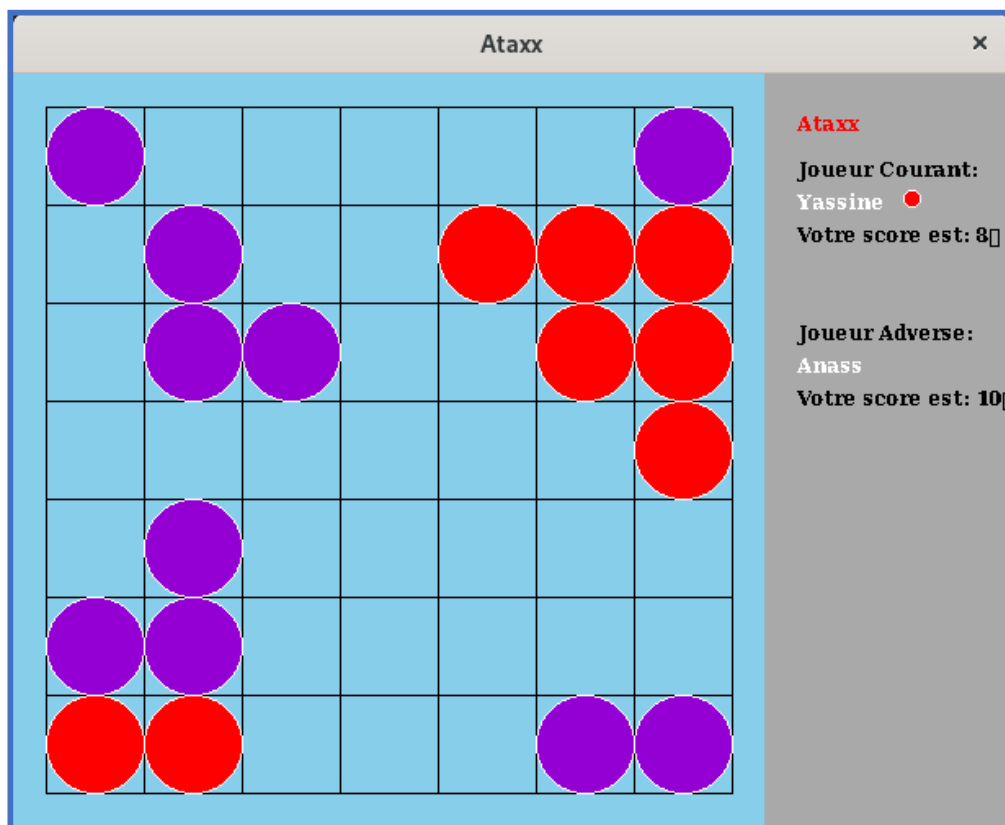


Ataxx

RAPPORT DU PROJET

РАББОКЛ ДУ ПРОЕКТ



Introduction

Le Reversi est un jeu de réflexion (stratégie et tactique), également connu sous le nom d'Othello, qui se joue à deux joueurs sur un plateau quadrillé de 8 cases sur 8 cases à l'aide de jetons blancs sur une face et noirs sur l'autre.

Le but du jeu étant d'obtenir le plus grand nombre possible de jetons de la couleur qui vous est attribuée.

Dans ce projet, on a implémenté ce fameux jeu avec le langage C en version terminal et graphique à l'aide de la bibliothèque MLV

I. Présentation du Projet

Le but est d'implémenter le jeu en version graphique et terminal en respectant toutes les contraintes suivantes :

- Un coup consiste à placer un pion de sa couleur sur une des huit cases voisines d'un pion adverse déjà posé.
- La case doit être vide pour que l'on puisse poser un pion dessus
- Après avoir placé un pion, tous les pions adverses sur les cases voisines de la case du pion posé changent de couleur en devenant de la couleur du pion posé.
- La partie est finie lorsque le plateau est rempli ou si un joueur ne possède plus de pions dans le plateau.

