

PROJET EN ÉQUIPE | **Cahier des charges et spécifications fonctionnelles détaillées
du jeu MORPION 8X8**

CAHIER DES CHARGES

(SUGGESTIONS DE FELIPE) :

Petite version du classique jeu Morpion avec une grille 8x8 programmé en C. Dans le jeu, participent deux personnes et l'objectif est de combiner 5 symboles égaux (des cercles ou des croix) dans 5 cases consécutives, indépendamment de la direction choisie (ça peut être dans la diagonal, vertical ou horizontal). Le joueur 1 commence par choisir quel symbole il va utiliser et dans quelle case il veut dessiner le premier symbole. Les tours se succèdent et il est possible de choisir n'importe quelle case qui soit vide sur la grille. Qui réussit à compléter toutes ses cases d'abord est le vainqueur.

SPÉCIFICATIONS FONCTIONNELLES DÉTAILLÉES

---->>> Cheik, Viet et Killian : vérifier le cahier des charges ci-dessus (et donner vos avis - dire si c'est bon ou non - ou suggestion de modification utilisant l'outil en haut à droite) et chercher les informations manquantes pour les spécifications fonctionnelles. Je cherche de mon côté aussi et je vais mettre le document à jour avec ce que j'ai trouvé / pensé à faire (vous pourrez aussi modifier le fichier et/ou donner vos suggestions, il faut que ça soit fait avant mercredi, c'est à rendre jeudi prochain).

// EXEMPLES CI-DESSOUS //

- Les bibliothèques à utiliser :

EXEMPLE:

#include ...

```
#include ...  
#include ...  
#include ...
```

(SUGGESTIONS DE FELIPE) :

- #include <stdio.h>
- #include <stdlib.h>
- #include <conio.h> (facultatif, pour la fonction getch() - interrompt l'exécution et la retome quand une touche est entrée)

- *Les variables à utiliser :*

EXEMPLE:

```
type1 variable1;  
type2 variable2;  
type3 variable3;
```

(SUGGESTIONS DE FELIPE) :

- int i; => indique s'il y a un vainqueur ou pas
- char position[65] = {0,
1, 2, 3, 4, 5, 6, 7, 8,
9,10,11,12,13,14,15,16,
17,18,19, 20, 21, 22, 23, 24,
25, 26, 27, 28, 29, 30, 31, 32,
33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48,
49, 50, 51, 52, 53, 54, 55, 56,
57, 58, 59, 60, 61, 62, 63, 64}; => simule les positions où le
joueur peut dessiner son symbole dans la grille
- int joueur = 1 ou 2; => indique quel joueur va dessiner à chaque tour;
- int choix = un nombre de 1 à 64; => indique dans quelle position le
symbole sera dessiné
- char symbole1, symbole2; => indique le symbole de chaque joueur 'X' ou
'O'

- *Les fonctions (en-têtes / noms) :*

EXEMPLE:

```
fonctions1 -> nom / type de la fonction1  
fonctions2 -> nom / type de la fonction2
```

(SUGGESTIONS DE FELIPE) :

- `int a_gagne();` => return la valeur de i, en déterminant s'il y a un gagnant ou pas
- `void grille();` => ne retourne aucune valeur, mais dessine la grille à chaque tour
- `int main(void);` => fonction principale qui va appeler les autres

- *Les contenus des fonctions (paramètres à passer, résultats, etc) :*

EXEMPLE:

fonction1 -> "prend ça et ça en paramètre et nous retourne ça"

fonction2 -> "prend machin et machin en paramètre et nous retourne machin"

