

Documentation Puissance 4

AUTHOR

Version

Dimanche 27 Novembre 2022

Table des matières

Table of contents

Index des fichiers

Liste des fichiers

Liste de tous les fichiers avec une brève description :

puissance4.c	3
---------------------------	---

Documentation des fichiers

Référence du fichier puissance4.c

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

Macros

```
#define COLONNE_DEBUT NBCOL / 2
    Constante pour définir la colonne de début qui est celle du milieu.
```

```
#define INCONNU ' '
    Constante pour le caractère inconnue.
```

```
#define NBCOL 7
    Constante pour définir le nombre de colonne.
```

```
#define NBLIG 6
    Constante pour définir le nombre de ligne.
```

```
#define PION_A 'X'
    Constante pour définir le pion du prénom1.
```

```
#define PION_B 'O'
    Constante pour définir le pion du prénom2.
```

```
#define VIDE ' '
    Constante pour le caractère vide.
```

Définitions de type

```
typedef int grille[NBLIG][NBCOL]
```

Fonctions

```
void afficher (grille g, char pion, int colonne)
    Affiche la grille.
```

```
int choisirColonne (grille g, char pion, int colonne)
    Un joueur voit son pion au-dessus de la grille et cette fonction doit lui permettre de
    "déplacer" son pion d'une colonne vers la gauche (par la touche 'q') ou d'une colonne
    vers la droite (par la touche 'd'). Après chaque déplacement, la grille est réaffichée. Le
    joueur peut finalement choisir la colonne où il souhaite faire tomber son pion (par la
    touche ESPACE).
```

```
bool estVainqueur (grille g, int ligne, int colonne)
    Indique si le pion situé dans la case repérée par les paramètres ligne et colonne a gagné
    la partie, c'est-à-dire s'il y a une ligne, une colonne ou une diagonale formée d'au moins
```

4 de ses pions (la ligne et la colonne passées en paramètres correspondent à la case où le joueur vient de jouer, c'est-à-dire la case à partir de laquelle il faut rechercher 4 pions successifs identiques).

void **finDePartie** (char pion, char nom1[20], char nom2[20])
Affiche le résultat d'une partie lorsqu'elle est terminée.

bool **grillePleine** (grille g)
Test si toutes les cases de la grille sont occupées.

void **jouer** (grille g, char pion, int *ligne, int *colonne)
permet à un joueur de jouer son pion. La procédure fait appel à choisirColonne, afin que le joueur indique la colonne dans laquelle il veut jouer ; puis fait appel à trouverLigne pour définir la case où ajouter le pion.

int **main** ()
Code Principale du puissance 4.

void **pinitgrille** (grille g)
Procédure qui va servir pour initialisé le tableau en tableau vide.

int **trouverLigne** (grille g, int colonne)
Consiste à trouver la première case non occupée de la colonne. Si la colonne est pleine, la fonction retourne -1.

Documentation des macros

#define COLONNE_DEBUT NBCOL / 2

Constante pour définir la colonne de début qui est celle du milieu.

#define INCONNU ' '

Constante pour le caractère inconnue.

#define NBCOL 7

Constante pour définir le nombre de colonne.

#define NBLIG 6

Constante pour définir le nombre de ligne.

#define PION_A 'X'

Constante pour définir le pion du prénom1.

#define PION_B 'O'

Constante pour définir le pion du prénom2.

#define VIDE ''

Constante pour le caractère vide.

Documentation des définitions de type

typedef int grille[NBLIG][NBCOL]

Documentation des fonctions

void afficher (grille *g*, char *pion*, int *colonne*)

Affiche la grille.

Note

Status : Done

Paramètres

<i>g</i>	: Grille, représente la grille de jeu
<i>pion</i>	: Caractère, représente le pion à afficher au-dessus de la grille
<i>colonne</i>	: Entier, représente l'indice de la colonne au-dessus de laquelle le pion doit être affiché

int choisirColonne (grille *g*, char *pion*, int *colonne*)

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

Note

Status : Done

Paramètres

<i>g</i>	: grille, représente la grille de jeu
<i>pion</i>	: caractère, représente le pion à tester
<i>colonne</i>	: int, colonne de départ

Renvoie

int : renvoie le numéro de la colonne

booléen estVainqueur (grille *g*, int *ligne*, int *colonne*)

Indique si le pion situé dans la case repérée par les paramètres ligne et colonne a gagné la partie, c'est-à-dire s'il y a une ligne, une colonne ou une diagonale formée d'au moins 4 de ses pions (la ligne et la colonne passées en paramètres correspondent à la case où le joueur vient de jouer, c'est-à-dire la case à partir de laquelle il faut rechercher 4 pions successifs identiques).

Paramètres

<i>g</i>	: grille, représente la grille de jeu
<i>ligne</i>	: entiers représentant la case à partir de laquelle rechercher une série de 4 successifs identiques
<i>colonne</i>	: entier, entiers représentant la case à partir de laquelle rechercher une série de 4 successifs identiques

Renvoie

booléen : VRAI s'il y a une telle ligne de 4 à partir de la case indiquée, FAUX sinon.

void finDePartie (char *pion*, char *nom1*[20], char *nom2*[20])

Affiche le résultat d'une partie lorsqu'elle est terminée.

Note

Status : Done

Paramètres

<i>pion</i>	: caractère, qui représente le pion gagnant ou bien vide si match nul
-------------	---

bool grillePleine (grille *g*)

Test si toutes les cases de la grille sont occupées.

Note

Status : Done

Paramètres

<i>g</i>	: Grille, représente la grille de jeu
----------	---------------------------------------

Renvoie

Booléen : Vrai si toutes les cases de la grille sont occupées par les pions, Faux sinon

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

permet à un joueur de jouer son pion. La procédure fait appel à choisirColonne, afin que le joueur indique la colonne dans laquelle il veut jouer ; puis fait appel à trouverLigne pour définir la case où ajouter le pion.

Note

Status : Done

Paramètres

<i>g</i>	: grille , représente la grille de jeu
<i>pion</i>	: caractère, correspong au pion jouer
<i>ligne</i>	: entier, correspond à la ligne où est tombé le pion
<i>colonne</i>	: entier, correspond à la colonne où est tombé le pion

int main ()

Code Principale du puissance 4.

void pinitgrille (grille *g*)

Procédure qui va servir pour initialisé le tableau en tableau vide.

Note

Status : Done

Paramètres

<i>g</i>	: tableau "g" de type grille
----------	------------------------------

int trouverLigne (grille *g*, int *colonne*)

Consiste à trouver la première case non occupée de la colonne. Si la colonne est pleine, la fonction retourne -1.

Note

Status : Done

Paramètres

<i>g</i>	: grille, représente la grille de jeu
<i>entier</i>	: indice de la colonne dans laquelle le pion doit tomber

Renvoie

entier, indice de la liigne où le pion devra être ajouté ou -1 si la colonne est pleine.

Index

INDEX