

## **Puissance 4 Emma Clugery**

AUTHOR  
Version  
Fri Nov 25 2022



# Table of Contents

Table of contents



# Général

**Author:**

Clugery

**Version:**

1.0

**Date:**

21 Novembre 2022

Ce programme propose un jeu de puissance 4 pour 2 joueur en langage C. On retrouve dans ce programme l'utilisation de procédures et de fonctions qui permettent le bon fonctionnement du jeu.

# File Index

## File List

Here is a list of all files with brief descriptions:

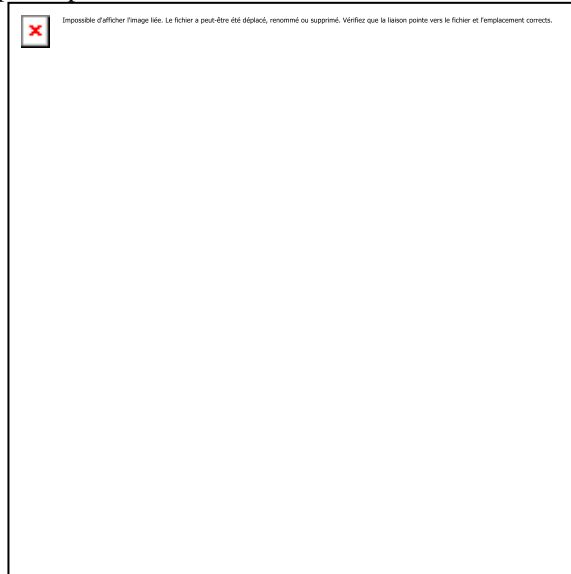
<b>puissance4.c</b>	4
<b>Puissance4Doxy.c</b>	9

# File Documentation

## puissance4.c File Reference

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

Include dependency graph for puissance4.c:



## Macros

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

## Typedefs

- `typedef char grille[NBLIG][NBCOL]`  
*Type tableau/grille à deux dimensions NBLIG et NBCOL caractères.*

## Functions

- `void initGrille (grille)`  
*Procédure qui initialise la grille.*
- `void afficher (grille, char, int)`  
*Affichage.*
- `bool grillePleine (grille)`  
*Fonction qui indique si la grille est pleine.*
- `void jouer (grille, char, int *, int *)`  
*Procédure qui permet de jouer grâce à l'appelle de la fonction choisirColonne et trouverLigne.*
- `float choisirColonne (grille, char, int)`  
*Fonction qui permet de choisir la colonne dans laquelle le pion doit tomber.*
- `float trouverLigne (grille, int)`  
*Fonction qui recherche les cases vides des colonnes.*
- `bool estVainqueur (grille, int, int)`

- void **finDePartie** (char)  
*Affiche le résultat d'une partie lorsque celle-ci est terminée.*
- int **main** ()  
*Programme principal.*

## Variables

- const char **PION\_A** = 'X'
- const char **PION\_B** = 'O'
- const char **VIDE** = '\_'
- const char **INCONNU** = ''
- const int **COLONNE\_DEBUT** = NBCOL/2

---

## Macro Definition Documentation

**#define NBCOL 7**

**#define NBLIG 6**

---

## Typedef Documentation

**grille**

Type tableau/grille à deux dimensions NBLIG et NBCOL caractères.

---

## Function Documentation

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

Affichage.

### Parameters:

<i>g</i>	: paramètre d'entrée/sortie qui représente la grille de caractère.
<i>pion</i>	: paramètre d'entrée qui représente le pion correspondant au joueur.
<i>colonne</i>	: paramètre d'entrée qui représente la colonne au dessus de laquelle le pion va s'afficher

Procédure qui commence par effacer l'écran puis qui affiche la grille avec les pions déjà joués. Elle affiche aussi au dessus de celle-ci le pion du joueur entrain de jouer pour que celui-ci puisse tomber dans la colonne choisie.

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

Fonction qui permet de choisir la colonne dans laquelle le pion doit tomber.

