AP4B

Rapport de projet

Allainé Hugo – Gachelin Estouan – Py Guillaume

# 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 :

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

*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

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

*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.

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

*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).

Une image contenant texte, capture d’écran, Police, diagramme

Description générée automatiquement

*Figure 4 – Diagramme d’utilisation (Combat)*

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

# Modules et Fonctionnalités

Parler de partie backend / partie interface

(Je ferai ça plus tard)

# 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.

# Conclusion