

UNIVERSITÉ CAEN NORMANDIE

Rapport Méthodes de conception (L3 informatique)

Soumis par :

David Tom

Huron Emilien

Pépin Florian

2 décembre 2024

Campus 2 CAEN

1^{er} Semestre



UNIVERSITÉ
CAEN
NORMANDIE

Table des matières

1	Démarrer le jeu	2
1.1	Configuration	2
1.2	Exécuter le jeu	2
2	Règles du jeu	2
2.1	Règles générales	2
2.2	Choisir sa spécialisation	2
2.3	Déterminer son tour de jeu	3
3	Les patterns	3
3.1	Pattern Factory Method	3
3.2	Pattern Proxy	3
3.3	Pattern Observer	4
3.4	Pattern Strategy	5
3.5	Pattern Adapter	6
4	Annexe des images	7

1 Démarrer le jeu

1.1 Configuration

Avant de lancer l'application, vous pouvez personnaliser certains aspects du jeu en modifiant le fichier de configuration **config.properties** situé dans le dossier **src/resources**. Ce fichier contient des paramètres tel que le nombre de joueurs graphiques, le nombre de joueurs en ligne de commandes, etc...

La stratégie des joueurs robots et la génération de la grille de jeu peuvent être personnalisés en modifiant le fichier **Main.java** situé dans le package **src/gridarena**.

1.2 Exécuter le jeu

Pour lancer le jeu, ouvrez un terminal à la racine du projet, dans le dossier livraison et exécutez le commande **ant run**.

2 Règles du jeu

Notre jeu de combat repose sur des mécaniques stratégiques au tour par tour, où les joueurs doivent utiliser tactique et planification pour survivre et éliminer les adversaires.

2.1 Règles générales

Nous avons modifié les règles du jeu afin de les améliorer.

Une action possible est :

- déplacement d'une case (4 directions possibles seulement).
- dépôt d'une mine sur l'une des 8 cases voisines.
- dépôt d'une bombe sur l'une des 8 cases voisines.
- utilisation d'un tir horizontal ou d'un tir vertical.
- déclenchement du bouclier, qui rend invincible au prochain tour.
- utilisation d'une hache sur la case voisine, vers le haut, le bas, la gauche ou la droite.
- passer son tour.

Lors de sa création, une grille peut contenir des murs, des kits de soins, des boîtes de munitions et des barils explosifs.

Par défaut une bombe explose au bout de trois tours.

Par défaut un tir horizontal ou vertical n'a pas de portée limitées.

Par défaut une mine n'est visible que par celui qui l'a posé.

2.2 Choisir sa spécialisation

Lors d'une partie, que vous soyez un joueur graphique ou un joueur en ligne de commandes, vous devrez choisir la spécialité de votre héro. En fonction de votre choix vous aurez un inventaire différent. Le choix de votre spécialité se fait également en tour par tour, c'est le but du jeu.

2.3 Déterminer son tour de jeu

Si vous êtes un joueur graphique, un cadre vert autour de votre écran indiquera que c'est à votre tour de jouer. Si vous êtes un joueur sur ligne de commandes, un message sera envoyé dans le terminal.

3 Les patterns

3.1 Pattern Factory Method

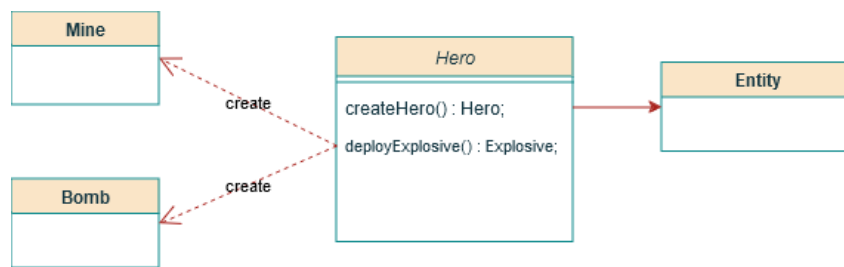


FIGURE 1 – Factory

Le pattern Factory Method nous sert à créer une instance de Hero depuis la classe Hero. Il nous sert aussi à créer une instance de Bombe ou de Mine depuis le Hero.

