PROJET

DILEMME DU PRISONNIER

Dimanche 26 décembre 2021

Projet réalisé par :

BOURGET Hugo

CADORET Vincent

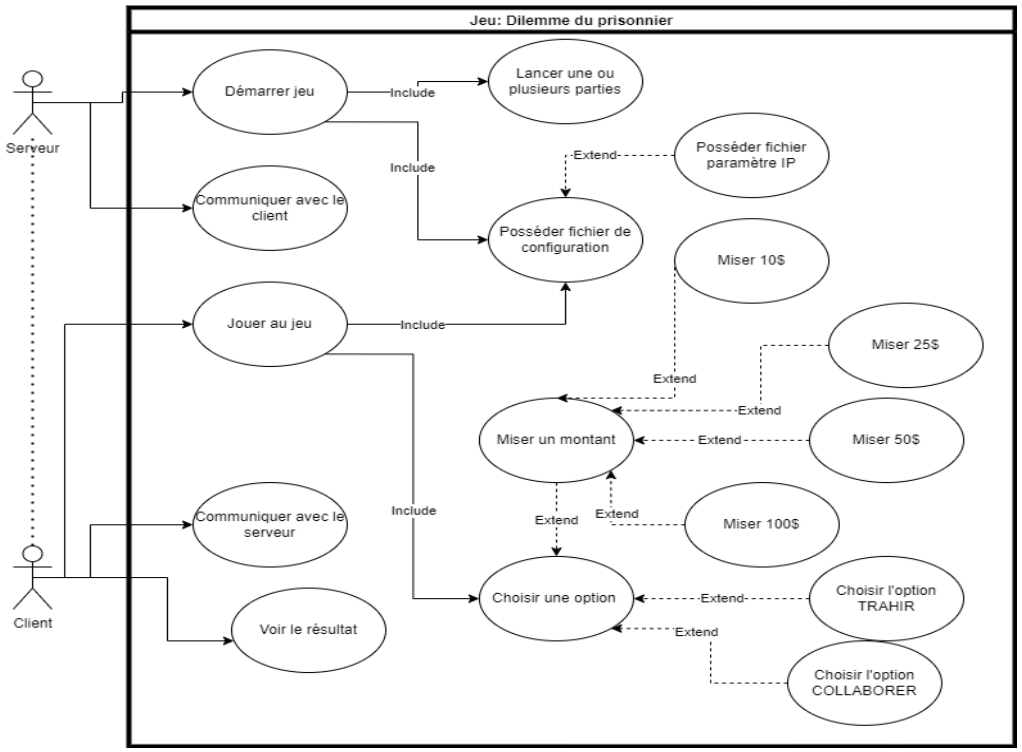
CARTELIER Alexis

HARBUTOGLU Mustafa

Table des matières

L’organisation du travail

Après avoir reçu les consignes et les directives du projet, nous avons commencé ce dernier en réalisant un digramme de cas d’utilisation afin de mieux visualiser l’architecture de fonctionnement.



De plus, afin d’être mieux organisé, un diagramme de GANTT a aussi été établi.

Après avoir tous étés concertés au début sur les choix d'architecture du projet, l'organisation du travail dans le groupe c'est scindé en deux parties. La partie client, illustré notamment par l'interface graphique et ses actions, puis la partie serveur, à travers la communication entre les deux entités et les règles du jeu.
La partie client a été réalisé en grande partie par Alexis et Moustafa tandis que le côté serveur a été attribué à Vincent et Hugo.

La méthodologie utilisée dans le développement

Le développement de l'application a été au début sous la forme de deux entités distinctes. Celles-ci ont été fusionnés en une seule grâce à l'élaboration complète d'un Makefile prenant en charge la construction de deux programmes.
Lors du processus de développement, le code a été versionné sur Github.
Une javadoc et des commentaires prenant soin d'expliqué chaque partie du code et permettre sa maintenance future ont été mis en place.
Le code a été factorisé pour faciliter sa lecture et ses potentielles évolutions.
Des fichiers de configurations ont été mis à disposition pour faciliter la prise en main du programme par le Doctorant.

La justification de certains choix et l'évaluation de la pertinence à posteriori

Lors du choix de la bibliothèque de configuration, notre premier choix a été inih,
un parser simple de fichier .ini. Cependant, il n'était pas possible de rendre la configuration modulable (exemple : ajouts illimités de rooms).
Après concertation, notre décision finale s'est tournée vers libconfig. Un puissant outil d'écriture, de lecture et de manipulation de configurations structurés.
Ce dernier permet notamment d'avoir un nombre de rooms illimités.

Pour la communication entre le serveur et client, nous avons utilisé des structures. Ce choix se justifie par la simplicité et l'efficacité lors de la manipulation de celles-ci, notamment grâce aux attributs.

La réalisation

Le projet fonctionne, il est parfaitement possible de jouer tout en suivant le protocole d'utilisation fournit dans le README.md

Après avoir cliqué sur « Connection », le joueur choisi alors une mise et une action. Ensuite il valide son choix avec le bouton « Validate ».
Une fois la partie terminée, il est indiqué au client s'il a gagné ou perdu.
Les résultats de chaque round sont stockés dans le fichier results.csv à la racine.

Ce qu'il reste à faire

Certaines choses restent à faire comme l'enregistrement du temps de réponse de chaque client lors du round, faire en sorte que les boutons soient grisés tant que l'on n'est pas connecté à sa room et l'affichage ou non de certaines données sur l'interface client (ex : balance initiale).

Les difficultés rencontrées

La grande difficulté rencontrée lors de ce projet fut le temps imparti.
En effet, ayant un groupe avec des niveaux hétérogènes, la compréhension du code et de son fonctionnement n'était pas simple pour Alexis et Mustafa. Cependant, l'aide nécessaire à ceci a été apportée dans la mesure du possible.

Conclusion

En plus du bagage technique que ce projet à apporter à chacun dans la conception logicielle, celui-ci a permis à chacun de s'améliorer tant en compétences d'organisation qu'en capacités de travail au sein d'un groupe. Il a permis aux membres de se faire confiance sur le travail effectué.