### Parameters:

<i>g</i>	: paramètre d'entrée/sortie qui représente la grille
<i>pion</i>	: paramètre d'entrée qui correspond au pion affiché au dessus de la grille.
<i>colonne</i>	: paramètre d'entrée qui correspond à la colonne où le pion doit être affiché au



	dessus
--	--------

#### Returns:

l'indice correspondant à la colonne choisie

Fonction qui permet au joueur de choisir sa colonne en tappant sur les touche 'q' et 'd' pour aller à gauche et à droite ainsi que la touche espace pour valider son choix. Le pion se déplace au dessus de la grille et tombe dans la colonne choisie. Si le joueur tape sur une autre touche ou encore si celui-ci continue d'appuyer sur les touches 'q' et 'd' alors que le pion est au extrémité, des messages d'erreurs correspondant s'afficheront.

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

Vérification d'une ligne verticale

Vérification d'une ligne horizontale

Vérification d'une ligne en diagonale en bas à droite vers le haut à gauche

Vérification d'une ligne en diagonale en bas à gauche vers le haut à droite

Vérification d'une ligne verticale

Vérification d'une ligne horizontale

Vérification d'une ligne en diagonale en bas à droite vers le haut à gauche

Vérification d'une ligne en diagonale en bas à gauche vers le haut à droite

**void finDePartie (char pion)**

Affiche le résultat d'une partie lorsque celle-ci est terminée.

#### Parameters:

pion	: paramètre d'entrée représentant le caractère du pion gagnant
------	--

Procédure affichant la fin d'une partie. Le message s'adapte au gagnant ou s'il n'y a pas de gagnant affiche "Match nul !".

**bool grillePleine (grille g)**

Fonction qui indique si la grille est pleine.

#### Parameters:

g	: paramètre d'entrée/sortie qui représente la grille
---	--

#### Returns:

true si la grille est pleine et false si celle-ci n'est pas pleine.

Fonction qui consiste à renvoyer un booléen vrai si la grille est pleine pour savoir si la partie doit continuer.

**void initGrille (grille g)**

Procédure qui initialise la grille.

#### Parameters:

g	: paramètre d'entrée/sortie qui représente la grille de caractère.
---	--

Initialisation des cases de chaque colonne et de chaque ligne de la grille par la constante VIDE.

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

Procédure qui permet de jouer grâce à l'appelle de la fonction choisirColonne et trouverLigne.

**Parameters:**

<i>g</i>	: paramètre d'entrée/sortie qui représente la grille
<i>pion</i>	: paramètre d'entrée qui correspond au joueur entrain de jouer.
<i>ligne</i>	: paramètre de sortie qui correspond à la ligne où le pion est tombé
<i>colonne</i>	: paramètre de sortie qui correspond à la colonne où le pion est tombé

Dans cette procédure, le pion peut être joué avec l'appel des fonctions choisirColonnes et trouverLigne qui servent à définir la colonne dans laquelle le joueur veut jouer et la case où le pion doit se placer.

**int main ()**

Programme principal.

Le programme principal traite les fonctions et les procédures nécessaires pour faire tourner la partie. Dans ce programme on retrouve une boucle while qui consiste à faire tourner la partie tant que personne n'a gagné et que la grille n'est pas pleine ainsi que des boucles if et else if permettant de faire jouer à tour de rôle. On retrouve aussi la variable vainqueur qui permet de savoir qui a gagné.

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

Fonction qui recherche les cases vides des colonnes.

**Parameters:**

<i>g</i>	: paramètre d'entrée/sortie qui représente la grille.
<i>colonne</i>	: paramètre d'entrée qui indique l'indice de la colonne dans laquelle le pion doit tomber.

**Returns:**

l'indice de la ligne dans laquelle le pion pourra être placé ou -1 si celle-ci est pleine.

Fonction qui consiste à trouver la première case inoccupée de la colonne pour ensuite renvoyer l'indice de la ligne où le pion pourra être placé. Si une colonne est pleine et qu'aucune case n'est disponible la fonction renverra -1.

