Rapport de conception

Allainé Hugo – Gachelin Estouan – Py Guillaume

Sommaire

AP4B

Introduction

Ce projet consiste en la création d'un programme en langage Java reprenant les bases du jeu de société « Munchkin ». L'objectif à travers la création de ce programme est de comprendre et d'appréhender la programmation orienté objet en utilisant le langage Java. Nous allons vous détailler nos choix et détails d'implémentation avec des diagrammes UML, schémas et autres explications techniques.

Vue d'ensemble de l'implémentation

Le programme est implémenté selon les diagrammes suivants :

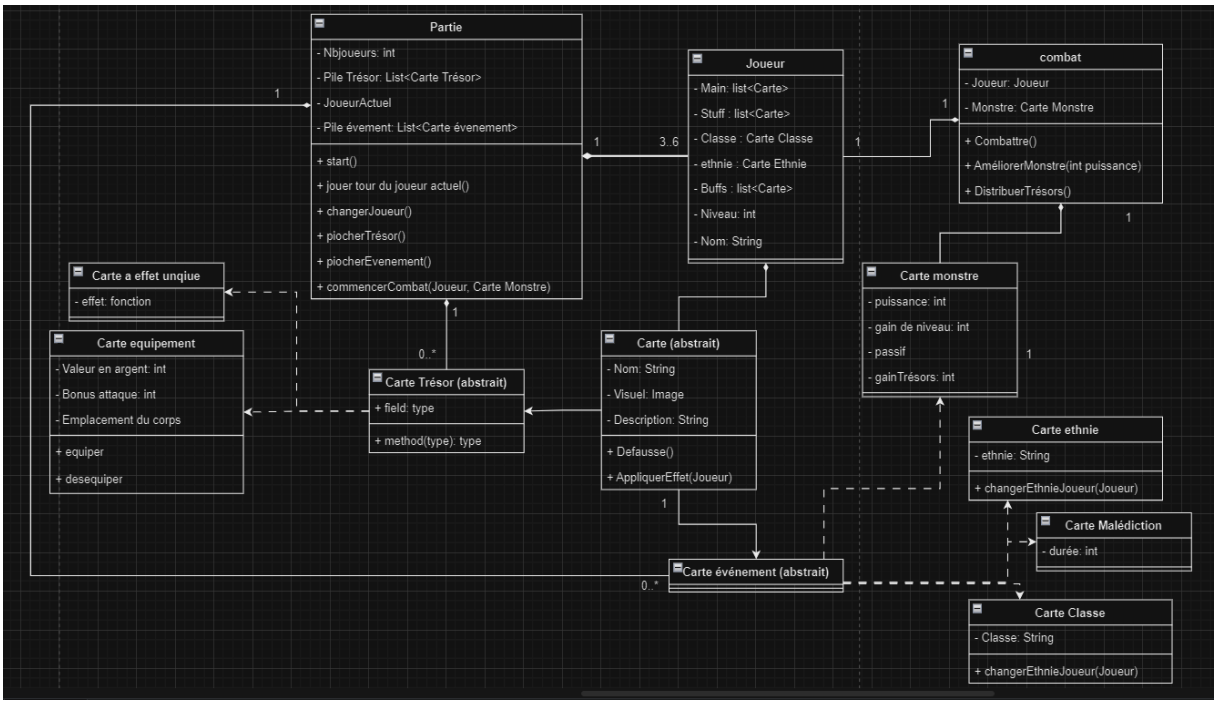

Figure 1 - Diagramme de classe

Listes des classes :

- Partie
- Joueur
- Combat
- Carte
    - Carte Trésor
        - Carte à effet unique

        - Carte équipement
    - Carte événement
        - Carte monstre
        - Carte ethnie
        - Carte malédiction
        - Carte classe

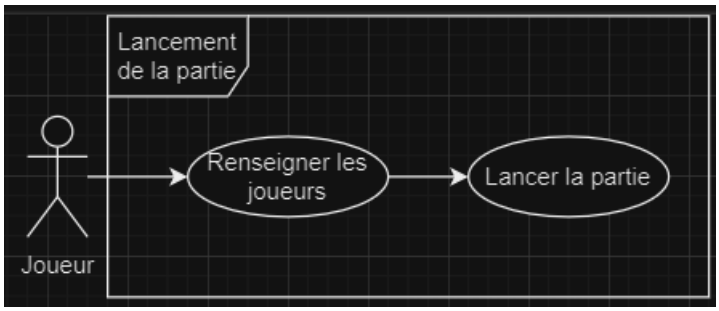

Figure 2 – Diagramme d'utilisation (Menu Principal)

Le diagramme précédent explique les choix de l'utilisateur au moment de lancer le programme. Il peut renseigner les joueurs et ensuite lancer la partie.

