



*Le logo du jeu Motus - FTV PROD*

# Développement du jeu Motus en langage C

*Mémoire projet d'études*

*Première année du cycle d'ingénieur informatique*

Réalisé par :  
MOUNTASSER Massine  
JARIDI Mustapha  
BAGHROSSE Rachid

Encadré par:  
Mme. GUIBADJ Rym Nesrine

Le : 23/12/2019

---

# Table des matières

<b>Table des matières</b>	<b>1</b>
<b>Introduction</b>	<b>2</b>
<b>Conception</b>	<b>4</b>
II. 1. Gestion du projet	4
II. 2. Organisation	6
<b>Réalisation</b>	<b>7</b>
III. 1. Structure du jeu	7
III. 2. Manipulation de fichiers	9
III. 3. Traitement de texte	11
<b>Conclusion</b>	<b>12</b>
IV. 1. Connaissances acquises	12
IV. 2. Difficultés rencontrés et Possibilités d'améliorations	13
IV. 3. Synthèse	14

## I. Introduction

Motus est un jeu télévisé inspiré de sa version originale américaine Lingo. Ce dernier a été diffusé pour la première fois aux Etats-Unis en 1987, repris donc par Motus en 1990 par la chaîne Antenne 2. C'est un jeu de lettres où l'émission oppose deux binômes de candidats en compétition pour décrocher une cagnotte.



*Plateau Motus du 15/06/2018 - Intégrale*

*Diffusion de Motus Officiel - France 2 sur Youtube*

Le principe est de chercher un mot de nombre de lettres fixe quand on a la main. Les deux équipes, bleue et jaune, sont en défi et tentent chacune de garder la main le plus longtemps et augmenter son score par rapport à l'autre en trouvant des mots. L'équipe gagnante poursuit donc le jeu par une "super-partie" où ses deux candidats doivent trouver 10 mots en jouant contre la montre.



