

Documentation Puissance 4

MOALIGOU Elise

Version 1

Lundi 21 Novembre 2022

Table des matières

Table of contents

Index des fichiers

Liste des fichiers

Liste de tous les fichiers avec une brève description :

`code_source_final.c` (Ce programme permet de jouer au puissance 4)3

Documentation des fichiers

Référence du fichier `code_source_final.c`

ce programme permet de jouer au puissance 4

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
```

Graphes des dépendances par inclusion de `code_source_final.c`:
IMAGE

Macros

```
#define NBCOL 7
#define NBLIG 6
```

Définitions de type

```
typedef char t_Grille[NBLIG][NBCOL]
```

créer le type tableau pour la grille du puissance 4, permet de stocker les indices des pions joués

Fonctions

```
void afficher (t_Grille, char, int)
```

Realise l'affichage a l'ecran du contenu de la grille avec les pions deja joues. Cette procedure affiche aussi, au-dessus de la grille, le prochain pion a tomber : il sera affiche au-dessus de la colonne dont le numero est donne en parametre. Cette procedure commencera par effacer l'ecran.

```
int choisirColonne (t_Grille, char, int)
```

Un joueur voit son pion au-dessus de la grille et cette fonction doit lui permettre de "deplacer" son pion d'une colonne vers la gauche (par la touche 'q') ou d'une colonne vers la droite (par la touche 'd'). Apres chaque deplacement, la grille est reaffichee. Le joueur peut finalement choisir la colonne ou il souhaite faire tomber son pion (par la touche ESPACE).

