

Rapport Pokedeck - Java

Deck
deck (ArrayList<Card>)
addCard() getCard() updateCard() removeCard()

Card
name (String) lifePoints (String) type (String) description (String)
getName() getLifePoints() getType() getDescription() setName() setLifePoints() setType() setDescription()

Afin de mettre en avant les différentes fonctionnalités du projet, j'ai opté pour une interface assez simple. Les boutons permettant le déclenchement d'actions par l'utilisateur servent de «menu» et sont regroupés donc en haut de la fenêtre. En dessous on retrouve une zone qui contient à gauche la carte pokémon et une liste des cartes présentes dans le deck créé par l'utilisateur. Ces deux parties seront développées ultérieurement dans le rapport.

Dans mon cas, la carte aura une structure similaire à une carte pokémon classique. On retrouve ainsi le nom de la carte pokémon sur la première ligne ainsi que le nombre de points de vie et le type sur la deuxième en ligne de la carte respectivement à gauche et à droite.

L'une des fonctionnalités est de pouvoir ajouter la description à une nouvelle carte. Cette fonctionnalité a été développée par le biais d'un formulaire qui lorsque l'on clique sur le bouton «Add card» de celui-ci ajoute une nouvelle carte dans la liste des cartes existantes dans le deck. Néanmoins, une carte ne possède pas seulement, comme information, une description. En effet, elle contient également un nom, le type du pokémon ainsi que ses points de vie. Le formulaire permet donc également de préciser ces informations lors de la création d'une nouvelle carte.

La fonctionnalité permettant de modifier une carte est très ressemblante à celle permettant d'en créer une. En effet, ces deux fonctionnalités possèdent le même formulaire et la seule différence réside dans le fait que le formulaire pour la modification d'une carte est déjà rempli avec les modifications de celle-ci.

Une autre fonctionnalité est de pouvoir supprimer une carte du deck de l'utilisateur. Celle-ci se déclenche lors du clic sur le bouton «Remove card». Contrairement, aux deux précédentes fonctionnalités énoncées, cette fonctionnalité n'ouvre pas de nouvelle fenêtre pour réaliser cette action. Elle supprime directement la carte du deck dont les informations sont affichés à l'écran.

Il était également demandé de permettre à l'utilisateur de consulter sa collection de cartes. L'idée de départ que j'avais était de partir sur une liste montrant l'ensemble des cartes contenues dans le deck. Néanmoins, je n'ai pas réussi à créer d'interactions à partir de celle-ci, j'ai donc choisi finalement d'utiliser, pour la navigation entre les cartes, des boutons permettant de changer le contenu de la carte actuellement affichée. Les boutons permettent donc de changer le contenu affiché par celui de la carte placée soit juste avant, soit juste après celle en cours de visionnage.

Si je devais faire un bilan sur ce que m'a apporté le projet, je dirais qu'il m'a permis de mieux comprendre les bases de la programmation orientée objet alors qu'auparavant c'était le naufrage complet. En effet, j'ai réussi à développer des fonctionnalités nécessaires au fonctionnement même si ces dernières ne sont pas parfaites. Je suis donc satisfaite des progrès que j'ai pu réaliser. De plus, j'ai également réussi à avoir une vision d'ensemble de la structure du projet ce qui fait qu'à l'ajout d'une nouvelle fonctionnalité, je ne devais pas modifier l'entièreté de mon code mais seulement ajouter de nouvelles méthodes.

Néanmoins, les fonctionnalités comme la sauvegarde de fichier et l'utilisation du json sont encore trop complexes pour que je parvienne à développer quelque chose de fonctionnel.