

INSA RENNES

JEU SMALL WORLD

Rapport de conception



Auteurs :
Damien CRÉMILLEUX
Lauriane HOLY

Enseignants :
Eric ANQUETIL
Arnaud BLOUIN
Manuel BOUILLON
Grégoire RICHARD
Maud MARCHAL

Table des matières

Introduction	5
1 Principes et utilisation	5
1.1 Règles du jeu	6
1.2 Les peuples	6
1.3 La Carte du Monde	6
1.4 Les Combats	6
1.5 La Vue	7
1.6 Début de Partie	7
1.7 Tour de jeu	7
2 Analyse	7
2.1 Fonctionnalités globales	7
2.2 Lancement d'une partie	8
2.3 Déroulement d'un tour de jeu	11
2.4 Déroulement d'un combat	13
2.5 Cycle de vie d'une unité	14
3 Modélisation du diagramme de classe	15
3.1 Patrons de conception	15
3.1.1 Monteur - Création de la partie	15
3.1.2 Stratégie - Création des différents types de cartes	16
3.1.3 Poids-mouche - Modélisation de la carte	17
3.1.4 Fabrique - Gestion des peuples et des unités	18
3.1.5 Observateur	18
3.2 Diagramme de classe	18
Conclusion	20

Introduction

Le projet de conception et de programmation orienté objet a pour objectif de concevoir un jeu inspiré de Small World, un jeu de stratégie au tour par tour où chaque joueur dirige un peuple. Les joueurs bougent leurs unités sur une carte et peuvent combattre celles de l'adversaire. Le but est d'obtenir plus de points que les autres peuples.

Ce rapport traite les thèmes essentiels abordés lors de la phase d'analyse et de conception du jeu. Son plan est fortement inspiré du polycopié d'énoncé. L'énoncé du projet est repris dans ce rapport afin de faire le parallèle entre les tâches à réaliser et notre modélisation.

Ce rapport explique nos choix de modélisation et de conception du jeu Small World. Ainsi, les principes et les règles du jeu sont d'abord rappelés, afin de comprendre et de cerner les différentes interactions possibles. Cette partie étant tirée du polycopié, il est possible de commencer directement à partir de la deuxième partie, page 7.

Nous y faisons figurer les différentes étapes d'analyse ainsi que les différents diagrammes de classe, d'utilisation et de séquences qui viendront illustrer le fonctionnement et l'organisation du jeu.

Nous expliquons également les choix d'utilisation de patrons de conception afin de résoudre les problèmes de conception de notre jeu. Ils nous permettent de décrire les procédés de conception de notre jeu au niveau de l'architecture, d'expliquer sous forme de diagrammes les arrangements et le rôle joué par les différents modules.

Enfin, nous donnons le diagramme de classe complet de notre jeu.

Tous les diagrammes sont joints à ce rapport sous forme d'images. En effet, certains diagrammes sont trop volumineux pour pouvoir s'afficher correctement sur ce document.

Cette phase de conceptions et de modélisation nous servira ensuite lors de l'implémentation.

1 Principes et utilisation

Comme énoncé dans l'introduction Small World est un jeu tour-par-tour où chaque joueur dirige un peuple. Le but du jeu est de gérer des unités sur une carte pour obtenir le plus de points possible à la fin d'un certain nombre de tours. Le placement de chaque unité rapporte plus ou moins de points. Les unités d'un joueur peuvent également attaquer les unités d'un autre joueur pour détruire des unités (limitant ainsi l'acquisition de points de l'adversaire) et occuper une case de la carte. Les points sont comptés à la fin de la partie, c.-à-d. après un nombre prédéfini de tours. Le jeu se déroule sur une carte du monde sur laquelle les unités se déplacent.

1.1 Règles du jeux

1.2 Les peuples

Il existe trois peuples, Gaulois, Viking et Nain, ayant des caractéristiques très différentes influant sur les stratégies de jeu :

- Gaulois.
 - Le coût de déplacement sur une case Plaine est divisé par deux.
 - Une unité Gauloise fournit 1 point de plus lorsqu'elle occupe une case du type plaine.
 - Une unité Gauloise n'acquière aucun point sur les cases de type montagne.
- Vikings.
 - L'unité Viking a la capacité de se déplacer sur l'eau. L'occupation d'une case eau ne rapporte cependant aucun point.
 - Une unité Viking fournit 1 point de plus lorsqu'elle occupe une case au bord de l'eau.
 - Une unité Viking n'acquière aucun point sur les cases de type désert.
- Nains.
 - Lorsqu'elle se trouve sur une case montagne, une unité Nain a la capacité de se déplacer sur n'importe quelle case montagne de la carte à condition qu'elle ne soit pas occupée par une unité adverse.
 - Une unité Nain fournit 1 point de plus lorsqu'elle occupe une case forêt.
 - Une unité Nain n'acquière aucun point sur les cases de type plaine.

