

Projet Mijnlieff

Pour le projet de création d'un jeu de Mijnlieff informatique en C, nous avons commencé par créer une première version avec le nécessaire du jeu : un plateau, deux camps: rouge et blanc, les 4 types de tuiles, les decks de tuiles de chaque joueurs, l'information de position des emplacements et l'information du coups joué. Le matériel nécessaire est défini par des énumérations et structures.

La fonction main et les fonctions basiques y ont été mises.

Dans le main seulement quelques lignes qui seront utilisé pour demandé au joueur si il souhaite jouer contre une autre personne tour a tour ou si il veut joué contre l'ordinateur.

Le choix renvoi a une des deux fonctions faite plus tard.

Les fonctions basiques sont :

- Les fonctions d'initialisation : des decks des joueurs, du plateau.
- Les fonction d'affichage : du decks du joueurs a son tours, du plateau actuelle a chaque début de tours. Le plateau affiche sur les bordures les numéros des colonnes et lignes correspondantes, les emplacements vides sont marqués par un points et les tuiles pas la première lettre de leur type et coloré en fonction du joueur rouge ou blanc.
- La fonction de mise a jour du deck des joueurs après la pose d'une tuile.
- Les fonctions de vérification du coup : que le coup est bien dans le plateau et respecte les règles.

La partie de la fonction de jeu en 1v1 a été faite en lançant les initialisations, puis le morceau de code de la partie demandant la position ou le joueur souhaite poser la tuile et quel type de tuile. Et garde en mémoire le dernier coup joué puis change de joueur. Le code est le meme pour les deux et la différence ne se fait que en fonction de la variable tmp qui indique le joueur actif et change a la fin de chaque tour. Le Code de jeux est dans une boucle while qui tourne tant qu'un coup est possible.

Dans la deuxième version le système de point et résultat des scores a été ajouté .

Une fonction résultat qui appelle tour a tour une fonction comptant le score des blanc en colonne ligne et diagonal puis renvoyant le score puis appelle de la fonction de compte des rouge et enfin la fonction résultat compare les deux scores et indique le gagnant ou égalité.

Dans la troisième version la fonction de jeux contre le bot a été implanté.

Pour la partie joueur il s'agit du même code que pour le 1v1.

Le bot est très simpliste, il teste les cases dans l'ordre pour voir si elles sont jouable et pose une tuile aléatoire sur le premier emplacement jouable.

Une réparation du système de point a été faite.

Problèmes

- Un problème mineur a été que dans la première énumération des tuiles elles ont toutes un numéro positif et pour les différencier dans la fonction d'initialisation du rouge les tuiles ont été associées à un -tuile. On aurait donc dû pouvoir appeler les tuiles par -1 -2 -3 -4 mais la fonction ne les reconnaissait pas il a donc fallu ajouter une fonction de conversion de la tuile sélectionnée « 1 2 3 4 » ou « -1 -2 -3 -4 » en « -CROIX -DIAG -TIREUR -POUSSEUR » pour le joueur rouge
- Lors du compte des scores un problème qui s'est posé est que la fonction comptait les cases vides comme cases correctes ajoutant ainsi des points malgré une condition n'acceptant que les « -CROIX -DIAG -TIREUR -POUSSEUR » ou « CROIX DIAG TIREUR POUSSEUR ». La fonction de vérification des tuiles pour les scores a été remplacée avec une appelle de fonction vérifiant si la case appartient au rouge ou blanc.
- Le bot devait initialement jouer une case au hasard parmi les coups possibles mais la fonction ne marchait pas pour tous les coups. La raison identifiée est que quand la fonction principale au tour du bot recevait la position de la case qu'il doit jouer les positions étaient totalement hors tableau n'ayant pas pu résoudre ce problème le mode de jeu du bot a été changé pour un plus simpliste qui ne cause pas d'erreur.