



# Présentation de Jeu de Hex

2023-2024



Réalisé par:

GOURRI ABDELLATIF  
NOUREDDINE YASSINE  
HICHAME KHALID

# Plan

1. Introduction
2. Présentation du jeu
3. Présentation de la méthodologie abordée
4. Présentation de la réalisation
5. Démonstration du jeu réalisé
6. conclusion

# 1-Introduction:

Nous présentons notre projet de développement du jeu Hex en langage C.

L'objectif est de programmer cette jeu de stratégie en langage C en utilisant notre connaissance pendant ce module



# Présentation du Jeu



# Forme du Jeu :

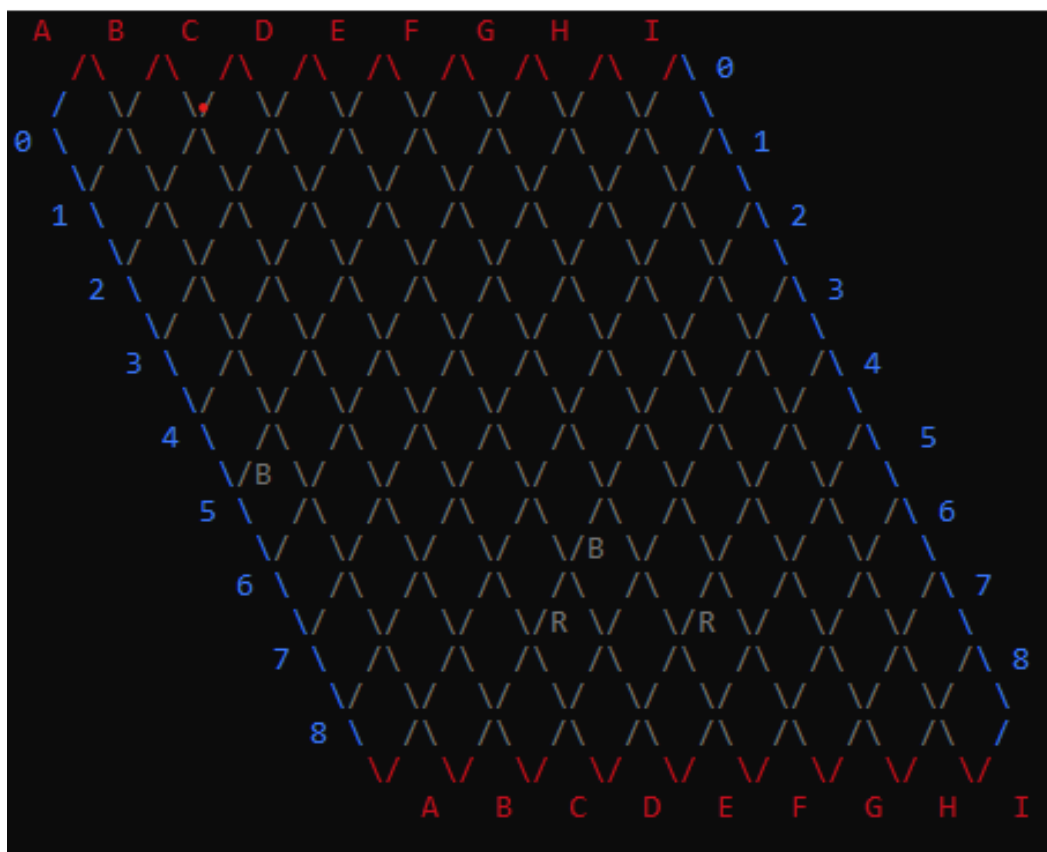
**Grille :** Plateau de dimensions 9x9 comportant des cases en forme de losange.

**Nombre de joueurs :** 2

**Jetons :** Bleu 'B' et Rouge 'R'

**Règles :** Le premier joueur qui relie ses deux bords par un chemin de jetons de sa couleur gagne.

## Le plateau de Jeu:





# Présentation de la méthodologie abordée




# Les questions posant problème :

- Comment représenter le plateau de jeu ?
- Comment déterminer la victoire d'un joueur ?
- Comment distribuer équitablement les tours entre les joueurs ?
- Comment créer une IA compétitive pour jouer contre un joueur humain ?
- Quelles améliorations apporter à l'expérience des joueurs (logo ,couleurs) ?



