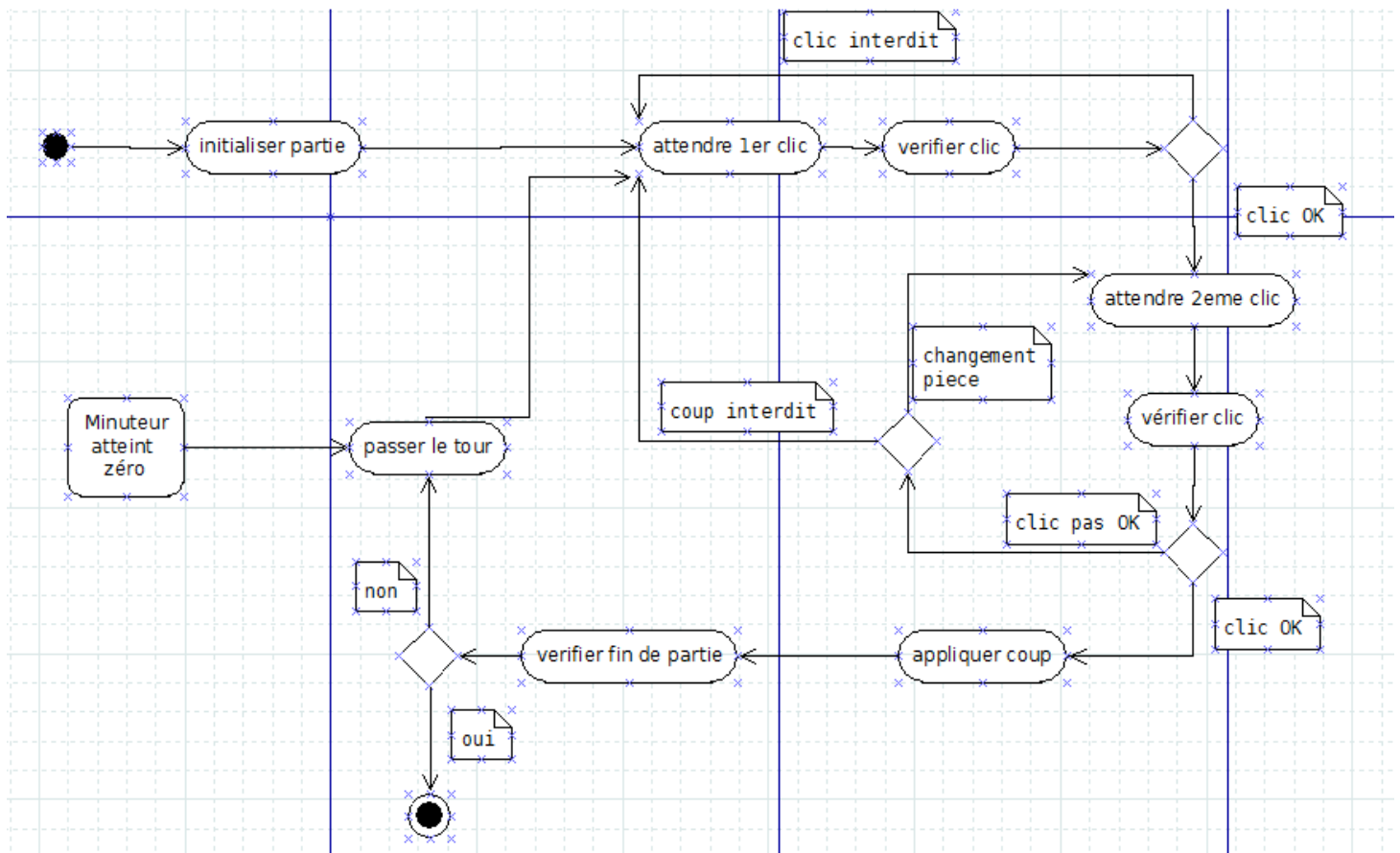
Ensuite, je me suis penché sur l'aspect logique du jeu, les déplacements possibles de chaque pièce et les conditions de fin de tour/fin de partie, ce qui m'a pris environ six heures. J'expliquerai cette partie dans un diagramme d'activité.

Enfin, je me suis occupé de la vue. J'ai créé notamment une énumération qui déterminait si on devait afficher l'echiquier ou bien le menu d'accueil par exemple. Puis j'ai également fait une fonction dans la vue permettant d'aller charger en mémoire la bonne image pour chaque pièce en fonction du type de la pièce et du joueur (pour la couleur) :

**public boolean setIcône(String type, String couleur)**

Ce travail m'a pris deux heures.

