

INSA de Rennes  
Quatrième année - Informatique

---

Projet d'Analyse, Conception et POO

Rapport de conception

---

Mickaël OLIVIER, Benoit TRAVERS

16 novembre 2013

# Table des matières

<b>1</b>	<b>Cas d'utilisation</b>	<b>4</b>
1.1	Création de partie . . . . .	4
1.2	Tour de jeu . . . . .	5
<b>2</b>	<b>Diagrammes de classe</b>	<b>7</b>
2.1	Modélisation globale du jeu . . . . .	7
2.2	Patrons utilisés . . . . .	7
<b>3</b>	<b>Diagramme d'états-transitions</b>	<b>10</b>
<b>4</b>	<b>Diagrammes d'interaction</b>	<b>11</b>
4.1	Création de partie . . . . .	11
4.2	Déplacer ou attaquer une unité . . . . .	12
	<b>Appendices</b>	<b>14</b>
	<b>Annexe A Diagramme de classes</b>	<b>14</b>
	<b>Annexe B Diagramme de séquence : Création de partie</b>	<b>15</b>
	<b>Annexe C Diagramme de séquence : Déplacer ou attaquer une unité</b>	<b>16</b>

## **Introduction**

Ce rapport marque la fin de la phase d'analyse et de conception, regroupant les différents diagrammes de cas d'utilisation, diagrammes de classe, diagramme d'états-transitions et diagrammes d'interaction.

# 1 Cas d'utilisation

Premièrement, nous retrouvons un diagramme de cas d'utilisation illustrant la globalité du jeu (Figure 1). En se penchant sur ce diagramme, on remarque q'un joueur peut créer une partie ou il peut jouer. Les diagrammes de cas d'utilisation pour la création de la partie et le tour de jeu sont décrit par la suite.

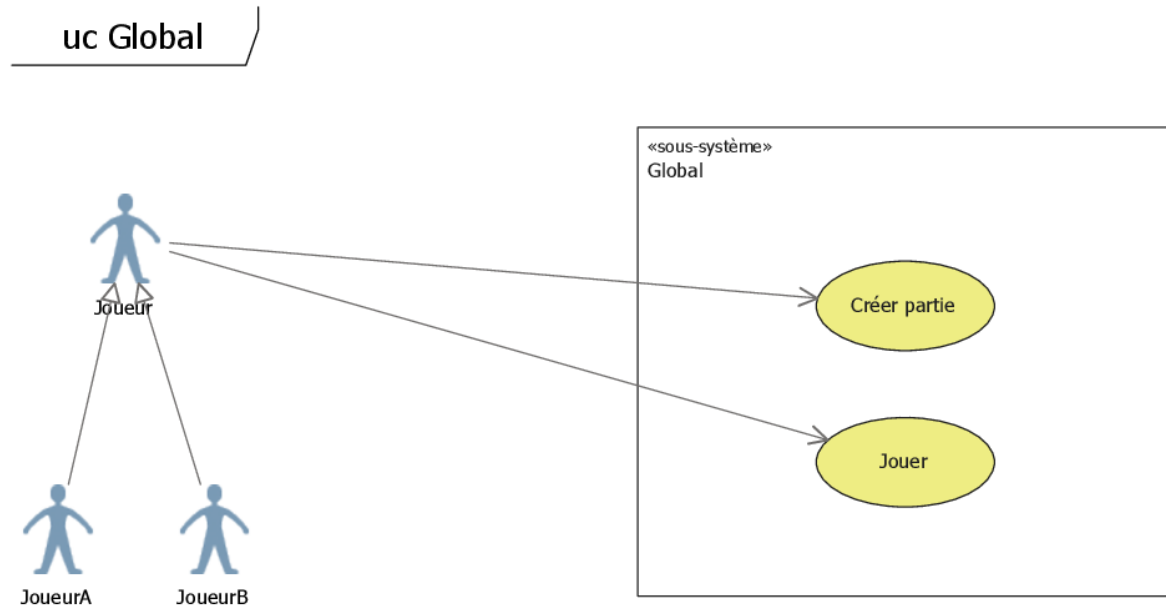


FIGURE 1 – Diagramme de cas d'utilisation global

## 1.1 Création de partie

Avant de pouvoir jouer, il faut créer la partie. Pour cela, le joueur A choisit la carte, à savoir, soit une carte de démonstration, soit une petite carte, soit une carte de taille normale. Ensuite, les joueurs devront choisir leur peuple parmi les Vikings, les Gaulois et les Nains. Ces actions sont illustrées dans le diagramme de cas d'utilisation de la Figure 2.

