

**Les Génies du Ponant**

# **Machine à courber les bananes**

**Version 2.7.4 - 18/01/2024**

# Description du code

---

Le programme sert à trier le contenu d'un tableau par la méthode du tri par selection et qui affiche le tableau avant et après le tri par selection Un exemple de code en C avec des structures, constantes, fonctions et procédures. Ce programme incroyable correspond au jeu du puissance 4. Chaque joueur joue à tour de rôle jusqu'à ce que l'un d'eux aligne 4 pions à lui dans la grille, auquel cas il gagne.

# DEFINES

---

- N 10 (*La taille du tableau*)
- ZERO 0 (*la valeur 0*)
- MAX\_POINTS 14 (*Nombre max de point*)
- ROWS 6 (*Nombre de lignes dans la grille*)
- COLS 7 (*Nombre de colonnes dans la grille*)

# STRUCTURES

---

## client

Structure d'un client

- int no\_client (*numéro du client*)
- char nom[20] (*nom du client*)
- char prenom[20] (*prenom du client*)
- int age (*age du client*)

## ocean

Structure d'un ocean

- char nom[20] (*nom de l'ocean*)
- int superficie (*superficie de l'ocean*)

## Point2D

Structure représentant un point dans l'espace 2D.

- float x (*Coordonnée x du point.*)
- float y (*Coordonnée y du point.*)

## Rectangle

Structure représentant un rectangle.

- Point2D coinSuperieurGauche (*Coin supérieur gauche du rectangle.*)
- Point2D coinInferieurDroit (*Coin inférieur droit du rectangle.*)

## Player

Structure de joueur

- char name[50] (*Nom du joueur*)
- char symbol (*Symbole du pion du joueur*)

## Board

Structure de tableau de jeu

- char grid[ROWS][COLS] (*Grille du Jeu*)

# GLOBALES

---

- int MAX\_NOTE = 20; (*Note maximale*)
- int notes[NB\_NOTES] (*Liste des notes d'un étudiant*)
- float PI = 3.14159265359; (*Constante représentant la valeur de pi.*)
- int LARGEUR\_MAX = 100; (*Constante représentant la largeur maximale d'un rectangle.*)
- int currentPlayer (*Joueur actuel*)
- Player player1 (*Structure du joueur 1*)
- Player player2 (*Structure du joueur 2*)
- Board gameBoard (*Structure du tableau de jeu*)

# FONCTIONS

---

## tri\_selection

le tableau passé en paramètre est trié par sélection si le tableau est constitué de plus qu'une case remplie.

Il parcourt toute la partie non trié du tableau (à gauche) en comparant les valeurs entre elles, avant de les placer les plus à gauche de la zone non triée.

### Paramètres

- tab T *(le tableau à trier)*

### Valeur Renvoyé

- none

## existe

à l'aide d'une boucle for, on parcourt le tableau afin de vérifier si la valeur passée en paramètre se trouve dans le tableau passée aussi en paramètre.

On arrête la recherche dans le tableau si :

- la valeur est trouvée dans le tableau
- OU le tableau a fini d'être parcouru

### Paramètres

- tab T *(le tableau à parcourir)*
- int nb *(la valeur à trouver dans le tableau)*

### Valeur Renvoyé

- bool *(true si la valeur est trouvé, false sinon)*

## somme

on additionne 2 entiers entre eux pour en retourner le résultat

### Paramètres

- int a *(le 1er entier)*
- int b *(le 2e entier)*

### Valeur Renvoyé

- int (*la somme des 2 entiers*)

## afficherPoint

affiche les coordonnées d'un point grâce a la structure Point2D qui est pris en paramètre

### Paramètres

- Point2D point (*Coordonné x et y d'un point*)

### Valeur Renvoyé

- none

## calculerDistance

Calcule la distance entre deux points mis en paramètre en fesant une soustraction les coordonné x des points puis y et ensuite les additionne

### Paramètres

- point2D a (*Coordonné x et y d'un point b*)

### Valeur Renvoyé

- float (*: La distance entre les deux points*)

## calculerAire

Calcule l'aire d'un rectangle mis en paramètre en calculant la largeur puis la hauteur et ensuite les multiplie entre

### Paramètres

### Valeur Renvoyé

- float (*largeur \* float hauteur Le calcul de l'aire*)

## main

-initialise deux point(pointA, PointB) de type Point2D -affiche les coordonné des deux points -calcul la distance entre les points -Création d'un rectangle monRectangle de type Rectangle -Calcul de l'aire du rectangle

### Paramètres

### Valeur Renvoyé

- none

## initializeGame

Fonction qui initialise le tableau de jeu. Chaque case est remplie par une chaîne de caractère vide. Initialise également le nom de chaque joueur ainsi que le symbole qui représente son pion.

**Paramètres**

**Valeur Renvoyé**

- none

**printBoard**

Affiche la grille sous la forme d'un tableau avec des séparateurs verticales "|" pour la mise en page

**Paramètres**

**Valeur Renvoyé**

- none

**dropDisc**

Fonction qui permet au joueur dont c'est le tour de placer un pion au sommet de la colonne sélectionnez par column. Renvoie true si la colonne est libre et false sinon.

**Paramètres**

- int column (*indice de la colonne*)

**Valeur Renvoyé**

- bool (*colonne est libre.*)

**checkWin**

Fonction qui parcourt la grille et vérifie si 4 pions sont alignés. Si c'est le cas, renvoie true et fait gagner le joueur dont c'est le tour. Sinon, retourne false.

**Paramètres**

**Valeur Renvoyé**

- bool (*un joueur a aligné 4 pion*)

**isBoardFull**

Fonction qui parcourt la grille et vérifie si le tableau est complet. Si c'est le cas, alors renvoie true et la partie est considérée comme un match nul. Sinon, renvoie false.



## Paramètres

## Valeur Renvoyé

- bool (*un joueur a aligné 4 pion*)