II. Guide d'utilisation

Pour pouvoir jouer au jeu suffit de compiler le fichier « main.c » après l'exécuter avec l'une des commandes suivantes (-g -h) ou (-g -o) ou (-a -h) ou (-a -o) avec :

- a : Mode terminal
- g : Mode graphique
- o : Jouer contre un ordinateur
- h : Jouer contre un autre joueur humain

```
(base) padawan@DebianUPEM2019:~/Bureau/projetC/projet$ ./a.out -a -h
Quel est le nom du premier joueur (symbol o) :Yassine
Quel est le nom du second joueur (symbol x) :Anass
* * * * *
* O . . . . X *
* . . . . . *
* . . . . . *
* . . . . . *
* . . . . . *
* . . . . . *
* . . . . . *
* X . . . . O *
* * * * *
Yassine (o), veuillez saisir les coordonnées où jouer (entre 1 et 7) :■
```

Pour placer un pion, il suffit d'indiquer les coordonnées de la case d'abord on indique la ligne après la colonne comme ceci :

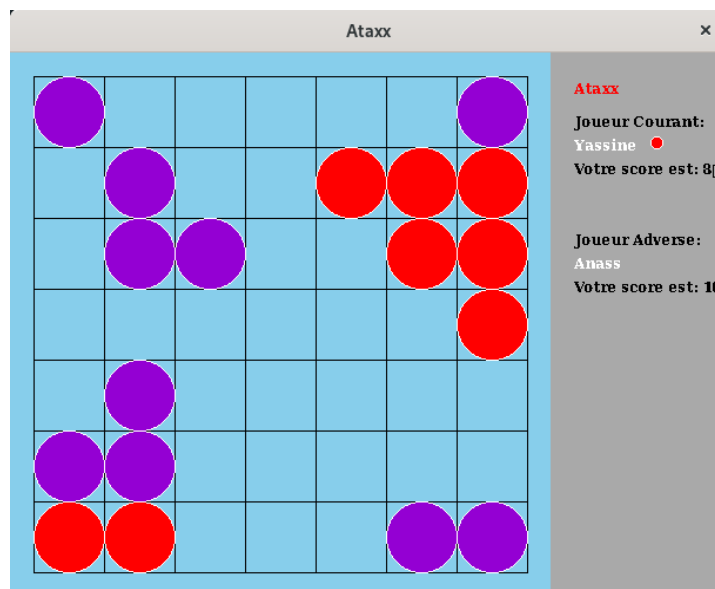
```

Quel est le nom du premier joueur (symbol o) :Yassine
Quel est le nom du second joueur (symbol x) :Anass
* * * * *
* O . . . . X *
* . . . . . *
* . . . . . *
* . . . . . *
* . . . . . *
* . . . . . *
* . . . . . *
* X . . . . O *
* * * * *
Yassine (o), veuillez saisir les coordonnées où jouer (entre 1 et 7) :2 6
* * * * *
* O . . . . O *
* . . . . O *
* . . . . . *
* . . . . . *
* . . . . . *
* . . . . . *
* . . . . . *
* X . . . . O *
* * * * *
Score actuel : Yassine(o) 4 – Anass(x) 1
Anass (x), veuillez saisir les coordonnées où jouer (entre 1 et 7) :

```

Le score du jeu est affiché après chaque emplacement d'un pion.

Pour le mode graphique, pour placer un pion il faut cliquer sur la case correspondante (le pion ne sera pas placé s'il vérifie pas les conditions du jeu) comme ceci :



Au début du jeu on peut choisir si le joueur veut jouer une nouvelle partie ou bien jouer la partie déjà sauvegardé une fenêtre apparaîtra au début indiquant ceci :



III. Structure du code

Nous avons utilisé les structures données dans l'énoncé mais on a dû rajouter une nouvelle structure `Coord` qui contient le couple de coordonnées où l'utilisateur a cliqué pour faciliter la manipulation de ce dernier dans le but de détecter ces voisins.

