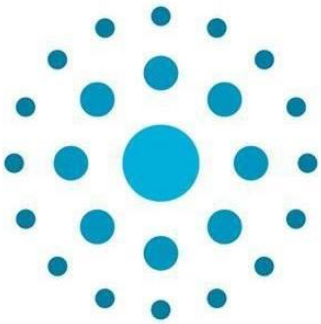
Rapport du projet puissance 4 en langage c



UNIVERSITÉ
CÔTE D'AZUR

Réalisé par : Brahim Saadaoui

Années universitaire : 2020/2021

Encadré par : Mme Marie Pelleau

Introduction au Jeu:

Notre but dans ce projet était de pouvoir jouer au jeu « Puissance 4 » à un ou deux joueurs ayant pour but d'aligner 4 jetons verticalement, horizontalement ou en oblique soit par les X ou O , la grille de jeu aura 7 lignes et 7 colonnes et de pouvoir sauvegarder puis charger une partie.

À chaque tour, chaque joueur peut ajouter un jeton dans une colonne, ce jeton ira à la position la plus basse dans cette colonne.
Si toutes les cases de la grille sont remplies et qu'aucun joueur n'a réussi à aligner 4 jetons, la partie est déclarée nulle.

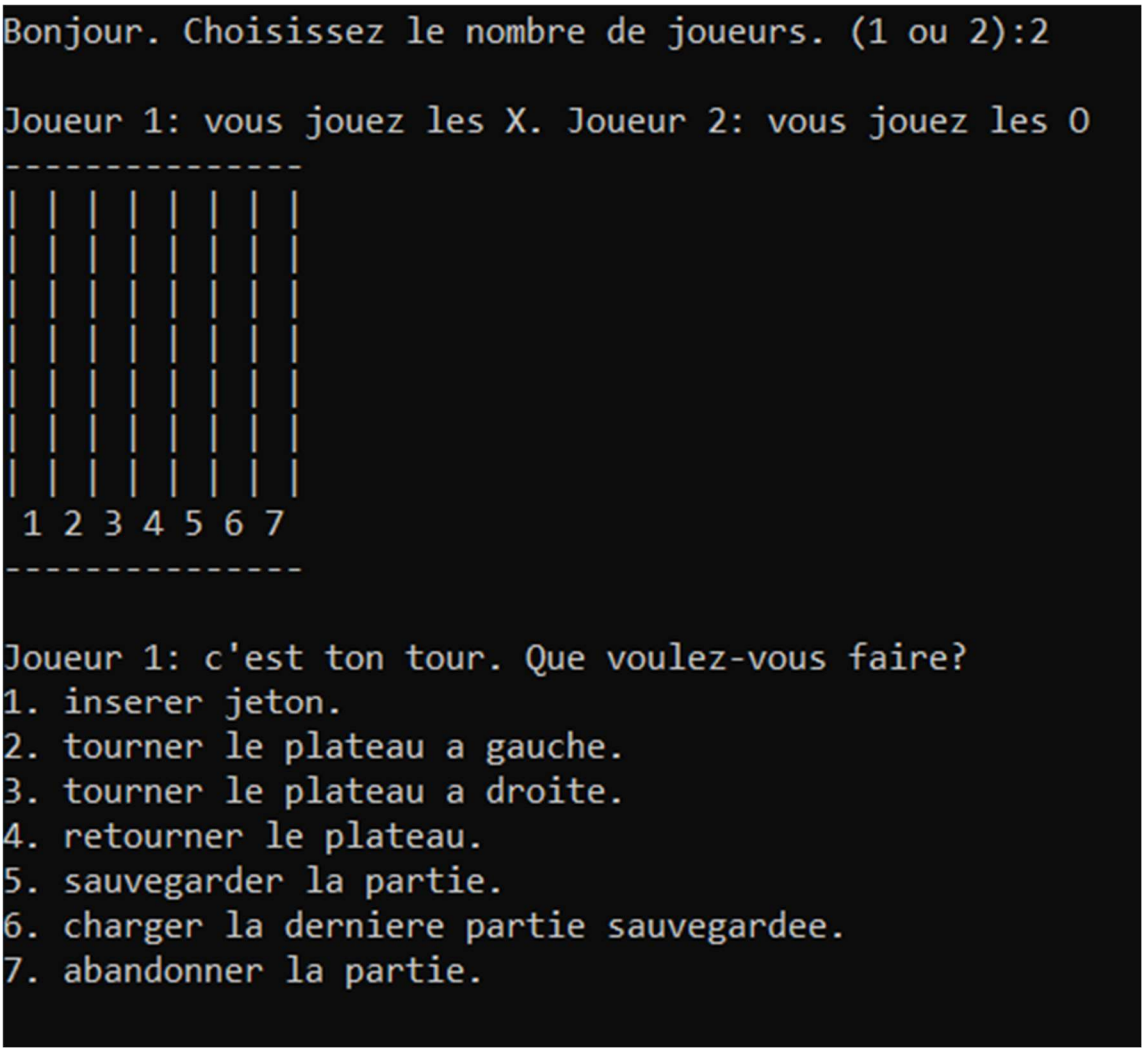
Chaque joueur aussi la possibilité de :

-Placer un jeton dans une colonne,

-Tourner le plateau de 90 degrés sur la gauche,

-Tourner le plateau de 90 degrés sur la droite,

-Ou retourner le plateau.



Première étape : le jeu

Le programme commence par initialiser la grille à partir de la fonction initialise() (tous les caractères qui la compose sont des espaces au début, symbolisant une case vide) et l'afficher en utilisant la fonction affiche_tableau().

