

Descriptif des fonctions du code :

initialiser :

Cette fonction sert à mettre la valeur VIDE à chaque cellule de la matrice

placer_pion :

Cette fonction met les valeurs BLEU ou ROUGE ou VIDE aux cellules de la matrice.

placer_un_pion_sur_deux :

Fonction semblable à placer_pion . Fait partie d'une des extensions.

Cela place les pions une ligne sur 2.

placer_deux_pion_sur_un :

Fonction semblable à placer_pion . Fait partie d'une des extensions.

Cela place les pions un sur deux.

afficher_solution :

Cette fonction à pour rôle d'afficher le plateau de jeu en fonction des valeurs des cellules de la matrice. Les valeurs possibles sont 1 pour BLEU, 2 pour ROUGE, 6 pour VIDE, 7 pour USE, 13 pour BLEU_BOUGE (une case visitée par un pion bleu) et 14 pour ROUGE_BOUGE.

Jouable :

Cette fonction permet de savoir si le pion ciblé peut jouer (donc s'il y a une case libre adjacente).

Mangeable :

Cette fonction regarde si le pion sélectionné peut manger un pion adverse que ce soit en diagonale, en avant ou en retrait, en prenant en compte s'il peut faire ou non des diagonales.

(Cela check donc les 8 positions adjacentes si le pion peut faire des mouvements en diagonale et les 4 positions cardinales s'il peut pas).

coord_area :

Cette fonction permet de voir si les coordonnées de déplacement données sont adjacentes aux coordonnées du pion

Demande :

Simple fonction qui demande au joueur s'il veut manger en avant ou en arrière quand il peut.

Mangeation :

En plus d'avoir un nom parfaitement francé, cette fonction a pour rôle d'éliminer les pions adverses en ligne tant qu'ils sont à la suite les uns les autres.

Diagonal :

Cette fonction permet de checker si le pion peut faire des mouvements en diagonale.

Elle permet d'alléger certaines fonctions.

Déplacement :

Cette fonction permet de faire le déplacement. Elle contient plusieurs restrictions comme vérifier si le pion peut bouger et manger et si les coordonnées données pour le déplacement sont bien adjacentes etc.

Check_vie_BLEU / Check_vie_ROUGE :

Fonction simple qui calcule le nombre de pion qui reste par joueur.

fin_tour :

Cette fonction demande à chaque tour si le joueur veut finir son tour si un autre déplacement impliquant le mangéage d'un pion est possible (sauf quand il ne peut de toute façon pas continuer)

doit_manger :

Cette fonction appelée dans le cas d'un déplacement n'incluant pas de mangeage (jolis mot n'est-ce pas ?) vérifie que pion n'a pas la possibilité de manger un pion adverse. Si c'est le cas, un message d'erreur est retourné.

devant_devant :

En plus de traduire un profond manque d'imagination, cette fonction permet d'empêcher un joueur d'aller deux fois dans la même direction

mouvement :

Si le nom de cette fonction vous fait penser à déplacement, c'est pour une bonne raison. Cette fonction se lance quand un pion peut faire plusieurs déplacements consécutifs. Les seuls moyens d'en sortir sont de répondre oui à la demande de fin de tour (cf fonction fin_tour) ou de ne plus pouvoir jouer en mangeant.

Reinit_use :

Cette fonction permet à la fin d'un tour de jeu de donner la valeur VIDE à toutes les cellules ayant la valeur USE.

Victoire:

Cette fonction permet en premier lieu de supprimer la sauvegarde. elle est appelée quand une des vies tombe à zéro et regarde la valeur de la vie des bleus : si elle est à zéro c'est le joueur bleu qui a perdu, sinon c'est l'autre.

sauvegarde:

Cette fonction sauvegarde sur un fichier texte si sur chaque case du jeu la cellule est BLEU,ROUGE ou VIDE.

load_sauvegarde:

Fonction se basant sur le procédé inverse de la sauvegarde, elle attribue à chaque case de la matrice les valeurs sauvegardées.

check_sauvegarde:

Cette fonction regarde si la taille du plateau choisie pour jouer est compatible avec la taille du plateau chargé.

menu:

Fonction graphique (et jolie !) qui demande au joueur des options de lancement du jeu.

jeu_sans_ia:

Cette fonction fait jouer les deux joueurs quand il n'y a pas d'IA.

Elle comprend un certain nombre des fonctions précédentes.

diagonale_ia:

Une version simplifiée de la fonction diagonale qui permet d'alléger la fonction déplacement_IA

score_dpl:

Cette (grosse) fonction calcule le score du mouvement donné. Le score donne 1 point par pions mangés. Elle prend seulement en compte les déplacements des pions pouvant faire des mouvements en diagonales.

score4:

Fonction identique à score_dpl, elle prend ici les pions ne pouvant pas faire de mouvements en diagonales.

mouvement_random :

Quand l'IA ne peut pas faire de déplacements pouvant manger, elle panique beaucoup et fait un déplacement aléatoire et stupide.

deplacement_IA :

Fonction très similaire à la suite de fonction contenue dans jeu_sans_ia, elle permet de faire jouer l'IA tout en respectant les règles du jeu (dura lex, sed lex).

jeu_ia_debile :

Cette fonction permet à l'IA (qui n'est pas SI stupide) de déterminer les meilleurs coups grâce aux fonction score_dpl/score4 et transmet ces coordonnées à déplacement_ia

jeu_joueur:

Cette fonction permet aux joueurs de jouer en comprenant toutes les règles de jeux, mais ne possédant pas les spécificités d'alternance de joueur car elle est faite pour le jeu avec IA.

jeu_avec_ia:

cette fonction permet de faire jouer en alternance le joueur puis l'IA ou inversement.

main :

Dedans il y a juste le choix de jouer avec ou sans IA. Elle appelle donc jouer_avec_IA ou jouer_sans_IA.

Extensions Réalisé

Nous avons réalisé différentes extensions.

Nous avons donné le choix à l'utilisateur de choisir entre 5 tailles différentes de plateau.

L'utilisateur peut aussi choisir deux règles différentes:

- un pion sur deux
- une ligne sur deux

Il y a une fonction de sauvegarde automatique a chaque fin de tour.

Il y a un check si la sauvegarde ne correspond pas à la taille du plateau sélectionné.

L'IA que nous avons fait le calcul du meilleur coup et l'exécute.