

# JEU SMALLWORLD

## RAPPORT DE CONCEPTION

Hyuk-Chan KWON  
Florent MALLARD

12 novembre 2014



## Table des matières

<b>1</b>	<b>Principes et But du jeu</b>	<b>3</b>
1.1	Règles du jeu . . . . .	3
1.2	Tours de jeu . . . . .	3
<b>2</b>	<b>Analyse</b>	<b>4</b>
2.1	Lancement d'une partie . . . . .	4
2.2	Tour de jeu . . . . .	7
2.3	Combat . . . . .	10
2.4	Unités . . . . .	11
<b>3</b>	<b>Modélisation et diagrammes de classe</b>	<b>12</b>
3.1	Fabrique . . . . .	12
3.2	Monteur . . . . .	13
3.3	Stratégie . . . . .	14
3.4	Poids-mouche . . . . .	15
3.5	Diagramme de classes global . . . . .	16
<b>4</b>	<b>Conclusion</b>	<b>17</b>

# Introduction

Le projet de Programmation et de Modélisation Orientée Objet se porte sur la réalisation d'un jeu inspiré de SmallWorld. Il s'agit d'un jeu de stratégie à deux joueurs dans lequel chacun dirige un peuple. Les unités des peuples bougent sur les cases de la carte afin de les conquérir, et combattent les unités ennemies. Le but est de contrôler plus de cases que son adversaire.

Nous allons aborder les thèmes principaux de la phase d'analyse et de conception. Nous y expliquerons nos choix de conception et de modélisation de notre jeu. Nous commencerons par effectuer un rappel des règles afin de distinguer les actions possibles. Les étapes d'analyse ainsi que les différents diagrammes (classe, séquence, cas d'utilisation) figurent également dans ce rapport. Nous expliquerons également notre utilisation des patrons de conception.

## 1 Principes et But du jeu

### 1.1 Règles du jeu

#### 1.1.1 Peuples

Les joueurs ont le choix entre trois peuples : les Orcs, les Elfs et les Nains. Chacun dispose de bonus et malus influant sur la façon de les jouer.

**Elfs** Les Elfs ont un coût de déplacement réduit de moitié sur une case Forêt, tandis que le déplacement sur une case Déserte est deux fois plus coûteux. Lors d'un combat dont l'issue conduit à la mort de l'unité, celle-ci a 50% de chance de s'échapper avec 1 point de vie.

**Orcs** Les Orcs ont un coût de déplacement de 50% sur une case Plaine. Ils ne gagnent aucun point sur une case Forêt, mais ont un bonus propre à l'unité lorsque celle-ci en tue une autre.

**Nains** Les Nains ont un coût de déplacement divisé par deux sur une case Plaine. Ils n'acquièrent aucun point sur les cases de ce type. Ils peuvent se déplacer d'une case Montagne à une autre à condition que celle-ci ne soit pas occupée par une unité adverse.

#### 1.1.2 Cartes

Le joueur, à la création de la partie, a le choix entre trois cartes :

- Small : 2 joueurs, 5 cases x 5 cases, 5 tours, 4 unités par peuple.
- Medium : 2 joueurs, 10 cases x 10 cases, 20 tours, 6 unités par peuple.
- Large : 2 joueurs, 15 cases x 15 cases, 30 tours, 8 unités par peuple.

#### 1.1.3 Déroulement d'une partie

Au début du jeu, chaque joueur choisit son peuple. Chaque peuple débute la partie avec toutes ses unités sur la même case de la carte, choisie de manière à ce que les joueurs ne soient pas trop proches. L'ordre de jeu est déterminé aléatoirement en début de partie. Les joueurs jouent chacun leur tour sur le même ordinateur.

### 1.2 Tours de jeu

Lorsqu'un joueur peut jouer (c.-à-d. une fois par tour), il peut déplacer toutes ses unités suivant leur nombre de points de déplacement (à part dans le cas d'un bonus lié au peuple, un déplacement sur une case coûte un point de déplacement). Il est possible pour chaque unité de passer son tour (généralement par le biais de la touche espace). Une unité combattante peut engager un combat

s'il lui reste au moins un point de mouvement. Lorsqu'un joueur a fini son tour, il clique sur le bouton correspondant ("Fin tour"). C'est alors au joueur suivant de commencer son tour. La partie se termine lorsque le nombre de tours prédéfini en début de partie à été effectué, ou lorsqu'il ne reste qu'un seul joueur sur le plateau.

## 2 Analyse

Après avoir lancé le jeu, l'utilisateur a la possibilité de créer une nouvelle partie, ou bien d'en charger une existante.

### 2.1 Lancement d'une partie

La Figure 1 représente la création d'une partie. Le joueur A choisit son nom de joueur puis son peuple. Le joueur B fait de même, puis définit la taille de la carte. Celui-ci termine en lançant la partie. Nous avons donc défini la machine à état globale du jeu, et le diagramme de séquence associé, respectivement en Figure 2 et en Figure 3

