

# Programmation Orientée Objet

## Mini-projet

Olivier Gérard

21 Mai 2024

Dans ce mini-projet, vous devrez appliquer les notions vues au cours de ce semestre dans le domaine de la Programmation Orientée Objet sur les différents exercices proposés ci-dessous. Quatre exercices sont proposés, et pour cet examen, vous devrez choisir et répondre à deux d'entre eux. Le but est de créer les différentes classes et relations permettant de répondre au mieux au sujet proposé. Si certaines parties des sujets nécessitent des précisions, n'hésitez pas à demander des clarifications ou des conseils à votre enseignant.

Le projet est à réaliser en binôme ou trinôme selon la composition des groupes. Vous porterez encore une fois une attention toute particulière à la propreté et à la qualité du code rendu, qui devra être clair, commenté et compréhensible par quelqu'un d'extérieur à votre groupe. De même, le diagramme UML devra être le plus clair possible, de préférence réalisé avec un logiciel type Draw.io. Vous serez notés sur plusieurs aspects : le code et le diagramme, et dans un second temps sur un oral où vous serez amenés à présenter rapidement vos applications, à expliquer certains de vos choix ou certaines portions de code.

### Exercice 1

Dans cet exercice, vous devrez créer une application console simulant au mieux un jeu de cartes où deux joueurs s'affrontent en s'opposant différentes créatures jusqu'à ce qu'un joueur n'en possède plus aucune sur son terrain de jeu. Chaque carte de créature précise son nom, son nombre de points de vie, les différentes attaques qu'elle peut effectuer et éventuellement un talent qu'elle possède, et enfin son type. Ce type va déterminer les modificateurs appliqués sur les attaques reçues, certains étant plus efficaces sur d'autres et infligeant donc plus de dégâts. D'autres types de carte existent : des cartes Objets pouvant être associées à des créatures pour leur accorder des bénéfices divers et variés, des cartes Énergie nécessaires au lancement de certaines attaques, et les cartes créatures elles-mêmes. Chaque joueur dispose d'un ensemble de cartes, et en pioche un certain nombre au lancement de la partie. Il choisit une créature active, et d'autres qui seront inactives sur son terrain. Une partie se déroule au tour par tour, et se termine lorsqu'un joueur ne possède plus aucune créature.

Vous devrez créer les classes et leurs relations pour simuler le plus fidèlement possible le déroulement d'une partie. Certaines règles n'ont pas été définies ici, vous êtes libres d'inventer les vôtres ou de vous inspirer de jeux existants.

### Exercice 2

Pour ce second exercice, vous devrez également créer une application console pour représenter le jeu du Blackjack. Dans ce jeu, que l'on retrouve souvent dans les casinos, les joueurs affrontent la banque et cherchent à battre le score des cartes de la main du croupier, sans dépasser le score de 21. Chaque carte a une valeur selon son type : les cartes de 2 à 9 ont la valeur nominale de la carte, les « têtes » valent toutes 10 points et l'as vaut soit 1 point soit 10 points, selon le choix du joueur. Avant la partie, la banque fixe les mises minimales qu'un joueur doit parier pour pouvoir jouer. Au début d'une partie, le croupier distribue une carte face visible à chaque joueur ainsi qu'à lui-même, puis une seconde carte visible pour les joueurs et une carte face cachée pour lui. Ensuite, chaque joueur décide de ce qu'il veut faire : observer les cartes du croupier, demander une nouvelle carte ou conserver dont il dispose et attendre la révélation de la carte cachée du croupier. Il gagne si il parvient à battre le croupier sans que ses cartes dépassent le score de 21.

Ici aussi, vous devrez créer les différentes classes permettant de simuler une ou plusieurs parties. Certaines règles ne sont pas détaillées ici, mais vous trouverez facilement des renseignements sur tous les aspects du Blackjack. Néanmoins, vous pouvez vous limiter aux règles de mises les plus simples.

## Exercice 3

Le but de ce troisième exercice est de concevoir une application permettant de gérer la collection de livres d'une bibliothèque. Chaque livre ou document mis à disposition par l'établissement possède plusieurs caractéristiques : son titre, l'ISBN, une catégorie ou éventuellement un type, qui changerait les conditions d'emprunt / de consultation du document. Les utilisateurs sont également présents dans le système, où on peut y retrouver certaines informations les concernant, par exemple les livres qu'ils ont empruntés ou un historique. Certains types d'utilisateurs peuvent également avoir des conditions d'emprunt plus avantageuses. Vous devez proposer différentes visualisations des livres / documents de la bibliothèque : lesquels sont actuellement empruntés, avec quel retard, et par quel utilisateur, par exemple.

Dans un premier temps, vous devez vous concentrer sur le fonctionnement de base d'un tel établissement. Pour complexifier votre application, vous pouvez ajouter la possibilité de générer des rapports (des fichiers) résumant les emprunts de livres et les différentes factures liées aux emprunts, ou encore ajouter la possibilité pour les utilisateurs de réserver un livre qui n'est pas encore ou plus disponible.

## Exercice 4

Dans ce dernier exercice, il vous est proposé de créer une application console permettant aux utilisateurs de chiffrer et de déchiffrer des messages. Les utilisateurs peuvent soumettre un texte à chiffrer, que ce soit directement par le biais de l'application ou en soumettant un fichier. L'application devra proposer au moins deux méthodes de chiffrement, qui peuvent être relativement simple, par exemple : le chiffrement de César. L'application devra retourner le texte chiffré soit directement soit dans un fichier, en fournissant également le moyen de déchiffrer le message.

Pour cet exercice, vous êtes libres de proposer des fonctionnalités supplémentaires, comme des moyens de déchiffrer un message ou des méthodes de chiffrement supplémentaire. Le choix de ces méthodes est laissé à votre discrétion, vous pouvez vous tourner vers des chiffrements simples ou n'importe quelle méthode qui vous intéresse et que vous souhaitez étudier et appliquer.