3.2 Pattern Proxy

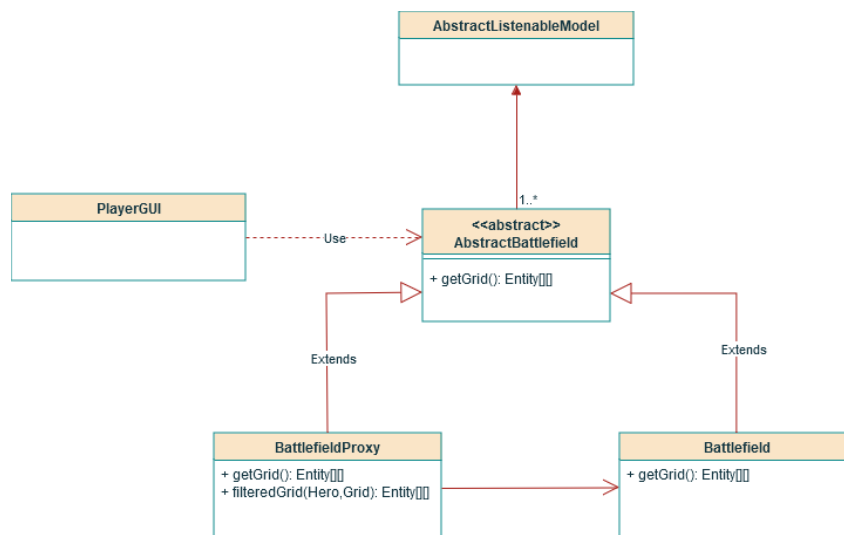


FIGURE 2 – Proxy

Dans notre projet, nous avons utilisé le pattern proxy. Dans notre cas, nous avons créé une classe abstraite **AbstractBattlefield** et non une interface car il était intéressant de lui faire hériter directement le **ModelListener**.

PlayerGUI et GameGUI ont tous les deux une vue sur la grille, mais une vue différente.

En effet PlayerGUI aura une vue "filtrée" des entités présentes dans la grille contrairement à GameGUI qui aura une vue de l'ensemble des entités présentes dans la grille.

3.3 Pattern Observer

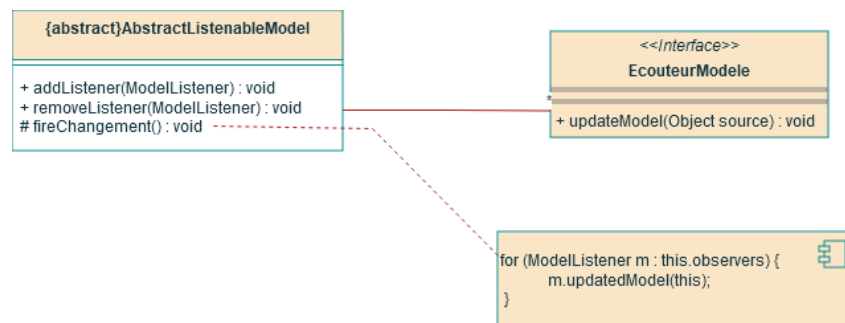


FIGURE 3 – Observer

Nous avons mis en place le pattern observer au sein de notre jeu MVC. Notre modèle de jeu est isolé de la vue et du contrôleur. Il possède juste des "listeners" de ses éventuels changements.