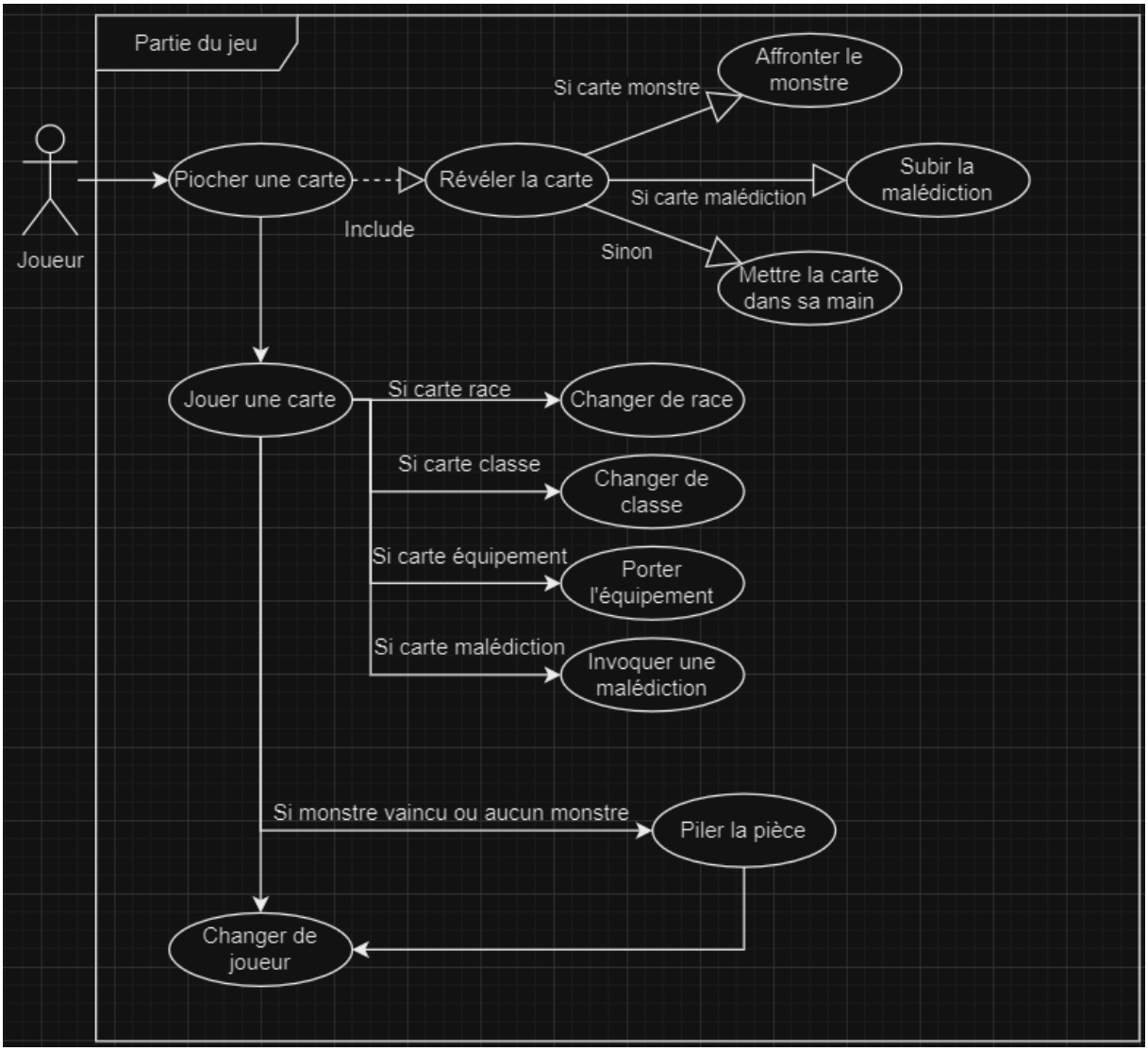

Figure 3 – Diagramme d'utilisation (Exemple des actions d'un joueur)

Comme le montre ce diagramme, un joueur doit piocher une carte au début de son tour puis en fonction de la carte piochée, il pourra faire différentes actions. Si la carte est une carte monstre, il doit le combattre. Si la carte est une carte malédiction, il subit la malédiction. Sinon, la carte est ajoutée à la main du joueur. Il peut ensuite jouer des cartes de sa main. Si la carte piochée était un monstre et qu'il l'a battu, le joueur peut piller la pièce (piocher un nombre de carte associé à la récompense du monstre).