À chaque tour, toutes unités peuvent être déplacées ou attaquer. Par défaut (c.-à-d. hors bonus), chaque unité peut se déplacer d'une case par tour. Chaque unité possède 2 d'attaque, 1 de défense et 2 points de vie.

1.3 La Carte du Monde

La carte du monde se compose de cases carrées. La largeur d'une case est de 50 pixels. Il existe différents types de case : montagne, plaine, désert, eau, forêt. La carte sera créée en début de partie de manière aléatoire.

Il existe 3 types de cartes :

- Démon : 2 joueurs, 5 cases x 5 cases, 5 tours, 4 unités par peuples.
- Petite : 2 joueurs, 10 cases x 10 cases, 20 tours, 6 unités par peuples.
- Normale : 2 joueurs, 15 cases x 15 cases, 30 tours, 8 unités par peuples.

1.4 Les Combats

Pour qu'une unité puisse lancer une attaque contre une unité d'un autre peuple, elles doivent se situer sur des cases juxtaposées (attaque en diagonale impossible cependant). Lorsqu'une unité attaque une case contenant plusieurs unités, la meilleure unité défensive est choisie. Une unité attaquée possédant 0 en défense meurt immédiatement. Sinon, un certain nombre de combats a lieu. Ce nombre est choisi aléatoirement à l'engagement (entre 3 et le nombre de points de vie de l'unité ayant le plus de points de vie + 2 points). Le combat s'arrête lorsque ce nombre est atteint ou lorsque l'une ou autre des unités n'a plus de vie. Chaque combat calcule les probabilités de perte d'une vie de l'attaquant. Par exemple, Si l'attaquant a 4 en attaque et l'attaqué a 4 en défense (en tenant compte des bonus de terrain et du nombre de points de vie restant), l'attaquant

à 50% de (mal-)chance de perdre une vie. S'il a 3 att. contre 4 déf., le rapport de force est de $75\% : 3/4 = 25\%$, $25\% \text{ de } 50\% = 12.5\%$, $50\% + 12.5\% = 62.5\%$ chance pour l'attaquant de perdre une vie.

Explications du calcul : par défaut 2 unités égales ont 50% de gagner. Puisque dans le cas présent un écart de 25% est constaté entre les deux unités, il est nécessaire de pondérer le 50% par ces 25% ce qui donne 62.5% contre 37.5%. S'il a 4 att. contre 2 déf., le taux baisse à 25% ($2/4 = 50\%$, 50% de 50% = 25%, $25\% + 50\% = 75\%$ pour l'attaqué, $100\% - 75\% = 25\%$ pour l'attaquant). Évidemment, lorsque l'attaquant gagne cela signifie que l'adversaire perd un point de vie. Les points de vie entrent en compte dans le calcul des probabilités : si une unité attaquante ayant 4 en attaque possède 75% de sa vie, alors son attaque sera au final de $4 * 75\% = 3$. L'unité attaquée suit le même calcul pour sa défense.

À la fin d'un combat gagné par l'attaquant et si la case du vaincu ne contient plus d'unité, ce dernier se déplace automatiquement sur cette case. Lorsqu'un joueur n'a plus d'unité, il est éliminé. Lorsqu'il ne reste plus qu'un seul joueur dans une partie, celui-ci a gagné. Une unité ne regagne pas ses points de vie d'un tour à un autre.

1.5 La Vue

La carte, ses ressources, les unités de tous peuples sont visibles par tous les joueurs. Le jeu doit permettre de voir la carte du dessus (vue plateau) contrairement à beaucoup de jeux une vue isométrique.

1.6 Début de Partie

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

1.7 Tour de jeu

Lorsqu'un joueur peut jouer (c.-à-d. une fois par tour), il peut déplacer toutes ses unités suivant leur nombre de déplacements (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

2.1 Fonctionnalités globales

Au lancement du jeu, le joueur a la possibilité soit de créer une nouvelle partie soit de charger une partie existante. Le diagramme de cas d'utilisation

présenté Figure 1 illustre ces possibilités.

