

Rapport de Projet - JAVA

Projet dont vous êtes le Héros



Projet réalisé par

Léo Rams

Alexandre You

Table des matières

I. La structure du code	3
II. Extensions utilisées.....	5
III. La difficulté du projet	6
IV. Bilan.....	7

I. La structure du code

La fonction main se trouve dans la classe Application, elle-même dans le package appli, son rôle est simplement de lancer un nouveau « Jeu ».

Le jeu a pour rôle de gérer tout ce qui est nécessaire au fonctionnement de l'application et de faire le lien entre toutes les autres classes. Lorsqu'un nouveau Jeu est créé, il va se connecter à la base de données, puis créer les interfaces graphiques et enfin se lancer.

Le lancement du jeu ne fait en réalité qu'ouvrir la première interface « Menu », et ce sont les boutons associés à chaque interface qui exécuteront des fonctions de la classe « Jeu » pour faire évoluer la partie.

Lorsque le joueur commence une partie, que ce soit en reprenant une sauvegarde ou en commençant une toute nouvelle, la première chose qui sera faite sera de créer le graphe contenant toutes les possibilités du scénario à partir des informations données par la base de données.

Une fonction récursive va alors initialiser chaque « Node » et réitérer le processus sur chacun de ses descendants, jusqu'à n'avoir que des nœuds finaux (« TerminalNode »).

Chaque « InnerNode » pouvant être un « DecisionNode » ou un « ChanceNode » aura alors une liste de successeurs ainsi que les options pour s'y rendre (du texte pour DecisionNode et des taux pour ChanceNode).

Chaque TerminalNode aura un « Personnage », qui contient toutes les informations liées à ce nœud final, et qui permettra donc d'afficher un résumé des performances du joueur une fois la partie terminée.

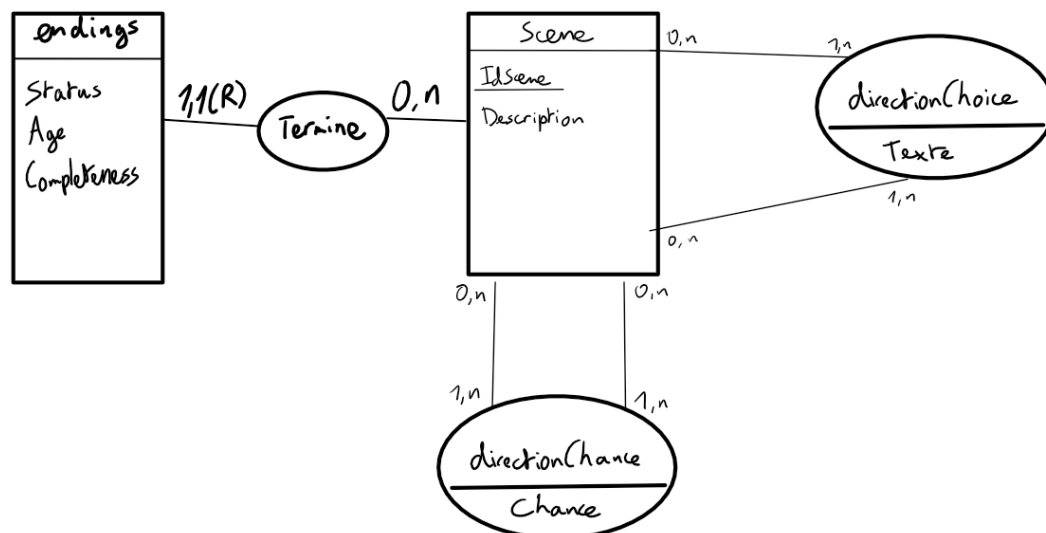
Les Interfaces héritent toutes de « IHM » qui elle-même hérite de « JFrame », une classe de l'extension Swing. C'est cette extension Swing qui a permis de faire l'ensemble des interfaces. Ces interfaces interagissent entre elles via la classe « Jeu », en appelant des fonctions de cette même classe qui vont ensuite ouvrir/fermer certaines interfaces en fonction du besoin.

