2015-2016



RAPPORT DE PROJET DE PROGRAMMATION EN C

ESIEA | 74 bis Avenue Maurice Thorez, 94200 Ivry-sur-Seine

Semestre 2 (Mr françois)

Rafael Boban / CéDRIC SEBOGO (1A\_11)

**SOMMAIRE**

1. **Introduction : Présentation du projet**………….…......2
   1. Consignes……………......……………......…………….........2
   2. Règles du jeu……………......……………......……………....2
2. **Réalisation et fonctions du programme**……………...2
   1. L’interface du menu.………………………………….…….2-3
   2. Fonctions de calculs de gains…..…………………..…3-4
   3. Ouverture lettres et chiffres ................................5-6
   4. Fonctions affichages des designs…..…………………..6
   5. Fonctions de déroulement du jeu………………...…7-8
   6. Fonction de sécurisation.……………………………....…..9
   7. Bibliothèques et librairies....…………………………..10-11
   8. Second thème…………………………………………………..11
3. **Bibliographie / Webographie**………………….....12
4. **Remerciements**………………………………...……13
5. **Conclusion**………………………….......……………13
6. **Introduction**
   1. **Consignes**

Ce projet consiste à programmer en langage C un simulateur de machine à sous.

Le projet est à réaliser en binôme sous environnement GNU/Linux, pour une durée d’environ 2 mois et demi.

* 1. **Règles du jeu**

Au début du jeu le joueur dépose un certain nombre de crédits, par exemple 100 crédits.

Avant chaque lancement de la machine, le joueur peut choisir de miser soit 1, 2 ou 3 crédits. Après chaque lancement, la machine indique le gain obtenu en fonction de la combinaison qui est apparue. Bien entendu, le gain obtenu correspond au gain de base lié à la combinaison multiplié par la mise du joueur.

Le jeu se poursuit tant que le joueur le souhaite et que son nombre de crédits est positif. Le jeu se termine automatiquement dans le cas où le joueur ne dispose plus de crédits.

Le joueur peut également décider d’encaisser les crédits disponibles à tout moment du jeu.

1. **Réalisation et fonctions du programme**
   1. **L’interface du menu**

Plusieurs étapes nous ont été nécessaires à la conception du code du jeu. Nous avons fait appel à un paquet de notions afin de confectionner cette interface de menu.

Extended ASCII, couleurs, caractères spéciaux, site de générations de lettre ( <http://www.patorjk.com/> ) , nous ont été nécessaires. Cela peut être résumé par la fonction *void print\_cadre():*