uc Lancement d'une partie

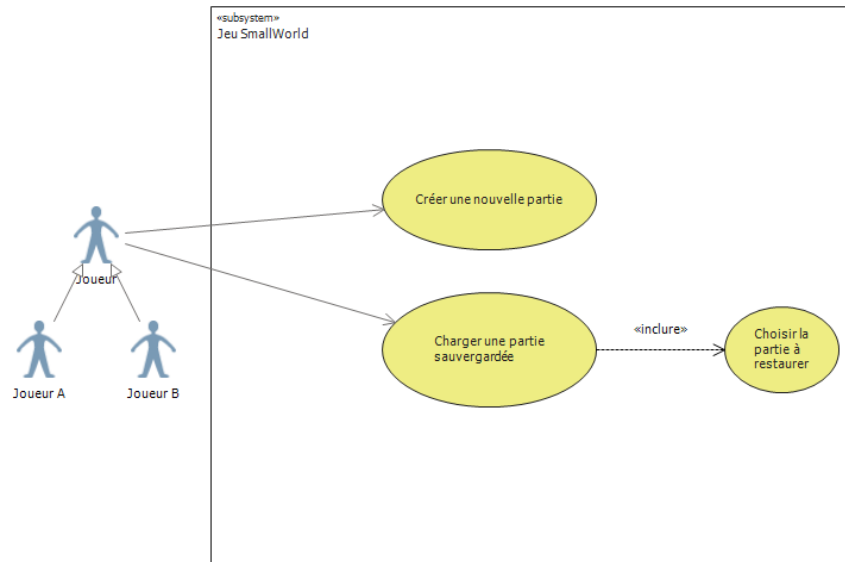


FIGURE 1 – Diagramme de cas d'utilisation global du jeu Small World

2.2 Lancement d'une partie

La création ou le chargement d'une partie est illustré de façon plus précise à la Figure 2. Lorsque l'on crée une partie, le joueur A commence par choisir son nom de joueur puis son peuple. Ensuite le joueur B fait de même et procède au choix de la taille de la carte, avant de lancer la partie. Le joueur B est donc plus spécialisé que le joueur A. Cette suite d'interactions est représentée Figure 3.

Enfin, un diagramme de séquence, Figure 4, représente les communications entre les joueurs et le jeu ainsi que de donner une représentation temporelle des interactions entre les objets. Ce diagramme de séquence nous permet de représenter la création d'une partie à partir des données (noms et peuples) entrées par les joueurs.