# Gérer les questions délicates :

Travail en équipe : implique une distribution équilibrée des tâches, une communication fluide et fréquente, ainsi qu'une pratique de révision du code entre pairs pour garantir la qualité et la cohérence du projet.





## -Utilisation de fonctions pour répondre aux questions précédents:

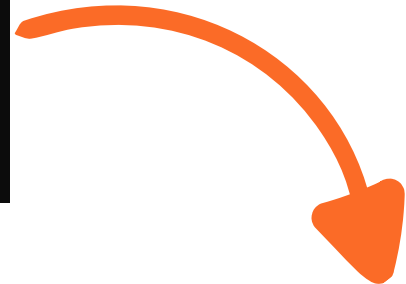
**-void affichage menu();** pour afficher le menu dès le début.



**-void rules();** pour afficher les règles de jeu.



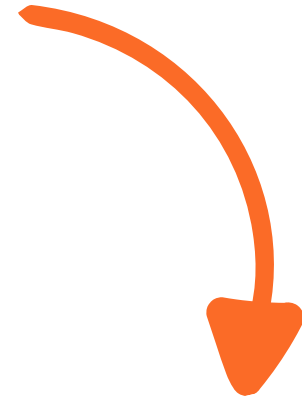
**-void jouer(int choixMenu);** pour afficher les modes de jeu.



**-void joueur\_information();**demander à chacun d'entrer leur nom

```
1>Joueur vs Joueur
2 >Joueur VS Machine
3 >Retour

|-->ENTRER VOTRE CHOIX: 1_
```

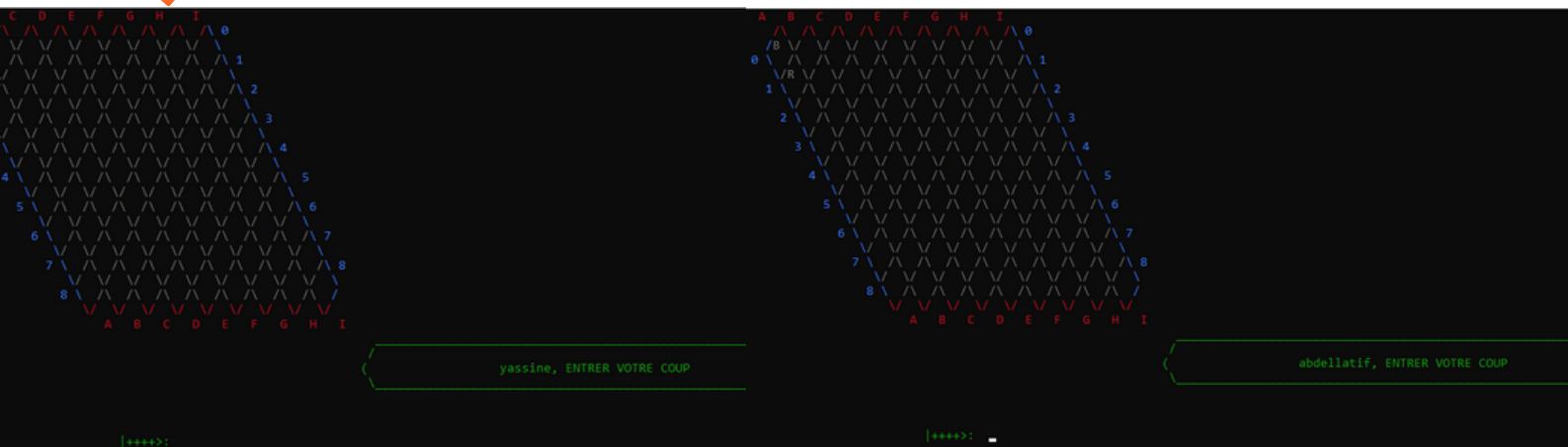
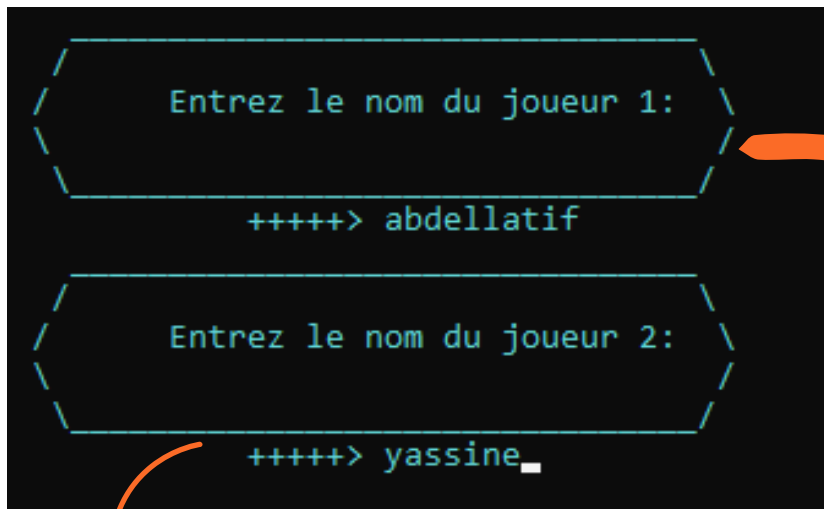


```
Entrez le nom du joueur 1:
++++> abdellatif

Entrez le nom du joueur 2:
++++> yassine_
```

