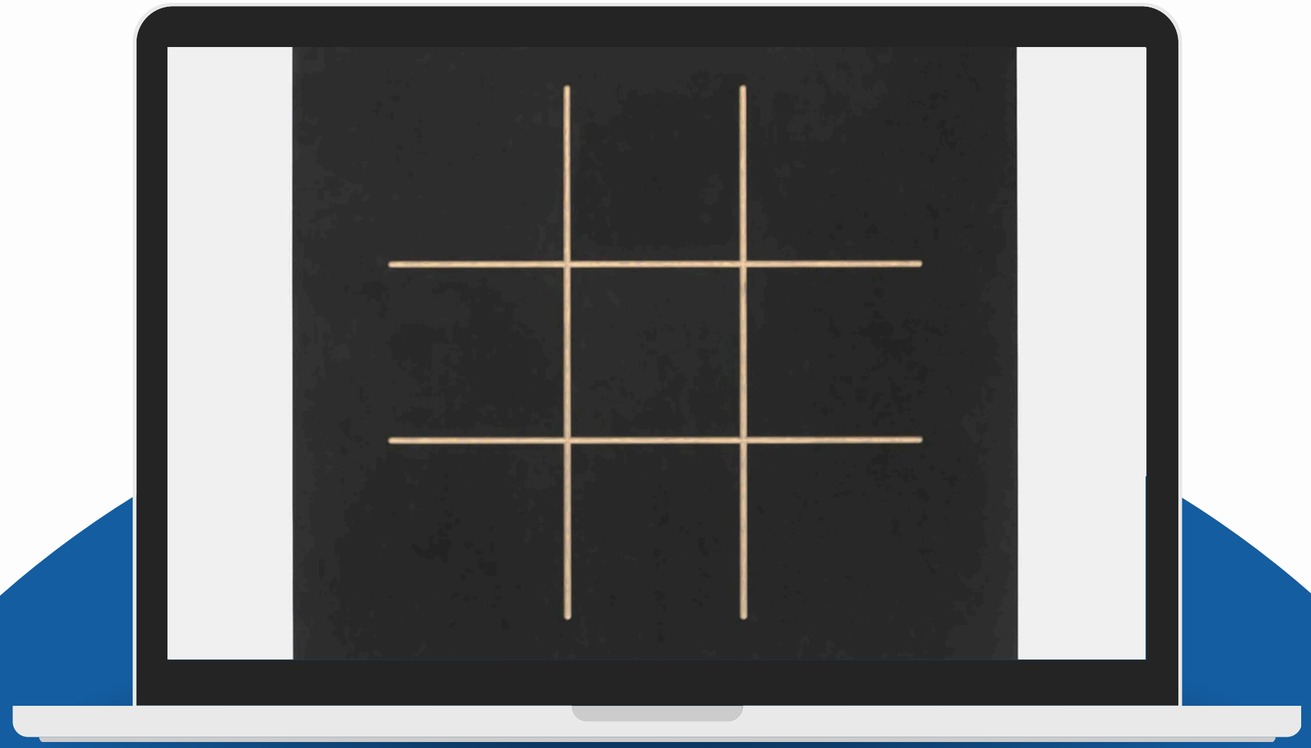
TIC TAC TOE

By: MOHAMED AZIZ KOCHT



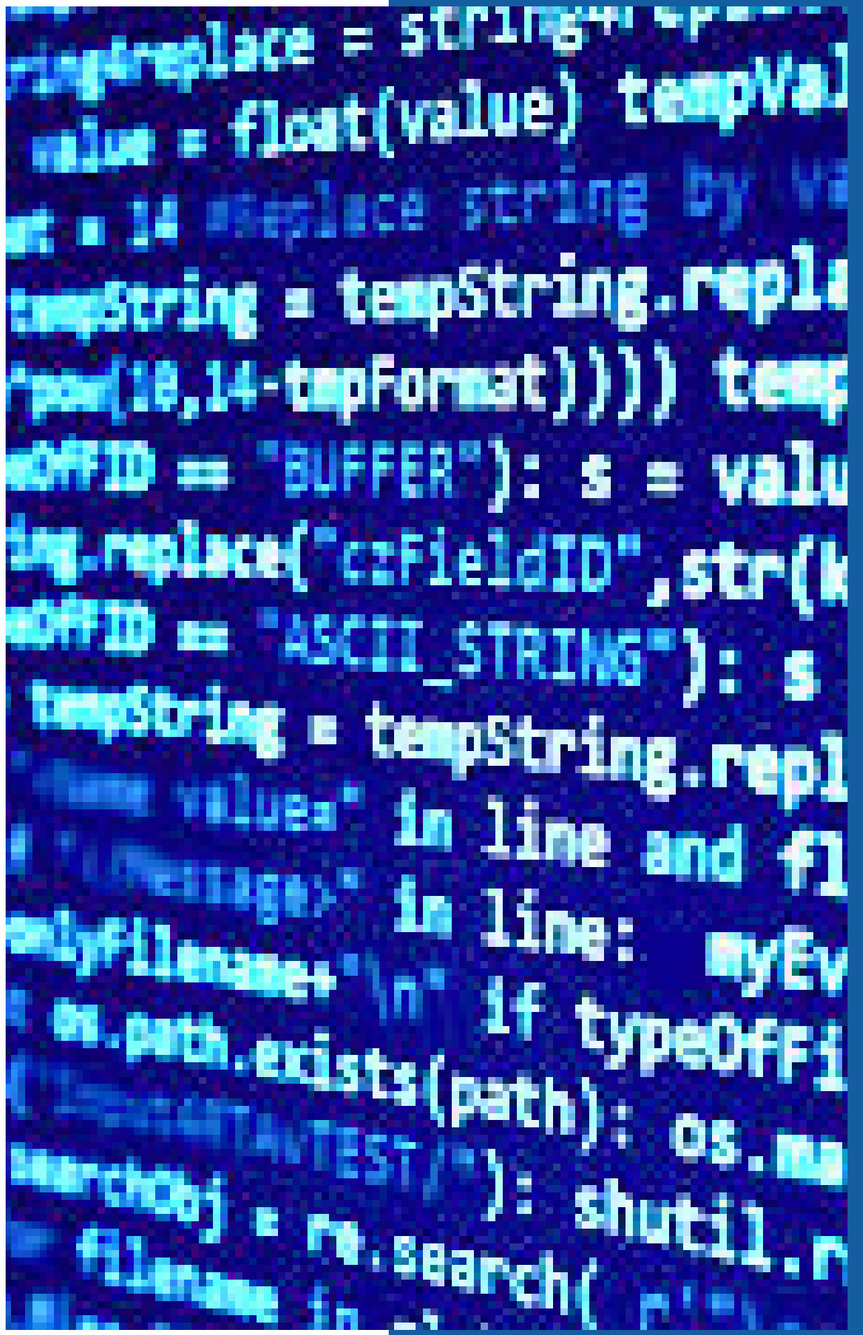
Introduction

C'est un jeu de tic tac toe, également connu sous le nom de X O.
Le morpion (ou Tic Tac Toe) est un jeu simple pour deux joueurs :

- Grille : 3x3 cases.
- Objectif : Aligner 3 symboles (X ou O) horizontalement, verticalement ou en diagonale.
- Tour par tour : Chaque joueur place son symbole dans une case vide.
- Fin : Le premier à aligner 3 symboles gagne, sinon c'est une égalité si toutes les cases sont remplies.

Rapide et stratégique ! 😊

Séminaire



1. Déclarations globales et constantes

```
char board[3][3];
const char PLAYER = 'X';
const char COMPUTER = 'O';
```

- board[3][3] : C’est la grille de jeu. Elle est composée de 9 cases (3x3), où chaque case peut être vide, contenir un 'X' (joueur) ou un 'O' (ordinateur).
- PLAYER et COMPUTER : Ce sont les symboles associés au joueur et à l'ordinateur. Le joueur utilise 'X' et l'ordinateur utilise 'O'.

2. Fonctions déclarées

Ces lignes déclarent les différentes fonctions utilisées dans le programme. Elles sont responsables de :

- Réinitialiser la grille.
- Afficher la grille.
- Vérifier s’il reste des espaces vides.
- Gérer les déplacements du joueur et de l'ordinateur.
- Vérifier s’il y a un gagnant.
- Afficher le résultat de la partie.

```
void resetBoard();
void printBoard();
int checkFreeSpaces();
void playerMove();
void computerMove();
char checkWinner();
void printWinner(char);
```