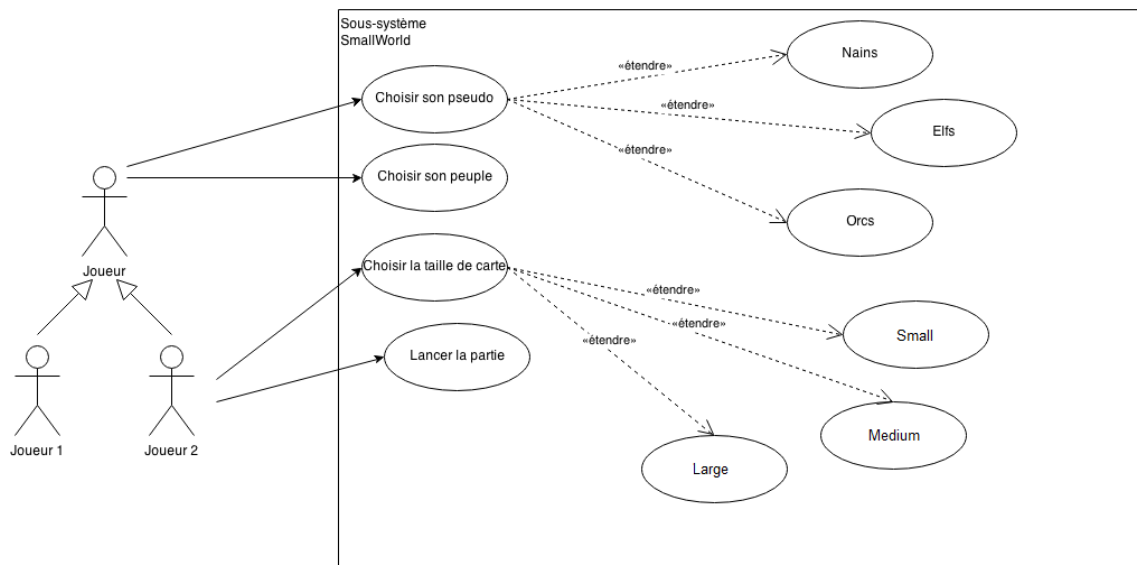


FIGURE 1 – Diagramme de cas d'utilisation de création du jeu

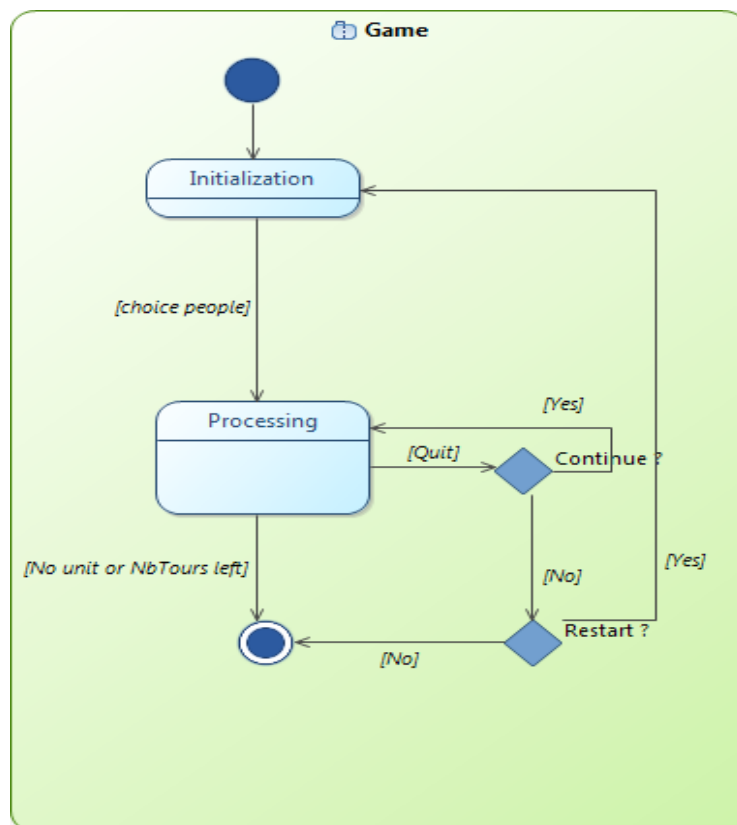


FIGURE 2 – États globaux du jeu



## 2.2 Tour de jeu

Nous avons représenté un tour de jeu avec la Figure 4. Le joueur peut déplacer chaque unité selon son nombre de points de déplacement, et le coût de celui-ci. En cas de rencontre d'une unité adverse, un combat a lieu. Il peut également passer le tour de l'unité, ou terminer totalement son tour. L'annulation d'un déplacement est prévue, ainsi que la sauvegarde de la partie. Le déroulement de ces actions est représenté sur la Figure 5, et sur la Figure 6.