*Partie Motus du 15/06/18 - Intégrale*

*Diffusion de Motus Officiel - France 2 sur Youtube*

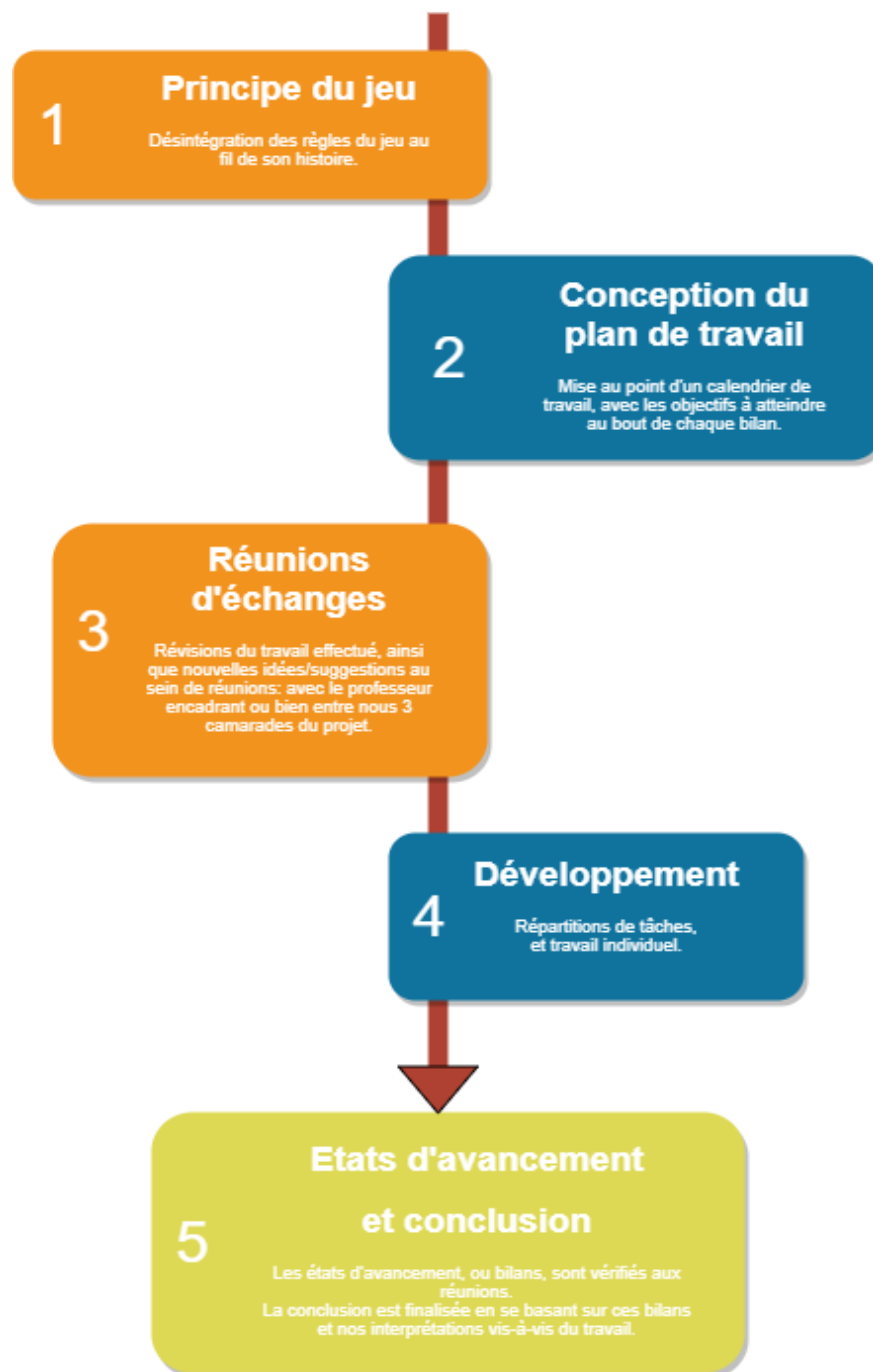
Dans le cadre du projet bureau d'études, nous allons faire le tour de l'ensemble du travail effectué dont l'objectif est de réaliser le jeu choisi en le programmant sur langage C. Voici comment nous avons procédé en tant qu'équipe de trois collègues chargés du projet: nous commençons par poser un plan de travail qui respecte les contraintes nécessaires, nous abordons ensuite et développons comme décrit par notre conception tout en effectuant un suivi de travail; ce qui nous permettra de faire des bilans ponctuels et un compte-rendu final.