3. Fonction main()

- Variables winner et response :
- winner stocke le résultat du jeu (gagnant, égalité ou aucun).
- response est utilisée pour demander à l’utilisateur s’il veut jouer à nouveau.
- Boucle principale do-while :
- La boucle continue tant que l'utilisateur répond 'Y' (pour jouer à nouveau). Elle réinitialise la grille, fait jouer le joueur et l'ordinateur, puis affiche le résultat.
- Après chaque tour, elle vérifie s'il y a un gagnant ou s'il n'y a plus de cases libres.

```
nt main()

  char winner = ' ';
  char response = ' ';

  // Initialisation du générateur de nombres aléatoires
  srand(time(0));

  do
  {
      winner = ' ';
      response = ' ';
      resetBoard();

      while(winner == ' ' && checkFreeSpaces() != 0)
      {
          printBoard();
          playerMove();
          winner = checkWinner();
          if(winner != ' ' || checkFreeSpaces() == 0)
          {
              break;
          }

          computerMove();
          winner = checkWinner();
          if(winner != ' ' || checkFr aces() == 0)
          {
```

4. Fonction resetBoard()

```
void resetBoard()
{
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            board[i][j] = ' ';
        }
    }
}
```

- Cette fonction réinitialise toutes les cases du tableau board en les remplissant avec des espaces vides (' '). Cela permet de commencer une nouvelle partie avec une grille vide.

5. Fonction printBoard()

```c
void printBoard()
{
    printf(" %c | %c | %c \n", board[0][0], board[0][1], board[0][2]);
    printf("---|---|---\n");
    printf(" %c | %c | %c \n", board[1][0], board[1][1], board[1][2]);
    printf("---|---|---\n");
    printf(" %c | %c | %c \n", board[2][0], board[2][1], board[2][2]);
}
```

- Cette fonction affiche l’état actuel de la grille à chaque tour. Elle imprime chaque ligne de la grille avec les séparations appropriées (---).

6. Fonction checkFreeSpaces()

```
int checkFreeSpaces()
{
    int freeSpaces = 9;

    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            if(board[i][j] != ' ')
            {
                freeSpaces--;
            }
        }
    }
    return freeSpaces;
}
```

Cette fonction compte le nombre de cases vides (représentées par ' '). Elle retourne le nombre d'espaces libres restants sur la grille. Elle est utilisée pour vérifier si la partie peut continuer.

7. Fonction playerMove()

```c
void playerMove()
{
    int x, y;

    do
    {
        printf("Enter row #(1-3): ");
        scanf("%d", &x);
        x--; // Convertir en indice 0
        printf("Enter column #(1-3): ");
        scanf("%d", &y);
        y--; // Convertir en indice 0

        if(board[x][y] != ' ')
        {
            printf("Invalid move!\n");
        }
```

Cette fonction permet au joueur de faire son mouvement en choisissant une ligne et une colonne (1 à 3). Si la case choisie est déjà occupée, elle demande de refaire le choix. Une fois le mouvement validé, elle place un 'X' dans la case.

8. Fonction computerMove()

Cette fonction permet à l'ordinateur de faire un mouvement aléatoire dans une case vide. Elle génère des coordonnées aléatoires pour la ligne et la colonne (0 à 2), et vérifie que la case est libre avant de placer un 'O'.

```
void computerMove()
{
    int x, y;

    if(checkFreeSpaces() > 0)
    {
        do
        {
            x = rand() % 3;
            y = rand() % 3;
        } while (board[x][y] != ' ');

        board[x][y] = COMPUTER;
    }
```

9. Fonction checkWinner()

Cette fonction vérifie si un joueur a gagné en vérifiant les lignes, colonnes et les deux diagonales. Si un joueur a aligné 3 symboles consécutifs, elle retourne ce symbole ('X' ou 'O'). Sinon, elle retourne ' ' (pas de gagnant)

```
char checkWinner()
{
    for(int i = 0; i < 3; i++)
    {
        if(board[i][0] == board[i][1] && board[i][0] == board[i][2])
        {
            return board[i][0];
        }
    }
    for(int i = 0; i < 3; i++)
    {
        if(board[0][i] == board[1][i] && board[0][i] == board[2][i])
        {
            return board[0][i];
        }
    }
    if(board[0][0] == board[1][1] && board[0][0] == board[2][2])
    {
        return board[0][0];
    }
    if(board[0][2] == board[1][1] && board[0][2] == board[2][0])
```

10. Fonction printWinner()

- Cette fonction affiche le résultat de la partie :
- "YOU WIN!" si le joueur a gagné.
- "YOU LOSE!" si l'ordinateur a gagné.
- "IT'S A TIE!" si la partie se termine par une égalité.

```c
void printWinner(char winner)
{
    if(winner == PLAYER)
    {
        printf("YOU WIN!\n");
    }
    else if(winner == COMPUTER)
    {
        printf("YOU LOSE!\n");
    }
    else
    {
        printf("IT'S A TIE!\n");
    }
}
```

THANK YOU!

MOHAMED AZIZ KOCHT

58785649

azizkocht28@gmail.com