## Variable Documentation

`const int COLONNE_DEBUT = NBCOL/2`

`const char INCONNU = ' '`

`const char PION_A = 'X'`

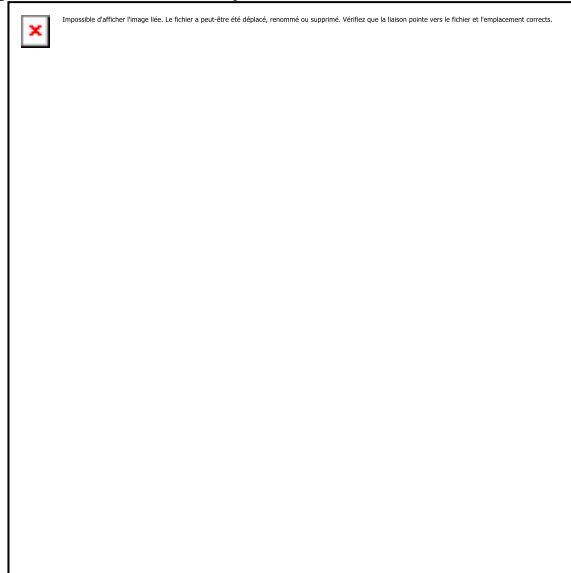
`const char PION_B = 'O'`

`const char VIDE = ' _'`

## Puissance4Doxy.c File Reference

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

Include dependency graph for Puissance4Doxy.c:



## Macros

- `#define NBLIG 6`  
*Nombre maximal de lignes dans la grille de jeu.*
- `#define NBCOL 7`  
*Nombre maximal de colonnes dans la grille de jeu.*
- `#define PION_A 'X'`  
*Désigne le caractère attribué au joueur PION\_A.*
- `#define PION_B 'O'`  
*Désigne le caractère attribué au joueur PION\_B.*
- `#define VIDE '_'`  
*Désigne le caractère attribué à VIDE.*
- `#define INCONNU ''`  
*Constante INCONNU qui est appliquée tant que personne n'a gagné et que la partie n'est pas terminée.*
- `#define DROITE 'd'`  
*Constante qui permet de déplacer le pion à droite par la touche 'q' entrée au clavier.*

## Typedefs

- `typedef char grille[NBLIG][NBCOL]`

## Functions

- `void initGrille (grille)`
- `void afficher (grille, char, int)`
- `bool grillePleine (grille)`
- `void jouer (grille, char, int *, int *)`

- int **choisirColonne** (grille, char, int)
- int **trouverLigne** (grille, int)  
*Fonction qui recherche les cases vides des colonnes.*
- bool **estVainqueur** (grille, int, int)
- void **finDePartie** (char)  
*Affiche le résultat d'une partie lorsque celle-ci est terminée.*
- int **main** ()

---

## Macro Definition Documentation

### **#define DROITE 'd'**

Constante qui permet de déplacer le pion à droite par la touche 'q' entrée au clavier.

### **#define INCONNU ''**

Constante INCONNU qui est appliquée tant que personne n'a gagné et que la partie n'est pas terminée.

### **#define NBCOL 7**

Nombre maximal de colonnes dans la grille de jeu.

### **#define NBLIG 6**

Nombre maximal de lignes dans la grille de jeu.

### **#define PION\_A 'X'**

Désigne le caractère attribué au joueur PION\_A.

### **#define PION\_B 'O'**

Désigne le caractère attribué au joueur PION\_B.

### **#define VIDE '\_'**

Désigne le caractère attribué à VIDE.

---

## Typedef Documentation

### **typedef char grille[NBLIG][NBCOL]**

---

## Function Documentation

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

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

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

Vérification d'une ligne verticale

Vérification d'une ligne horizontale

Vérification d'une ligne en diagonale en bas à droite vers le haut à gauche

Vérification d'une ligne en diagonale en bas à gauche vers le haut à droite

**void finDePartie (char *pion*)**

Affiche le résultat d'une partie lorsque celle-ci est terminée.

### Parameters:

<i>pion</i>	: paramètre d'entrée représentant le caractère du pion gagnant
-------------	--

Procédure affichant la fin d'une partie. Le message s'adapte au gagnant ou s'il n'y a pas de gagnant affiche "Match nul !".

**bool grillePleine (grille *g*)**

**void initGrille (grille *g*)**

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

**int main ()**

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

Fonction qui recherche les cases vides des colonnes.

### Parameters:

<i>g</i>	: paramètre d'entrée/sortie qui représente la grille.
<i>colonne</i>	: paramètre d'entrée qui indique l'indice de la colonne dans laquelle le pion doit tomber.

### Returns:

l'indice de la ligne dans laquelle le pion pourra être placé ou -1 si celle-ci est pleine.

Fonction qui consiste à trouver la première case inoccupée de la colonne pour ensuite renvoyer l'indice de la ligne où le pion pourra être placé. Si une colonne est pleine et qu'aucune case n'est disponible la fonction renverra -1.

# **Index**

INDEX