## II. Conception

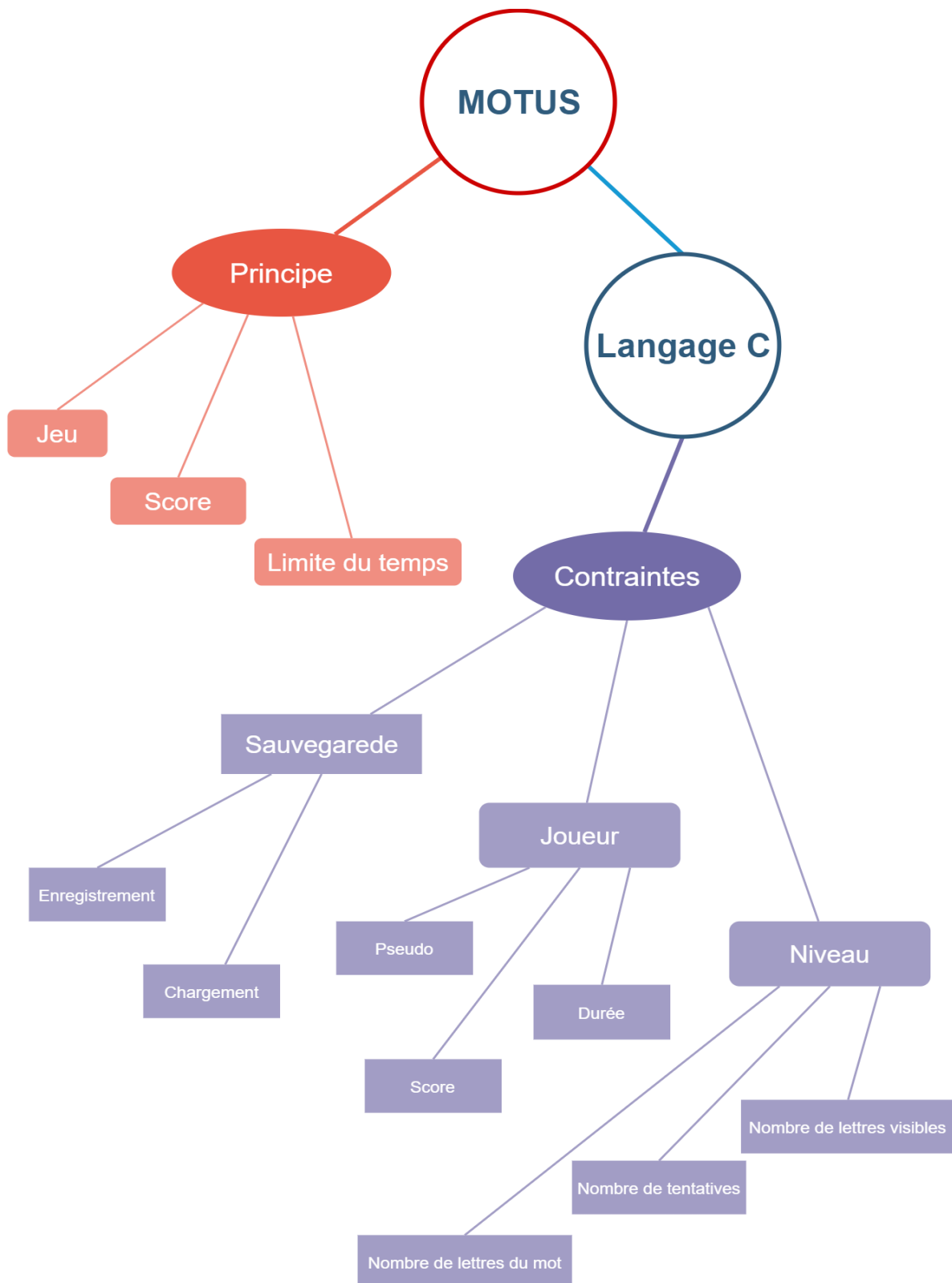
### II. 1. Gestion du projet

C'est le premier pas à franchir pour assurer le bon déroulement du travail.

Dans un premier temps, il est nécessaire de commencer par analyser l'objectif final à atteindre et effectuer un plan global.

