

Séance 12 : Jeudi 21/03/2019

Jeu du morpion

Dans cette partie nous allons programmer le jeu du morpion. Pour cela, vous avez besoin d'une grille 3*3 et de 2 joueurs ayant des pions différents (les croix et les ronds).

X	X	O
O	X	X
X	O	O

Grille sans
gagnant

X	O	X
O	X	O
O	X	X

Grille avec
gagnant

A tour de rôle, chaque joueur positionne un de ses pions sur la grille. Le jeu se finit quand un joueur a réalisé une ligne, une colonne ou une diagonale avec ses pions (c'est le gagnant) ou quand la grille est pleine (pas de gagnant).

La grille est représentée par un tableau à 2 dimensions de caractères dont chaque case contiendra soit '_', soit 'O', soit 'X'. Pour réaliser l'implémentation de ce jeu, écrivez les sous-programmes suivants.

1. Initialisation de la grille du morpion à vide (caractère '_')
`void initialiseGrille(char grille[3][3])`
2. Affichage de la grille du morpion : _ indique case vide, O pion joueur 1 et X pion joueur 2 :
`void afficheGrille(char grille[3][3])`
3. Saisie des coordonnées du nouveau pion à mettre sur la grille. Si les coordonnées sont en dehors de la grille ou si la case possède déjà un pion, la saisie est refusée, un message d'erreur est affiché, et le joueur doit rejouer. Dans le cas où les coordonnées sont correctes, placer le pion sur la grille à cet emplacement.
`void metUnPionSurLaGrille(char grille[3][3], char &joueur)`
4. Teste si l'un des joueurs a gagné (ligne, colonne ou diagonale remplie de pions semblables). Dans ce cas, affiche un message pour indiquer le joueur qui a gagné. S'il n'y a pas de gagnant, teste que la grille n'est pas pleine. Si elle est pleine, affiche un message indiquant qu'aucun des joueurs n'a gagné. Retourne TRUE si la grille est pleine ou si un joueur a gagné, FALSE sinon.
`bool testeFinJeu(char grille[3][3], char joueur)`
5. Écrivez ensuite le programme principal permettant de dérouler la partie. En voici son algorithme :
 Algorithme principal :
 Initialisation de la grille à vide
 Tant que (pas de gagnant ou pas grille pleine)
 Afficher grille
 Mettre un pion sur la grille

Correction

```
#include <iostream>
#include<math.h>
using namespace std;

void init_grille(char grille [3][3]){
    int i,j;
    for (i=0;i<3;i++)
    {
```

```

        for (j=0;j<3;j++)
        {
            grille[i][j]= '-';
        }
    }
}

void affiche_grille(char grille [3][3]){
    int i,j;
    for (i=0;i<3;i++)
    {
        for (j=0;j<3;j++)
        {
            cout<<grille[i][j]<<" | ";
        }
        cout<<endl;
    }
}

void metunptionsurgrille (char grille [3][3], char & joueur) {
    int l,c;
    do
    {
        cout<<"Donnez les coordonnees de la case à remplir"<<endl;
        cin>>l>>c;
    }while (l<0 || l>2 || c<0 || c>2 || grille[l][c] !='-');
    grille[l][c]=joueur;
    if (joueur == 'X') joueur = 'O';
    else joueur = 'X';
}

bool test_finjeu (char grille[3][3], char joueur){
    bool fin = false;
    int i;
    for (i=0;i<3;i++)
    {
        if
        ((grille[i][0]==grille[i][1])&&(grille[i][1]==grille[i][2]) &&
        (grille[i][0]!='-'))
            fin = true;

        if
        ((grille[0][i]==grille[1][i])&&(grille[1][i]==grille[2][i])&&
        (grille[0][i]!='-'))
            fin = true;
    }

    if ((grille[0][0]==grille[1][1])&&(grille[1][1]==grille[2][2])&&
    (grille[1][1]!='-'))
        //première diagonale avec le meme symbole
        fin = true;
    if ((grille[0][2]==grille[1][1])&&(grille[1][1]==grille[2][0])&&
    (grille[1][1]!='-'))
        //deuxième diagonale avec le meme symbole
        fin = true;

    /*
    if
    ((grille[0][0]==grille[0][1])&&(grille[0][1]==grille[0][2]) &&
    (grille[0][0]!='-'))
        //première ligne avec le meme symbole
        fin = true;

    if
    ((grille[1][0]==grille[1][1])&&(grille[1][1]==grille[1][2])&&
    (grille[1][1]!='-'))
        //deuxieme ligne avec le meme symbole