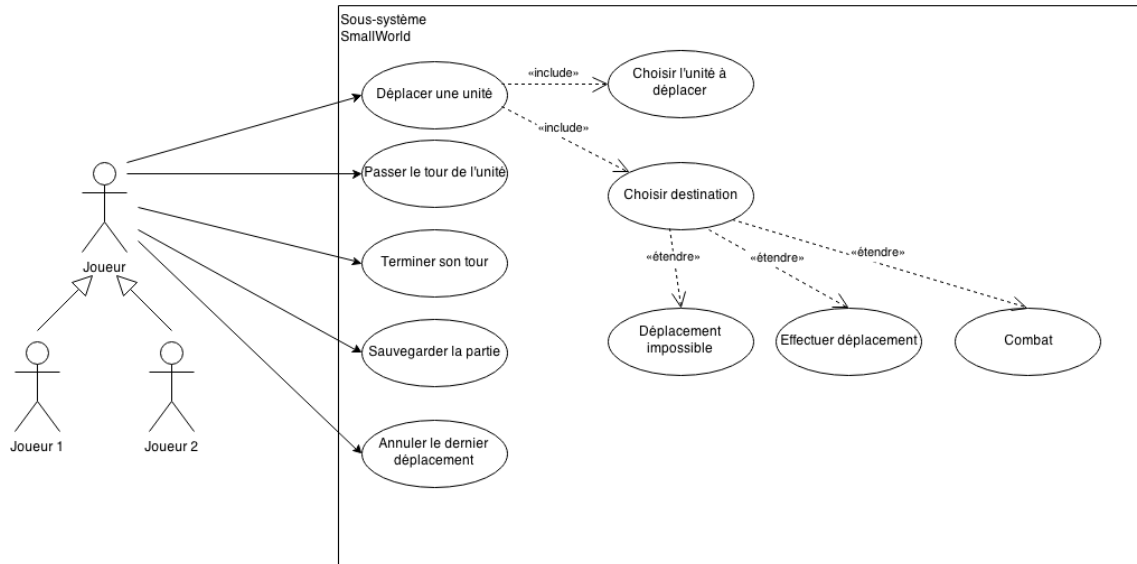


FIGURE 4 – Diagramme de cas d'utilisation d'un tour de jeu

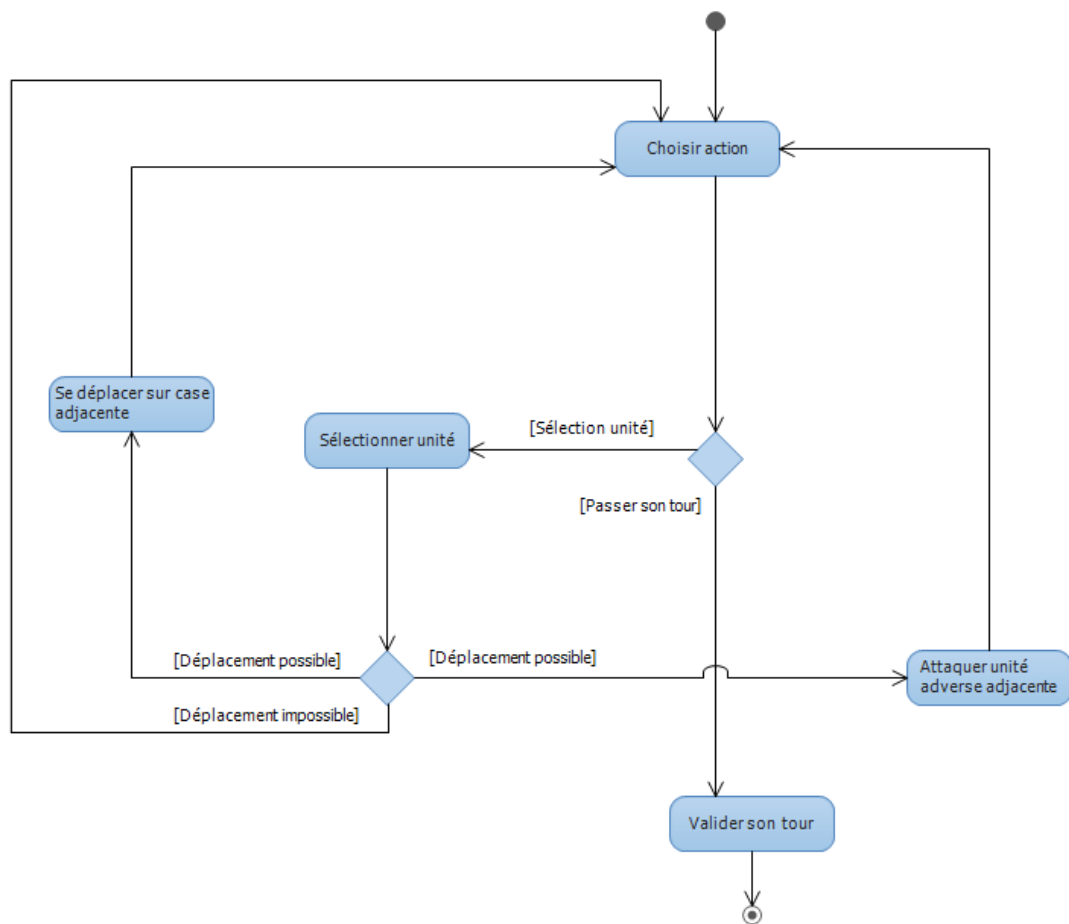


FIGURE 5 – Diagramme d'interaction d'un tour de jeu



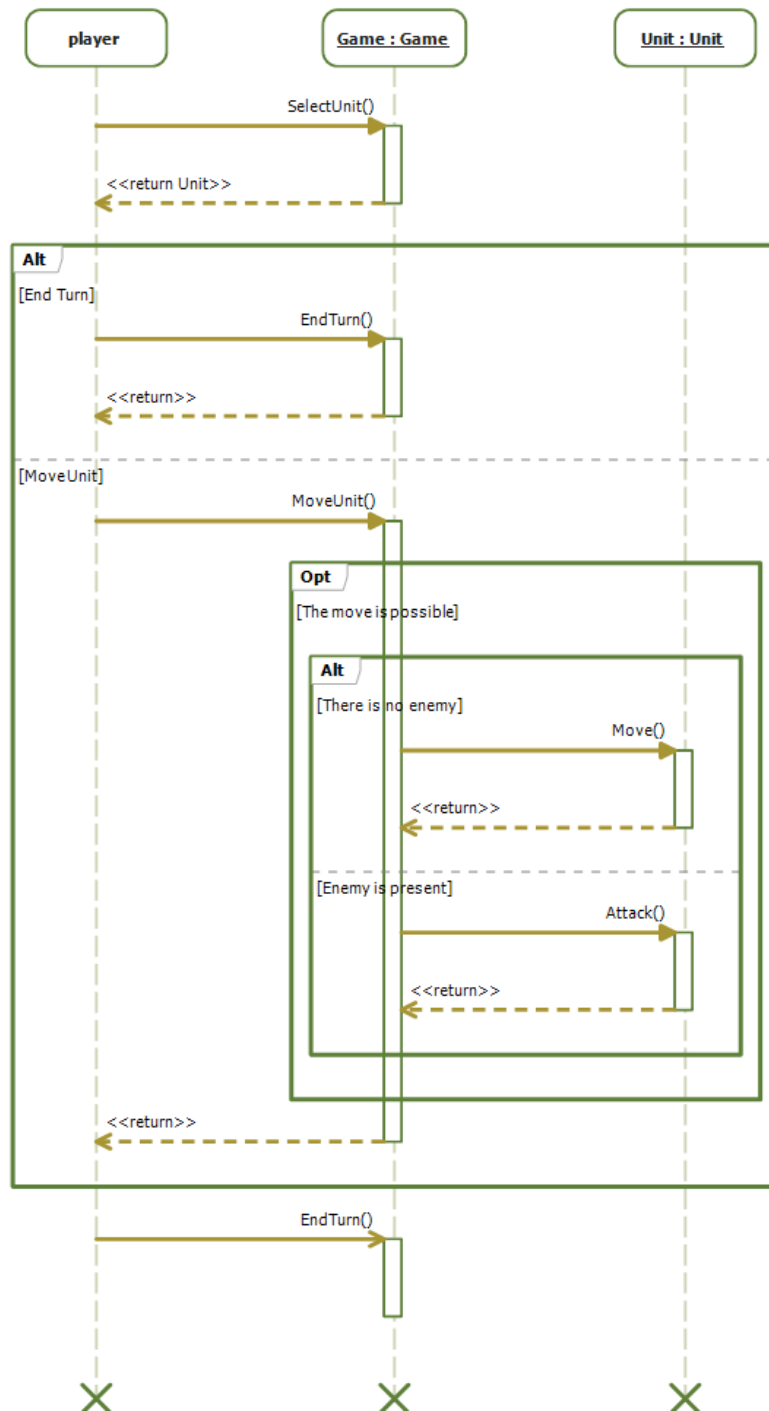


FIGURE 6 – Diagramme de séquence d'un tour de jeu

## 2.3 Combat

A la rencontre d'unités adverses, le joueur peut décider de combattre à l'aide de ses unités. Le déroulement d'un combat est montré à l'aide de la Figure 7.