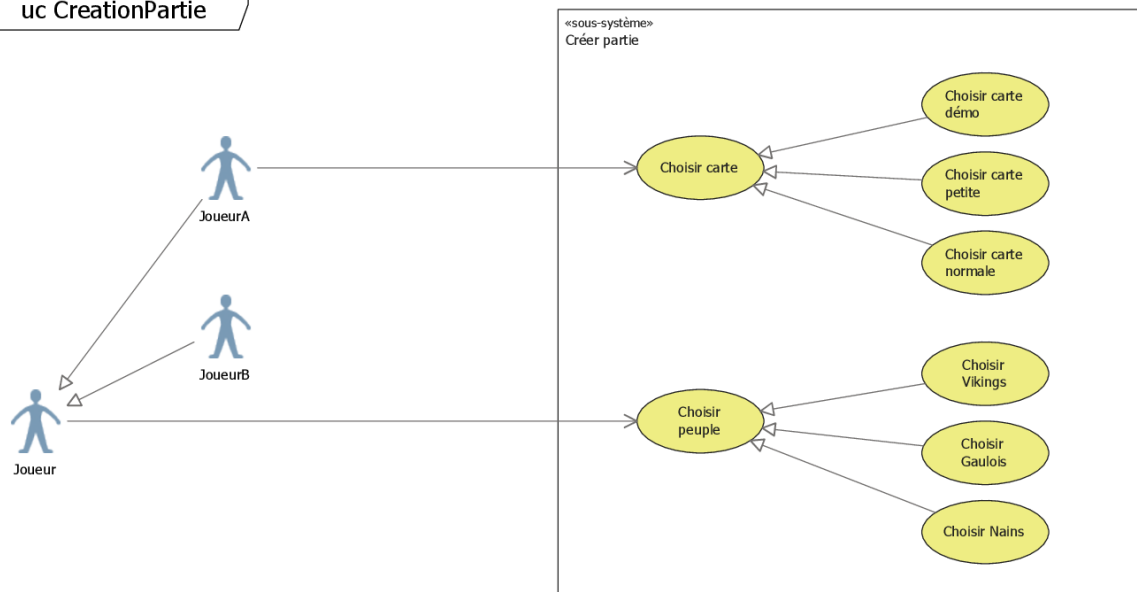


FIGURE 2 – Diagramme de cas d'utilisation : Création de la partie

## 1.2 Tour de jeu

Lors de son tour de jeu, un joueur peut effectuer différentes actions. Il peut déplacer une unité et pour se faire, il doit sélectionner la case avec l'unité qu'il souhaite déplacer, il doit choisir l'unité sur la case sélectionnée (il peut y avoir plusieurs unités d'un même joueur sur une même case) puis il doit sélectionner une case de destination. Un joueur peut aussi attaquer une unité. Pour attaquer une unité, un joueur doit sélectionner la case avec l'unité qu'il souhaite engager au combat, il doit choisir l'unité sur la case sélectionnée puis il doit sélectionner la case de destination avec l'unité adverse qu'il souhaite attaquer. Un joueur peut aussi passer le tour d'une unité en sélectionnant la case avec l'unité puis en choisissant l'unité sur cette case ou il peut simplement passer son tour. Ces actions sont illustrées dans le diagramme de cas d'utilisation de la Figure 3.

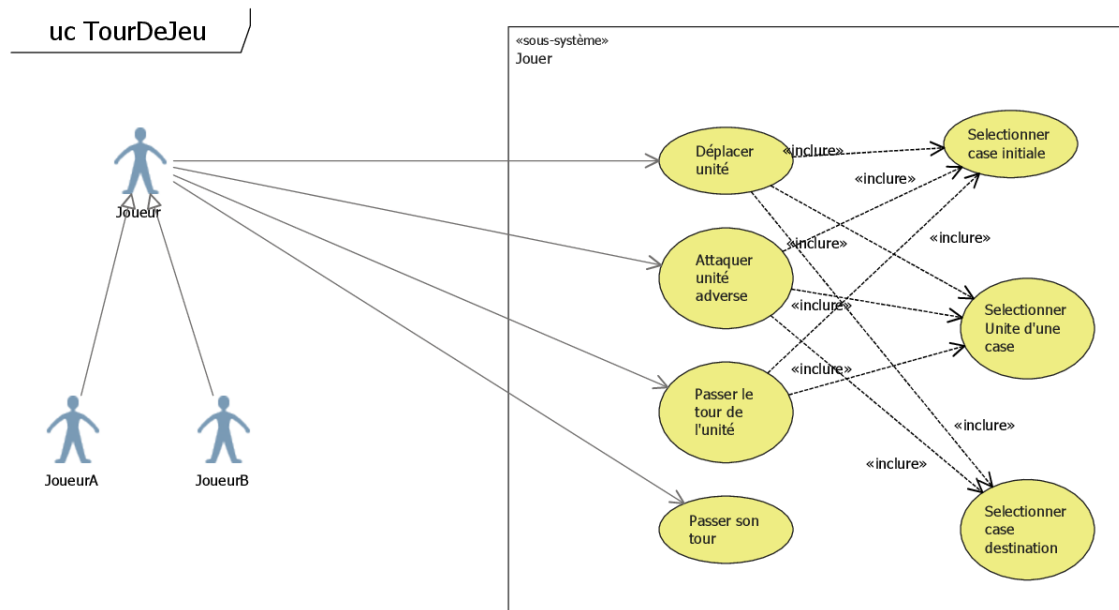


FIGURE 3 – Diagramme de cas d'utilisation : Tour de Jeu

## 2 Diagrammes de classe

### 2.1 Modélisation globale du jeu

En annexe A, on retrouve le diagramme de classe de notre jeu avec les différents patrons utilisés. Par la suite, nous allons détailler les différents patrons utilisés.

### 2.2 Patrons utilisés

Le premier patron utilisé est la fabrique. La classe `Peuple` est considérée comme une fabrique d'Unité.