## Diagramme d'activité représentant un tour de jeu



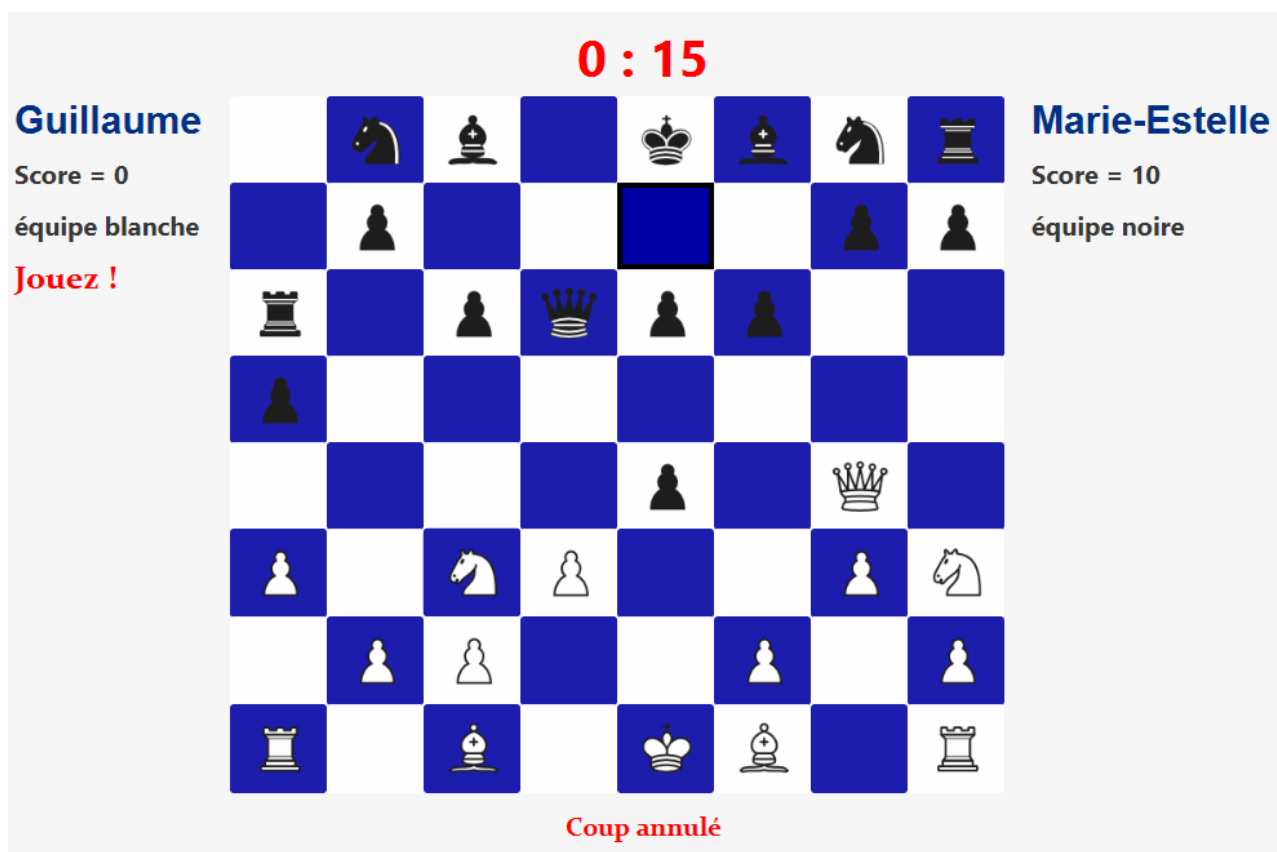
Ce diagramme représente le déroulement d'une partie.

Un tour de jeu est composé d'un coup, qui a une position initiale et une position finale, représenté par le premier et le second clic.

Le coup est mis à jour après vérification de chaque clic.

Lorsque le coup est appliqué, on vérifie si la partie est terminée, s'il n'y a pas échec, sinon on passe le tour au joueur adverse.

## Aperçu d'une partie en cours



On remarque plusieurs choses sur cette capture d'écran :

Le nom des joueurs ainsi que leur score

Un label “Jouez!” se met du côté du joueur dont c'est le tour.

Un label sous l'échiquier permet d'indiquer plusieurs informations :

- Coup annulé lorsqu'on reclique sur la même pièce
- Coup réussi lorsque le coup est vérifié et exécuté

- Temps écoulé
- Déplacement impossible
- Changement de pièce lorsqu'on clique sur une pièce de notre équipe après avoir déjà sélectionné une de nos pièces.
- Cette pièce n'est pas à vous !