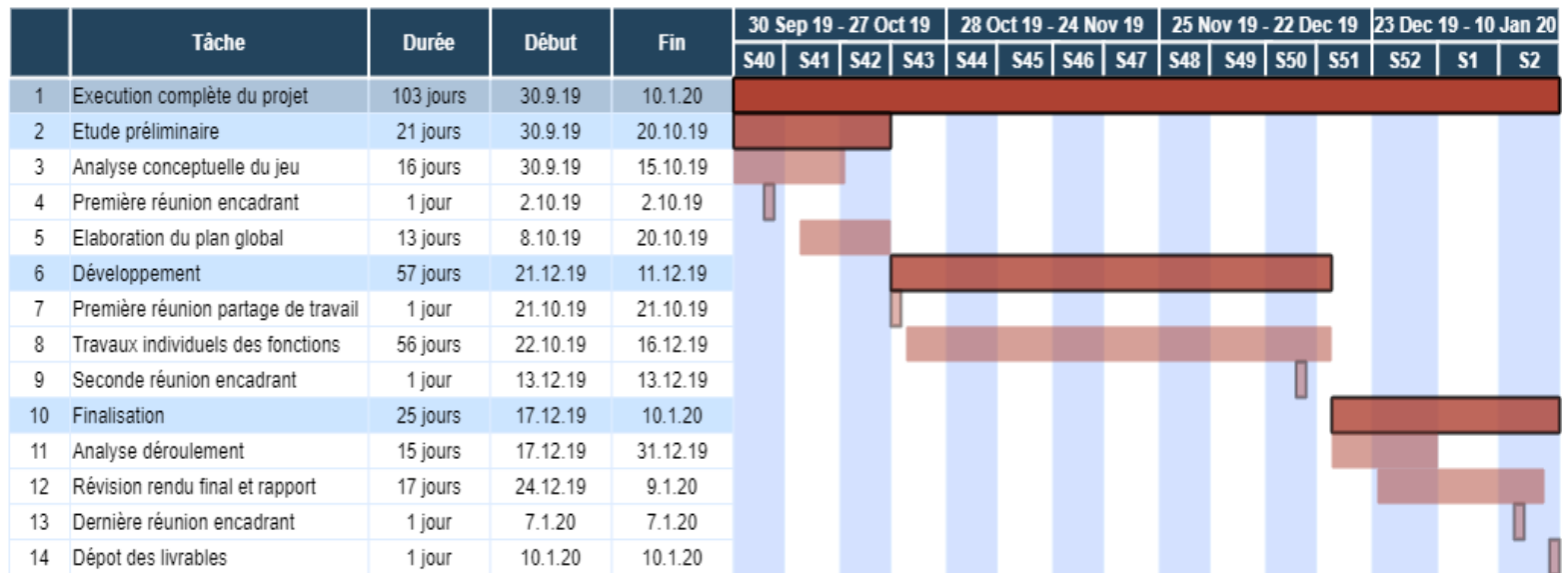


Nous allons voir comment on décide de désintégrer le but principal dans un schéma en forme d'idées en Mind Mapping:



## II. 2. Organisation

Après avoir départagé l'ensemble des points intéressant concernant les principes du jeu Motus ainsi que les contraintes imposées par le cahier de charges, nous décidons de travailler en direction des idées tout en respectant la limite des règles et surtout du temps, ceci représenté dans un diagramme Gantt qui suit:



## III. Réalisation

### III. 1. Structure du jeu

En forme de parties commentées de notre code écrites en C, voici les principales fonctions réalisées qui concernent le côté structurel du jeu :

```
>> typedef struct liste_mot      //structure historique mots entrés
{
    int id_mot;
    char mot[10];
    struct liste_mot *suivant;
}liste_mot;
```

```
>> typedef struct niveau        //structure niveau difficulté
{
    int nbreTenta;
    int tailleMot;
    int nbreLettreV;
} Niveau;
```

```
>> struct joueur                //structure joueur
{
    char pseudo[Max];
    int score;
    int nbPartieG;
    float duree;
    char niveau[MAXCHAINE];
    int enregistrer;
};
typedef struct joueur Joueur;
```



---

```
>> void Menu();           //Fonction du menu principal seule appelée au main

>> void nouvellePartie(); //Début d'une partie

>> void chargerPartie();  //Chargement d'une partie existante

>> void seulJoueur(); void joueurOrdinateur(); void deuxJoueurs();
    //Fonctions appelées selon le mode choisi

>> void niveauDifficulte(); //Choix du niveau

>> Joueur * creerJoueur(char nom[]) //Création d'un élément Joueur
{
    Joueur * joueur=NULL;
    joueur=(Joueur*)malloc(sizeof(Joueur));
    joueur->score=0;
    joueur->nbPartieG=0;
    joueur->duree=0.0;
    joueur->enregistrer=0;
    strcpy(joueur->pseudo,nom);
    joueur->pseudo[strlen(joueur->pseudo)+1]='\0';
    return joueur;
}
```

## III. 2. Manipulation de fichiers

Les fonctions qui assurent la sauvegarde/chargement de parties :

```
>> void afficher_Partie_Sauv(char * pseudo);           //Affichage des parties existantes

>> int creerDossier(char * pseudo);                   //Création de dossier d'un nouveau joueur

>> void lire_Partie_Existe(char* pseudo)               //Affichage et lecture des parties
                                                         existantes pour le chargement

>> void sauvegarder_Partie(Joueur * joueur, char**tab, int ligne, int colonne, char *motChoisi,
int nbtentative,int Mode);                           //Fonction de sauvegarde d'une partie

>> void Charge_seulJoueur(Joueur*monjoueur,char **table,int ligne,int colonne,
char*motDico, int nbtenta);

>> void Charge_joueurOrdinateur(Joueur*monjoueur,char **table,int ligne,int
colonne,char*motDico,int nbtenta);

>> void Charge_deuxJoueurs(Joueur*monjoueur,char **table,int ligne,int
colonne,char*motDico,int nbtenta);                   //Fonctions de chargement de parties existante
```