uc Création d'une partie

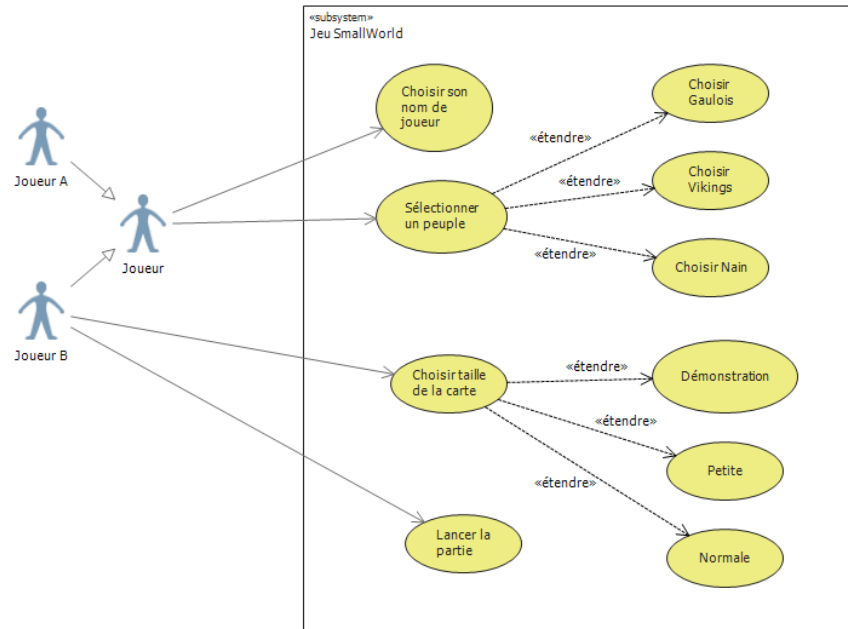


FIGURE 2 – Diagramme de cas d'utilisation de création d'une partie

act Lancement d'une partie

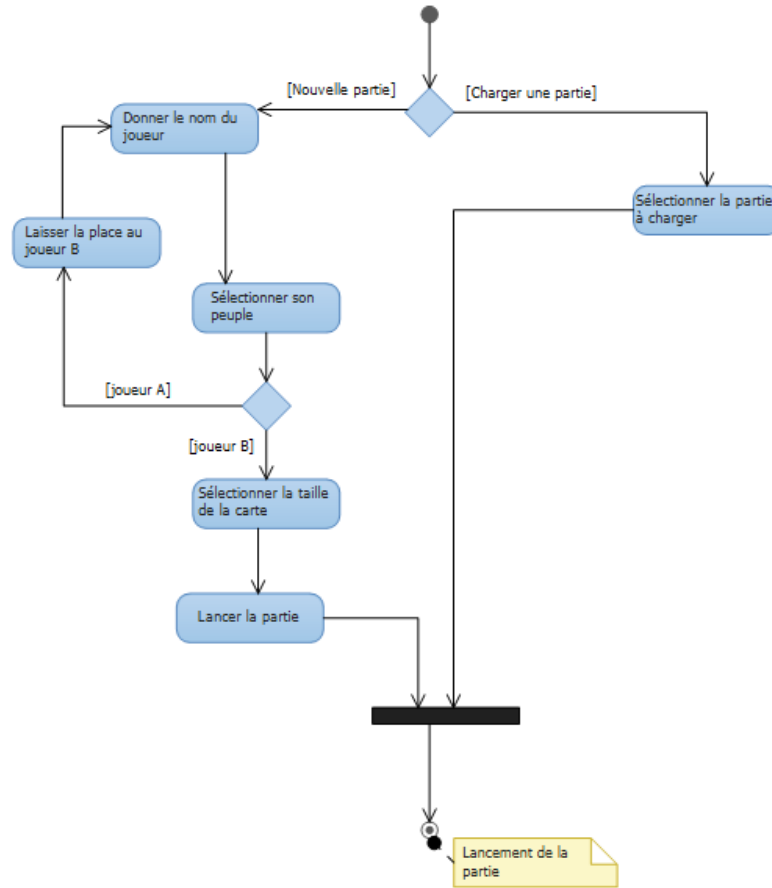


FIGURE 3 – Diagramme de d'activité de création d'une partie

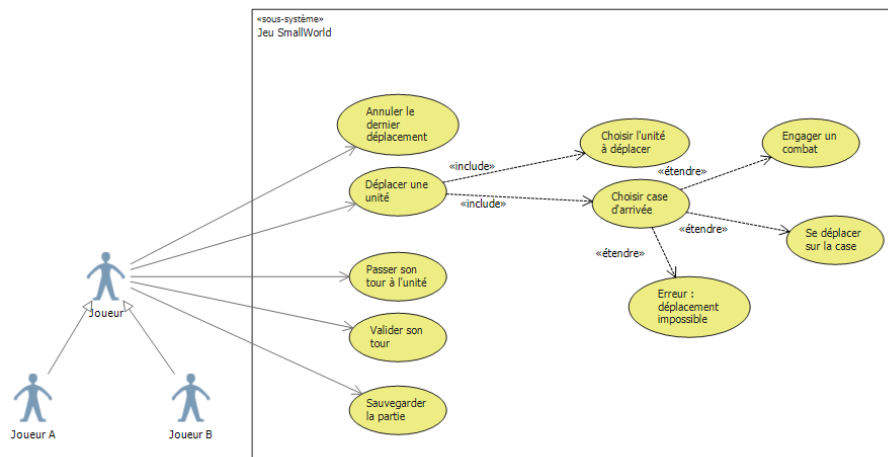


FIGURE 5 – Diagramme de cas d'utilisation d'un tour de jeu