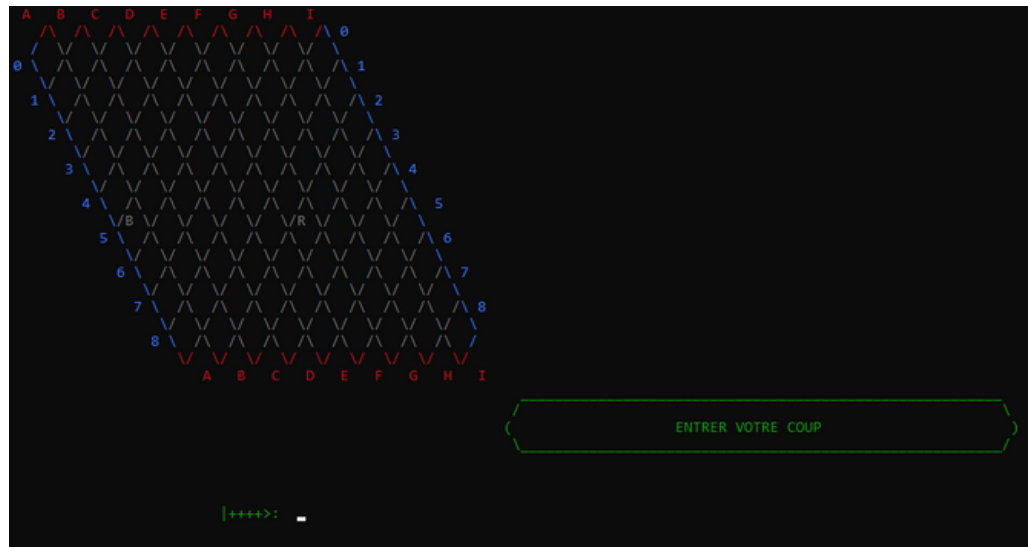
```
void play_joueur_vs_joueur(char *joueur1,char *joueur2);
```

Pour permettre une alternance des choix entre deux joueurs.



- void changement de grille(pion \*\*\*p,char[9][9]);
- void update(0.5);
- showgrille(tab);
- test\_winner(P,joueur1,joueur2);

**-void jouer(int choixMenu);** pour afficher les modes de jeu.



- void jouerCoup1(char tab[TAILLE\_GRILLE][TAILLE\_GRILLE],int x,int y,char \*joueur);
- int coupmachine();
- char colonnerAleatoire();
- void Changement\_grille(P, tab);
- void update(0.5);
- void showgrille(tab);
- void test\_winner(P, joueur, adversaire);



# Présentation de la réalisation



## Les fonctionnalités du jeu :

Fonctionnalités de la forme du jeu	Implémentées	Non implémentées	Concepts C utilisés
Création de la grille	✓		Boucles, tableaux
une alternance des choix entre deux joueurs	✓		conditionnement
Mode joueur vs joueur	✓		Récursion, listes chaînées
Mode joueur vs machine facile	✓		Récursion, listes chaînées
Mode joueur vs machine difficile		✗	