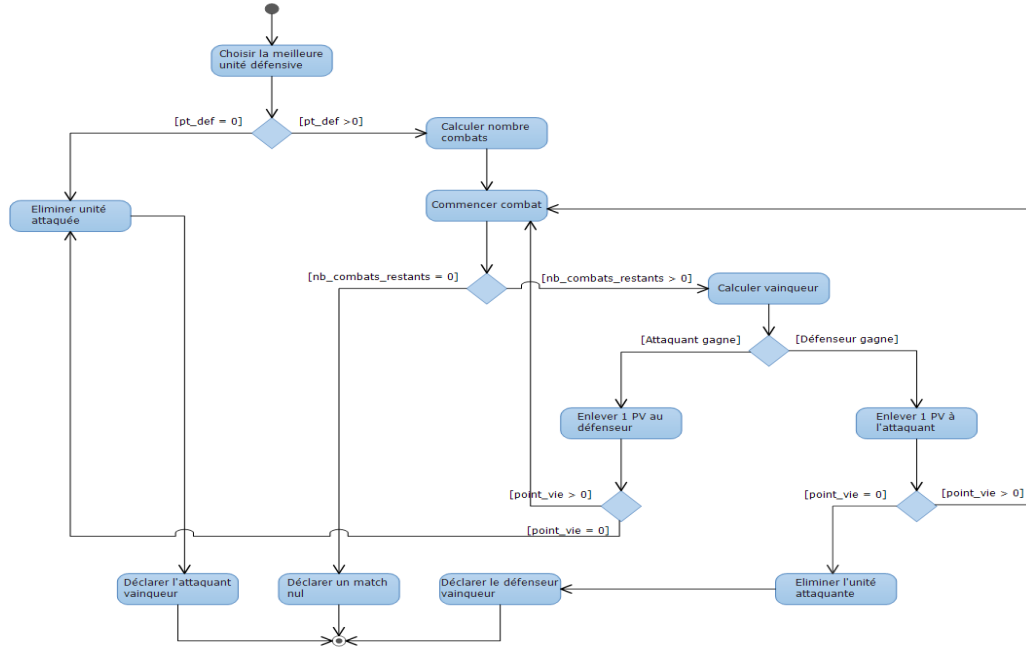


FIGURE 7 – Diagramme d'interaction représentant un combat

Les calculs des combats prennent en compte les points de vie restants de chaque unité en combat, pour déterminer des probabilités de remporter le combat. Une attaque est composée de 3 au maximum des points de vie des unités en combat + 2. Chaque combat fait perdre un point de vie à l'unité perdante, jusqu'à ce que le nombre de combats soit atteint ou qu'une unité soit tuée.