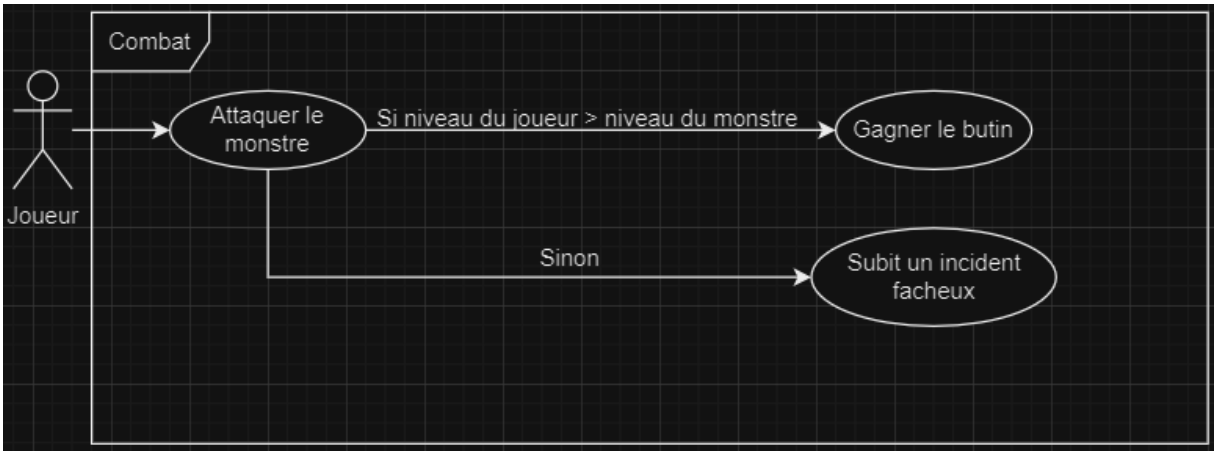

Figure 4 – Diagramme d'utilisation (Combat)

Ce diagramme montre le déroulé d'un combat lors du tirage d'une carte monstre.

Modules et Fonctionnalités

Nous avons fait le choix d'implémenter les classes précédentes dans un dossier distinct « jeu » et d'implémenter en parallèle les classes dédiées à l'interface graphique dans un second dossier « gui » afin de bien séparer chaque partie du code de façon efficace.

L'interface graphique est développée à l'aide de la bibliothèque Swing.

L'avantage du langage Java qui est la programmation orienté objet a permis une meilleure modularité pour l'organisation du code en différents fichiers, simplifiant le développement du programme et la réutilisation des différentes classes.

Problèmes rencontrés

Les problèmes majoritaires rencontrés sont :

- Comment associer différents effets au même type de cartes, par exemple la carte malédiction peut avoir plusieurs types d'effets différents. Pour cela nous avons fait le choix de créer une interface fonctionnelle permettant ainsi d'assigner différentes fonctions d'effet en rapport avec le type de la carte.
- Comment organiser l'interface pour avoir un jeu facilement compréhensible et "modulaire". Pour cela, l'interface est composée de différents menus contenant les composants de l'interface tels que les boutons, les informations du joueur, etc.

- Quel type de fichier choisir pour la configuration du jeu ? Il existe différents types de fichiers pour sauvegarder la configuration du programme (CSV, XML, JSON, etc…) et nous ne sommes pas encore fixé sur le type définitif.

Conclusion

Ce projet nous a permis de nous familiariser avec la conception de diagramme UML ainsi que la réalisation d'un programme en langage Java avec interface graphique. Nous avons aussi amélioré notre capacité à travailler en équipe et notre maitrise d'outil de versionnage come Git. L'utilisation de diagramme UML nous a aidé à partager les différentes tâches de développement du programme et à gagner du temps dans le début de l'écriture du code.