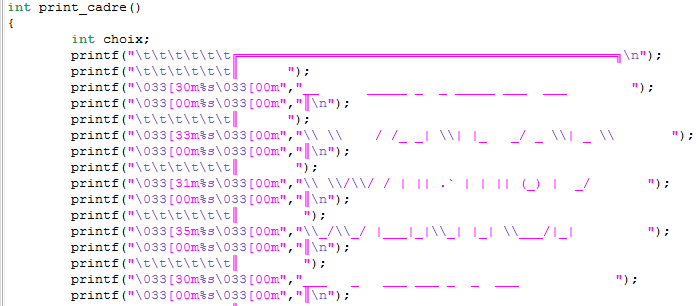
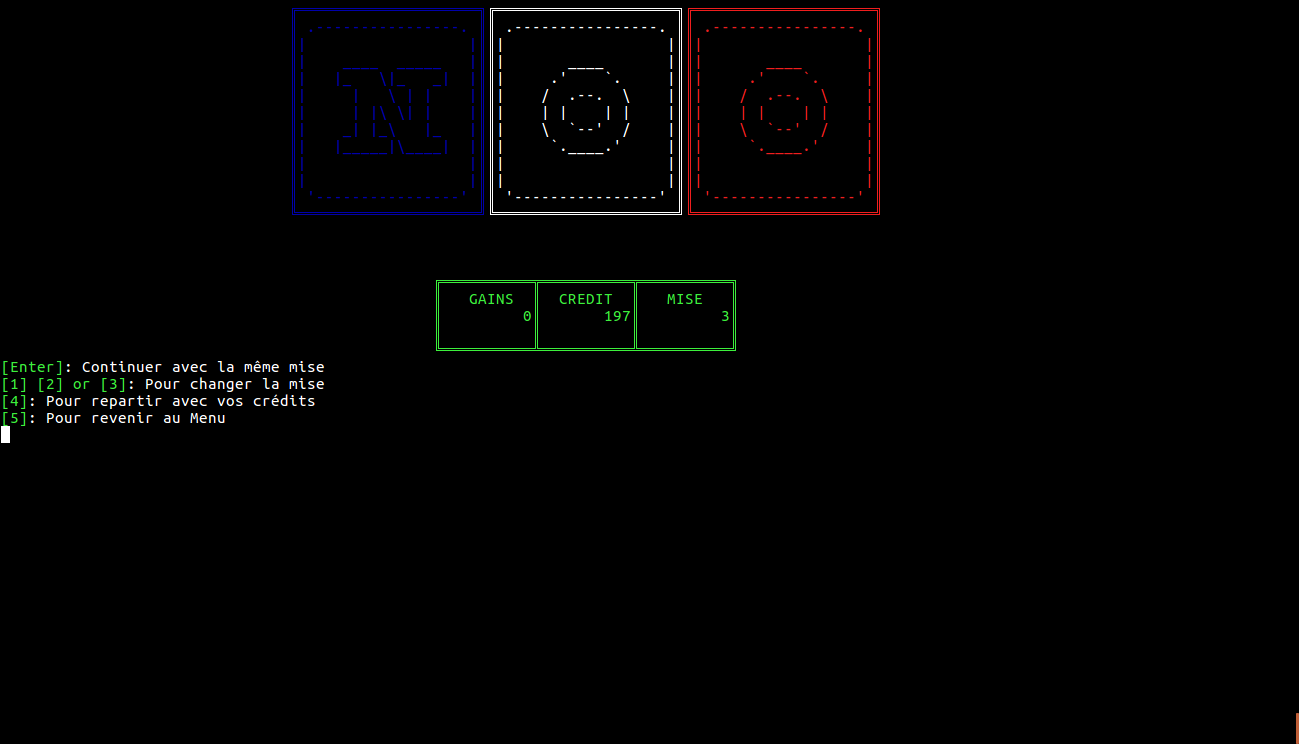
****

Figure 1 : Extrait de la fonction print\_cadre

* 1. **Fonctions de calculs de gains**

Plusieurs fonctions nous ont été nécessaires pour le calcul des gains, et des crédits. En effet plusieurs facteurs devraient être pris en compte : Mise, combinaisons, mode, calculs du crédit... Le but était d’avoir les informations sur la mise, le crédit et le gain à la fin de chaque jeu.



Gain : Affiche la somme que rapporte la combinaison. Crédit : Affiche le crédit restant.

Pour se faire, nous avons décidé de choisir 15 combinaisons gagnantes. Ensuite, nous avons créé, pour les deux modes, et pour les 3 mises des fonctions permettant d’avoir les informations. Au total 6 fonctions permettent de faire les calculs.

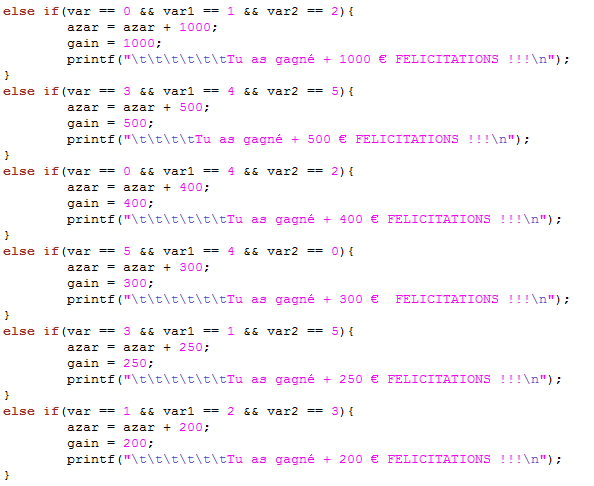
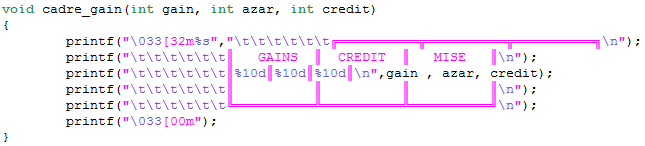


Figure 2: Fonction de gestion crédits/gains

Le bout de code ci-dessus ajoute le gain au crédit puis retourne le gain afin que nous puissions afficher le gain d’une combinaison à travers un printf. Une fonction que nous avons nommé *cadre\_gain* permet d’afficher crédit, gains et mises.