(SUGGESTIONS DE FELIPE) :

- ```
int a_gagne() {
 if (position[1] == position[2] && position[2] == position[3] && position[3] ==
 position[4] && position[4] == position[5])
 {
 return 1; // quelqu'un a gagné
 }
 .
 .
 .
 .
 // répéter pour toutes les combinaisons gagnantes possibles
 else if (position[1] != '1' position[64] != '64')
 {
 return 0; // match nul
 } else
 {
 return -1 // personne n'a encore gagné, le jeu continue
 }
}
```
- ```
grille() {
    printf(" %c | %c | %c | %c | %c | %c | %c | %c | ", position[1], position[2],
        position[3], position[4], position[5], position[6], position[7], position[8]);
    .
    .
    .
    // répéter pour toutes les 8 lignes de la grille
}
```
- ```
int main(void) {
 // début du jeu
 // on récupère le symbole de chaque joueur avec scanf et if
 printf("Joueur 1, entrez votre symbole);
```

```

scanf("%c", &symbole1);
if (symbole1 == 'X') ...

//boucle do/while pour repeter les tours jusqu'au moment où i = 1 (on a alors un vainqueur), on stock la case choisi dans la variable choix
do {
 grille();
 joueur = 1 ou 2
 printf("Joueur %d, entrez votre case", joueur);
 scanf("%c", &choix)
 if (joueur == 1 && choix == 'a' && position[1] == 'a')
 {
 position[1] = symbole1;
 } else if (...) // répéter pour toutes les positions et aussi quand joueur == 2
 .
 .
 .
 . else
 {
 printf (" Option invalide ! "); // si aucune case correcte est sélectionnée
 }

 i = a_gagne(); // on retourne cette fonction pour modifier la valeur de i à chaque tour et vérifier s'il y a un gagnant ou pas
 ++joueur; // on passe au joueur suivant à chaque tour

} while(i == -1) // i = -1 signifie que personne a gagné pour l'instant

if (i == 1) // si on a un gagnant
{
 printf("Félicitations, joueur %d ! Vous avez gagné !", joueur);
} else // si on a un match nul
{
 printf("Ça a été un match nul :(! ");
}
return 0;
}

```

### (SUGGESTIONS DE VIET) :

- #include <stdio.h>
  - #define max 500
- ```
/*Dit a l'utilisateur d'entrer la taille de l'echiquier et initialiser l'etat de l'echiquier a 0.*/
```
- void lancement(char a[][max],int m)
{
 printf("Veuillez entrez la taille de la ligne et colonne: \n");
 scanf("%d",&m);
 while(m<=5)
 {
 printf("Veuillez entrez la taille de la ligne et colonne\n");
 scanf("%d",&m);
 }
 for(int i=0; i < m; i++)
 for(int j=0;j < m; j++)
 {
 a[0][j]=j;
 a[i][0]=i;
 if(i!=0||j!=0)
 {
 a[i][j]='.';
 }
 }
 }

 /*verifier si les pieces d'echecs sont bleines sur l'echiquier.*/
 - int isfull(char a[][max],int m)
{
 int count=0;
 for(int i=0;i< m;i++)
 {
 for(int j=0;j< m;j++)
 {
 if(i!=0&&j!=0)
 {
 if(a[i][j]=='.')
 {
 count++;
 }
 }
 }
 }
 if(count==0){
 return 0;
 }

```

    }
    return 1;
}

```

/*La fonction de joueur entre dans la pièce d'échecs et vérifie sa validité lors de la sélection de la case souhaitée*/

- void input1(char a[][max],int m,int i1,int j1,char s1[])


```

{
    printf("\n Veuillez %s entrez la case pour taper: ",s1);
    printf("  entrer la ligne:");
    scanf("%d",&i1);
    printf("  entrez dans la colonne:");
    scanf("%d",&j1);
    if((i1==100&&j1==100)||i1==0&&j1==0))
        return;
    while(i1<=0||j1<=0||a[i1][j1]=='X'||a[i1][j1]=='O'||i1>=m||j1>=m)
    {
        printf("\n Veuillez %s entrez la case pour taper: ",s1);
        scanf("%d",&i1);
        scanf("%d",&j1);
        if((i1==100&&j1==100)||i1==0&&j1==0))
            return;
    }

    a[i1][j1]='O';
}

```
- void input2(char a[][max],int m,int i2,int j2,char s2[])


```

{
    printf("\n Veuillez %s entrez la case pour taper: ",s2);
    printf("  entrer la ligne:");
    scanf("%d",&i2);
    printf("  entrez dans la colonne:");
    scanf("%d",&j2);
    if(i2==100&&j2==100)
    {
        return;
    }
    while(i2<=0||j2<=0||a[i2][j2]=='O'||a[i2][j2]=='X'||i2>=m||j2>=m)
    {
        printf("\n Veuillez %s entrez la case pour taper: ",s2);
        scanf("%d",&i2);
        scanf("%d",&j2);
        if(i2==100&&j2==100)
            return;
    }

    a[i2][j2]='X';
}