```

```

        fin = true;
    if
((grille[2][0]==grille[2][1])&&(grille[2][1]==grille[2][2])&&
(grille[2][2]!='-'))
        //troisieme ligne avec le meme symbole
        fin = true;
    if ((grille[0][0]==grille[1][0])&&(grille[1][0]==grille[2][0])&&
(grille[0][0]!='-'))
        //premiere colonne avec le meme symbole
        fin = true;
    if ((grille[0][1]==grille[1][1])&&(grille[1][1]==grille[2][1])&&
(grille[1][1]!='-'))
        //deuxieme colonne avec le meme symbole
        fin = true;
    if ((grille[0][2]==grille[1][2])&&(grille[1][2]==grille[2][2])&&
(grille[2][2]!='-'))
        //troisieme colonne avec le meme symbole
        fin = true;
    if ((grille[0][0]==grille[1][1])&&(grille[1][1]==grille[2][2])&&
(grille[1][1]!='-'))
        //premiere diagonale avec le meme symbole
        fin = true;
    if ((grille[0][2]==grille[1][1])&&(grille[1][1]==grille[2][0])&&
(grille[1][1]!='-'))
        //deuxieme diagonale avec le meme symbole
        fin = true;
*/
    return fin;
}

int main (void)
{
    char morpion[3][3];
    char j = 'X';
    int i, nb_tour = 0;
    bool test;
    init_grille(morpion);
    affiche_grille(morpion);
    do
    {
        metunpionsurgrille(morpion,j);
        nb_tour++;
        affiche_grille(morpion);
        test = test_finjeu(morpion,j);
        if (test) cout<<"fini"<<endl;
    } while ((!test)&&(nb_tour<9));

    if (test==false) cout<<"aucun gagne, la grille est
pleine"<<endl;
    else if (j == 'X') cout <<"Les O ont gagne"<<endl;
        else cout <<"Les X ont gagne"<<endl;
    return 0;
}

```

CM Chaines de caractères

Exemple du cours => le jeu du pendu

```

#include <iostream>
#include<math.h>

```

```

using namespace std;
const int MAX = 15 ;

void saisir_mot (char mot[MAX]){
    cout<<"Donnez la chaine de caracteres à faire deviner "<<endl;
    cin>>mot;
}

int creer_solution (char mot[MAX], char sol[MAX]) {
    int i = 0;
    while (mot[i]!='\0')
    {
        sol[i] = '-';
        i++;
    }
    sol[i]='\0';
    return i;
}

int jouer (char mot[MAX], char sol[MAX], char lettre){
    int i = 0,nb = 0;
    while (mot[i] != '\0')
    {
        if ((mot[i]==lettre)&&(sol[i]!=lettre))

        {
            sol[i] = lettre;
            nb++;
        }
        i++;
    }
    return nb;
}

int main (void)
{
    char m[MAX], s[MAX], l;
    int reste,essai=4, rempl;
    saisir_mot(m);
    reste = creer_solution(m,s) ;
    do
    {
        cout<<"Il y a "<<reste<<" caracteres a decouvrir"<<endl;
        cout<<"Quelle lettre ?"<<endl;
        cin>>l;
        rempl = jouer(m,s,l);
        cout<<l<<" apparait "<<rempl<<" fois "<<endl;
        cout<<s<<endl;
        if (rempl == 0 ) essai--;
        else reste-=rempl;
    } while ((reste!=0)&&(essai>0));
    if (reste == 0) cout << "GAGNE"<<endl;
        else cout <<"PERDU ! "<<endl;

    return 0;
}

```