TP7 – PERFECTIONNEMENT EN C

BOGGLE

SOMMAIRE :

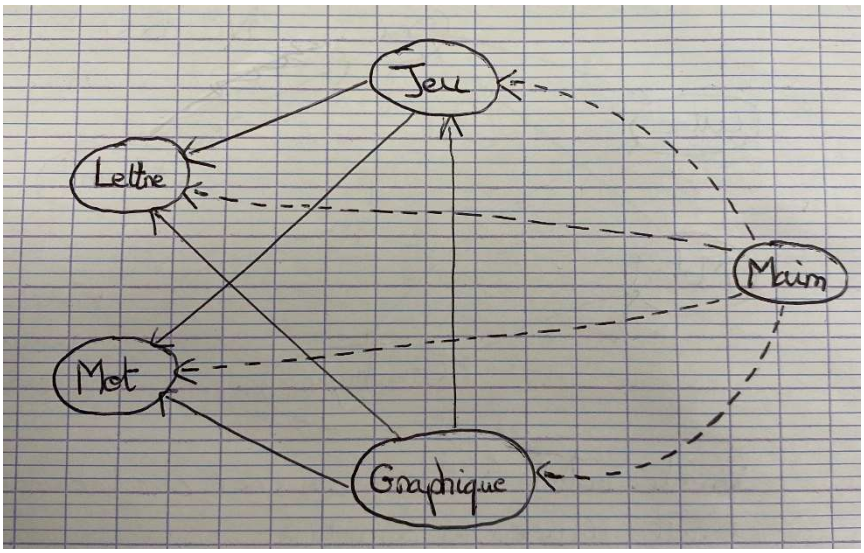
INTRODUCTION :

Durant ce TP, nous devons concevoir le jeu du Boggle. En général, le jeu se joue à plusieurs (dans notre cas, il n'y aura que 1 seul joueur). Il est constitué de 16 dés au total de 6 faces chaque une et d'une grille 4x4. Lors, du début du jeu on tire au hasard les 16 dés, puis chaque dé tiré, on tire une lettre au hasard que l'on place sur la grille. Ensuite, chaque joueur note sur une feuilles les mots qu'il trouve sur la grille pendant un temps imparti. A la fin, on compte le nombre de points pour chaque joueur si le mot est correct et qu'il contient plus de 2 lettres, alors le mot lui rapport des points. En revanche, si le mot est faux il perd une vie. Ainsi, pour ce TP, nous verrons la modularisation du jeu, puis l'implémentation des dés, de la grille, des mots et du dictionnaire en mémoire. Puis pour finir, nous verrons la partie graphique de jeu et l'implémentation du jeu.

MODULARISATION DU PROJET :

Tous d'abord, nous avons modulariser le projet en divers modules pour nous simplifier la tâche. Ceci passe par la création des fichiers et l'écriture du Makefile pour compiler le projet. (Nous verrons l'utilisation du Makefile dans l'annexe de compilation du projet). Ainsi, nous avons utilisé un module Lettre pour gérer l'initialisation des dés et de leur liste chainée. Dans ce même module, il y a une structure qui permet de sauvegarder les mots déjà joué par le joueur. Ceci permet d'interdire le joueur de rejoué le même mot (sinon c'est trop facile …)

Ensuite, dans un deuxième module Mot, nous avons implémenter le dictionnaire qui permet de valider le mot joué par le joueur. Le quatrième module Jeu, permet de gérer la validation des mots, la fin du jeu, la construction du mot que le joueur souhaite jouer et l'attribution des points pour chaque mot juste du joueur. Enfin, le dernier module Graphique gère les fonctions graphiques du jeu à partir de la bibliothèque graphique NCURSES.



LES DES ET LES MOTS :

Pour gérer les dés et les mots déjà joué par l'utilisateur nous avions crée deux structures qui sont des listes chainées. Pour les dés, la liste contient comme un tableau dynamique de 6 caractères. Et pour les mots, la liste contient un mot de type char passé par adresse. Ensuite, il pour gérer les listes, nous avions écrits les fonctions d'allocation, d'insertion en tête de la liste, de libération de liste et de vérification si un mot est déjà présent dans la liste des mots. Pour les dés, nous avions utilisé la « version internationale » du jeu, composée des 16 dés suivants :

- E, T, U, K, N, O ;
- E, V, G, T, I, N ;
- D, E, C, A, M, P ;
- I, E, L, R, U, W ;
- E, H, I, F, S, E ;
- R, E, C, A, L, S ;
- E, N, T, D, O, S ;
- O, F, X, R, I, A ;
- N, A, V, E, D, Z ;
- E, I, O, A, T, A ;
- G, L, E, N, Y, U ;
- B, M, A, Q, J, O ;

- T, L, I, B, R, A ;
- S, P, U, L, T, E ;
- A, I, M, S, O, R ;
- E, N, H, R, I, S.

Nous avions donc inséré les dés comme des tableaux dynamiques dans la liste chainée. Puis nous avions écrits une fonction qui prend un dé dans la liste et replace le dé à la fin de liste. Puis, prend une lettre dans ce dé et renvoie cette lettre. Enfin, il a besoins d'une fonction qui tire au sort un dé et une lettre dans ce dé renvoie la lettre tirée au sort (cette fonction utilise la fonction précédente). Enfin, nous avons écrit une structure Plateau qui contient un tableau dynamique à deux dimensions de 4x4 pour la grille. Puis, nous avons rempli ce tableau à l'aide des fonctions précédentes.