```
bool estVainqueur (t_Grille, int, int)
```

Indique si le pion situe dans la case reperee par les parametres ligne et colonne a gagne la partie, c'est-a-dire s'il y a une ligne, une colonne ou une diagonale formee d'au moins 4 de ses pions (la ligne et la colonne passees en parametres correspondent a la case ou le joueur vient de jouer; c'est-a-dire la case a partir de laquelle il faut rechercher 4 pions successifs identiques).

void **finDePartie** (char)

Affiche le resultat d'une partie lorsqu'elle est terminee.

bool **grillePleine** (t_Grille)

Teste si toutes les cases de la grille sont occupees ou non.

void **initGrille** (t_Grille)

Initialise la grille en affectant la constante VIDE a chacun de ses elements.

void **jouer** (t_Grille, char, int *, int *)

Permet a un joueur de jouer son pion. La procedure fait appel a choisirColonne, afin que le joueur indique la colonne dans laquelle il veut jouer ; puis fait appel a trouverLigne pour definir la case ou ajouter le pion.

int **main** ()

int **trouverLigne** (t_Grille, int)

Consiste a trouver la premiere case non occupee de la colonne. Si la colonne est pleine, la fonction retourne -1.

bool **verif_col** (t_Grille, int, int, char)

Indique si le pion situe dans la case reperee par les parametres ligne et colonne, est aligne verticalement a trois autres pions egaux.

bool **verif_diago_ne_so** (t_Grille, int, int, char)

Indique si le pion situe dans la case reperee par les parametres ligne et colonne, est aligne a trois autres pions egaux en diagonale du type : nord-est vers sud-ouest de la grille.

bool **verif_diago_no_se** (t_Grille, int, int, char)

Indique si le pion situe dans la case reperee par les parametres ligne et colonne, est aligne a trois autres pions egaux en diagonale du type : nord-ouest vers sud-est de la grille.

bool **verif_lig** (t_Grille, int, int, char)

Indique si le pion situe dans la case reperee par les parametres ligne et colonne, est aligne horizontalement a trois autres pions egaux.

Variables

const int **COLONNE_DEB** = NBCOL/2

const char **INCONNU** = ' '

const char **PION_A** = 'X'

const char **PION_B** = 'O'

const char **VIDE** = ' '

Description détaillée

ce programme permet de jouer au puissance 4

Auteur:

Elise MOALIGOU

Version:

1.0

Date:

20 novembre 2022

Ce programme permet de jouer une partie de puissance 4 avec deux utilisateurs reels. Chaque joueur choisi un pion entre le pion X et le pion O au prealable et jouent chacun leur tour en deplacant leur pion au dessus de la colonne choisie a l'aide des touches 'd' 'q' et 'espace'. A la fin de la partie le programme annonce le joueur gagnant.

Documentation des macros

#define NBCOL 7

defini le nombre de colonne pour le tableau

#define NBLIG 6

defini le nombre de ligne pour le tableau

Documentation des définitions de type

t_Grille

creer le type tableau pour la grille du puissance 4, permet de stocker les indices des pions joues

Documentation des fonctions

void afficher (t_Grille grille, char pion, int bonne_colonne)

Realise l'affichage a l'ecran du contenu de la grille avec les pions deja joues. Cette procedure affiche aussi, au-dessus de la grille, le prochain pion a tomber : il sera affiche au-dessus de la colonne dont le numero est donne en parametre. Cette procedure commencera par effacer l'ecran.

Paramètres:

<i>grille</i>	Grille, represente la grille de jeu
<i>pion</i>	represente le pion a afficher au-dessus de la grille
<i>bonne_colonne</i>	indice de la colonne ou doit se trouver le pion au debut de chaque tour

int choisirColonne (t_Grille grille, char pion, int colonneDessus)

Un joueur voit son pion au-dessus de la grille et cette fonction doit lui permettre de "deplacer" son pion d'une colonne vers la gauche (par la touche 'q') ou d'une colonne vers la droite (par la touche 'd'). Apres chaque deplacement, la grille est reaffichee. Le joueur peut finalement choisir la colonne ou il souhaite faire tomber son pion (par la touche ESPACE).

Paramètres:

<i>grille</i>	Grille, represente la grille de jeu
<i>pion</i>	caractere, represente le pion a tester
<i>colonneDessus</i>	colonne de depart (celle au-dessus de laquelle se trouve le pion initialement)

Renvoie:

indice de la colonne choisie par le joueur

bool estVainqueur (t_Grille grille, int ligne, int colonne)

Indique si le pion situe dans la case reperee par les parametres ligne et colonne a gagne la partie, c'est-a-dire s'il y a une ligne, une colonne ou une diagonale formee d'au moins 4 de ses pions (la ligne et la colonne passees en parametres correspondent a la case ou le joueur vient de jouer, c'est-a-dire la case a partir de laquelle il faut rechercher 4 pions successifs identiques).

Paramètres:

<i>grille</i>	Grille, represente la grille de jeu
<i>ligne</i>	indice de la ligne dans laquelle le pion se trouve
<i>colonne</i>	indice de la colonne dans laquelle le pion se trouve

Renvoie:

VRAI s'il y a une telle ligne de 4 a partir de la case indiquee, FAUX sinon

void finDePartie (char vainqueur)

Affiche le resultat d'une partie lorsqu'elle est terminee.

Paramètres:

<i>vainqueur</i>	caractere qui represente le pion gagnant (PION_A ou PION_B) ou bien VIDE si match nul
------------------	---

bool grillePleine (t_Grille grille)

Teste si toutes les cases de la grille sont occupees ou non.

Paramètres:

<i>grille</i>	Grille, represente la grille de jeu
---------------	-------------------------------------

Renvoie:

VRAI si toutes les cases de la grille sont occupees par les pions, FAUX sinon

void initGrille (t_Grille grille)

Initialise la grille en affectant la constante VIDE a chacun de ses elements.

Paramètres:

<i>grille</i>	Grille, represente la grille de jeu
---------------	-------------------------------------

void jouer (t_Grille grille, char pion, int * ligne, int * colonne)

Permet a un joueur de jouer son pion. La procedure fait appel a choisirColonne, afin que le joueur indique la colonne dans laquelle il veut jouer ; puis fait appel a trouverLigne pour definir la case ou ajouter le pion.

Paramètres:

<i>grille</i>	Grille, represente la grille de jeu
<i>pion</i>	Caractere, correspond au pion a jouer
<i>ligne</i>	entier, correspond a la ligne ou est tombe le pion
<i>colonne</i>	entier, correspond a la colonne ou est tombe le pion

int main ()

int trouverLigne (t_Grille grille, int ind_col)

Consiste a trouver la premiere case non occupee de la colonne. Si la colonne est pleine, la fonction retourne -1.

Paramètres:

<i>grille</i>	Grille, represente la grille de jeu
<i>ind_col</i>	indice de la colonne dans laquelle le pion doit tomber

Renvoie:

indice de la ligne ou le pion devra etre ajoute ou -1 si la colonne est pleine

bool verif_col (t_Grille grille, int ligne, int colonne, char pion)

Indique si le pion situe dans la case reperee par les parametres ligne et colonne, est aligne verticalement a trois autres pions egaux.

Paramètres:

<i>grille</i>	Grille, represente la grille de jeu
<i>ligne</i>	indice de la ligne dans laquelle le pion se trouve
<i>colonne</i>	indice de la colonne dans laquelle le pion se trouve
<i>pion</i>	caractere, represente le pion a tester

Renvoie:

VRAI s'il y a une telle ligne de 4 a partir de la case indiquee, FAUX sinon

bool verif_diago_ne_so (t_Grille grille, int ligne, int colonne, char pion)

Indique si le pion situe dans la case reperee par les parametres ligne et colonne, est aligne a trois autres pions egaux en diagonale du type : nord-est vers sud-ouest de la grille.

Paramètres:

<i>grille</i>	Grille, represente la grille de jeu
<i>ligne</i>	indice de la ligne dans laquelle le pion se trouve
<i>colonne</i>	indice de la colonne dans laquelle le pion se trouve
<i>pion</i>	caractere, represente le pion a tester

Renvoie:

VRAI s'il y a une telle ligne de 4 a partir de la case indiquee, FAUX sinon

bool verif_diago_no_se (t_Grille grille, int ligne, int colonne, char pion)

Indique si le pion situe dans la case reperee par les parametres ligne et colonne, est aligne a trois autres pions egaux en diagonale du type : nord-ouest vers sud-est de la grille.

Paramètres:

<i>grille</i>	Grille, represente la grille de jeu
<i>ligne</i>	indice de la ligne dans laquelle le pion se trouve
<i>colonne</i>	indice de la colonne dans laquelle le pion se trouve
<i>pion</i>	caractere, represente le pion a tester

Renvoie:

VRAI s'il y a une telle ligne de 4 a partir de la case indiquee, FAUX sinon

bool verif_lig (t_Grille *grille*, int *ligne*, int *colonne*, char *pion*)

Indique si le pion situe dans la case reperee par les parametres ligne et colonne, est aligne horizontalement a trois autres pions egaux.

Paramètres:

<i>grille</i>	Grille, represente la grille de jeu
<i>ligne</i>	indice de la ligne dans laquelle le pion se trouve
<i>colonne</i>	indice de la colonne dans laquelle le pion se trouve
<i>pion</i>	caractere, represente le pion a tester

Renvoie:

VRAI s'il y a une telle ligne de 4 a partir de la case indiquee, FAUX sinon

Documentation des variables

const int COLONNE_DEB = NBCOL/2

const char INCONNU = ' '

const char PION_A = 'X'

const char PION_B = 'O'

const char VIDE = ' '

Index

INDEX