3.4 Pattern Strategy

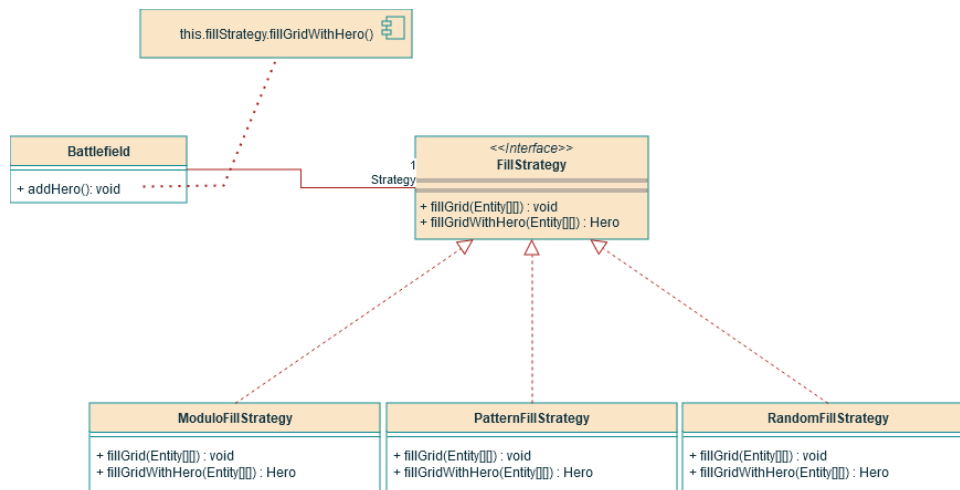


FIGURE 4 – Strategy

La génération de la grille de jeu se fait avec le pattern Strategy. Celui-ci nous a permis de définir différentes stratégies pour remplir la grille. Nous avons une stratégie de remplissage aléatoire, une stratégie de remplissage avec un pattern précis et une stratégie de remplissage utilisant modulo.

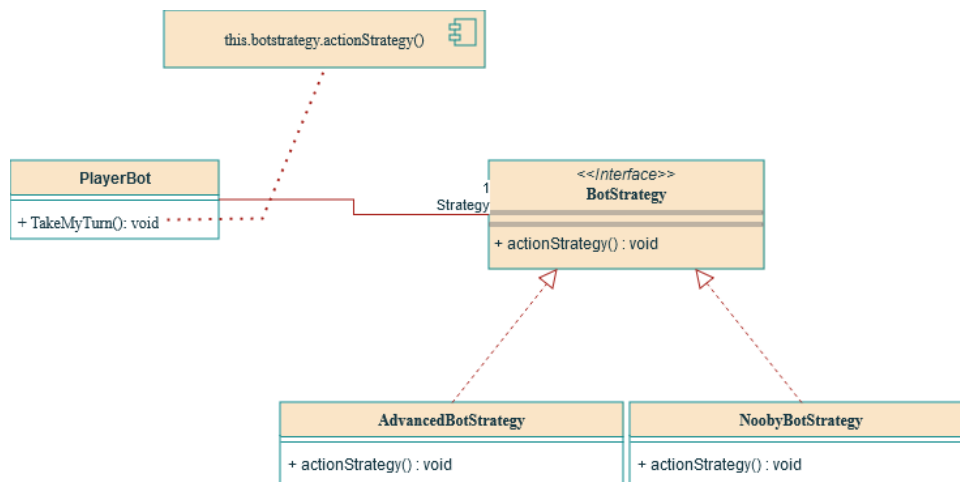


FIGURE 5 – Strategy

Nous avons également utilisé le pattern strategy pour mettre en place les joueurs robots. Nous avons mis en place une stratégie totalement aléatoire et nous avons mis en place une stratégie un peu plus fine. La stratégie un peu plus finie va, par exemple, regarder si le joueur peut tirer sur un autre joueur.

3.5 Pattern Adapter



FIGURE 6 – Adapter

Nous avons repris le même principe qu'au TP08. D'un côté nous avons un groupe de Hero et de l'autre un adapter qui permet de prendre les informations du groupe de Hero et de les adapter à un JTable. Evidemment ceci est mis à jour automatiquement via des listeners.

4 Annexe des images








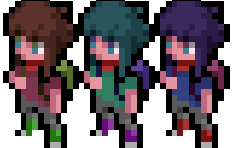
	Générateur d'image IA Bing
	Générateur d'image IA Bing
	Générateur d'image IA Bing
	Générateur d'image IA Bing
	Bombe - Vecteezy.com
	munitions - Freepik
	décor - Cupnooble
	Héros - ssary.itch

TABLE 1 – Annexe des images