Nous avons travaillé beaucoup avec les adresses pour éviter les copies du plateau à chaque fois qu'on l'utilise dans une fonction et on a implémenté une option de sauvegarde du plateau au cas où le joueur voulait restaurer sa partie.

A. Fonctions générales :

Une grande majorité de fonctions sont communes entre le mode graphique et le mode terminal (fonction d'emplacement des pions – fonction de changement de pions – fonction de détection des voisins – fonction de sauvegarde du fichier- fonction d'initialisation du plateau comme indiqué dans l'énoncé).

Ce sont des fonctions qui s'occupent du fonctionnement du jeu.

- **Void echangeTour (Plateau *p)** : cette fonction s'occupe de changement du joueur courant en effectuant un échange entre les deux joueurs dans le but de toujours considérer le premier joueur dans le tableau comme le joueur courant.
- **Void detecterVoisins (Coord *voisins, int *nbVoisins, int lig, int col)** : Cette fonction s'occupe de la détection des voisins de la case saisie par l'utilisateur en les affectant à un tableau de voisins qui est sous la forme des couples pour chaque élément grâce à la structure `Coord`.
- **Int coupValide (Plateau *p, int lig, int col)** : Cette fonction est la responsable de la vérification du coup du joueur en faisant appel à la fonction détecter voisins et elle procède à une vérification de chaque voisins s'elle valide grâce à la **fonction coordonneeValide** et

EmplacementVide après on effectue un changement de pions adverse avec la **fonction changementPions** qui s'occupe du changement des pions adverses.

- **Void resteUnTypePions (Plateau *p) et Void plateauRemplit (Plateau *p)** ce sont les fonctions responsables des conditions de victoire du jeu

B. Mode terminal

A l'aide des fonctions générales l'implémentation de la partie terminale était assez facile vu la structure du jeu est déjà prête les principales fonctions d'affichage du mode terminale étaient :

- **Void affichePlateauAscii (Plateau *p)** : la fonction responsable d'affichage du plateau à chaque tour pareil que les autres fonctions **afficheGagnant** et **afficheScore** même principe.
- **Void jeuAsciiBinome () et Void jeuAsciiPc ()** : les fonctions du mode terminales qui effectuent des boucles du jeu tant qu'il n'est pas fini.
- **Void choixEmplacement (Plateau *p, int *lig, int *col)** : fonction responsable de l'affectation des coordonnées saisies par l'utilisateur dans le plateau du jeu.
- **Void initJoueurHumain (Plateau *p, Joueur *j1, Joueur *j2)** : effectue la saisie des noms des joueurs pareils pour la fonction qui s'occupe de l'ordinateur.

C. Mode Graphique

Pareil que le mode terminal les fonctions du mode graphique s'occupaient de la fenêtre et l'emplacement des coordonnées dans leur propre case correspondante. Les fonctions ont été implémentées à l'aide de la bibliothèque graphique MLV, Les principales fonctions sont :

- **Int coordVersIndice (int coor)** : la fonction responsable de transfert des coordonnées du clic de l'utilisateur vers un indice valide du plateau pour faciliter la manipulation des données.
- **Int indiceVersCoord (int i)** : C'est le contraire de la fonction précédente il s'occupe du changement de l'indice du tableau vers des coordonnées on l'utilise au moment du placement des pions du plateau déjà sauvegardé.
- **Void initPlateauSauvegard (Plateau *p)** : Restauration de fichier du plateau déjà sauvegardé et affiche de ce dernier comme plateau du jeu au cas où le joueur restaure sa partie.
- **Void sauvegardePlateau (Plateau *p)** : Fonction responsable de la sauvegarde du plateau dans un fichier texte pour le réutiliser dans la restauration du jeu.
- **Void jeuGraphiqueBinome et jeuGraphiquePc** : Selon le mode du jeu choisi on fait appel à ces fonctions qui effectuent une boucle tant que qui vérifie toutes les conditions du jeu dans le mode graphique.