## 2.4 Unités

Le diagramme d'état-transitions, Figure8, montre le cycle de vie d'une unité. Une unité peut simplement être vivante ou non. A sa création, elle possède cinq points de vie. Au début d'un tour, l'unité possède un point de déplacement. Elle a alors la possibilité de se déplacer. Si elle rencontre une unité ennemie, elle combat et en ressort soit vainqueur (ou match nul), soit perdante. Dans ce dernier cas l'unité meurt et atteint son état final.

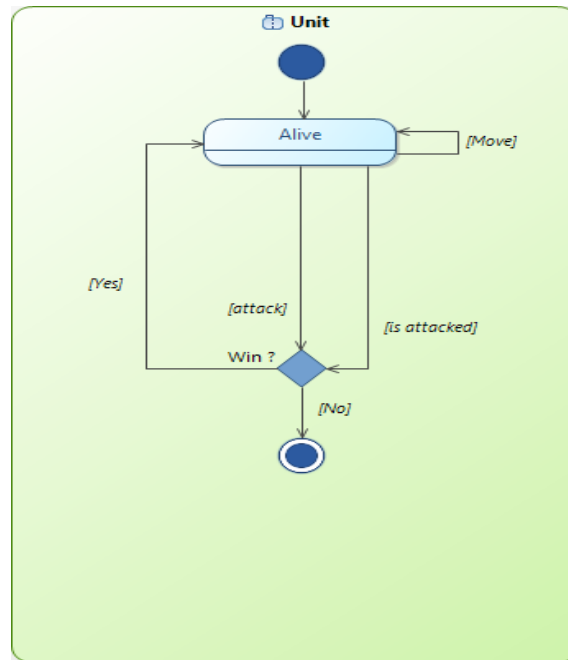


FIGURE 8 – Machine à état pour une unité

### 3 Modélisation et diagrammes de classe

Nous allons dans cette section expliquer nos utilisations des différents patrons de conception, accompagnés de diagrammes de classe montrant leur mise en place au sein de notre de projet.

#### 3.1 Fabrique

Le patron de conception *Fabrique* est utilisé pour la création des différents peuples. En effet, le jeu créera simplement un peuple, sans savoir duquel il s'agit. Une interface réunit les points commun entre les différents peuples. Les unités étant toutes les mêmes, elles n'ont pas besoin de *Fabrique*. Les classes sont représentées dans la Figure 9