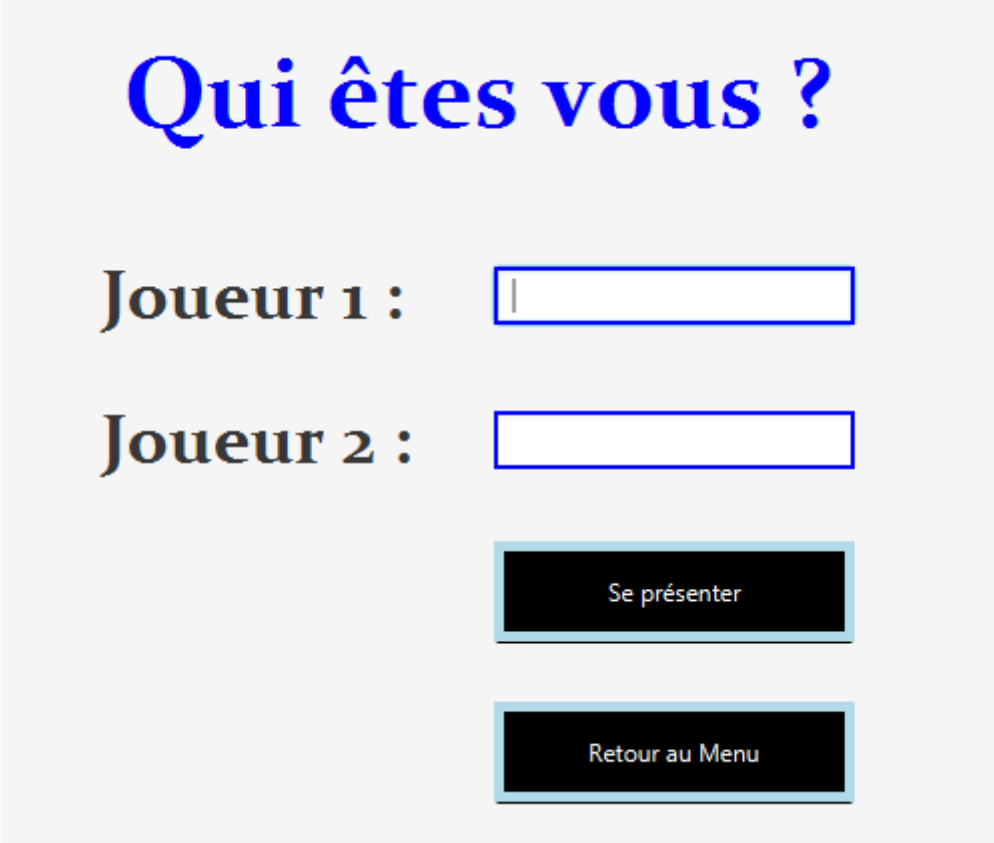
Le minuteur, qui passe en rouge sous une certaine valeur et fait passer le tour lorsqu'il arrive à zéro.

D'ailleurs, le minuteur fait partie d'une extension dont j'aimerais parler.

J'ai la classe Minuteur dans le model qui hérite de Thread et qui, chaque seconde, utilise la fonction update pour notifier son Observer qui est MinuteurView, hérité de Label.

Le thread minuteur est démarré à chaque début de partie par le model.

## Aperçu lorsqu'on selectionne 'partie à deux joueurs'



The screenshot shows a light gray rectangular area with a blue title "Qui êtes vous ?" at the top. Below the title, there are two labels "Joueur 1 :" and "Joueur 2 :". Each label is followed by a white rectangular input field with a blue border. Below the input fields, there are two black rectangular buttons with white text. The first button is labeled "Se présenter" and the second button is labeled "Retour au Menu".

**Qui êtes vous ?**

**Joueur 1 :**

**Joueur 2 :**

**Se présenter**

**Retour au Menu**

## **Améliorations suite à la présentation**

J'ai ajouté la vérification d'echec :

On ne peut pas se mettre en echec tout seul ni le rester lorsque l'adversaire nous a mis en echec.

Cela m'a pris trois heures.

J'ai fait attention à l'encapsulation (j'avais parfois omis des private)

En conclusion, ce travail m'a permis de créer ma première véritable application et d'améliorer ma méthodologie en ce qui concerne la modélisation ainsi que la pratique avec javaFX.

La maîtrise du MVC me permet désormais de mieux organiser mon code, et cela m'a donné les clefs pour créer mon projet de l'UE LIFPROJET, qui est une discussion instantanée en javaFX.