



UNIVERSITÉ
DE LORRAINE

IUT Metz
Informatique

Département Informatique
Promotion 2019-2021

Projet de programmation répartie en RMI



Projet soutenu le 31 mars 2021

Membres du Groupe

Maxime JOURNET

Robin FROLIGER

Angéla RAVELLI

Projet effectué au département informatique de l'IUT de Metz.

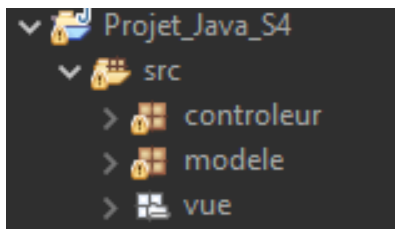
I Sujet et jeux choisis

Ce projet concerne le développement d'une application Java en RMI (Remote Methode Invocation). Nous devons réaliser un serveur de jeux en réseau: pour cela, nous avons opté pour la bibliothèque JavaFX, qui permet de produire des interfaces graphiques de façon beaucoup plus simple qu'avec la bibliothèque Swing.

Nous avons le choix entre 4 jeux proposés et d'autres jeux additionnels, pour un total de 3 jeux minimums à implémenter. Nous avons choisi le jeu du pendu, le jeu des allumettes et le tic-tac-toe car il s'agit de jeux avec lesquels nous sommes le plus familiers, ainsi nous avons pu gagner du temps sur la réflexion du fonctionnement du jeu. Othello, étant un jeu que nous ne connaissions pas, il nous aurait été beaucoup plus long à développer, car il fallait déjà comprendre le fonctionnement, l'organisation et les règles du jeu ainsi que les algorithmes nécessaires pour la réalisation de ce jeu.

II Architecture du projet

Le projet se décompose en plusieurs paquets, en utilisant le modèle MVC (Modèle, Vue, Contrôleur). Les classes de bases se trouvent dans le modèle, les contrôleurs des différentes interfaces graphiques dans la partie contrôleur et enfin les interfaces graphiques dans le paquet vue.



III Structure de données

Une structure de données c'est :

- une ArrayList
- une HashMap
- un BinaryTree

Nous n'avons pas utilisé d'ArrayList, HashMap ou encore d'arbre binaire dans ce projet. Cependant, dans l'interface du jeu du pendu, nous avons utilisé une ListView pour afficher l'historique des lettres choisies par l'utilisateur.



Ici, nous pouvons bien voir que les lettres D,E et F ont été choisies par l'utilisateur.

IV Fonctionnalités importantes

Dans la partie modèle, certaines fonctions importantes sont essentielles au bon fonctionnement du serveur de jeux. Tout d'abord, on retrouve le client et le serveur, sans qui le projet ne pourrait jamais se lancer. Ensuite, l'interface `MethodesInterface` contient l'entête des fonctions utilisées dans le projet pour jouer aux différents jeux proposés. Enfin, la classe `Methodes` contient le corps de ces fonctions, tout en implémentant l'interface `MethodesInterface`. Par exemple, la fonction `motPendu` est utilisée pour lire le contenu du fichier `dictionnaire.txt`, et renvoie un mot du dictionnaire ne contenant ni d'accent ou d'accent circonflexe (car eclipse a du mal à lire ces caractères et affiche "?" à la place) au hasard.

V Difficultés rencontrées

Comme tout projet, celui-ci, développé en Java, nous a mis à l'épreuve de par différents obstacles que nous avons pu rencontrer tout au long de notre travail. Tout d'abord, nous avons eu certaines difficultés par rapport à la conception du projet, comme arriver à trouver une architecture cohérente en fonction des différents jeux. Appliquer la RMI au sein d'un projet complet a été un nouveau défi à relever car cela diffère d'un simple exercice de TD. Nous avons également rencontré un nouvel obstacle dans la conception du Tic-Tac-Toe, ici la difficulté est de synchroniser le jeu pour les deux clients (joueurs) et de gérer les conflits (deux joueurs ne peuvent pas jouer en même temps, un joueur ne doit pas pouvoir jouer deux fois de suite, etc).

VI Organisation du travail

Pour réaliser ce projet, les groupes étaient séparés en trinômes. Ainsi, chaque membre du groupe a pu travailler sur un jeu en particulier, se concentrer dessus, ce qui a pu simplifier la réalisation du projet. Cependant pour la création du serveur et du client, nous nous y étions mis tous ensemble.

En ce qui concerne la communication entre les membres du groupe, nous avons créé un serveur Discord pour pouvoir échanger autour de ce projet, se donner des conseils, ou simplement discuter des problèmes rencontrés, ou encore des dates pour se réunir et faire un point tous ensemble pour voir où nous en étions.

Nous avons aussi créé un répertoire sur Github pour partager facilement le code du projet, déposer ce que nous avons terminé et avoir un espace de stockage en commun.

La méthode AGILE, qui consiste à réaliser un projet en le séparant en plusieurs tâches différentes réparties sur un temps donné (des rushs), nous a été très utile pour notre organisation. En effet, nous nous sommes fixés des objectifs et nous nous sommes donnés des rushs d'une semaine, après lesquels nous nous réunissons afin de faire un point, pour découvrir ce que chacun avait pu réaliser, et nous fixer les objectifs pour la semaine suivante. Travailler ainsi nous a permis de ne pas être dépassé par la charge de travail, de la répartir équitablement sur le temps qui nous avait été donné.