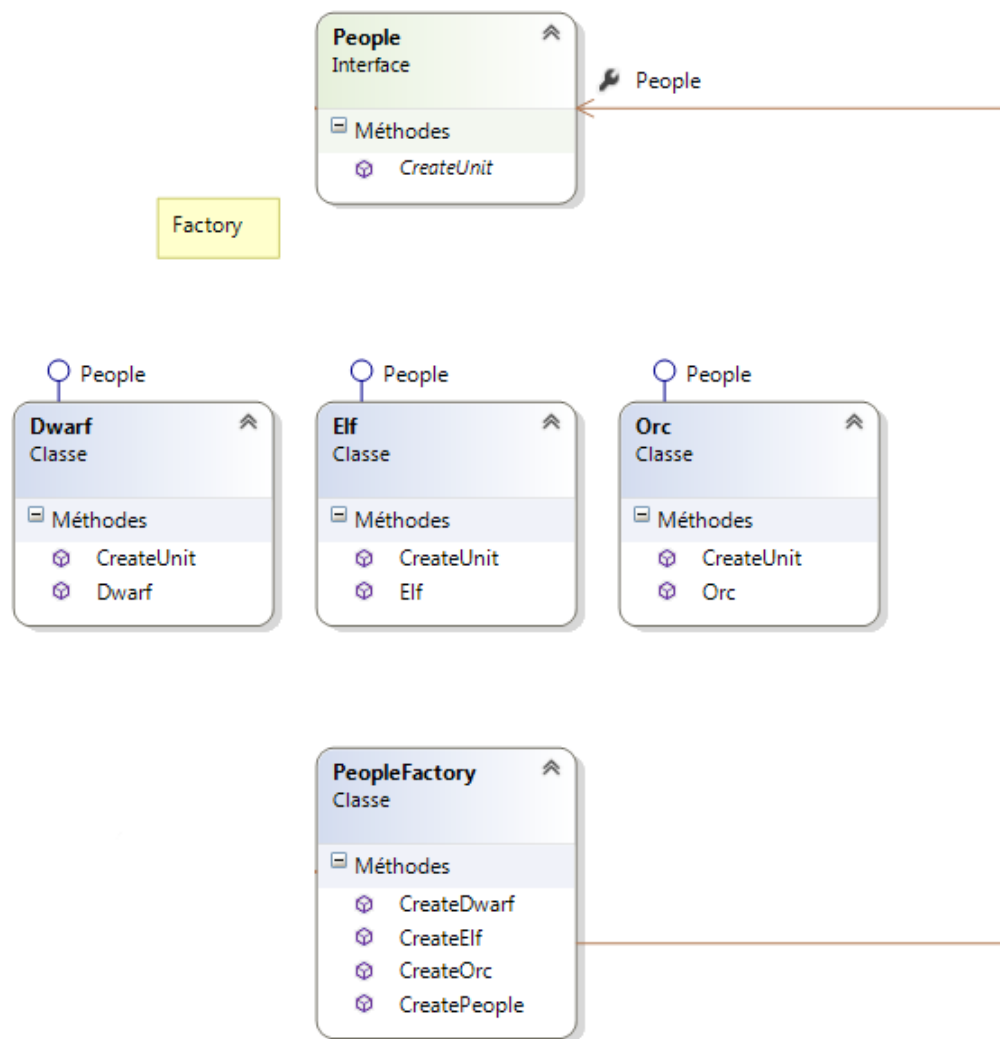


FIGURE 9 – Fabrique des différents peuples

### 3.2 Monteur

Nous avons utilisé un *Monteur* pour la création d'une partie. En effet, la création d'une partie est un assemblage de différents objets complexes. Son utilisation extériorise la création d'une partie, et permettra de pouvoir implémenter les différentes configurations de partie. Il est représenté à la Figure 10.

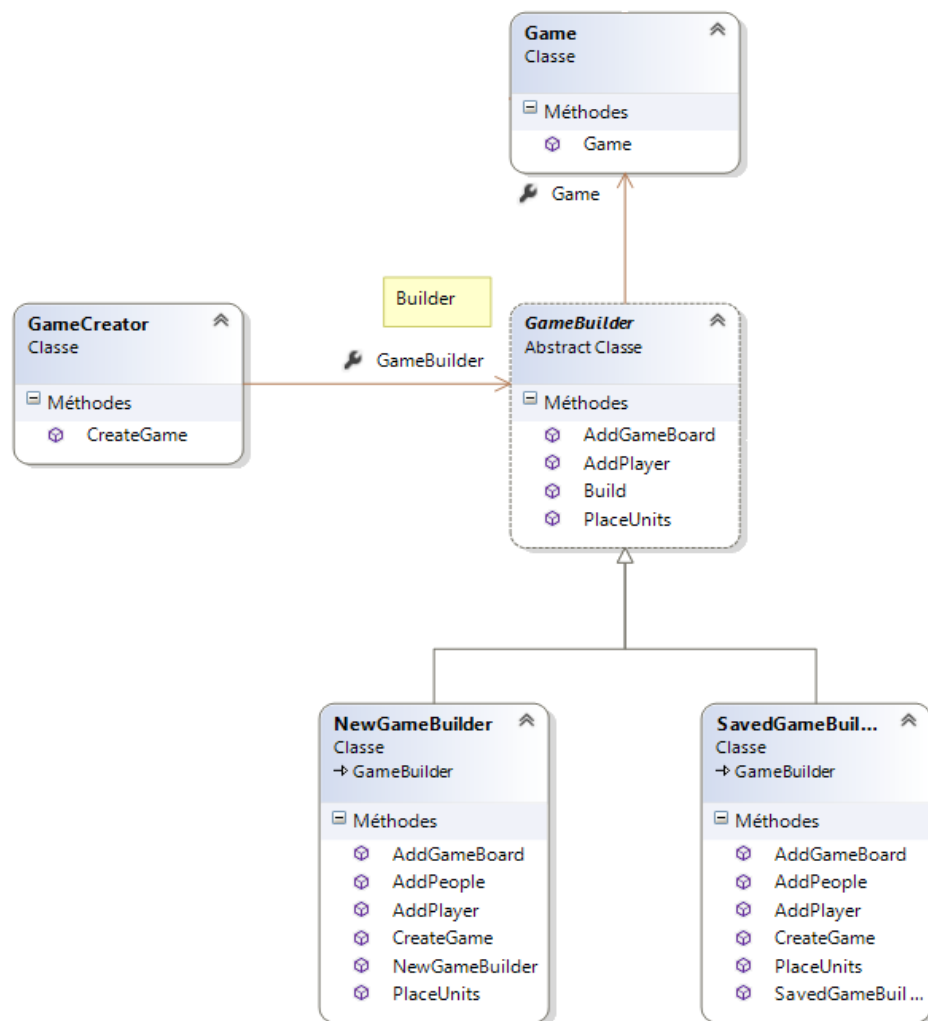


FIGURE 10 – Monteur de la partie

### 3.3 Stratégie

Le patron de conception *Stratégie* est utilisé pour la création des cartes. En effet, le joueur a la choix entre trois tailles : Petite, Moyenne et Grande. Cette implémentation nous permettra de changer facilement d'algorithmes en fonction de ce choix, et d'en implémenter de nouveaux au besoin. Cette modélisation est représentée à la Figure 11.