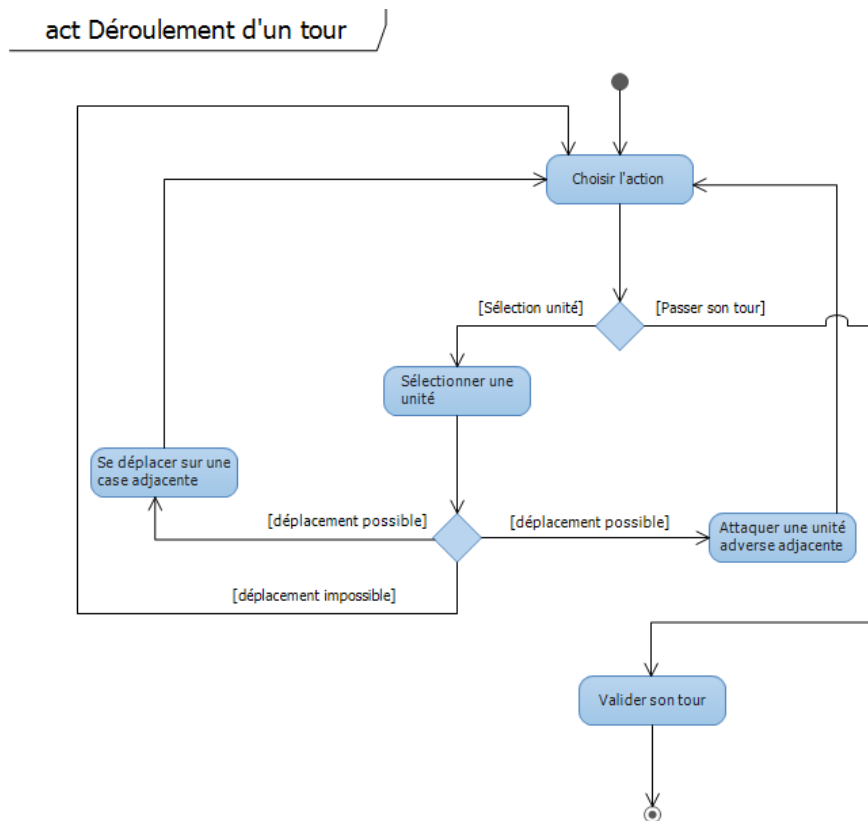


FIGURE 6 – Diagramme d'activité d'un tour de jeu

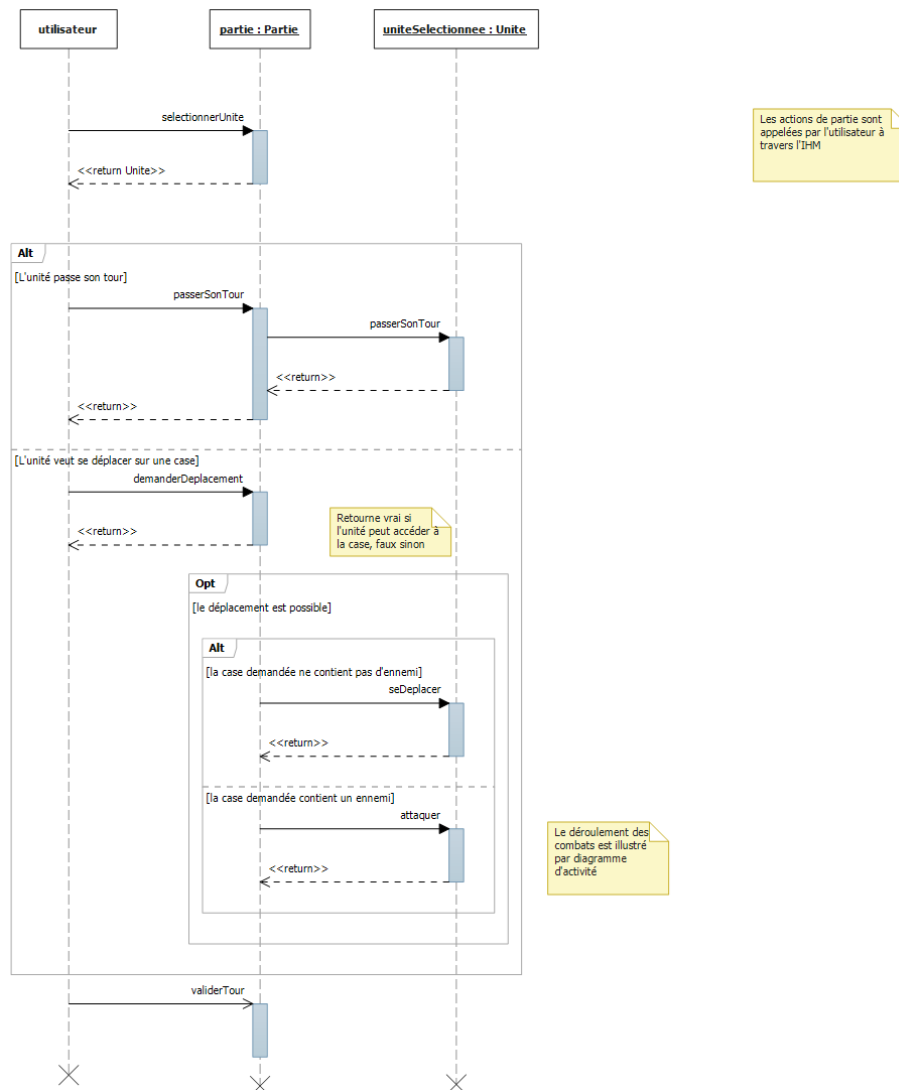


FIGURE 7 – Diagramme de séquence du déplacement d'une unité

2.4 Déroulement d'un combat

Au cours d'un tour de jeu, le joueur peut décider de combattre le joueur adverse à l'aide de ses unités. Le déroulement d'un combat est montré à l'aide de la Figure 8.

