

RAPPORT JEU D'ECHEC



SOMMAIRE

- Projet et manuel utilisateur - [page 3](#)
- Structure générale du programme, principales fonctions et leur rôle - [page 4](#)
- Principales variables utilisées - [page 5](#)
- Répartition du travail au sein du binôme - [page 6](#)

LE PROJET

Le projet que nous avons choisi est celui du jeu d'échec, le but de celui-ci était de créer un jeu qui permettrait à deux joueurs de s'affronter sur un plateau, suivant les règles du jeu d'échec normales. Ce projet n'a pas pour but de faire un jeu dans lequel un joueur peut jouer contre l'ordinateur mais bel et bien un jeu où deux joueurs réels s'affrontent.

MANUEL D'UTILISATION

Le jeu permet donc à deux joueurs de s'affronter sur un plateau composé de 8 pions de chaque camp, 2 tours, 2 fou, 2 cavalier, 1 roi et 1 reine.

Les règles de ce jeu sont les mêmes que celles d'un jeu lambda en ce qui concerne les déplacements.

Le joueur doit double cliquer sur un pion pour le sélectionner.

Le joueur peut afficher le menu Pause en appuyant sur la touche P. Ce menu lui offrira la possibilité de quitter le jeu et d'arrêter la partie ou d'annuler le coup qui a été effectué en cas de fausse manipulation.

Un pion qui arrive sur la dernière ligne du côté adverse du plateau peut se sacrifier pour faire apparaître au choix un fou, une tour, un cavalier ou une reine.

La partie est gagnée lorsque un Roi d'une des deux équipes est tué.

STRUCTURE DU PROGRAMME

Le programme est divisé en trois parties, la première est celle qui concerne toute l'interface du jeu. Cette partie est composée de plusieurs fonctions dont les plus importantes sont la fonction « affiche_pieces » et « plateau » qui sont les fonctions qui servent à afficher les pièces en fonction de leur placement, c'est-à-dire que chaque pion est une liste de coordonnées et que la fonction reçoit cette liste et affiche chaque pion à sa coordonnée attribuée soit initialement soit par les déplacements du pion, la seconde fonction principal sert à afficher les cases du plateau de couleur différente et prédéfinie.

De plus, la seconde partie du programme est celle qui concerne les déplacements des pions. Toutes les fonctions de déplacement sont importantes, ainsi nous allons en prendre 2 pour exemple. « regle_depl_fou » et « mouv_fou ». La fonction « regle_depl_fou » a été divisée en 2 car sinon elle allait être trop longue. Ainsi, celle-ci sert à affecter la case choisie comme arrivée au pion, celle-ci se met en place si la fonction « mouv_fou » qui reçoit les coordonnées de la case d'arrivée a validé le déplacement en se basant sur les règles de l'échec qui dit qu'un fou ne doit uniquement se déplacer en diagonale. Ainsi, la fonction « regl_depl_fou » remplace les anciennes coordonnées du fou dans la liste fou par les nouvelles, c'est-à-dire celles de la case choisie comme destination.

Enfin, la troisième et dernière partie du programme est celle qui contient toutes les « options » de jeu, qui pourraient être mises dans les deux premières parties. Par exemple la fonction « menu » qui propose d'afficher le menu du jeu lorsque la touche « P » (pause) sera utilisée. Nous pouvons également citer la fonction « choix_nouv_piece » qui permet à l'utilisateur de choisir une nouvelle pièce lorsqu'un pion atteint le côté opposé du plateau.

PRINCIPALES VARIABLES

Les principales variables utilisées pour ce programme sont les listes de chaque pion, par exemple « pion_blanc » ou « fou_noir » qui contiennent les coordonnées de chaque pion blanc ou fou noir sur le plateau. Ainsi, chaque type de pion de chaque couleur a une liste qui lui est attribué contenant les coordonnées de son type de pion et de sa couleur.

Nous avons également dû établir comme variables la hauteur (hauteur_plateau) et la largeur du plateau (largeur_plateau) pour la fonction qui va afficher le plateau et les cases

De plus, une variable « joueur » a dû être définie pour savoir quel joueur doit jouer, 1 correspond au joueur qui utilise le jeu blanc et 2 le jeu noir. Cette variable est essentielle à la boucle principale du programme.

Enfin, une variable « morts » a été établie contenant la liste de tous les pions tués sous la forme d'un string (« pion_blanc »)

REPARTITION DU TRAVAIL

s'est occupé de toute la seconde partie du programme (ce qui équivaut à la première et troisième partie du programme réunies), c'est-à-dire celle qui concerne les déplacements des pions et les règles de déplacement et les fonctions qui concernent l'élimination des pions.

s'est occupé de toute la première partie qui concerne l'interface du jeu et de la troisième partie, c'est-à-dire la partie qui contient les options comme « menu » ou par exemple « choix_nouv_piece ».

En ce qui concerne la boucle principale du programme, cette partie a été faite ensemble.