

Jeu de Yams

LAMBERT Chloé
Version
Dimanche 28 Novembre 2021

Table des matières

Table of contents

Index des fichiers

Liste des fichiers

Liste de tous les fichiers avec une brève description :

YAMS.c (Ce programme permet de jouer une partie de YAMS à deux)3

Documentation des fichiers

Référence du fichier YAMS.c

Ce programme permet de jouer une partie de YAMS à deux.

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

Graphes des dépendances par inclusion de YAMS.c:

IMAGE

Macros

```
#define NOMBRE_DES 5
```

Raccourci associé au nombre de dés présents dans le jeu.

```
#define NOMBRE_JOUEURS 2
```

Raccourci associé au nombre de joueurs pour une partie.

Définitions de type

```
typedef int t_des[NOMBRE_DES]
```

Représente les dés que le joueur va lancer.

```
typedef int t_score[13]
```

Représente le tableau de score.

Fonctions

```
void afficher_des (t_des des)
```

Cette procédure sert à afficher les dés lancés.

```
void afficher_scores (t_score scores, char nom_joueur[20])
```

Cette procédure sert à afficher les scores dans le tableau.

```
bool brelan (t_des des)
```

Cette fonction détermine si la combinaison effectuée par le joueur est un Brelan.

```
bool carre (t_des des)
```

Cette fonction détermine si la combinaison effectuée par le joueur est un Carré

```
void combinaisons_disponibles (t_des des, t_score scores)
```

Cette procédure permet de déterminer quelles sont les combinaisons disponibles dans la grille.

void **combinaisons_possibles** (t_des des, int combinaisons[13], int *nombre_combinaisons)
Cette procédure détermine quelles combinaisons sont possibles à marquer par le joueur en fonction de son lancer.

void **compter_des** (t_des des, int occurrences[6])
Cette procédure va compter les occurrences des faces des dés pour déterminer les combinaisons.

int **compter_points** (int combinaison, t_des des)
Cette fonction sert à compter les points que le joueur a fait avec son lancer.

void **entrer_booleen** (char *message, bool *reponse)
void **entrer_chaine** (char *message, char *reponse)
int **entrer_combinaison** (t_score scores)
Cette fonction permet d'entrer la combinaison que le joueur a réalisé

void **entrer_points** (t_score scores, int combinaison, int points)
Cette procédure sert à entrer les points marqués dans la feuille de score.

void **entrer_suite** (char *message, t_des reponse)
bool **full_house** (t_des des)
Cette fonction détermine si la combinaison effectuée par le joueur est un Full House.

bool **grande_suite** (t_des des)
Cette fonction détermine si la combinaison effectuée par le joueur est une Grande Suite.

void **lancer_des** (int nombre, t_des des)
Cette procédure sert à lancer les dés.

int **main** ()
bool **petite_suite** (t_des des)
Cette fonction détermine si la combinaison effectuée par le joueur est une Petite Suite.

void **trier_des** (t_des des, t_des resultat)
Cette procédure trie les dés.

int **trouver_vainqueurs** (t_score scores[NOMBRE_JOUEURS])
Cette fonction détermine quel joueur a remporté la partie.

bool **yams** (t_des des)
Cette fonction détermine si la combinaison effectuée par le joueur est un Yams.

Variables

const char **NOMS_COMBINAISONS** [13][13]
Représente les 13 combinaisons possibles du jeu.

Description détaillée

Ce programme permet de jouer une partie de YAMS à deux.

Auteur

LAMBERT Chloé

Version

0.1

Date

2021-11-28

Copyright

Copyright (c) 2021

Documentation des macros

#define NOMBRE_DES 5

Raccourci associé au nombre de dés présents dans le jeu.

#define NOMBRE_JOUEURS 2

Raccourci associé aux nombre de joueurs pour une partie.

Documentation des définitions de type

typedef int t_des[NOMBRE_DES]

Représente les dés que le joueur va lancer.

typedef int t_score[13]

Représente le tableau de score.