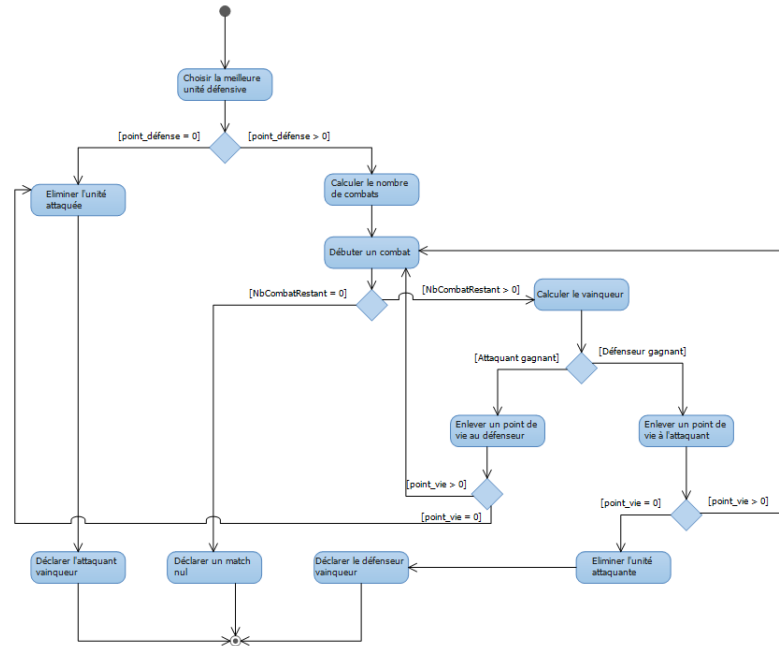


FIGURE 8 – Diagramme d'activité d'un combat

2.5 Cycle de vie d'une unité

Nous avons utilisé un diagramme d'état-transitions, Figure 9, afin de montrer le cycle de vie d'une unité et ses différents états possibles. À sa création, l'unité possède deux points de vie. Au début d'un tour, l'unité possède un point de déplacement. Elle a alors la possibilité de se déplacer, d'attaquer ou de passer son tour. À la fin de l'action, l'unité perd son point de déplacement et son tour est terminé. Lorsqu'elle a perdu ses deux points de vie, l'unité a été vaincue, elle meurt.

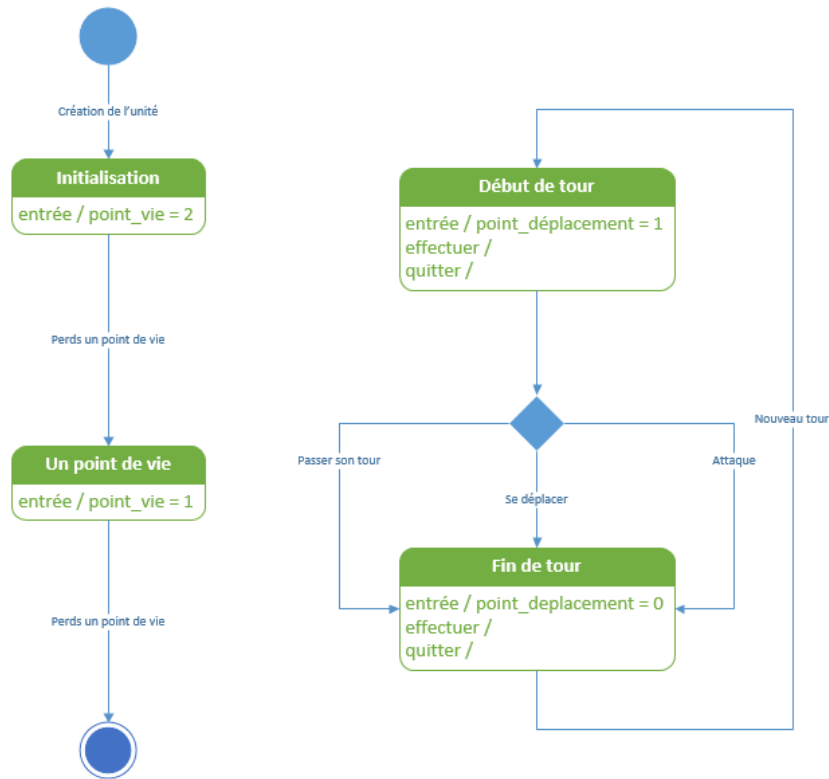


FIGURE 9 – Diagramme d'état-transitions

3 Modélisation du diagramme de classe

3.1 Patrons de conception

Les fonctionnalités du jeu étant définies, nous allons maintenant modéliser les différents aspects de celui-ci à l'aide des patrons de conception. Sur les diagrammes de classe qui suivent, les interfaces ne sont pas représentées, seulement les classes qui en dérivent.

3.1.1 Monteur - Création de la partie

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. L'utilisation de ce patron de conception extériorise la création d'une partie, sans alourdir la classe. En outre, le monteurt permettra de pouvoir implémenter différentes configurations de partie. Le monteurt est représenté à la Figure 10.