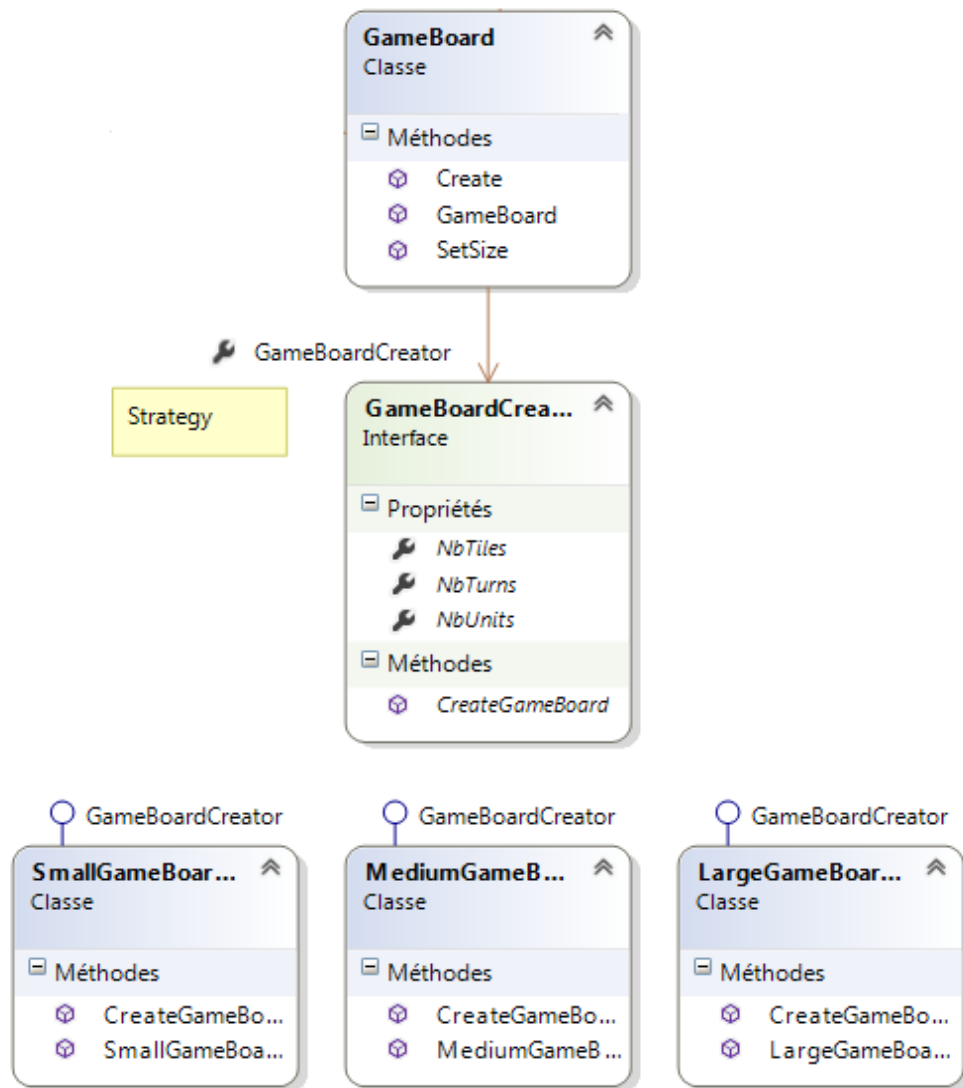


FIGURE 11 – Stratégie de création de la carte

### 3.4 Poids-mouche

La carte est composée de cases de quatre types : Montagne, Plaine, Désert et Forêt. Instancier chaque case consommerait beaucoup de mémoire, c'est pourquoi nous utilisons le *Poids-Mouche*. Ce modèle permet de ne pas créer plusieurs fois le même type de case. La Figure 12 illustre ce patron. Nous utilisons également une *Fabrique* pour la création aléatoire des cases de la carte.

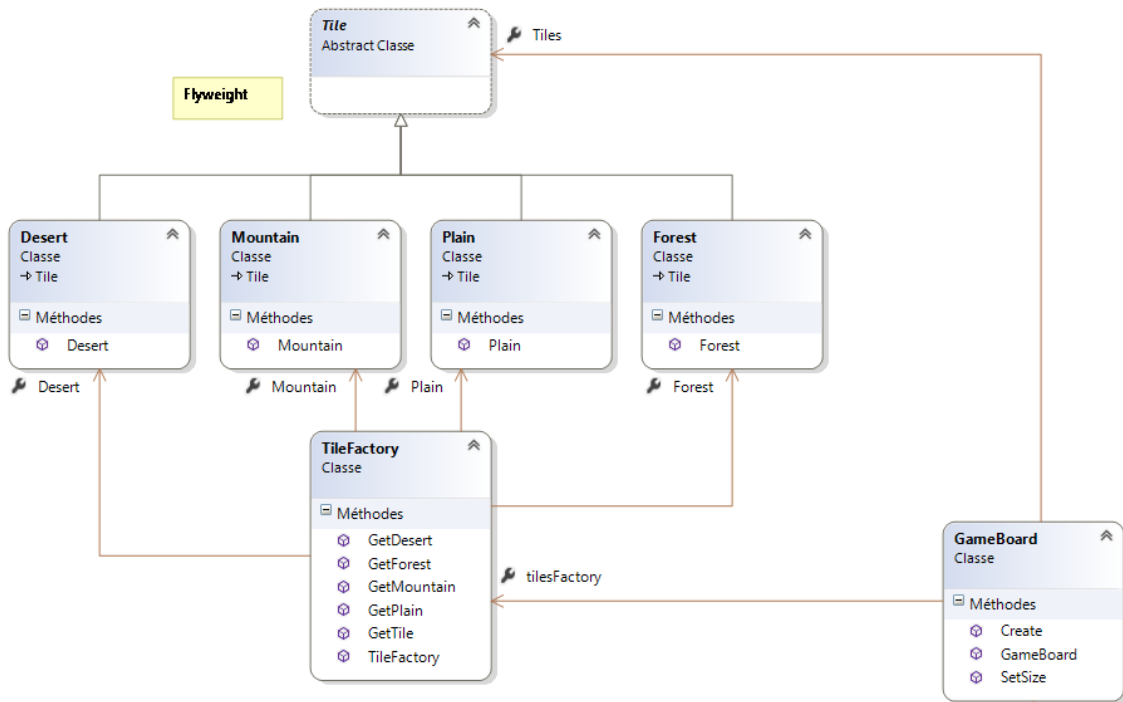


FIGURE 12 – Poids-Mouche pour les cases de la carte

### 3.5 Diagramme de classes global

Voici donc la représentation globale de notre projet, visible à la Figure 13.