L'ensemble des interfaces se trouve dans le package « ihm », il y a en tout 4 types d'interface :

- Menu, la première interface qui s'ouvre et qui permet de commencer une nouvelle partie en ouvrant « IHMJeu » ou d'en charger une avec « Saves »
- Saves, qui lorsqu'elle est ouverte va présenter au joueur 3 fichiers de sauvegarde allant de la plus récente (1) à la plus ancienne (3).
- IHMJeu, qui contient l'interface principale du jeu et permet donc au joueur d'effectuer une des actions proposées par le jeu à chaque tour, jusqu'à la fin, ainsi qu'à sauvegarder sa partie quand il le souhaite.
- Close, qui s'ouvre lorsque le joueur clique dans IHMJeu sur le bouton « Quitter » et qui va alors demander au joueur s'il souhaite sauvegarder ou non, avant de tout fermer.

Enfin, deux classes ont été testées, TerminalNode et Personnage, car elles étaient celles qui étaient le plus adaptées aux tests unitaires, ne demandant pas de lancer une interface graphique, ce qui permet ainsi de vérifier le plus important automatiquement, sans devoir jouer.

Pour ce qui est de la structure de la base de données (qui est aussi montrée dans le fichier ReadMe), elle a été construite comme suit :



II. Extensions utilisées

La seule extension utilisée a été Swing, qui a permis de réaliser toutes les interfaces graphiques du projet. Swing est décrit par [Wikipédia](#) comme « une bibliothèque graphique pour le langage de programmation Java, faisant partie du package Java Foundation Classes (JFC), inclus dans J2SE. »

Il est important aussi de mentionner que pour l'utilisation de Swing, [StackOverflow](#) ainsi que [ChatGPT](#) ont été d'une grande aide, le premier en donnant des solutions à des problèmes courants (Correctement placer chaque élément) et le second pour donner la syntaxe à utiliser pour faire fonctionner le tout.

III. La difficulté du projet

Les points suivant ont engendré des difficultés :

- Un point à souligner est la complexité de la création du scénario. Notre projet est énormément basé sur l'histoire, le contexte et les choix que l'on prend en tant que héros. Ecrire une histoire cohérente a été très difficile, on a donc dû piocher nos inspirations de diverses sources, réelles et fictives, en plus des idées apportées par notre imagination.
- Il a été difficile d'établir les meilleurs outils pour réaliser ce projet. Pour ce qui est du stockage des données, nous voulions d'abord utiliser un ou plusieurs fichiers json qui seraient lus grâce à du code java, avant de décider d'utiliser le langage SQL avec mySQL Workbench, qui apportait une plus grande liberté sur la sélection de données et rendait la saisie des informations plus facile. Pour ce qui est de l'interface, nous voulions d'abord réaliser le projet sur Android Studio pour en faire une réelle application, avant de nous raviser et de finalement utiliser Swing, ce qui en plus de rendre l'application utilisable sur un ordinateur, facilitait aussi les tests de ladite application.
- L'interface swing a été particulièrement difficile à réaliser. L'application a vite été utilisable, mais il a été plus long de la rendre présentable, car faire de l'interface un ensemble bien organisé et sans dysfonctionnements (Ne pas pouvoir appuyer sur « commencer » alors qu'une autre fenêtre est ouverte par exemple) a demandé beaucoup de tests et une bonne compréhension du fonctionnement de Swing.

Globalement ce projet n'aurait pas dû être bien difficile à réaliser, mais c'est le choix des outils qui a compliqué sa réalisation. L'objectif était de faire du code le plus propre possible (utiliser une base de données pour les nœuds au lieu de tout mettre en brut sur une classe Java) et rendre l'expérience de l'utilisation aussi agréable et facile qu'il se peut (utilisation d'une interface graphique au lieu de la ligne de commande).

IV. Bilan

Ce projet a globalement été enrichissant, en faisant appel en conséquence de nos choix d'outils à de nombreuses notions. La partie code en java a été plutôt simple à réaliser au départ, ne nécessitant que de relier des objets entre eux, mais c'est lors de l'implémentation de la base de données et de l'interface graphique que tout s'est compliqué. La fonctionnalité et la modifiabilité du code nous font penser que ces choix étaient les bons cependant.

Quant à la réalisation du scénario, elle a été plus longue que prévu dû au fait que l'évolution du nombre de nœuds à chaque ajout est exponentielle. Il a donc été décidé que plusieurs nœuds pourraient rediriger aux mêmes, ce qui a permis d'offrir un scénario plus long.