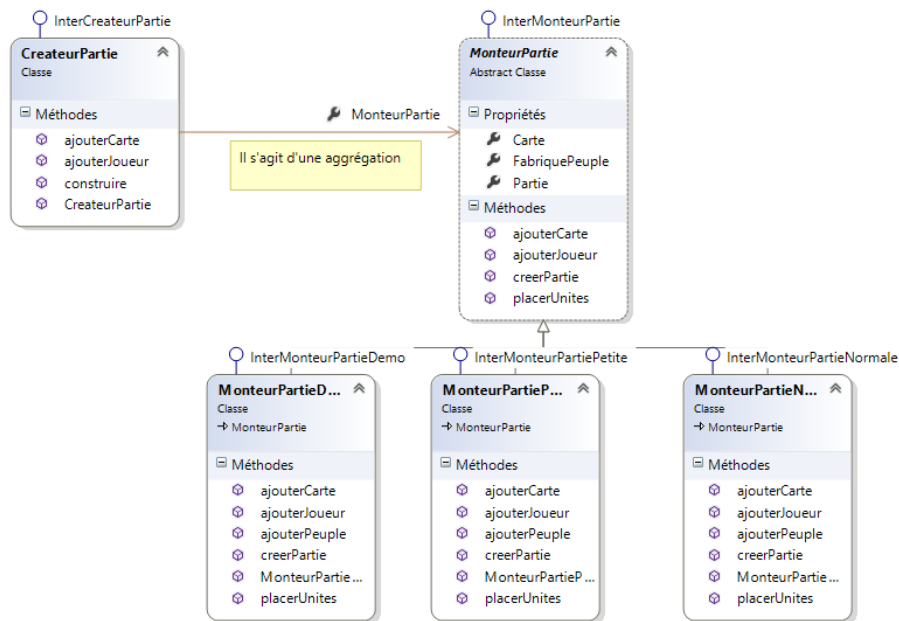


FIGURE 10 – Diagramme de classe du monteur

3.1.2 Stratégie - Création des différents types de cartes

Lors de la création d'une partie, les joueurs ont le choix entre trois tailles de cartes : démo, petite ou normale. Différents algorithmes seront développés pour ces tailles. Afin de modéliser ce changement de comportement, c'est-à-dire de changer d'algorithme en fonction du contexte, le patron de conception utilisé est la *stratégie*. De plus, il sera très facile d'implémenter une nouvelle taille de carte si besoin, simplement par l'ajout d'un nouvel algorithme. La *stratégie* est représentée à la Figure 11.

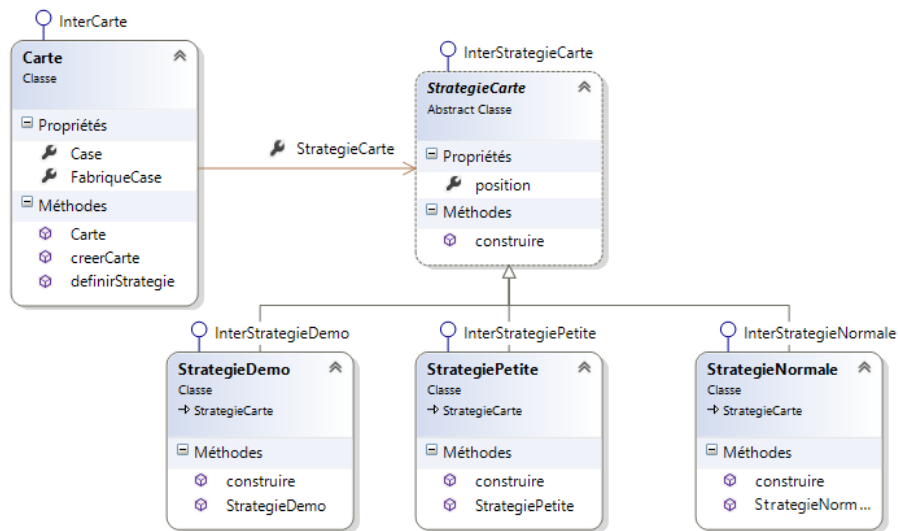


FIGURE 11 – Diagramme de classe de la stratégie

3.1.3 Poids-mouche - Modélisation de la carte

Il existe cinq types de cases : montagne, plaine, désert, eau et forêt. Pour ne pas instancier un grand nombre de case et devenir trop gourmand au niveau de la mémoire, le patron de conception *poids-mouche* est mis en place. Nous évitons ainsi la création répétée d'une même case. La Figure 12 illustre le poids-mouche. Pour obtenir la carte, nous utilisons une *fabrique*. La *fabrique* permet de créer aléatoirement des types des cases différents sur la carte. Son utilisation rend plus facile l'évolution des types de cases par simple ajout de classes concrètes.