*Azar* représente le crédit et *credit* représente la mise.

* 1. **Ouverture lettres et chiffres**

Pour l’ouverture et la fermeture de fichiers, fopen, fgets, memset et autres nous ont été nécessaires. Pour se faire nous avons créé au total 12 fichiers de type « .txt » contenant les lettres et chiffres utilisés dans le jeu.

* **Ouverture et fermeture**

Lorsque nous ouvrons un fichier avec la commande fopen, le système génère automatiquement un bloc du type FILE et nous fournit son adresse.

Ce que nous avons fait dans notre programme est :

\* déclarer un pointeur du type **FILE**\* pour chaque fichier dont nous avons besoin,

\* affecter l'adresse retournée par **fopen** à ce pointeur,

\* employer le pointeur à la place du nom du fichier dans toutes les instructions de lecture ou d'écriture,

\* libérer le pointeur à la fin du traitement à l'aide de **fclose**.

Par cette méthode, nous arrivons à ouvrir et fermer les fichiers « .txt ».

* **Affichage**

Afin d’afficher les lettres et chiffres, nous avons procéder par l’affichage ligne par ligne.

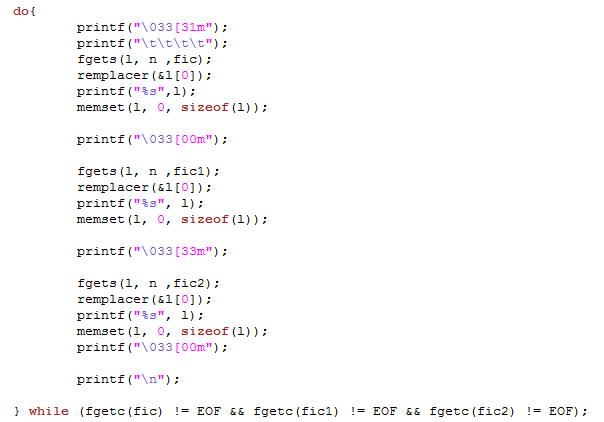


Figure 3 : Méthode ligne par ligne

Tout un processus, permettant l’affichage, se cache derrière ce code.

- fgets : celui-ci permet de lire la 1ère ligne du 1er fichier aléatoire ouvert.

- remplacer(&l[0]) : permet de charger cette ligne

- printf : permet d’afficher cette ligne

- memset : permet d’effacer la ligne afficher précédemment.

Ce processus se poursuit avec l’affichage de la 1ère ligne du 2ème fichier, ensuite la première ligne du 3ème fichier et ainsi de suite. C’est ainsi que l’on obtient un rendu qui ressemble à cela :

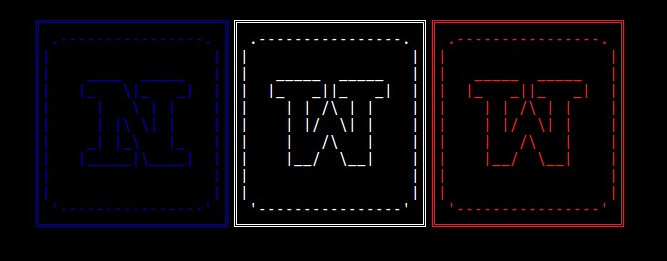


Figure 4: Affichage

* 1. **Fonctions affichages des designs**

Pour l’ouverture des différents designs aperçus dans le jeu, une méthode fut utilisée : fopen + fgets + printf. Illustrés par :

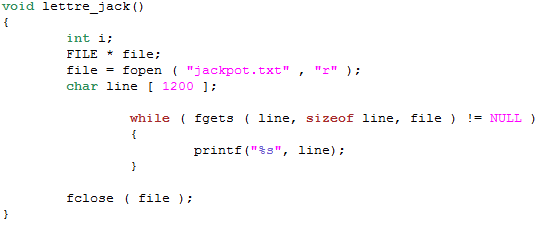


Figure 5: Designs

Ceci est un exemple permettant d’afficher le message jackpot contenu dans un fichier texte.