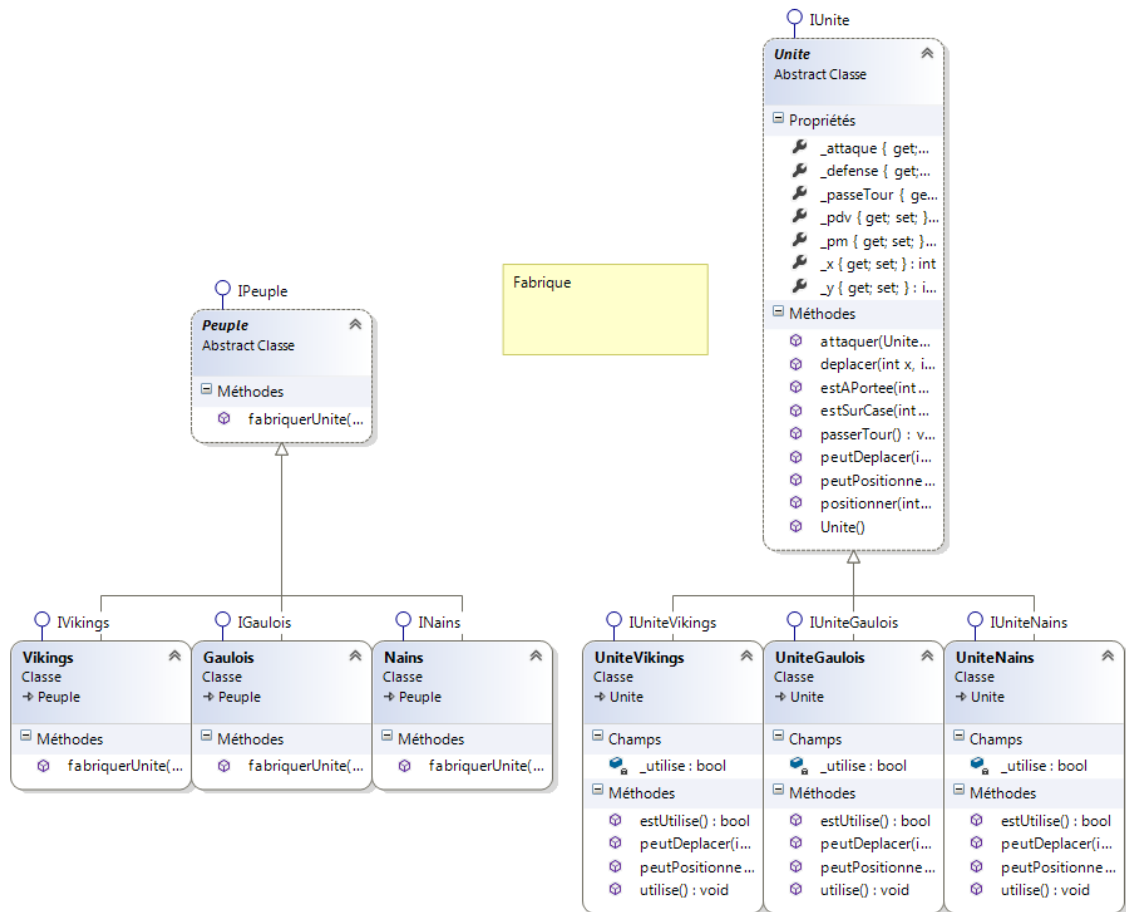


FIGURE 4 – Fabrique

La deuxième patron utilisé est le monteur. Ce patron permet l'assemblage des différents composant de la partie.

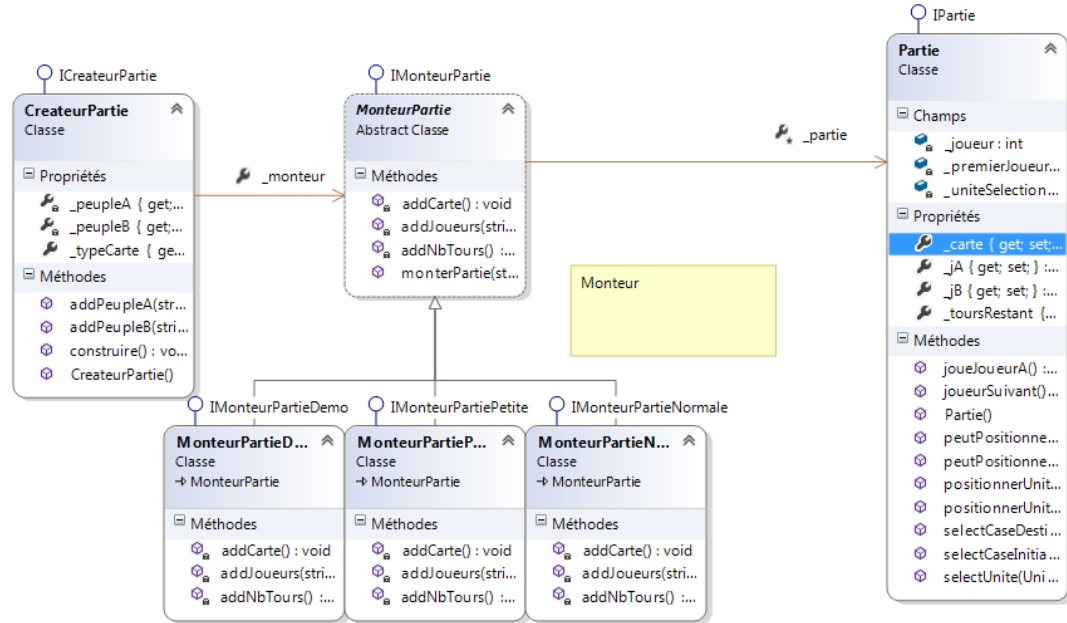


FIGURE 5 – Monteur

Un autre patron utilisé est le poids-mouche. La classe Carte contient un tableau de Case. Cependant chaque case n'est instanciée qu'une seule fois par la classe FabriqueCase dans l'objectif de ne pas consommer trop de mémoire.

