

```
// Abiven Salomé 1A1 - Jeu du puissance 4 - Code source langage C
```

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

#define NBCOL 7
#define NBLIG 6
#define PION_A 'X'
#define PION_B 'O'

const char VIDE = '_';
const char INCONNU = ' ';
const char GAUCHE = 'q';
const char DROITE = 'd';
const int COLONNE_DEBUT = NBLIG/2;

typedef char grillejeu[NBLIG][NBCOL];

void initGrille(grillejeu);
void afficher(grillejeu, char, int);
bool grillePleine(grillejeu);
void jouer(grillejeu, char, int*, int*);
int choisirColonne(grillejeu, char, int);
int trouverLigne(grillejeu, int);
bool estVainqueur(grillejeu, int, int);
void finDePartie(char);
```

```

int main(){

    char vainqueur;
    int ligne, colonne;
    grillejeu g;

    initGrille(g);
    vainqueur = INCONNU;
    afficher(g, PION_A, COLONNE_DEBUT);

    while (vainqueur==INCONNU && !(grillePleine(g))){
        jouer(g, PION_A, &ligne, &colonne);
        afficher(g, PION_B, COLONNE_DEBUT);
        if (estVainqueur(g, ligne, colonne)){
            vainqueur = PION_A;
        }
        else if (!grillePleine(g)){
            jouer(g, PION_B, &ligne, &colonne);
            afficher(g, PION_A, COLONNE_DEBUT);
            if (estVainqueur(g, ligne, colonne)){
                vainqueur = PION_B;
            }
        }
    }
    finDePartie(vainqueur);
    return EXIT_SUCCESS;
}

/*
    Initiation de la grille pour que toutes les cases du tableau soit affichées avec
    des '_' au début.
*/

void initGrille(grillejeu g){

    int i, j;
    for (i = 0; i < NBLIG; i++){
        for (j = 0; j < NBCOL; j++){
            g[i][j] = VIDE;
        }
    }
}

```

```

/*
    Affichage du jeu
*/

void afficher(grillejeu g, char pion, int colonne){

    int i, j;
    system("clear");

    printf("\nAu tour à %c de jouer\nFaite 'q' pour aller à gauche, 'd' pour aller à
    droite et 'espace' pour jouer.\n\n", pion);

    printf(" 1 2 3 4 5 6 7 \n");

    for (i = 0; i < colonne; i++){
        printf("  ");
    }
    printf("%2c\n", pion);

    for (i = 0; i < NBLIG; i++){
        printf("|");
        for (j = 0; j < NBCOL; j++){
            printf("%c|", g[i][j]);
        }
        printf("\n");
    }
    printf("\n");
}

/*
    Vérifie si il reste une case avec VIDE
*/

bool grillePleine (grillejeu g){

    int i = 0;
    bool plein = true;

    while ( i < NBCOL && plein==true){
        if (g[0][i] == VIDE){
            plein = false;
        }
        i++;
    }
    return plein;
}

```

```

/*
    Après avoir choisis colonne et trouver la ligne place le pion
*/

void jouer(grillejeu g, char pion, int *ligne, int *colonne){
    *colonne = choisirColonne(g, pion, COLONNE_DEBUT);
    *ligne = trouverLigne(g, *colonne);
    g[*ligne][*colonne] = pion;
}

/*
    Choix de la colonne avec les touches 'q', 'd' et 'espace'
*/

int choisirColonne(grillejeu g, char pion, int colonne){
    char droitegauche, entree;
    char message[100];
    int choix = colonne;

    scanf("%c%c", &droitegauche, &entree);
    while (droitegauche != ' ' || trouverLigne(g, choix) == -1){
        strcpy(message, "");
        switch (droitegauche){
            case 'q':
                if (choix > 0){
                    choix = choix - 1;
                }
                else {
                    choix = 6;
                }
                break;
            case 'd':
                if (choix < NBCOL - 1){
                    choix = choix + 1;
                }
                else {
                    choix = 0;
                }
                break;
            case ' ':
                if (trouverLigne(g, choix) == -1){
                    strcpy(message, "Erreur ! Colonne pleine\n");
                }
                break;
            default:
                strcpy(message, "Erreur de caractère ! (si vous avez fait un 'Entré' avant
votre choix de colonne refaite un 'Entré')\n");
                break;
        }
    }
    afficher(g, pion, choix);
    printf("%s\n", message);
}

```

```

        scanf("%c%c", &droitegauche, &entree);
    }
    return choix;
}

/*
    Recherche la ligne sur laquelle le pion va tomber
*/

int trouverLigne(grillejeu g, int colonne){

    int ligne = -1;

    while (ligne < NBLIG && g[ligne+1][colonne] == VIDE){
        ligne = ligne + 1;
    }
    return ligne;
}

/*
    Recherche du vainqueur
*/

bool estVainqueur (grillejeu g, int ligne, int colonne){

    int i = 0;

    int suite = 0;
    int suiteMax = 0;

    // Recherche sur une ligne en verticale
    while (ligne+i < NBLIG && suite < 4 && g[ligne+i][colonne] == g[ligne][colonne]){
        i++;
        suite++;
    }
    if (suiteMax < suite){
        suiteMax = suite;
    }
    else {
        suiteMax = suiteMax;
    }
}

```

```

// Recherche sur une ligne en horizontale
suite = 1;
i = 1;
while (colonne+i < NBCOL && suite < 4 && g[ligne][colonne+i] == g[ligne]
[colonne]){
    i++;
    suite++;
}

i = 1;
while (colonne-i >= 0 && suite < 4 && g[ligne][colonne-i] == g[ligne][colonne]){
    i++;
    suite++;
}
if (suiteMax < suite){
    suiteMax = suite;
}
else {
    suiteMax = suiteMax;
}

// Recherche sur une ligne en diagonale1
suite = 1;
i = 1;
while (colonne+i < NBCOL && ligne-i >= 0 && suite < 4 && g[ligne-i][colonne+i] ==
g[ligne][colonne]){
    i++;
    suite++;
}

i = 1;
while (colonne-i >= 0 && ligne+i < NBLIG && suite < 4 && g[ligne+i][colonne-i] ==
g[ligne][colonne]){
    i++;
    suite++;
}
if (suiteMax < suite){
    suiteMax = suite;
}
else {
    suiteMax = suiteMax;
}

```

```

// Recherche sur une ligne en diagonale2
suite = 1;
i = 1;
while (colonne-i >= 0 && ligne-i >= 0 && suite < 4 && g[ligne-i][colonne-i] ==
g[ligne][colonne]){
    i++;
    suite++;
}

i = 1;
while (colonne+i < NBCOL && ligne+i < NBLIG && suite < 4 && g[ligne+i][colonne+i]
== g[ligne][colonne]){
    i++;
    suite++;
}
if (suiteMax < suite){
    suiteMax = suite;
}
else {
    suiteMax = suiteMax;
}

return suiteMax >= 4;
}

/*
Affichage de la fin de partie
*/

void finDePartie(char pion){
    if (pion == INCONNU){
        printf("Match nul !\n");
    }
    else {
        printf("Le joueur %c a gagné ! Félicitations !\n", pion);
    }
}

```