

Rapport de projet

par Adam Bekkar et Antoine Bastide

Bot

Le Bot prend ses décisions selon son niveau de difficulté et sa stratégie adaptative. Les niveaux vont de BABY à IMPOSSIBLE, définissant combien de coups le Bot peut prévoir. Le bot BABY joue aléatoirement. Les stratégies adaptatives (EARLY, MID, LATE) ajustent les décisions selon l'évolution du jeu. Le Bot évalue les pions et huttes libres, les occupants sur le plateau et les tuiles placées. En début de partie (EARLY), il se concentre sur l'expansion rapide, tandis qu'en fin de partie (LATE), il optimise les points et limite les opportunités adverses. Cela permet au Bot de jouer de manière flexible et optimisée.

Méthodes principales :

- play : Exécute la meilleure action en fonction de l'état du jeu et de la stratégie adaptative.
- playRandomAction : Joue une action aléatoire (niveau BABY).
- simulateGame : Simule le jeu sur plusieurs coups pour déterminer la meilleure action.
-

Animation de Fin de Jeu

L'animation de fin de jeu offre une conclusion dynamique. Les cases du plateau deviennent plus petites, permettant d'en afficher davantage. Elles changent de couleur plusieurs fois par seconde, adoptant les couleurs des joueurs en vague, du coin supérieur gauche au coin inférieur droit. Ce mouvement ralentit progressivement, créant un effet de suspense. À la fin, l'écran affiche le score du gagnant teinté de la couleur du ou des gagnants. Si plusieurs joueurs gagnent, l'écran devient multicolore pour refléter leurs couleurs.

Composante Graphique du Dernier Coup Joué

Cette composante met en évidence la dernière action et son impact sur le plateau. Elle est visible uniquement durant la partie et aide à s'y retrouver lorsque le plateau devient très grand. Pour implémenter cela, une classe a été créée pour simplifier le message board.

Parseur JSON

Nous avons utilisé le parseur json que nous avons écrit au premier semestre pour pouvoir implémenter les deux prochaines extensions puisqu'elles renvoient des json strings et le fait que Java n'a pas de parseur json dans sa bibliothèque standard.

Il y a 5 classes dans ce parseur:

- JSONValue.java qui est la base du parseur et contient un string qui fait office de valeur
- JSONArray.java qui étend JSONValue et est une liste de JSONValue
- JSONObject.java qui étend JSONValue et est une Map qui prend en clef des chaînes de caractères et des JSONValue en valeurs
- JSONUtils.java qui contient des méthodes utilitaires nécessaire au fonctionnement du parseur
- JSONParser.java permettant de lire/écrire du json dans des fichiers

Jeu à distance

Pour implémenter le jeu à distance, nous avons utilisé une base de données nommée [Realtime Database](#). Ce choix d'utiliser une base de données plutôt qu'un serveur ou tunnel tcp dédié nous a permis de faciliter la connexion à une partie et de rajouter d'autres fonctionnalités par dessus le jeu à distance. Ces fonctionnalités incluses:

- re-rejoindre une partie en cours si on a quittée celle-ci par mégarde
- remplacé un joueur par un bot si il a quitté la partie

Pour implémenter cette extension, nous avons utilisé l'accès à la base de données via des requêtes HTTPS get, put et delete via une classe que nous avons appelée Database.java. Ces requêtes renvoient des Objets de types Response.java, qui contiennent le corps de la réponse de la requête HTTPS et son statut code. Cette classe utilitaire permet de faciliter l'utilisation des réponses des requêtes dans notre code.

Authentication

En plus du jeu à distance, nous avons étendu la classe Database.java avec la classe Authentication.java permettant de créer un compte, se connecter à son compte, se déconnecter et supprimer son compte. Un compte nécessite un nom d'utilisateur et un mot de passe, qui sont utilisés pour créer un identifiant unique pour le joueur (UUID). Cet identifiant permet au joueur de se reconnecter automatiquement lorsqu'il relance le jeu, de re-rejoindre une partie puisqu'elle est liée à son identifiant et de changer l'hôte de la partie si nécessaire.

Interface

Nous avons ajouté plusieurs interface différents afin de rendre ergonomique pour l'utilisateur, on retrouve des scènes :

- D'authentification générale permettant de choisir entre créer un compte et se connecter à son compte
- De "Compte" permettant de créer un compte ou de se connecter à son compte
- Générale qui affiche toute les boutons permettant d'accéder au autres scènes
- Pour rejoindre une partie qui affiche toutes les parties existantes avec leur statut et un bouton pour rafraîchir l'affichage
- Pour créer une partie qui demande un nom de partie
- Qui affiche les règles du jeu
- Qui affiche les statistiques lié au compte du joueur
- D'attente qui est affiché lorsque on a rejoint une partie, et le l'hôte à la possibilité de choisir le nombre de cartes et la difficulté du bot qui remplacera un joueur si nécessaire
- Création d'une partie contre un bot, similaire à la scène d'attente, sans communication avec la base de données.