Ensuite, il est demandé au premier joueur d'effectuer une action. Cela est réalisé via la fonction jeu() qui demande de sélectionner le nombre de joueurs (1 ou 2), si l'utilisateur sélectionne l'option 1 , le 2eme joueur sera l'ordinateur , sinon le premier joueur joue avec un autre humain(ami) , ici le premier joueur joue avec les X , l'autre avec O. Si le joueur entre autres chose un message d'erreur s'affiche : "CHOIX INVALIDE, TAPER UNE TOUCHE POUR QUITTER".

Dans le cas où le numéro de nombre de joueurs est valide, le programme donne 7 choix au joueur :

1. insérer jeton.
2. tourner le plateau à gauche.
3. tourner le plateau à droite.
4. retourner le plateau.
5. sauvegarder la partie.
6. charger la dernière partie sauvegardée.
7. abandonner la partie

Pour insérer un jeton il suffit d'utiliser la fonction inserer_jeton() en sélectionnent l'option 1 , qui prend en paramètre un tableau , numéro de colonne sélectionner et le caractère de jeton , la fonction inserer_jeton() vérifier au premier lieu si une case du tableau est vide , en suite la remplir avec un jeton, si une colonne est rempli , un message s'affiche : "ERREUR: colonne pleine , sélectionner une autre colonne. "

Pour tourner le plateau soit a gauche ou droite , il nous faut crée une nouvelle grille, puis parcourir chaque ligne de la grille initiale , quand on rencontre un jeton , on le remplace dans la ligne correspondante de la nouvelle grille.

Pour retourner le plateau , il nous faut crée une nouvelle grille, puis parcourir chaque ligne de la grille initiale , quand on rencontre un jeton , on le remplace dans la position la plus bas dans la même ligne correspondante de la nouvelle grille.

La fonction statut_jeu() entre alors en action et détermine si quatre jetons adjacents sont présents ou non ou si la grille est complète en faisant appelle à la fonction verifier_grille_complete() . Si aucune des deux conditions n'est remplie, c'est reparti pour un tour sinon la boucle est quittée et le programme se termine.

La fonction statut_jeu() , utilise 4 boucles pour vérifier si les 4 jetons son adjacents (score==4) :

La première boucle (déplacement horizontal) parcourt la grille , et calcule le nombre de jetons adjacents de la ligne (deux déplacements : un vers la gauche et un vers la droite) , si (score==4) retourner 1 sinon retourner 0.

La deuxième boucle (déplacement vertical) parcourt aussi la grille , et calcule le nombre de jetons adjacents de la colonne, si (score==4) retourner 1 sinon retourner 0.

et des deux obliques (deux déplacements pour chacune d'entre elles) :

-un déplacement diagonal.

-un déplacement inverse de la diagonal.

Deuxième étape : IA

1)Introduction :

Dans cette seconde partie, notre objectif est de construire une petite intelligence artificielle. Notre programme va donc devoir demander si un ou deux joueurs sont présents et jouer à la place du second joueur s'il n'y en a qu'un seul.

L'algorithme au début sélectionne l'une des 4 option au hasard:

1. inserer jeton.
2. tourner le plateau a gauche.
3. tourner le plateau a droite.
4. retourner le plateau.

Si l'algorithme à sélectionner l'option (tourner le plateau a gauche ou tourner le plateau a droite ou retourner le plateau) , l'option sélectionner s'exécute simplement en faisant appelle au boucles qui sont déjà défini.

Sinon l'algorithme sélectionne la première option, l'insertion de jeton ce fait aléatoirement pour chaque emplacement possible.

2)Tirage au sort :

La plupart de ceux-ci fonctionnent à partir d'un nombre de départ qui va déterminer tout le reste de la suite. C'est ce système qui a été choisi par la bibliothèque standard du C. Deux fonction sont nécessaire srand() et rand().

```
void srand(time(NULL));
int rand(void);
```

srand(time(NULL)) initialise la fonction srand sur le temps actuel.

La fonction rand() retourne un nombre pseudo-aléatoire compris entre zéro et RAND_MAX (qui est une constante également définie dans l'en-tête <stdlib.h>) suivant le nombre de depart fournie à la fonction srand().

C'est ici que la fonction time() (déclarée dans l'en-tête <time.h>) entre en jeu.

La fonction time() retourne la date actuelle. Le plus souvent, il s'agit du nombre de secondes écoulé depuis une date fixée arbitrairement.

Pour obtenir des nombres différents à chaque exécution, nous pouvons donc appeler srand() avec le retour de la fonction time() converti en entier non signé.

2.1)Tirer un nombre dans un intervalle donné :

Il est possible de générer des nombres dans un intervalle précis en procédant en deux temps. Tout d'abord, il nous est nécessaire d'obtenir un nombre compris entre zéro inclus et un exclus. Pour ce

faire, nous pouvons diviser le nombre pseudo-aléatoire obtenu par la plus grande valeur qui peut être retournée par la fonction rand() augmentée de un. Celle-ci nous est fournie via la RAND_MAX qui est définie dans l'en-tête <stdlib.h>.

(rand() % (7 - 1 + 1) + 1);

Troisième et dernière étape : un système de sauvegarde/restauration

À présent, durant la partie, un joueur peut entrer le numéro 5 afin d'effectuer une sauvegarde de la partie courante au sein d'un fichier qui s'appelle «sauvegarde.txt». Les données sont sauvegardées sous forme de texte avec, dans l'ordre : le tour de joueur, le contenu de la grille. Une fois la sauvegarde effectuée, il faut appuyer sur l'option 7 (abandonner la partie), pour bien sauvegarder la partie.

Il peut également entrer le numéro 6 pour charger une partie précédemment sauvegardée.