

Cahier de charge :

Problématique et objectif :

Retrouver le mot a choisi ,au hasard , par l'ordinateur dont la première lettre est visible .Le joueur a le droit d'essayer sept fois et il se déclare gagnant s'il trouve le bon mot au bout de ces sept tentatives

Description de la fonctionnement :

- ❖ L'ordinateur va choisir un mot au hasard et afficher la première lettre .
- ❖ Le joueur va essayer de retrouver ce mot à travers 7 tentatives successives .Au cours de ses essais :
 - Si la longueur du mot différente de celle de mot a recherché alors un message d'erreur va apparaitre et on passe à la tentative suivante.
 - Sinon , on fait l'analyse :
 - Si une lettre appartient au mot et elle est dans sa bonne place alors l'ordinateur affiche cette lettre dans le reste des tentatives.
 - Si une lettre appartient au mot mais elle n'est pas dans sa bonne place alors un signale sous forme d'un caractère s'apparait. (!)
 - Sinon "-" va apparaitre .

❖ Il est nécessaire d'utiliser :

- * Une fonction qui permet de récupérer un mot de 7 lettres choisis au hasard à partir du dictionnaire : fonction `choix_mot()`
- * une fonction qui compte le nombre de tentatives.
- * une fonction qui permet de faire l'analyse de mot choisi par le joueur.
- * une fonction qui vérifie l'égalité entre le mot saisi et la mot choisi par l'ordinateur.
- * une fonction jouer qui décrit les déroulements de jeu
- * Des autres procédures secondaires d'affichage

1/ la fonction qui récupère un mot: `choix_mot()` :

```
char* choix_mot(){  
    FILE *in;  
  
    int nb_lignes=0,i=0,j;  
  
    char c,*mot;  
  
    mot=(char*)malloc(7*sizeof(char));  
    srand(time(NULL)*100);  
    in = fopen ("5.txt","rt");  
    while ( c = fgetc(in),! feof (in) )  
    {  
        if(c == '\n')
```

```

        nb_lignes++;
    }
fclose(in);

in = fopen("5.txt", "rt");

j=rand()%nb_lignes;
j++;
do

{
    fscanf(in,"%s",mot);
    i++;
}
while(i!=j);
fclose(in);
return(mot);
}

```

2/ fonction qui compte le nombre d'essai

```

nt essay( char *tab2 , int cmt ){

```

```

int test ;

if (cmt>= 7) {printf("Game Over!!!\n  SOLUTION : \t"); test = 0 ;}

else {

    printf("Essay n   %d :",cmt+1);

    test=1;}

    return test ;

}

```

3/fonction d'analyse :

```

void analyse (char *tab1 , char *tab2 , char *tab3){

int i,j;

for(i=0;i<7;i++){

    if(tab3[i]=='!') tab3[i]='-' ;

    }

for ( i=0;i<7;i++){

    if((tab3[i]=='-') || (tab3[i]=='!')){

        if(tab1[i]==tab2[i]) tab3[i]=tab1[i]; // Si une lettre appartient
au mot et elle est dans sa bonne place alors l'ordinateur affiche
cette lettre dans le reste des tentatives

        else{for ( j=1;j<7;j++) if((tab1[i]==tab2[j])&&(i!=j))

```

// Si une lettre appartient au mot mais elle n'est pas dans sa bonne place alors un signal sous forme d'un caractère s'apparaît. (!)

```
if((tab3[i]=='-') || (tab3[i]=='!')){tab3[j]='!';}  
}
```

```
}
```

```
}
```

```
}
```

4/une fonction qui vérifie l'égalité entre le mot saisi et le mot choisi par l'ordinateur :

```
int vrf(char *tab1,char *tab3){  
    int i;  
    for ( i=0;i<7;i++) if(tab1[i]!=tab3[i]) return 1;  
    printf("BRAVO!!");  
    return 0;  
}
```

5/Fonction jouer :

```
void jouer() {  
    srand(getpid());  
  
    char ch ;  
  
    do{  
        printf("_____\\n");  
        printf("|                |\\n");  
        printf("|    Jeu de motus    |\\n");  
        printf("| By@ ALIBI-MOURAD    |\\n");  
        printf("|    & BEN AMOR AMANI |\\n");  
        printf("|  PROJET DE PROGRAMATION  |\\n");  
        printf("|  GI 1 - S4 2015-2016    |\\n");  
        printf("_____\\n");  
        printf("\\n vous avez 7 essay ! le mot se commance par : \\n");  
  
        char *tab1,*tab2,*tab3;  
  
        int i,test1=1,test2=1;  
  
        tab1=allouer_chaine (7);tab2=allouer_chaine  
        (7);tab3=allouer_chaine (7);  
  
        tab1=choix_mot();
```

```

tab3[0]=tab1[0];
for (i =1 ;i<7 ; i++) { tab3[i]='-';}
int cmt=0;
while(test1 && test2){
    printf("\t  ");
    puts(tab3);
    test1=essay(tab2,cmt);

    if(cmt>=7) goto out ;
    gets(tab2);
    cmt++;

    if( strlen(tab2)==7){
        analyse(tab1,tab2,tab3);
        test2=vrf(tab1,tab3);}
    else printf("longueur faus\n");
    out: i=0;
}

puts(tab1);

```

```
printf ("\n");  
  
free (tab1);free (tab2);free (tab3);  
  
do{  
  
    color_console(4, 3);  
  
    printf("Repeter une auter fois ? [o/n] ou [O/N] : \n");  
  
    ch=getch();}while(ch!='o'&&ch!='O'&&ch!='n'&&ch!='N');  
  
    system("cls");system("color 3f");  
  
} while( (ch=='o') || (ch== 'O' ));  
  
}
```