

# Rapport de projet

## Methode de Conception

### Jeu de stratégie au tour par tour

Fatoumata M'balou Diallo 22013282  
Mamoudou Camara 22012579  
Arafat Feical Idriss 21914879  
SOW Ismael 22014919

December 2021

## Table des matières

<b>1</b>	<b>Objectif du projet</b>	<b>2</b>
1.1	Detail du jeu . . . . .	2
<b>2</b>	<b>Diagrammes de classes-utilisation des differents Methode de conception</b>	<b>2</b>
2.1	Diagramme des packages . . . . .	2
2.2	Diagramme de classes pattern proxy . . . . .	3
2.3	Diagrammes de classes pattern stratégies . . . . .	3
2.4	Diagramme de classes pattern adapter . . . . .	4
2.5	Diagramme de classes pattern factory . . . . .	4
<b>3</b>	<b>les Methode de conception utilisées</b>	<b>5</b>
3.1	Description Methode de conception . . . . .	5
<b>4</b>	<b>Vue sur l'application</b>	<b>5</b>
4.1	Exemple d'execution pour deux combattants . . . . .	5
<b>5</b>	<b>Conclusion</b>	<b>8</b>

# 1 Objectif du projet

## 1.1 Detail du jeu

Dans le cadre de la licence trois informatique, nous avons un devoir en Methode de Conception. Il s'agit de réaliser une application de jeu de combat opposant entre 2 et n joueurs sur une grille en 2 dimensions, de taille paramétrable. Un combattant dispose initialement d'une certaine énergie et d'armes de différents types, avec des munitions en nombre limité.

chaque joueur dispose des points de vie et de points d'action, ils perdent lorsqu'ils n'ont plus de points de vie, et utilisent les points d'action pendant leur tour pour effectuer leurs actions. Les points d'action sont restitués au début du tour d'un joueur.

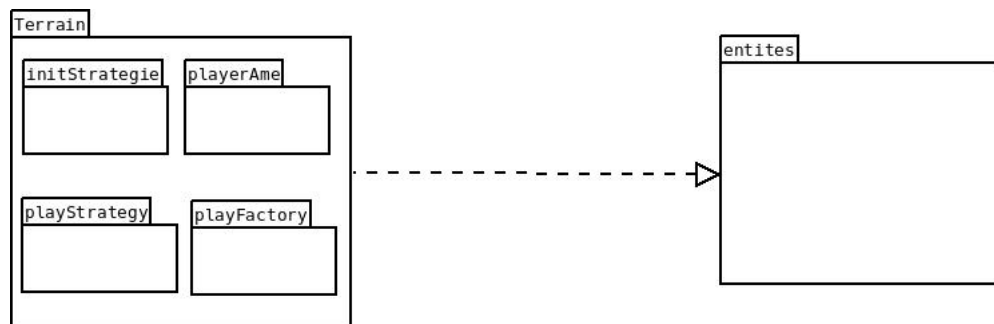
Pendant un tour, ils peuvent agir autant qu'ils le veulent, tant qu'ils ont des points d'action. Le tour d'un joueur fini quand il n'a plus de points d'actions ou quand il passe sans utiliser ces points.

Nous allons via ce rapport vous présenter la conception de ce projet vous décrire la façon dont le projet a été organisé. Détailler l'architecture du Projet : comment les différentes classes du jeu ont été reparties et comment nous les avons ordonnées dans différents packages. La conclusion résumera les objectifs qui ont été remplis et les éventuelles améliorations possibles qui pourraient être apportés au jeu

## 2 Diagrammes de classes-utilisation des differentes Methode de conception

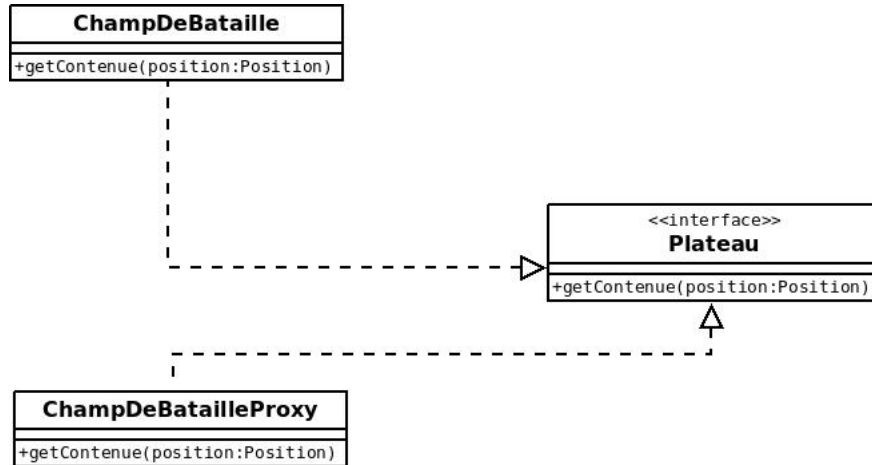
### 2.1 Diagramme des packages

Ce diagramme représente le diagramme de package qui englobe l'organisation de packages et des éléments.