Fonctions de manipulation des dictionnaires à mots :

```
>> void choisirMot(char *MotChoisi, int tailleMot);  
                                     //Choix d'un mot dans le dico  
                                     aléatoirement et copie dans MotChoisi  
  
>> void choisirMotPc(char *MotPc, char *mot_heuristique, liste_mot *mot_deja_entrer, int  
tailleMot);  
                                     //Choix d'un mot dans le dico  
                                     aléatoirement et le met dans MotPc  
  
>> int motExiste(char *mot);        //Verification d'existence d'un mot dans le dico
```

### III. 3. Traitement de texte

## Les fonctions concernant le traitement de textes en entrées et sorties :

```
>> void Color(int couleurDuTexte,int couleurDeFond); //Fonction d'affichage couleurs
```

```
>> void verifier(char *MotChoisi,char *MotEntrer,int *trouve,int taille);
```

```
//Vérification du mot secret vs le mot entré..
```

[illegible]

```
>> Void choisirMot(char *MotChoisi, int tailleMot) //Choisir mot secret  
aléatoirement dans le dico
```

---

## IV. Conclusion

### IV. 1. Connaissances acquises

Comme convenu par l'objectif principal du projet bureau d'études; le but est de mettre en pratique et développer les connaissances des camarades, notamment :

- la gestion d'un projet,
- le travail de groupe,
- la cohérence collective,
- la gestion du temps,

et d'un côté plus technique :

- la logique du langage généraliste C,
- l'utilisation de fonctions plus ou moins complexes,
- la gestion de mémoire,
- l'assemblage d'un projet d'équipe.

## IV. 2. Difficultés rencontrés et Possibilités d'améliorations

On représente le résultat de notre analyse du déroulement en phase finale du projet en deux listes de ce qu'on a rencontré comme principales difficultés lors de notre travail d'une part, et d'idées de développement envisageables de l'autre.

### Difficultés rencontrées

- Le premier défi relevé était de passer de notre documentation sur le jeu télévisé à un plan pratique de travail.
- Après avoir initié les tâches à accomplir par chacun, il n'a pas été facile de gérer certains bugs, chose qui a impactée notre gestion du temps.
- La partie sauvegarde/chargement de jeu a été un peu corsée à surmonter, du fait qu'elle nous a pris plus de temps et d'efforts à fournir avant de fonctionner correctement.



### Possibilités d'amélioration

- Enrichissement du programme par une interface graphique.
- Implémentation d'une table des meilleurs scores de joueurs enregistrés.
- Possibilité de faire des parties joueur contre joueur en ligne.
- Ajouter un nouveau paramètre monétaire fictif dans la structure joueur qu'il accumule en remportant des cagnottes de parties.

### IV. 3. Synthèse

Nous avons donc pu décomposer l'ensemble de notre travail effectué dans le cadre du projet d'études. Nous constatons que le fait de concrétiser une idée de programmer un concept sur logiciel est un processus qu'il faut surtout bien savoir démarrer pour pouvoir poser les limites et assurer le respect des contraintes nécessaires.

Suite à cela, nous avons les directions dans lesquelles le projet reste en développement continu. Ce développement n'est pas nécessairement fini, tellement il peut y avoir d'innovations et corrections qui s'imposent.

Pour terminer avec un petit mot, nous à trois tenons à nous remercier mutuellement pour l'effort, l'assiduité et le sérieux fourni par l'équipe. Nous devons cependant toute notre reconnaissance à nos professeures en matière d'algorithmique avancée et programmation; Mme AL ASSAAD Hiba, et notre encadrante Mme GUIBADJ Rym Nesrine. Merci pour votre dévouement à nous soutenir et nous guider tout au long de nos travaux, cours et nos projets d'études.

Nous vous remercions, cher lecteur, pour nous avoir prêté votre attention.