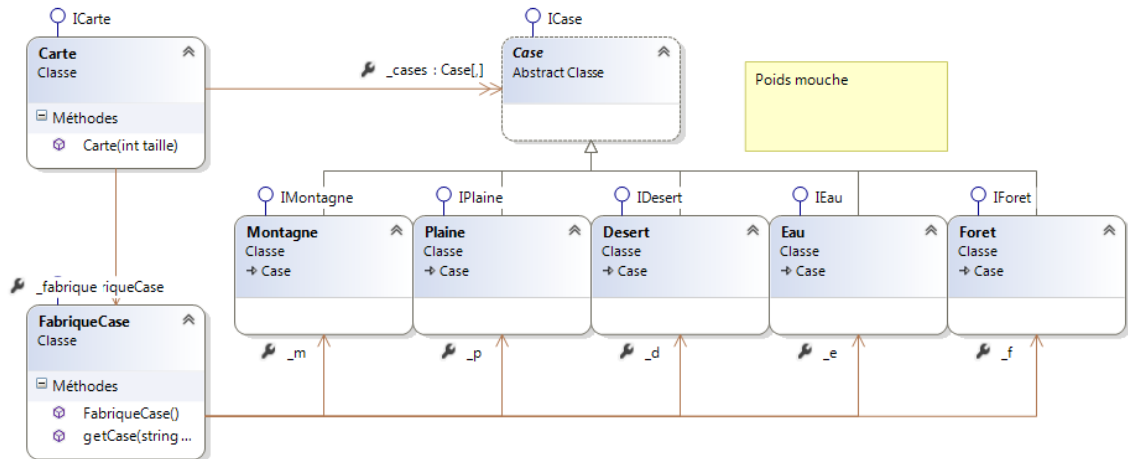


FIGURE 6 – Poids-Mouche

Le dernier patron utilisé est la stratégie. Le créateur carte va faire appel à l'algorithme de création de carte développé en C++. Mais l'appel a cet algorithme sera différente selon le type de carte à utiliser.



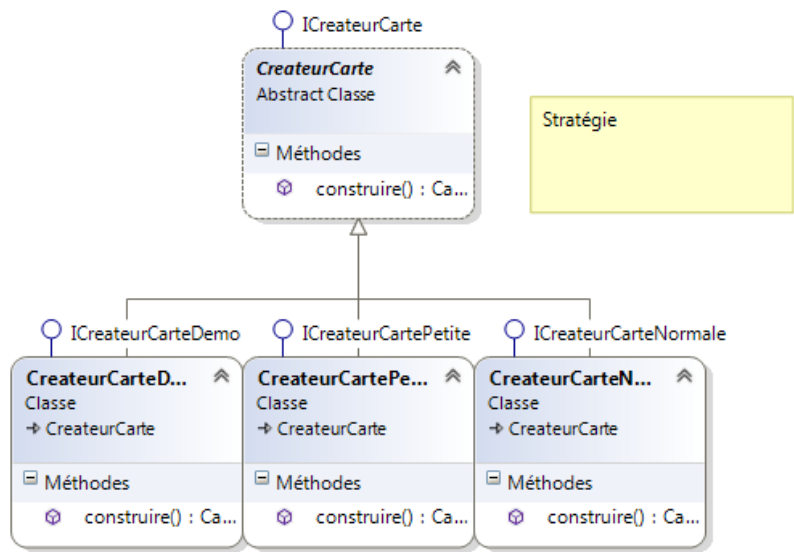


FIGURE 7 – Stratégie

### 3 Diagramme d'états-transitions

Le diagramme d'états-transitions représenté Figure 8 illustre le cycle de vie d'une unité. Pour commencer, une unité est positionnée sur la carte. Puis durant son tour de jeu, elle pourra soit être sélectionnée puis ensuite être déplacée, attaquer ou passer son tour, soit ne rien faire. Lorsque l'adversaire joue, une unité pourra se faire attaquer, dans ce cas, elle rentre dans l'état Défense.