```

```

//output
• void output(char a[][max],int m)
{

    printf("\n\nl'echiquier est:\n");
    for(int i=0;i< m;i++)
    {
        for(int j=0;j< m;j++)
        {
            if(i==0||j==0)
                printf("%4d",a[i][j]);
            else
                printf("%4c",a[i][j]);

        }
        printf("\n");
        printf("\n");
    }
}

//instruction
• void instruction()
{
    printf("Vous entrez a votre tour les coordonnees = (nombre de lignes, nombre de colonnes)\n");
    printf("Vous ne devez pas cliquer sur les coordonnées saisies, si vous entrez de manière incorrecte vous entrerez a nouveau\n");
    printf("Si l'un ou l'autre camp a>= 5 de ses mêmes pièces consecutivement, il gagne\n ");
    printf("EX: XXXXX->Win\n");
    printf("EX: OXXXXXO->NO WIN\n");
    printf("EX: OXXXXX O->WIN\n");
    printf("Si vous souhaitez quitter, exit = nombre de lignes = 100 et nombre de colonnes = 100\n");
    printf("Coordonnées mortes (100,100)\n");
    printf("Si vous voulez jouer à nouveau, entrez: ligne: 0 et colonne: 0\n");
    printf("(0,0) = replay\n");
    printf("Si toutes les cases sont cochees sans gagnant, une egalite\n");
    printf("thank you! enjoy the game(^_^\n");
    printf("  2-Start\n");
    printf("  3-Exit\n");
    printf("Votre choix:");
}

int main() {
    int k,n,win=0;
    char s1[20],s2[20],kt;
    char a[max][max];

```

```

int m,i1,j1,i2,j2;
printf("Veuillez choisir: \n");
printf("    1-Instruction\n");
printf("    2-Start\n");
printf("    3-Exit\n");
printf("Votre choix:");
scanf("%d",&n);
do{
    switch(n)
    {

        case 1:
            instruction();
            scanf("%d",&n);
            break;
        case 2:
            printf("\nGAME START\n");
            printf("ENTrez le nom player 1: ");
            scanf("%s",s1);
            printf("Entrez le nom player 2: ");
            scanf("%s",s2);
            lancement(a,m);
            output(a,m);
            do
            {

                k=isfull(a,m);
                if(k==1)
                {
                    input1(a,m,i1,j1,s1);
                }
                output(a,m);

            }

            if(win==1||k==0||(i1==100&&j1==100)||(i2==100&&j2==100)||(i1==0&&j1==0)||(i2
            ==0&&j2==0))
            {
                printf("Voulez-vous rejouer? Y/N \n");
                scanf("%s",&kt);
                if((i1==0&&j1==0)||(i2==0&&j2==0)||kt=='Y'||kt=='y'||k==0)
                {
                    if(k==0)
                    printf("%s NULL %s\n",s1,s2);
                    win=0;
                    break;
                }
                else
                {

```



```

        return;
    }
    }k=isfull(a,m);
    if(k==1)
    {
        input2(a,m,i2,j2,s2);
    }
    output(a,m);

if(win==1||k==0||(i1==100&&j1==100)||i2==100&&j2==100)||i1==0&&j1==0)||i2
==0&&j2==0))
{
    printf("Voulez-vous rejouer? Y/n \n");
    scanf("%s",&kt);
    if((i1==0&&j1==0)||i2==0&&j2==0)||kt=='Y'||kt=='y'||k==0)
    {
        if(k==0)
        printf("%s NULL %s\n",s1,s2);
        win=0;
        break;
    }
    else
    {
        return;
    }
}

}while(k==1&&win==0);
break;
case 3:
break;
default:
printf("Pas cette choix\n Veuillez donner une autre
choix:");
scanf("%d",&n);
}
}while(n!=3);

return 0;
}

```