## Règles du jeu

Règles du jeu	implémentée	Non implémentée
Placement des pions	✓	
Vérification du gagnant	✓	

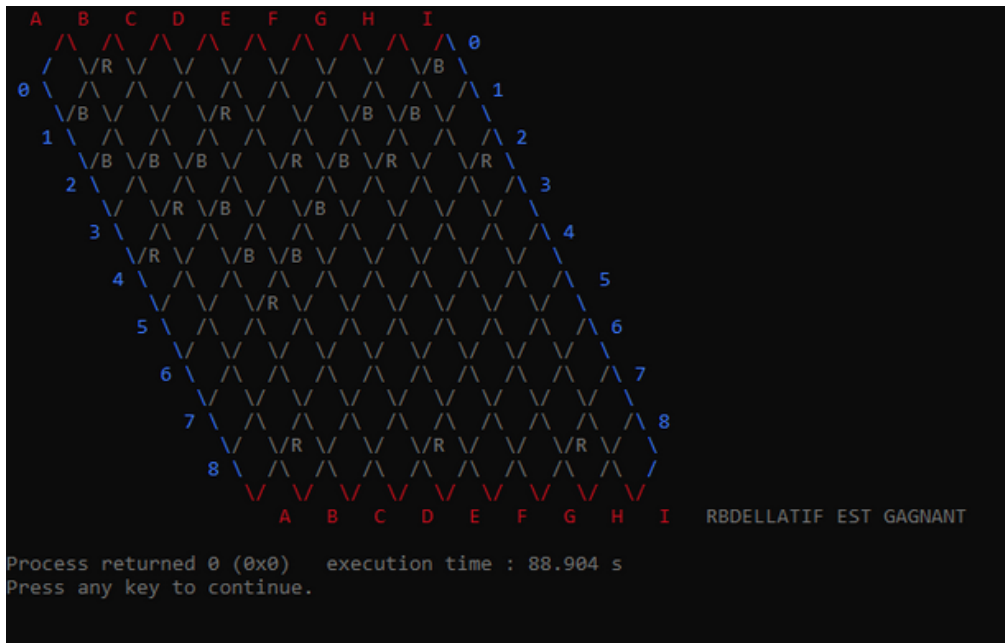


# Démonstration du jeu réalisé



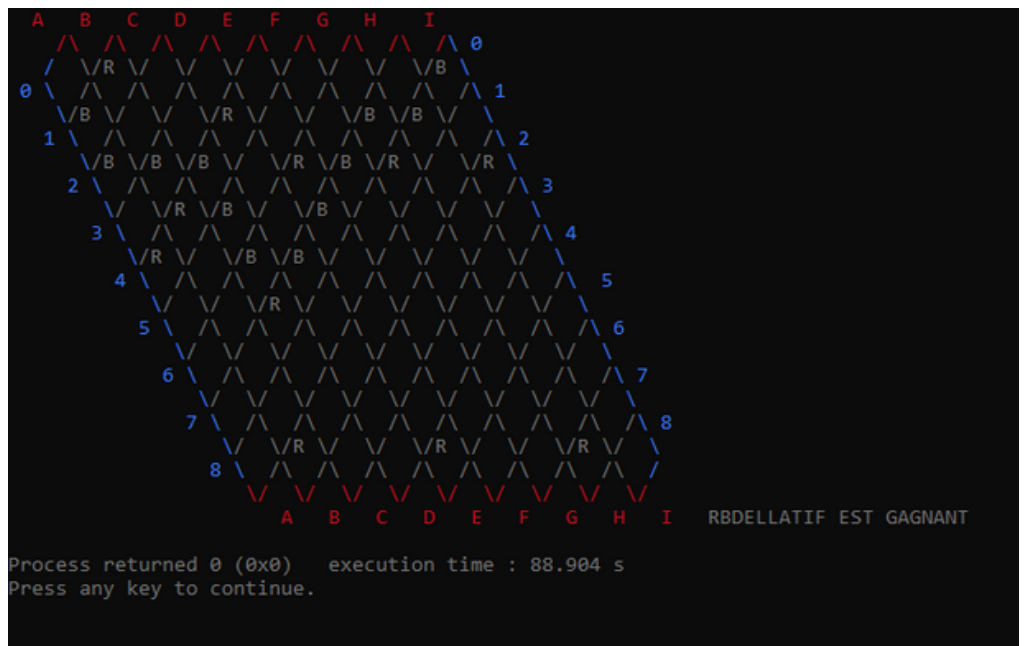


## Algorithme de détection de gagnant:



comme on a vu ci dessus le jeu est devenu capable de detecter la ligne qui relie entre les bords grace à un algorithme de recherche.

## Fonction utilisé:



```
A B C D E F G H I
^ ^ ^ ^ ^ ^ ^ ^ ^
/ \ R \ \ \ \ \ \ \ \ \ 0
0 \ / \ \ \ \ \ \ \ \ \ 1
1 \ / \ \ \ \ \ \ \ \ \ 2
2 \ / \ \ \ \ \ \ \ \ \ 3
3 \ / \ \ \ \ \ \ \ \ \ 4
4 \ / \ \ \ \ \ \ \ \ \ 5
5 \ / \ \ \ \ \ \ \ \ \ 6
6 \ / \ \ \ \ \ \ \ \ \ 7
7 \ / \ \ \ \ \ \ \ \ \ 8
8 \ / \ \ \ \ \ \ \ \ \
A B C D E F G H I RBDELLATIF EST GAGNANT

Process returned 0 (0x0) execution time : 88.904 s
Press any key to continue.
```

1. Cree\_Grille : Cette fonction alloue dynamiquement de la mémoire pour créer une grille de jeu. Elle retourne un pointeur vers la grille créée.
2. Changement\_grille :
  - Cette fonction met à jour la grille de jeu avec les valeurs d'une grille spécifique.
  - Elle prend en paramètres la grille de jeu à mettre à jour et une grille spécifique.
3. Cree\_Liste :
  - Cette fonction crée une liste chaînée vide.
  - Elle retourne un pointeur vers la liste créée.
4. Insert\_Element\_In\_Liste :
  - Cette fonction insère un élément dans une liste chaînée.
  - Elle prend en paramètres la liste dans laquelle insérer l'élément et l'élément à insérer.
5. Clonage\_Liste :
  - Cette fonction crée une copie d'une liste chaînée existante.