Ce tableau va permettre le stockage des scores marqués par les joueurs, à savoir les 13 combinaisons possibles

Documentation des fonctions

void afficher_des (t_des des)

Cette procédure sert à afficher les dés lancés.

Paramètres

<i>des</i>	Les 5 dés lancés
------------	------------------

Consiste à afficher la valeur des dés du joueur après un lancer

void afficher_scores (t_score scores, char nom_joueur[20])

Cette procédure sert à afficher les scores dans le tableau.

Paramètres

<i>scores</i>	La feuille de score
<i>nom_joueur</i>	Correspond au nom du joueur dont c'est au tour de jouer, chaque joueur ayant sa feuille de score

Consiste à afficher les scores du joueur dans les cases correspondantes du tableau de la feuille de score

bool brelan (t_des des)

Cette fonction détermine si la combinaison effectuée par le joueur est un Brelan.

Paramètres

<i>des</i>	Les 5 dés lancés par le joueur
------------	--------------------------------

Renvoie

true si la combinaison est un Brelan

false dans les autres cas

Consiste à déterminer si la combinaison réalisée après le lancer du joueur est un Brelan
C'est à dire 3 dés identiques

bool carre (t_des des)

Cette fonction détermine si la combinaison effectuée par le joueur est un Carré

Paramètres

<i>des</i>	Les 5 dés lancés par le joueur
------------	--------------------------------

Renvoie

true si la combinaison est un Carré

false dans les autres cas

Consiste à déterminer si la combinaison réalisée après le lancer du joueur est un Carré
C'est à dire 4 dés identiques

void combinaisons_disponibles (t_des des, t_score scores)

Cette procédure permet de déterminer quelles sont les combinaisons disponibles dans la grille.

Paramètres

<i>des</i>	Les dés lancés par le joueur
<i>scores</i>	La feuille de score

Consiste à vérifier quelles sont les combinaisons disponibles dans la grille, celles que le joueur n'a pas encore validées Par exemple, si le joueur réalise un score de 3 mais qu'il a déjà validé la case, alors celle-ci n'apparaîtra pas

void combinaisons_possibles (t_des des, int combinaisons[13], int * nombre_combinaisons)

Cette procédure détermine quelles combinaisons sont possibles à marquer par le joueur en fonction de son lancer.

Paramètres

<i>des</i>	Les 5 dés lancés par le joueur
<i>combinaisons</i>	Les 13 combinaisons possibles
<i>nombre_combinaisons</i>	Le nombre de combinaisons possibles

Consiste à déterminer quelles combinaisons sont possibles à valider par le joueur, en effet le résultat d'un lancer peut correspondre à plusieurs combinaisons, ce sont ces combinaisons qui seront indiquées

void compter_des (t_des des, int occurrences[6])

Cette procédure va compter les occurrences des faces des dés pour déterminer les combinaisons.

Paramètres

<i>des</i>	Les 5 dés lancés par le joueur
<i>occurrences</i>	Les 6 faces des dés

Consiste à compter les occurrences des faces des dés de chaque lancer pour déterminer quelles combinaisons ont été réalisées

int compter_points (int combinaison, t_des des)

Cette fonction sert à compter les points que le joueur a fait avec son lancer.

Paramètres

<i>combinaison</i>	Le numéro, ou indice de la combinaison dans le tableau
<i>des</i>	Les 5 dés lancés par le joueur

Renvoie

int le score obtenu

Consiste à compter les points marqués par le joueur avec son lancer en fonction des combinaisons réalisées s'il y en a ou sinon par les points indiqués sur les dés

void entrer_booleen (char * message, bool * reponse)

void entrer_chaine (char * message, char * reponse)

int entrer_combinaison (t_score scores)

Cette fonction permet d'entrer la combinaison que le joueur a réalisé

Paramètres

<i>scores</i>	La feuille de score qui contient les 13 combinaisons possibles
---------------	----------------------------------------------------------------

Renvoie

int l'indice de la combinaison dans le tableau des scores

Consiste à entrer le nom de combinaison que le joueur a réalisé et qu'il souhaite valider

void entrer_points (t_score scores, int combinaison, int points)

Cette procédure sert à entrer les points marqués dans la feuille de score.