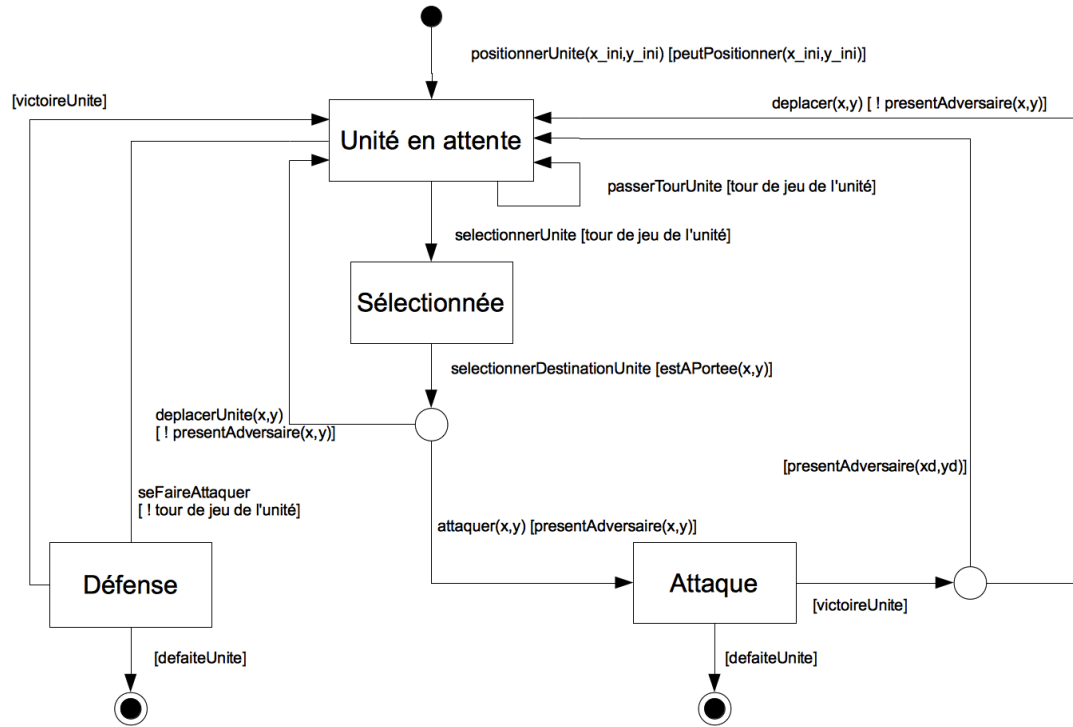


FIGURE 8 – Diagramme d'états-transitions

## 4 Diagrammes d'interaction

### 4.1 Création de partie

En annexe B, on retrouve le diagramme de séquence modélisant la création d'une partie. Pour commencer, le joueur A sélectionne la carte et son peuple, puis le joueur B sélectionne son peuple avant que le joueur A lance la création de la partie. La Figure 9 montre les actions effectuées lors de la sélection du peuple par le joueur A pour que le joueur B ne puisse sélectionner le même peuple.

Puis lors du montage de la partie, trois cas vont se présenter : il faut créer une partie de démonstration, une petite partie ou une partie normale. Pour chaque cas, on va utiliser le monte adéquate puis lancer le processus de création de partie. La Figure 10 montre les actions effectuées pour créer et ajouter la carte.