* 1. **Déroulement du Jeu**

Nous avons décidé d’appeler le jeu WINTOP CASINO car w, i, n, t, o, et p sont les lettres qui sont utilisés dans le thème principal. Pour le lancement du jeu, le joueur doit s’assurer d’être dans le dossier contenant les données du jeu. Ensuite, la commande make doit être utilisée pour le lancement. Après le message de bienvenue (welcome), nous avons le menu qui se présente :



Figure 6: Menu

Sur la capture d’écran ci-dessus, nous pouvons voir les différentes parties du jeu:

1. Jouer

Dans cette partie, le joueur devra choisir entre deux thèmes. Les principes des 2 thèmes reste le même mais la différence est que dans le thème1 (*mode classique*), il aura des lettres et le second (*mode chiffres*) le joueur aura des chiffres. Après avoir choisi le mode, il devra entrer successivement le nombre de crédit, puis sa mise.

Le joueur est par la suite guidé par des informations que nous avons laissées tout au long du jeu.

1. Règles

Cette partie informe les joueurs sur tout le jeu. C’est une sorte d’explication en quelques mots du fonctionnement de notre jeu.

1. Conditions de victoires

Nous avons inséré dans cette partie du jeu les combinaisons gagnantes des chiffres comme des lettres et qui permet au joueur d’en connaitre un peu plus sur notre chef d’œuvre.

1. Privacy policy

Cette partie du jeu nous « fais honneur » c’est-à-dire donne les noms de ceux qui l’ont conçu : *Cedric SEBOGO and Rafael BOBAN*.

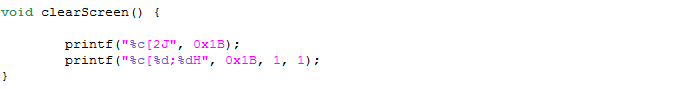
1. Mode jackpot

Ce mode a été créée afin que tout joueur puisse voir quelle serait l’animation s’il arrivait qu’il gagne le *JACKPOT.*

1. Exit

Celui-ci permet de quitter le jeu avec un message de fin que nous vous avons soigneusement concocter.

Pour que l’écran de jeu soit le plus clair possible, l’appel de fonctions « *ClearScreen* » est nécessaire.



Celle-ci permettait d’effacer l’écran et d’afficher que la partie suivant le *clearsreen*.

* 1. **Méthode de sécurisation**

Tout au long de notre programme, nous avons opter pour la sécurité *do...while* mais aussi celle du avec le *getchar.* La 1ère, celle *do..while*, permet de tout le temps poser une question lorsque les conditions que nous, Cédric et Rafael, avons requises ne sont pas remplies. Nous les avons pour la plupart du temps mis autour des *scanf* qui requière par exemple soit 1, 2, ou 3. Exemple :

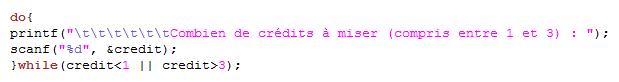


Figure 7: Sécurité avec scanf

La deuxième méthode permet des sécuriser grâce à *getchar*.

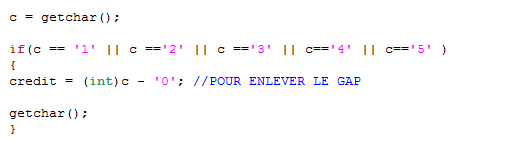


Figure 8: Sécurité avec getchar

Ce bout de code ci-dessus permet de refaire la saisir si celle-ci n’est pas valide. C’est une sorte de *ClearBuffer*.

* 1. **Bibliothèques et Librairies**

1. Bibliothèques

Nous avons placé nos bibliothèques dans le fichier fonction.h de notre makefile afin de réaliser une compilation séparée (via Cours 12 de Mr. François).

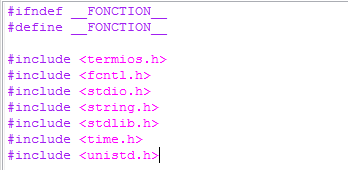
****

Figure 9 : Bibliothèques

*« time.h »* est utilisée pour les délais d’affichage des textes. Nous utilisons *usleep* pour le faire. Et pour les pauses, nous avons utilisé la fonction *my\_delay* décrite plus haut.

Les bibliothèques *termios.h* et *fcntl.h* permet de faire fonctionner la fonction *key\_pressed.*

« unistd.h » permettait d’utiliser fseek.