Paramètres

<i>scores</i>	La feuille de score
<i>combinaison</i>	L'indice de la combinaison dans le tableau
<i>points</i>	Points marqués par le joueur

Consiste à entrer le score marqué par le joueur dans le tableau/feuille de score

void entrer_suite (char * message, t_des reponse)

bool full_house (t_des des)

Cette fonction détermine si la combinaison effectuée par le joueur est un Full House.

Paramètres

<i>des</i>	Les 5 dés lancés par le joueur
------------	--------------------------------

Renvoie

true si la combinaison est un Full House

false dans les autres cas

Consiste à déterminer si la combinaison réalisée après le lancer du joueur est un Full House C'est à dire une Paire + un Breton

bool grande_suite (t_des des)

Cette fonction détermine si la combinaison effectuée par le joueur est une Grande Suite.

Paramètres

<i>des</i>	Les 5 dés lancés par le joueur
------------	--------------------------------

Renvoie

true si la combinaison est une Grande Suite

false dans les autres cas

Consiste à déterminer si la combinaison réalisée après le lancer du joueur est une Grande Suite C'est à dire 5 dés dont la valeur se suit

void lancer_des (int nombre, t_des des)

Cette procédure sert à lancer les dés.

Paramètres

<i>nombre</i>	Le nombre de dés qui est à lancer, il peut être de 5 ou moins si le joueur décide de relancer certains dés
<i>dés</i>	Les 5 dés à lancer

Consiste à lancer les dés d'un joueur

int main ()

bool petite_suite (t_des des)

Cette fonction détermine si la combinaison effectuée par le joueur est une Petite Suite.

Paramètres

<i>dés</i>	Les 5 dés lancés par le joueur
------------	--------------------------------

Renvoie

true si la combinaison est une Petite Suite

false dans les autres cas

Consiste à déterminer si la combinaison réalisée après le lancer du joueur est une Petite Suite C'est à dire 4 dés dont la valeur se suit

void trier_des (t_des des, t_des resultat)

Cette procédure trie les dés.

Paramètres

<i>dés</i>	Les 5 dés lancés par le joueur
<i>resultat</i>	Le résultat, les dés triés

int trouver_vainqueurs (t_score scores[NOMBRE_JOUEURS])

Cette fonction détermine quel joueur a remporté la partie.

Paramètres

<i>scores</i>	La feuille de score
---------------	---------------------

Renvoie

int l'indice du vainqueur, si c'est le joueur 1 ou 2

Consiste à dire lequel des deux joueurs à gagner la partie, celui qui à le score le plus élevé

bool yams (t_des des)

Cette fonction détermine si la combinaison effectuée par le joueur est un Yams.

Paramètres

<i>dés</i>	Les 5 dés lancés par le joueur
------------	--------------------------------

Renvoie

true si la combinaison est un Yams

false dans les autres cas

Consiste à déterminer si la combinaison réalisée après le lancer du joueur est un Yams
C'est à dire 5 dés de la même valeur

Documentation des variables

const char NOMS_COMBINAISONS[13][13]

```
Valeur initiale := {  
    "Total de 1", "Total de 2", "Total de 3",  
    "Total de 4", "Total de 5", "Total de 6",  
    "Brelan", "Carré", "Full House",  
    "Petite suite", "Grande suite", "Yams",  
    "Chance"  
}
```

Représente les 13 combinaisons possibles du jeu.

Ce tableau permet de stocker les 13 combinaisons jouables associées à leur noms

Index

INDEX