LE DICTIONNAIRE :

Pour la gestion du dictionnaire, nous avons utilisé un arbre ternaire de recherche (ATR) issue du projet d'Algorithmique des arbres écrit par Gaël PAQUETTE et Hervé NGUYEN. Ainsi, dans cette partie nous avons écrit toute la fonction nécessaire à la gestion de l'arbre. C'est-à-dire, la structure de l'arbre, la création d'un arbre vide, la libération de l'arbre, l'insertion d'un mot dans l'arbre, la suppression d'un mot dans l'arbre, l'affichage des mots de l'arbre.

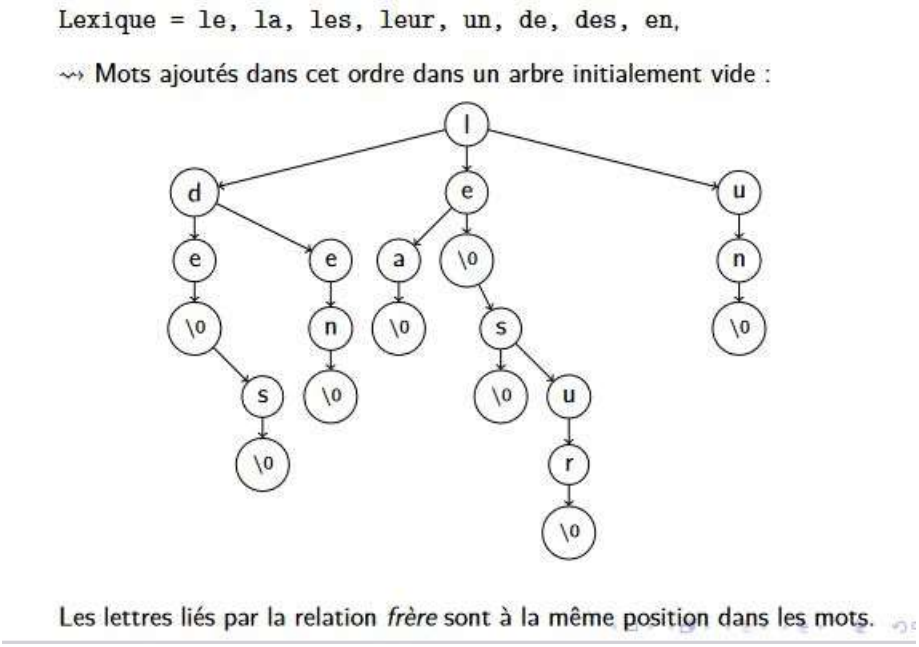


Image issue du cours d'Algorithmique de Arbres par Mr. BOUILLOT

LE JEU ET LES GRAPHISMES :

Dans cette partie, nous verrons comme nous avions implémenter le jeu et les graphismes du jeu. Pour le jeu, nous avions besoins de 4 fonctions pour valider un mot, compte le score, construire un mot (ajout d'une lettre au mot), et savoir si le jeu est fini. Pour les graphismes nous devons afficher les règles du jeu, le score, le mot en construction, la grille de jeu, la surbrillance de la lettre sélectionner, la saisie d'un mot graphiquement. Pour la saisie du mot, nous avions établi la liste de tâches à effectuer par la fonction dans le cas ou la touche entré soit sélectionné, la touche r, la touche a, la touche q et les touches du pavé directionnelle. Ensuite dans la fonction principale du jeu « main » nous avions implémenter l'algorithme suivants :

Initialisation des dés

Initialisation du plarteau

Tant que partie non finie :

    Afficher le score

    Afficher la grille

    Récupérer le mot saisi à l'écran

    Si le joueur décide de sortie de la partie :

        Arrêter tous à afficher la fin du jeu

    Si le mot n'est pas valide :

        Enlever une vie au joueur

    Sinon :

        Mettre le mot jouer dans la liste des mots déjà jouer par le joueur

Afficher la fin de la partie

Attendre 500000 mili-secondes

CONCLUSION :

Lors de ce TP, nous avons pu mettre en place le jeu du Boggle. Pour cela, nous avions procéder en 5 phases. La première consistait à découper convenablement le programme en divers modules. Ainsi que, la création des fichiers et du Makefile. Ensuite, nous nous somme occupé de la gestion des dés et des mots en mémoire. Ainsi que le tirage au sort des lettres des dé pour constituer la grille de jeu. Puis, nous nous somme occupé de la gestion de la arbre ternaire de recherche pour le dictionnaire. Enfin, nous nous sommes occupé de la gestion du jeu et des graphismes du jeu.

ANNEXE 1 : (Compilation du projet)

Pour compiler le projet, nous avons utilisé un Makefile qui permet de compiler les divers modules. Pour cela, il faut utiliser les commandes suivantes :

- make boggle (pour compiler)
- ./boggle Mots (Pour exécuter, «Mots » c'est le dictionnaire)
- make mrproper (pour supprimer l'exécutable et les fichiers .o)

ANNEXE 2 : (Manuel Utilisateur)

Lors de l'exécution du programme, un page de jeu apparait avec la grille le score et la mot construit. Pour ce déplacé sur la grille, il faut utiliser les touches directionnelles (attention si vous venez de sélectionner une lettre vous ne pouviez pas quitter le périmètre de 1 cases autour de la lettre). Pour sélectionner une lettre, il faut utiliser la touche q (attention vous ne pouviez pas écrire un mot de plus de 30 lettres car il n'existe aucun mot de plus de 30 lettres). Pour réinitialiser un mot veuillez appuyer sur la touche a. Pour valider le mot que vous avez écrit, il faut utiliser la touche entrée. Si votre mot est plus court ou égale à 2, il vous rapport zéro point. Si votre mot est vide vous perdez une vie (attention au bout de 4 vies perdu, la partie s'arrête et vous donne votre score). Enfin, vous ne pouviez pas saisir un mot déjà jouer.