  - Elle prend en paramètre la liste à cloner et retourne un pointeur vers la nouvelle liste.

## Fonction utilisé:

### **Verifier\_Element\_In\_Liste :**

Cette fonction vérifie si un élément est présent dans une liste chaînée donnée.

Elle prend en paramètres la liste dans laquelle effectuer la vérification et l'élément à vérifier.

Elle retourne 1 si l'élément est présent et 0 sinon.

### **test\_winner\_X :**

Cette fonction vérifie s'il y a un gagnant dans le jeu pour le joueur X (généralement Bleu).

Elle prend en paramètres la grille de jeu, une liste chaînée contenant les pions du joueur X, et les coordonnées actuelles du pion.

Elle utilise une approche récursive pour explorer toutes les possibilités de victoire pour le joueur X.

### **test\_winner\_O :**

Cette fonction vérifie s'il y a un gagnant dans le jeu pour le joueur O (généralement Rouge).

Elle prend en paramètres la grille de jeu, une liste chaînée contenant les pions du joueur O, et les coordonnées actuelles du pion.

Elle utilise une approche récursive pour explorer toutes les possibilités de victoire pour le joueur O.

### **test\_winner :**

Cette fonction vérifie s'il y a un gagnant dans le jeu en parcourant la grille de jeu.

Elle prend en paramètres la grille de jeu et les noms des joueurs.

Elle appelle les fonctions test\_winner\_X et test\_winner\_O pour chaque joueur, en commençant par le joueur X (Bleu) et le joueur O (Rouge) respectivement.

Ces fonctions constituent l'essentiel de la logique de détection du gagnant dans le jeu. Elles utilisent une approche récursive pour explorer toutes les possibilités de victoire pour chaque joueur.