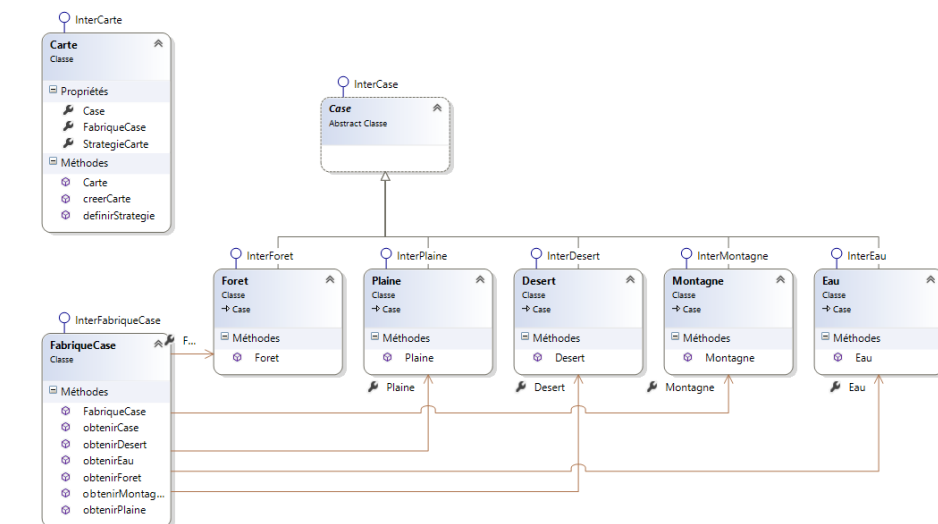


FIGURE 12 – Diagramme de classe du poids mouche

3.1.4 Fabrique - Gestion des peuples et des unités

Lors de la création d'une partie, le joueur choisit entre trois peuples. Le jeu devra donc manipuler les unités sans savoir s'il s'agit de vikings, gaulois ou nains. Le patron de conception *fabrique* est implémenté afin de résoudre ce problème, et de définir une interface pour la création de peuples. En outre, il sera maintenant facile de rajouter un nouveau peuple. Ces éléments sont présentés Figure 13. Les unités étant d'un seul type (il n'y pas d'unité spécialisée au sein d'un peuple), il n'y a pas besoin de *fabrique* pour les unités. On utilise une classe abstraite pour définir les éléments communs aux unités des différents peuples.

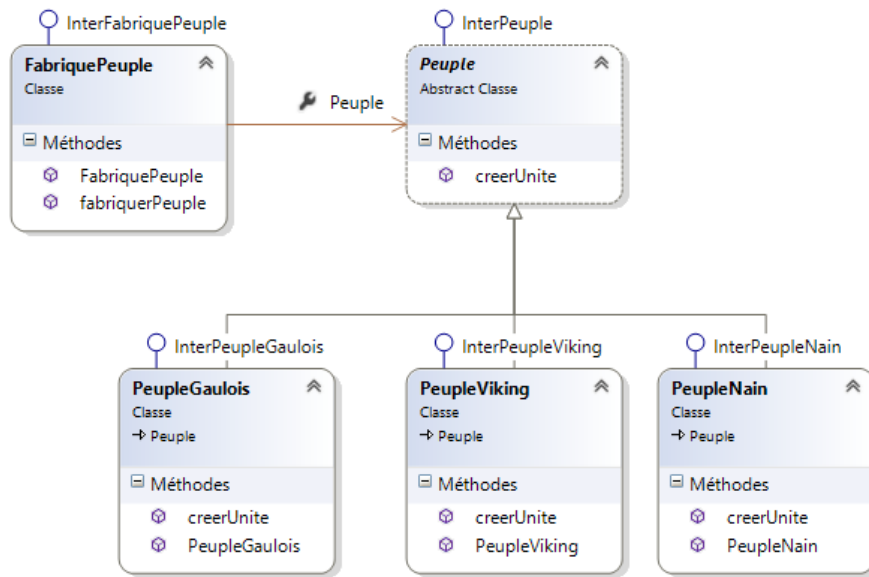


FIGURE 13 – Diagramme de classe de la fabrique

3.1.5 Observateur

Enfin, nous utiliserons un *observateur* afin de créer un lien entre la carte et les unités. En effet, lorsqu'une unité se déplace, il faut que la carte en soit informée afin qu'elle mette à jour l'affichage et les données associés. L'*observateur* permettra à la carte d'observer les unités et donc de créer un lien entre eux. Si une unité se déplace, la carte en sera alors avertie et se mettra à jour. Nous n'avons cependant pas modélisé cette notion, puisqu'en effet, il n'est pas possible d'importer une classe/interface C# dans un diagramme de classes. Nous en tiendrons cependant compte lors du codage.

3.2 Diagramme de classe

Enfin, la Figure 14 montre le diagramme de classes complet de notre jeu. Nous y avons représenté les différentes classes utilisées dans les patrons de conception précédemment cités ainsi que les différentes classes nécessaires au fonctionnement du jeu.

Conclusion

Le projet de conception et de programmation orienté objet a pour objectif de concevoir un jeu inspiré de Small World. La première phase, celle d'analyse et de conception, est donc terminée. Le cahier des charges nous a permis de trouver les différentes fonctionnalités du jeu et ensuite de modéliser les différentes classes nécessaires à son fonctionnement.

La prochaine étape de notre projet consiste donc à implémenter et compléter le code généré automatiquement. Il sera également important de tester les différentes parties du logiciel au fur et à mesure du développement.

Table des figures

1	Diagramme de cas d'utilisation global du jeu Small World	8
2	Diagramme de cas d'utilisation de création d'une partie	9
3	Diagramme de d'activité de création d'une partie	10
4	Diagramme de séquence de création d'une partie	11
5	Diagramme de cas d'utilisation d'un tour de jeu	12
6	Diagramme d'activité d'un tour de jeu	12
7	Diagramme de séquence du déplacement d'une unité	13
8	Diagramme d'activité d'un combat	14
9	Diagramme d'état-transitions	15
10	Diagramme de classe du monteur	16
11	Diagramme de classe de la stratégie	17
12	Diagramme de classe du poids mouche	17
13	Diagramme de classe de la fabrique	18
14	Diagramme de classe complet	19