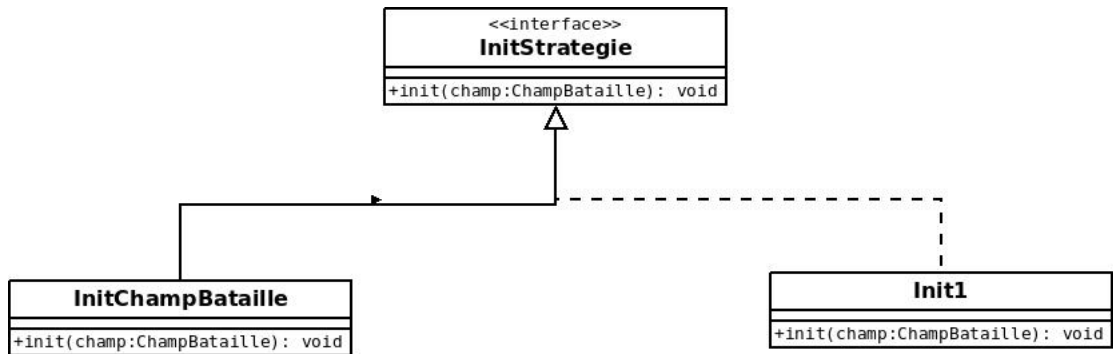
## 2.2 Diagramme de classes pattern proxy

Ce diagramme représente le diagramme de classes de la mise en pratique du pattern proxy. Il y'a une seule grille qui stocke toutes les informations. Chaque joueur n'a qu'une vision partielle de la grille de jeu, ne voyant que les explosifs qu'il a lui-même posés. Le pattern proxy permet de réaliser ces deux impératifs en même temps.

