

Christophe TARATIBU

Loïc BERGEOT

Rapport de développement en java du jeu de société

Cascadia

Cascadia est un jeu de société créé par Randy Flynn en 2021, faisant référence à la région nord-ouest de L'Amérique du Nord qui s'étend de la Californie jusqu'au confins de l'Alaska. Le jeu s'effectue au tour par tour entre les joueurs (entre 1 et 5 joueurs), qui ont pour but de constituer leur environnement avec les tuiles et jetons du jeu. Une partie de ce jeu s'arrête au bout de 20 tours avec pour gagnant le joueur qui a le plus de points. L'objectif ici est d'implémenter une version du jeu en java en utilisant l'architecture de notre choix. Voici alors le choix d'implémentation du jeu

I. Organisation des fichiers du jeu

Les fichiers sources sont contenus dans un package général nommé game. Ce dernier contient des sous packages qui structurent et organisent le jeu.

- game/display : contient les fichiers sources (AnimalColor.java, Display.java et DisplayTools.java) qui permettent l'affichages des tuiles et jetons colorés ou encore le choix des 4 tuiles et jetons.
- game/logic : contient les fichiers sources (Choice.java, GameLogic.java, Position.java, Scores.java) qui implémentent la logique du jeu : choix des tuiles et jeton pose des tuiles en fonction d'une tuile adjacente, interaction utilisateur.
- game/ main : Le fichier source Main.java qui inclus le programme principal du jeu.
- game/material : contient l'implémentation du matériel du jeu, tuiles, et environnement du joueur (composé de jetons et tuiles placés)
- game/player : ce package implémente les données d'une personne pour en faire un joueur.

Un fichier csv (Tuiles.csv) comprend toutes les tuiles du jeu.

II. Structure des éléments du jeu

Le matériel du jeu Cascadia a été implémenté de la manière suivante :

- Pour les tuiles, l'objet `Tile` les caractérise . Il englobe les animaux possibles (sous forme de liste) et l'unique place possible représenté de chaque par deux lettres. Pour l'affichage terminal les tuiles sont affichées en ASCII art.
- Tuiles de départ : `HashMap<Tile,Token>` qui implémentent des tuiles sans jetons dessus et `LinkedHashMap<HashMap<Tile,Token>,Position>` qui permet de poser une tuile de départ à des positions précises (0,0) (0,1) et (0,2) . Cette même structure est utilisée pour implémenter l'environnement du joueur.
- Tuiles du jeu (basiques) : `ArrayList<Tile>` comprend la liste des tuiles contenu dans le fichier csv.
- Dans le cas des jetons, le record `token` est présent, il représente les jetons avec le nom de l'animal et sa couleur. Dès lors qu'un jeton est posé dans l'environnement , la couleur en fonction de l'animal est mis en avant.
- Le sac du jeu pour mélanger les tuiles et jetons est représenté par la méthode `shuffle` de la classe `Collections` qui mélange les listes tuiles et jetons avant les choix des joueurs.
- L'environnement du joueur est représenté par cette structure `LinkedHashMap<HashMap<Tile,Token>,Position>` qui permet de garantir l'ordre d'insertion des tuiles et jetons avec leur position dans la grille.
- En ce qui concerne la grille, elle implémentée par une liste de liste de `Sti`
- Le calcul de score se fait par parcours en profondeur avec la recherche des groupes d'animaux avec stockage des positions déjà visitées.

III. Difficultés rencontrées.

La conception de la première partie du projet s'est heurtée à plusieurs problèmes :

- L'affichage du jeu : Problèmes pour l'affichage des tuiles et coloration des jetons
- Exploitation du fichier csv : Différents chemins de recherche du fichier pour l'exécution normale et celle du fichier jar.
- Conversions données : Des variables se partagent les mêmes données créant des mêmes environnements pour tous les joueurs.