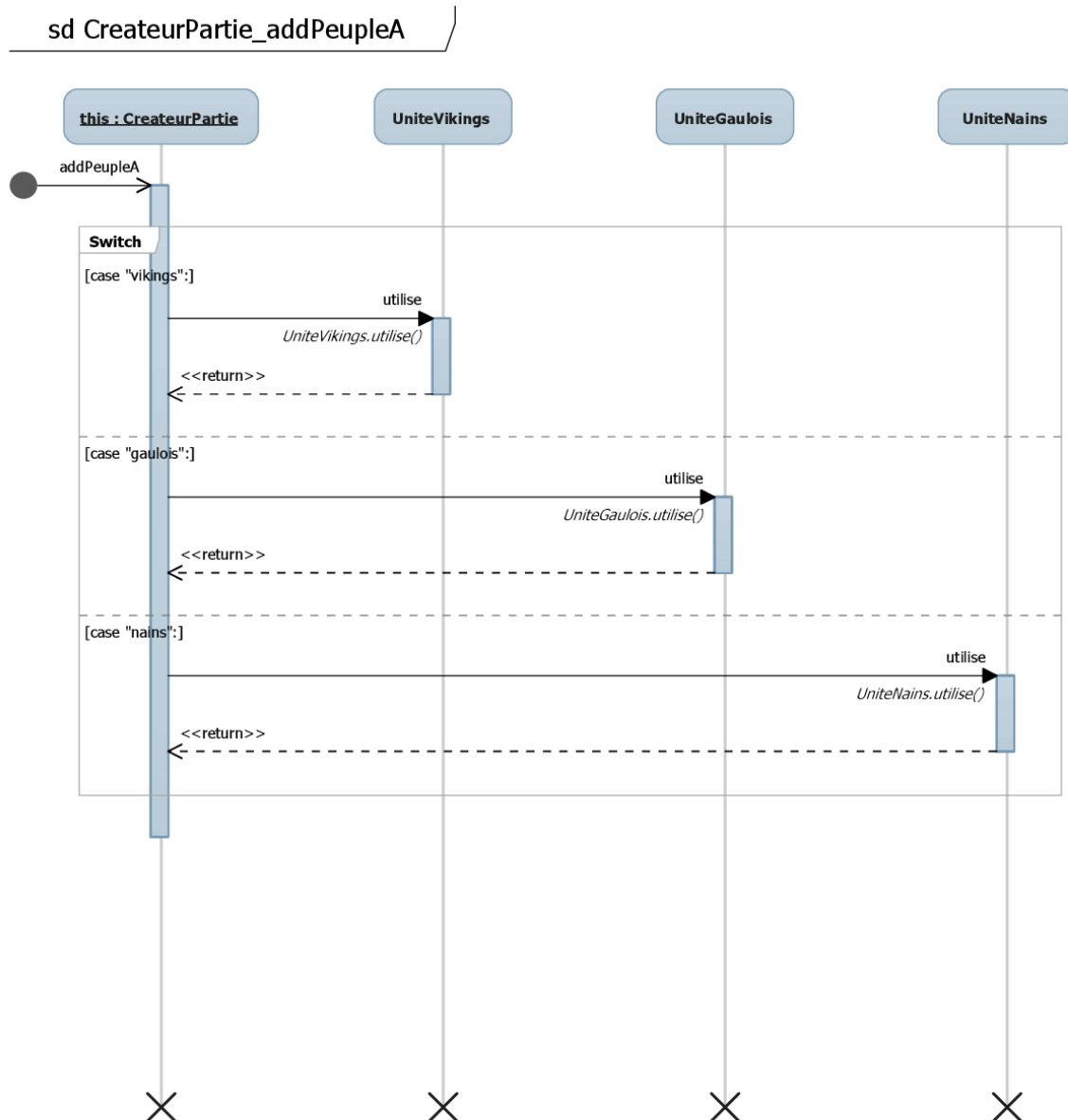


FIGURE 9 – Diagramme de séquence : Ajouter le peuple du joueur A

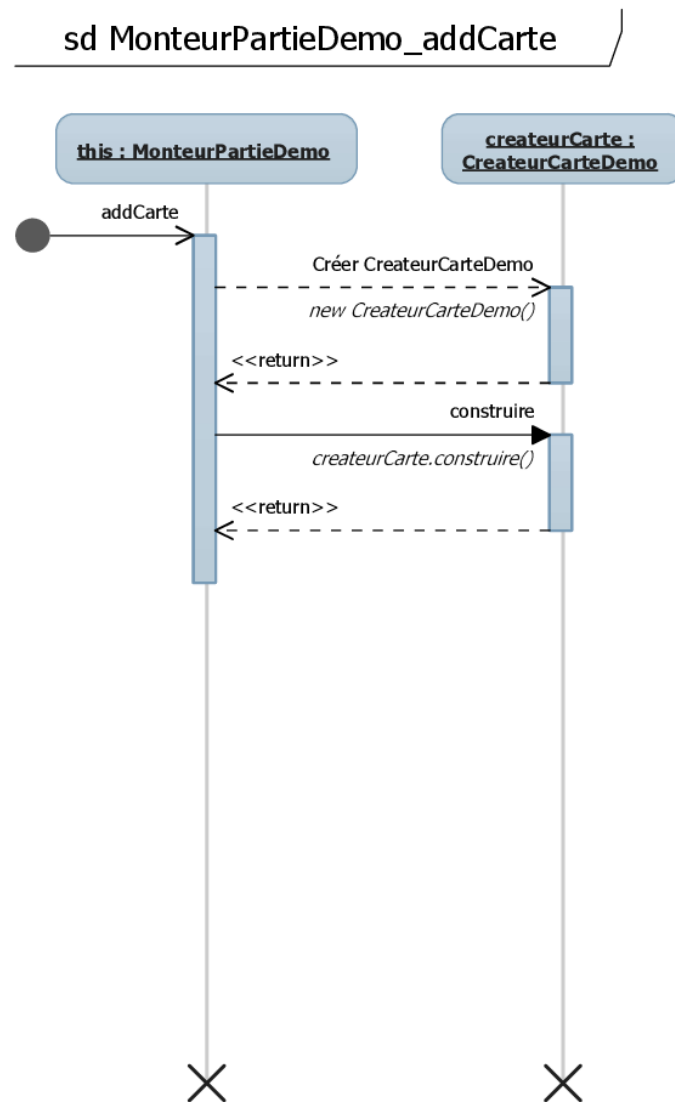


FIGURE 10 – Diagramme de séquence : Ajouter la carte

## 4.2 Déplacer ou attaquer une unité

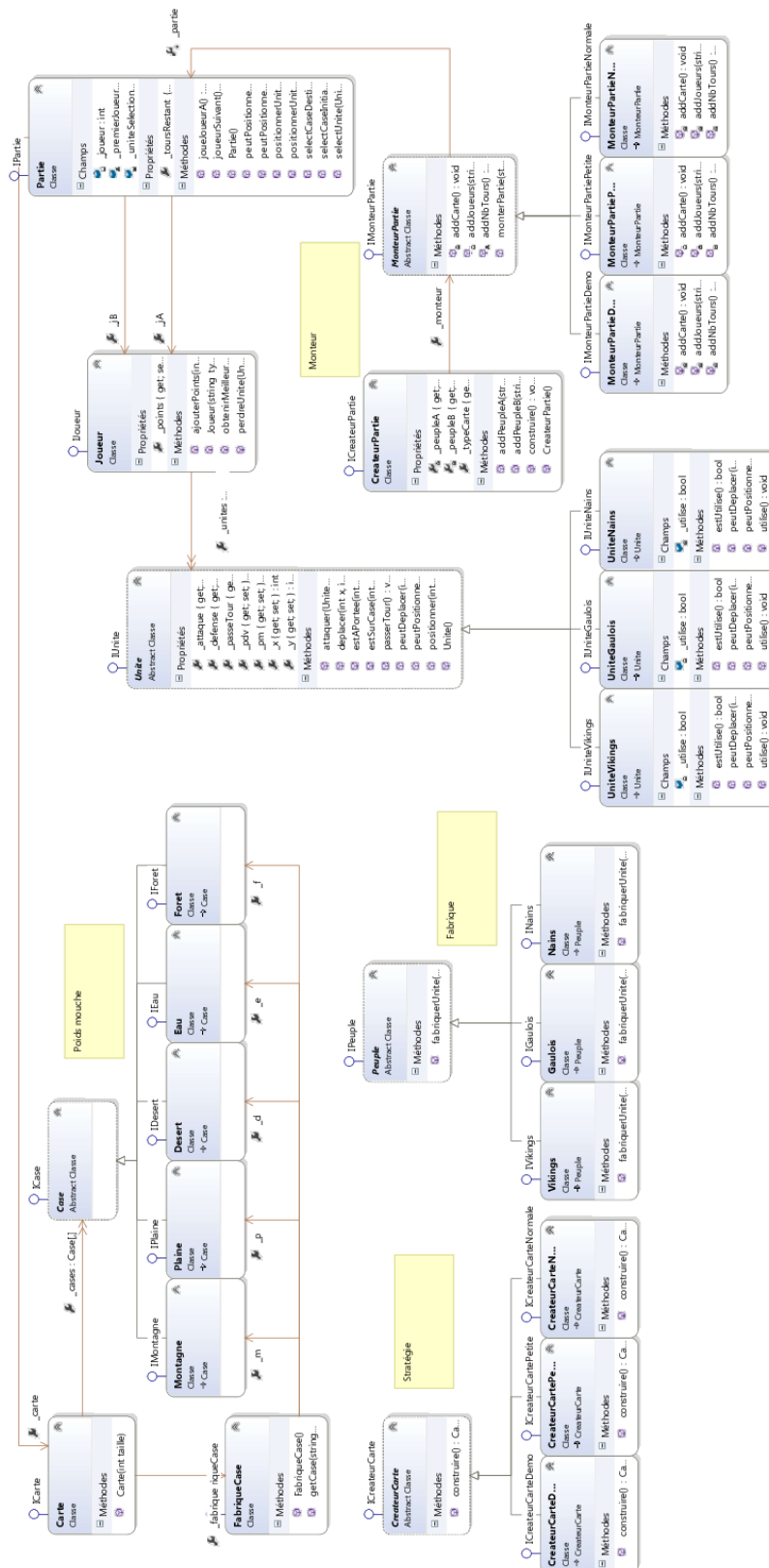
En annexe C, on retrouve le diagramme de séquence modélisant le déplacement ou le lancement d'un combat d'une unité. Pour commencer, le joueur sélectionne une case. Si ce joueur n'a pas d'unité sur cette case, il ne pourra pas poursuivre cette action.