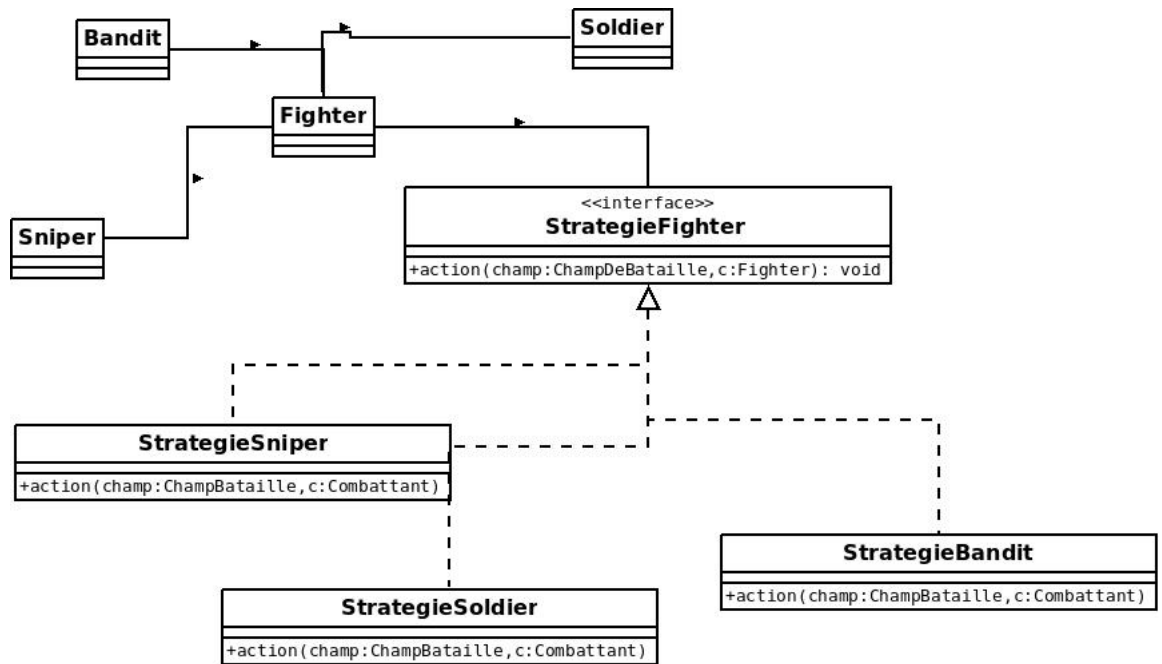


## 2.3 Diagrammes de classes pattern stratégies

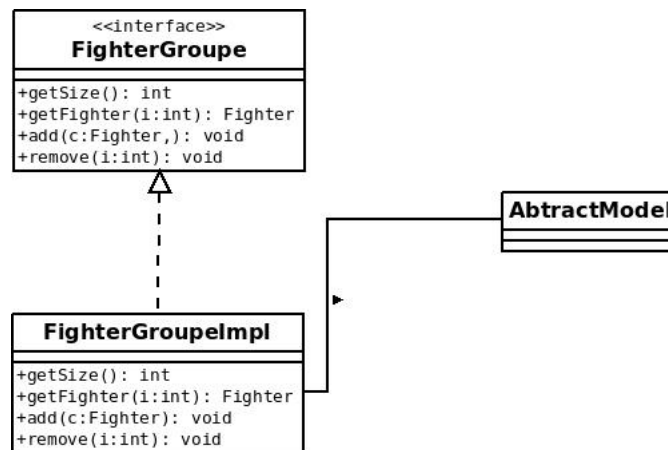
Ce diagramme représente le diagramme de classes de l'utilisation du pattern stratégie pour l'initialisation de la grille.



Ce pattern modifie radicalement le résultat d'une méthode sans modifier le coeur du code. Si on veut changer la façon dont on remplit la grille, il suffit d'ajouter une nouvelle implémentation.

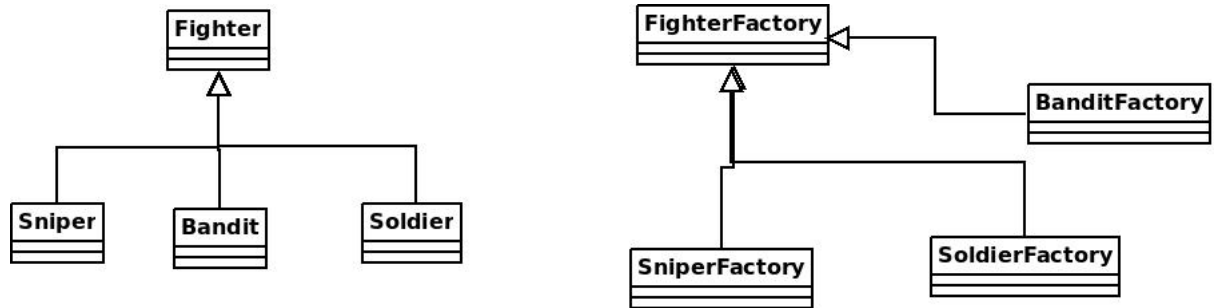


## 2.4 Diagramme de classes pattern adapter



## 2.5 Diagramme de classes pattern factory

Ce diagramme représente le diagramme de classes de la mise en ÷uvre du pattern factory.



### 3 les Methode de conception utilisées

#### 3.1 Description Methode de conception

les Methode de conception sont des solutions de conception pour un certain nombre de problème en programmation orientée objet. Dans ce projet nous avons utilisés quesles methode comme :

Pattern adapter : Il nous a permis degérer l’affichage d’un groupe de combattant dans un Jtable.

Pattern stratégie : il est utilisé pour l’initialisation de notre grille de jeu et pour définir les stratégies de déplacement et de combat pour chaque combattant.

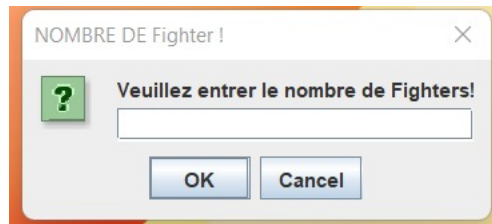
Pattern proxy : Il donne à chaque combattant une vue partielle du jeu.Seul les explosifs marqués visibles par les combattants qui les ont déposés sont visibles par les autres combattants. Les explosifs sont visible que par ceux qui les ont deposés.

Pattern factory : Il instancie les objets dont les types sont dérivés d’un type abstrait,nous l’avons utiliser pour la création des combattants à une autre classe.