1. Librairies

Nous avons rajouté la librairie SOX (avec la commande, sudo apt-get sox). Celle-ci nous a permis de mettre de la musique en arrière-plan tout en faisant fonctionner notre programme. Comme exemple :

C:\Users\Cédric\Dropbox\TP D'info\Projet\Son bruit.png

* **play** : Permet de jouer la musique.
* **-q** : Permet d’effacer toute les informations sur la musique.
* **‘&’**: fait fonctionner la musique en arrière-plan.

C:\Users\Cédric\Desktop\dc619a7f5abcd3c8a8da977b55a31ab9.png

* **killall** : permet de stopet toute musique en arrière plan.
  1. **Second thème**

Notre premier thème étant celui avec les lettres, nous devions réaliser notre jeu avec un second thème.

Notre choix s’est alors porté sur le thème des chiffres (1, 2, 3, 4, 5 et 6). Ainsi entre nos 2 thèmes le principe est resté le même mais les animations ont été un peu modifiés.

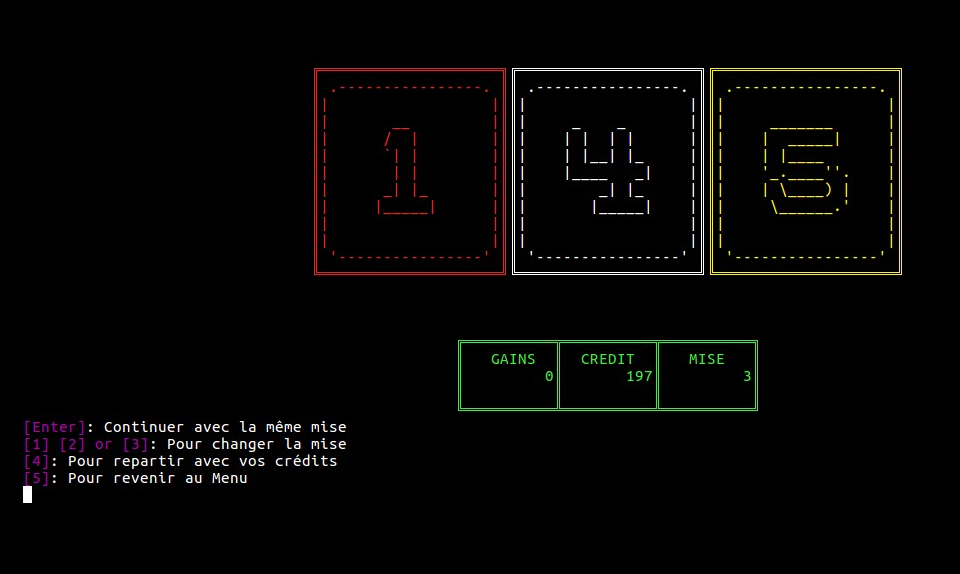


Figure 10: Second thème

1. **Bibliographie et Webographie**
2. Cours du 1er semestre 1A de Mr Beaudoin
3. Cours du 2nd semestre 1A de Mr François
4. Site internet Open Classrooms : <https://openclassrooms.com/forum/categorie/langage-c>
5. Site internet Iron Geek : <http://www.irongeek.com/alt-numpad-ascii-key-combos-and-chart.html>
6. Site internet The Ascii Code : <http://www.theasciicode.com.ar>
7. Site internet Table ASCII -I: [www.cdrummond.qc.ca/cegep/informat/Professeurs/Alain/files/ascii.htm](http://www.cdrummond.qc.ca/cegep/informat/Professeurs/Alain/files/ascii.htm)
8. **Remerciements**

Nous tenons à remercier toutes les personnes qui ont contribué au succès de notre projet et qui nous ont aidé lors de la réalisation de ce dernier.  
  
 Ainsi, nous adressons nos remerciements à notre professeur, **Mr François** qui nous a aidé dans notre recherche de perfectionnement de notre projet. Son écoute et ses conseils nous ont permis d’améliorer notre programme, et nos capacités en programmation afin de pouvoir finaliser notre programme tel que nous le souhaitions et nous l’imaginions au début de ce projet.

1. **Conclusion**

Pour conclure, nous estimons dans l’ensemble que nous sommes parvenus à l’objectif que nous nous étions fixés au début du semestre qui était de pouvoir réaliser un jeu en état fonctionnel, accessible à tous et design.

Ce projet de programmation nous a permis d’exploiter nos compétences et nos connaissances en C.

Ce fût un plaisir de pouvoir le réaliser en binôme car cela permet de développer l’esprit d’équipe et d’améliorer l’écoute de ses collèges.