Si, au contraire, le joueur a au moins une unité sur la case sélectionnée, il devra choisir l'unité à sélectionner qui sera conservé par l'attribut `_unitéSélectionnée` de la classe `Partie`. Ensuite, le joueur devra sélectionner une case de destination. On vérifie si cette case est à portée de l'unité sélectionnée, c'est-à-dire si il existe un chemin pour aller jusqu'à cette case sans dépasser le nombre de point de mouvement de l'unité. Ensuite, soit la case de destination sélectionnée n'a pas d'unités ennemies et l'unité sélectionnée va se déplacer sur cette case, soit il y a au moins une unité ennemie et l'unité sélectionnée attaque la meilleure unité ennemie présente sur la case.

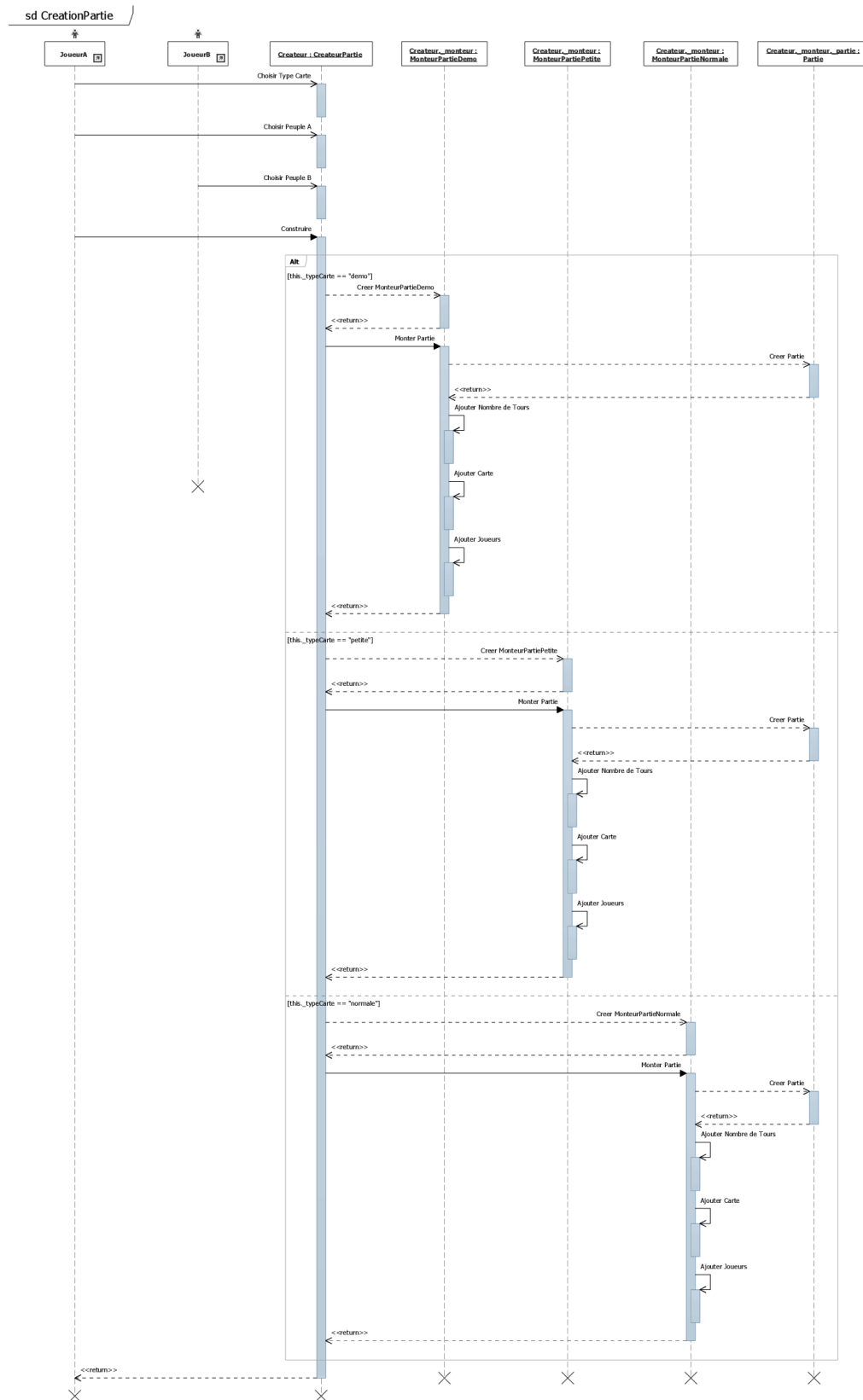
## Conclusion

Ce rapport nous donne une idée de la modélisation de notre jeu. Mais notre modèle n'est pas définitif, il évoluera par la suite avec la phase d'implémentation.

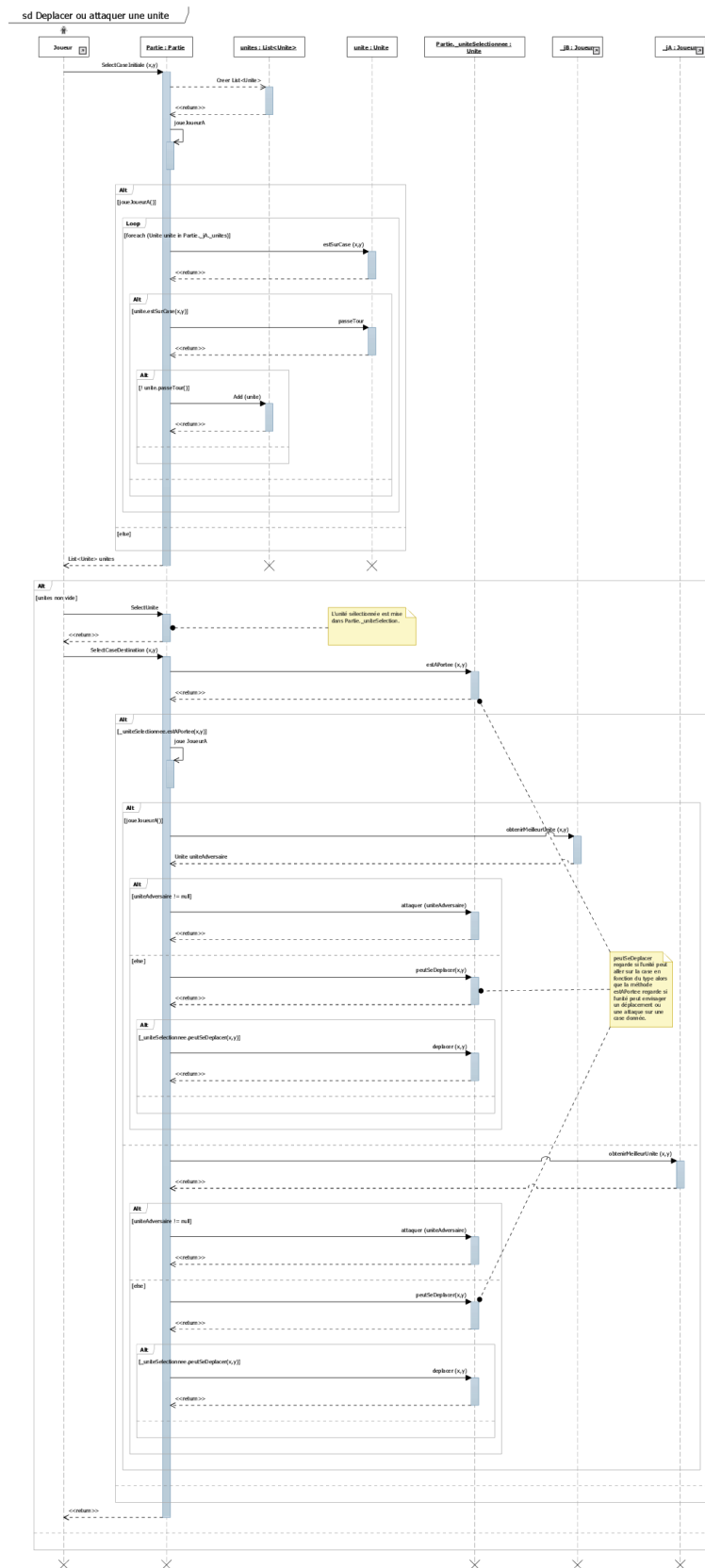
## Annexe A Diagramme de classes



## Annexe B Diagramme de séquence : Création de partie



## Annexe C Diagramme de séquence : Déplacer ou attaquer une unité





## Table des figures

1	Diagramme de cas d'utilisation global . . . . .	4
2	Diagramme de cas d'utilisation : Création de la partie . . . . .	5
3	Diagramme de cas d'utilisation : Tour de Jeu . . . . .	6
4	Fabrique . . . . .	7
5	Monteur . . . . .	8
6	Poids-Mouche . . . . .	8
7	Stratégie . . . . .	9
8	Diagramme d'états-transitions . . . . .	10
9	Diagramme de séquence : Ajouter le peuple du joueur A . . . . .	11
10	Diagramme de séquence : Ajouter la carte . . . . .	12