### 4 Vue sur l’application

#### 4.1 Exemple d’execution pour deux combattants

Le jeu commence avec une fenêtre de permet d’indiquer le nombre de combattant. Après ce choix , une autre s’affiche pour choisir le mode d’execution souhaité .Une fois le mode déterminé le jeu commence automatiquement et à chaque tour de jeu, les joueurs jouent l’un après l’autre selon un ordre défini. Un combattant est éliminé du jeu lorsque son énergie est négative ou nulle. Il disparaît alors du jeu et le gagnant est le dernier survivant.

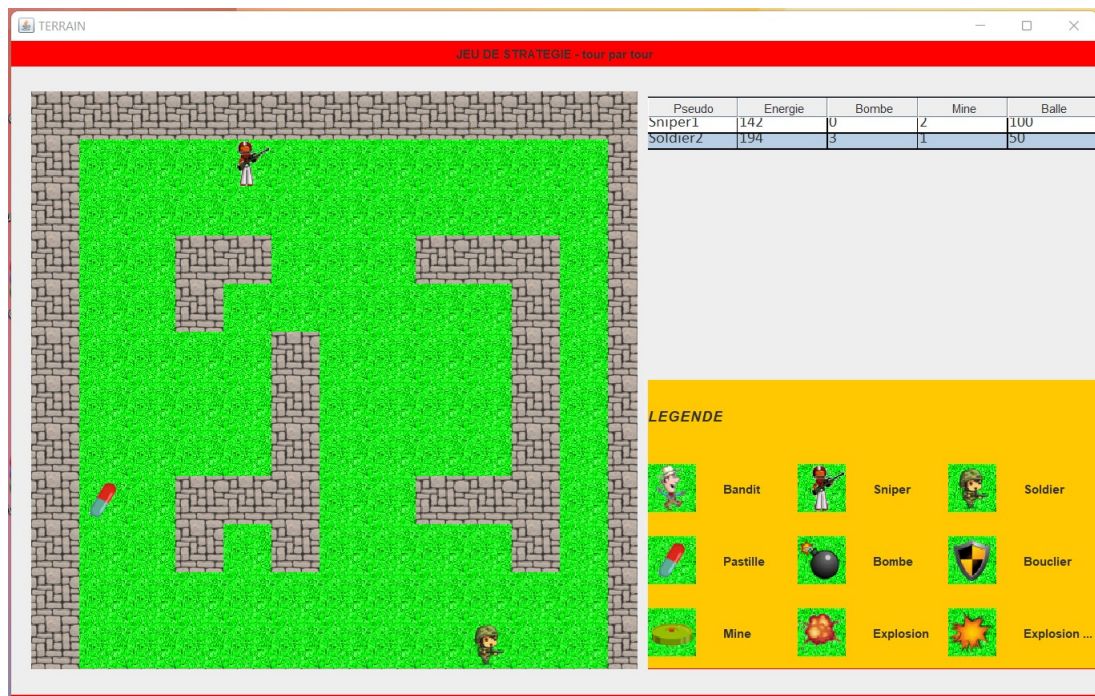


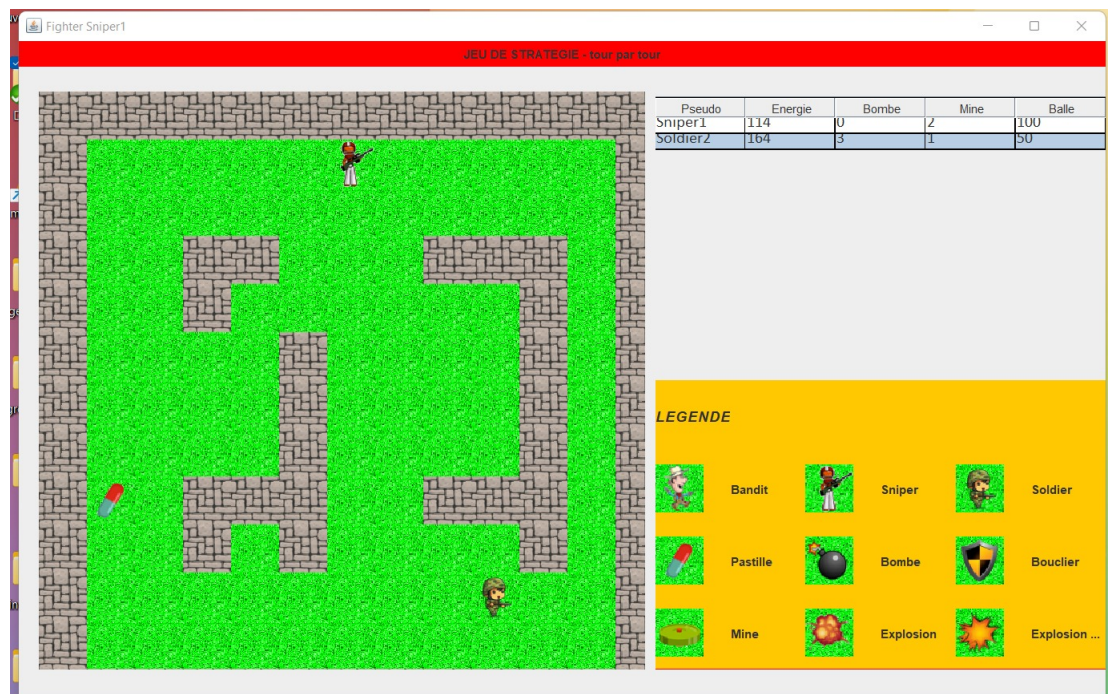
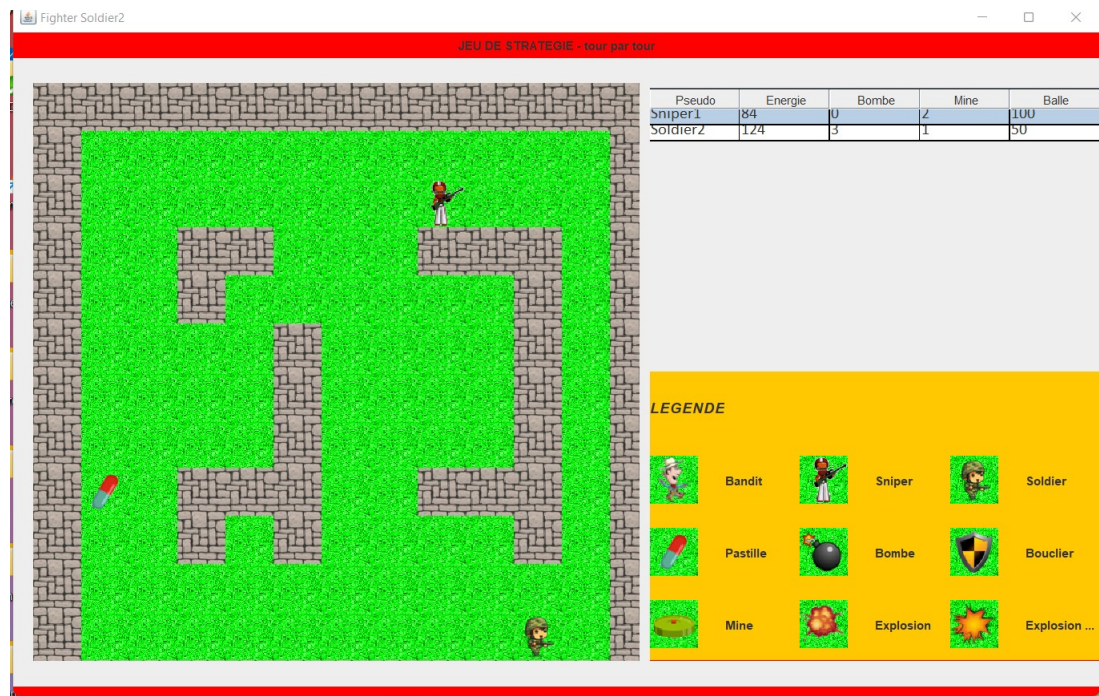
choix du mode d'exécution Ce schéma montre le choix du mode d'exécution que l'on souhaite utilisé



Déroulement du jeu

Ce schéma montre la phase de Déroulement du jeu





Fin du jeu

Ce schéma montre la phase terminal et le gagnant du jeu



## 5 Conclusion

L'implémentation des patterns (proxy, stratégie, factory et adapter) nous a facilité la réalisation de certaines fonctionnalités complexe,nous a permis aussi d'optimiser le code , avoir un code propre et lisible. Avec ce projet nous avons compris que la programmation orienté objet est très intéressante et permet de réaliser plusieurs taches que nous ne pourrons pas faire avec d'autres langages de programmation. Elle permet par exemple la création d'une instance en fonction d'une classe créée préalablement. De plus, le Pattern MVC(Modèle, Vue, Controleur) nous a permis de créer notre jeu en utilisant deux parties : la partie ligne de commande et la partie graphique.