Rapport du projet Tablut

Introduction

Pour l'implémentation de ce jeu, nous avons utilisé plusieurs Structures et Constantes réparties dans les différents fichiers .h .

La structure Plateau a un champ contenu qui est un tableau de pointeur de type Cellule qui contient des informations sur le contenu d'une case.

La structure Coord est une structure dédiée aux déplacements des pions avec des coordonnées de départs et d'arrivées, ainsi qu'un champ j qui permet de déterminer le camp du joueur et les pièces déplaçables ou non par ce dernier.

plateau.c

Nous avons fait une fonction afin d'allouer dynamiquement le plateau du jeu et une fonction pour libérer la mémoire en fin de programme.

On a ensuite une fonction placer_pions pour placer les pions sur le plateau qui est au préalable initialisé à la valeur 0 pour Château et VIDE pour Pièce.

Dans cette fonction, on parcours le plateau de gauche à droite puis de bas en haut pour placer les pions. On place un MOSCOVITE si nous sommes en dehors du château, un SUÉDOIS sinon.
On peut détecter les bords du château en utilisant une variable bord qui contient la taille du plateau divisé par 4.
On utilise une variable f qui s'incrémente lorsque que l'on touche un bord du plateau et qui détermine le sens du parcours.

deplacement.c

Dans ce fichier, nous avons créé une fonction afin de pouvoir déplacer les pions.
Nous obtenons la direction des pions grâce à la fonction dir.

Nous avons créé une fonction message, qui détermine les erreurs de déplacement (contre règlement du jeu), ainsi qu'une fonction par rapport aux validité des déplacements (par exemple : si nous jouons sur le bon pion, si les coordonnées sont dans le plateau ou aucun des pions ne peut aller sur le trône après que le roi soit parti, et ni même ce dernier).

Ensuite, nous avons créé une fonction qui détecte le bord du chateau, si nous sommes dans le château, la fonction nous retourne 2, sinon 0, et si nous sommes dans le bord, 1.

Enfin, les deux dernières fonctions concernent la capture du pion ou du roi:

- La capture du pion peut se faire par un encadrement de deux pions du même joueur à un autre pion de l'autre joueur. Nous regardons si un pion du joueur agresseur et un pion du joueur victime sont côte à côte, si oui, alors la fonction regarde à 2 cases avant selon la direction s'il y a un autre pino agresseur; si oui, la capture se fait.
- La capture du roi : si le pion en question est un moscovite, nous regardons à chaque direction du roi s'il y'en a un, car la capture ne se fait que par 4 moscovites qui l'entourent. num stocke la sortie de fonction de la détection du bord de château et return -1 indique le fin de jeu.

menu.c

Nous avons une fonction bot_rdm qui choisit des coordonnées aléatoires pour le Bot.

On a d'abord un choix aléatoire de la pièce de départ déplaçable par le Bot. puis un choix aléatoire pour la direction prise par le Bot, verticale ou horizontale, et enfin un choix aléatoire de la case d'arrivée, le tout conformément aux règles de déplacements.

main.c

Nous sélectionnons le mode dans lequel nous voulons jouer, contre un joueur ou contre un ordinateur. On déclare une instance plat de type plateau et nous l'initialisons à NULL ainsi que co de type Coord. Nous allouons dynamiquement plat puis à la fin du programme on libère la mémoire allouée.

Extensions

Nous avons fait seulement la variation de la taille du plateau.
Nous avons utilisé une variable globale SIZE, utilisée dans la fonction d'alloument dynamique.

Sauf pour une taille de plateau de 9, le château se place mal, probablement une mauvaise gestion des arrondies des divisions.