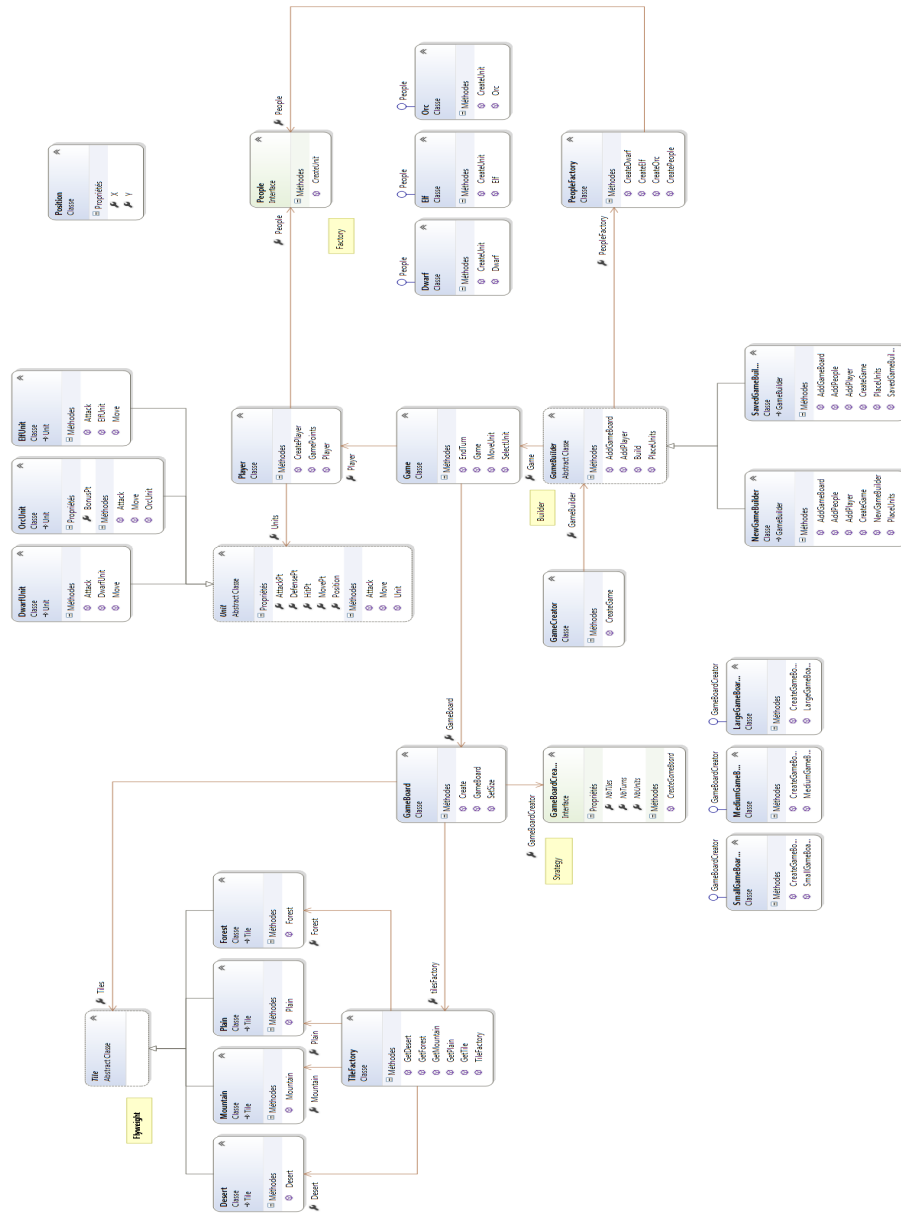


FIGURE 13 – Diagramme de classes complet



## 4 Conclusion

La phase d'analyse et de conception de notre projet de Programmation et de Modélisation Orientée Objet se termine avec ce rapport. Elle nous a permis, avec le polycopié contenant le cahier des charges, de penser à toutes les classes et aux attributs dont nous allons avoir besoin afin que notre version du jeu SmallWorld dispose des fonctionnalités nécessaires. Nous allons donc commencer à implémenter les méthodes décrites dans ce rapport, dont le code a, en partie, été généré automatiquement.

## Table des figures

1	Diagramme de cas d'utilisation de création du jeu . . . . .	4
2	États globaux du jeu . . . . .	5
3	Séquences pour la création du jeu . . . . .	6
4	Diagramme de cas d'utilisation d'un tour de jeu . . . . .	7
5	Diagramme d'interaction d'un tour de jeu . . . . .	8
6	Diagramme de séquence d'un tour de jeu . . . . .	9
7	Diagramme d'interaction représentant un combat . . . . .	10
8	Machine à état pour une unité . . . . .	11
9	Fabrique des différents peuples . . . . .	12
10	Monteur de la partie . . . . .	13
11	Stratégie de création de la carte . . . . .	14
12	Poids-Mouche pour les cases de la carte . . . . .	15
13	Diagramme de classes complet . . . . .	16