Cahier Des Specifications

Alabi Steve - Benyamna Younes - Capdenat Nicolas-
Chouipe Thibaut - El Harti Zakaria - Lienhardt Florian

Chef de projet : Benyamna Younes

Sous la direction de Mme Kloul

Outil automatique de décryptage

18 avril 2017

1 Introduction

Dans le cadre de notre troisième année de licence nous devons réaliser une application d'aide au décryptage par la méthode de Substitution ainsi que celle de Vigenère. Nous vous avons donc proposé l'application Dcrypt qui permet de décrypter mais également crypter un texte ou même de faire une analyse fréquentielle sur un texte. Pour cela, nous avons rédigé un cahier des charges afin de clarifier les objectifs et les exigences de ce projet. Dans la suite du cahier des charges, nous vous proposons donc le cahier des spécifications.

Pour développer notre projet et après étude des besoins les arguments en faveur du langage C sont les suivants :

-Tout d'abord, le C permet d'utiliser des expressions et des opérateurs qui sont très proches du langage machine, il est donc possible de développer des programmes efficients et rapides.

-La portabilité est aussi un argument de choix : En effet, en respectant le standard ANSI-C, il est possible d'utiliser le même programme sur tout autre système (autre hardware, autre système d'exploitation), simplement en le recompilant.

-De plus, pour assurer la maintenance du produit apres sa realisation et ce meme par des developpeurs qui ne sont pas ceux d'origine, il faut un langage des plus utilisés et maitrisés

-Enfin, selon l'IEEE(*), le langage C est depuis 2016 le meilleur langage de programmation(de par sa forte croissance et sa demande par les employeurs). C'est aussi le langage n=1 pour le développement d'applications d'entreprise, de bureau et d'applications scientifiques.

Nous avons également choisi une bibliothèque graphique du langage C : GTK+. Cette dernière nous permet de gérer efficacement la navigation entre les differentes fenêtres ainsi que nos accès memoires pour la sauvegarde et le chargement de nos fichiers. IEEE(*) : Institut des ingénieurs électriciens et électroniciens. C'est la plus grande organisation professionnelle technique du monde de l'évolution de la technologie

2 Organigramme

Interface graphique :
- bouton cryptage
- bouton decryptage
- bouton substitution
- bouton Vigenère
- affichage de texte(complet et partiel)
- affichage pour la clé de substitution
- bouton Francais(decryptage)
- bouton Anglais(decryptage)
- affichage pour l'analyse fréquentielle
- charger un fichier texte
- sauvegarder un fichier texte
- créer un nouveau fichier texte(resultats)
- Demander clef de Vigenère

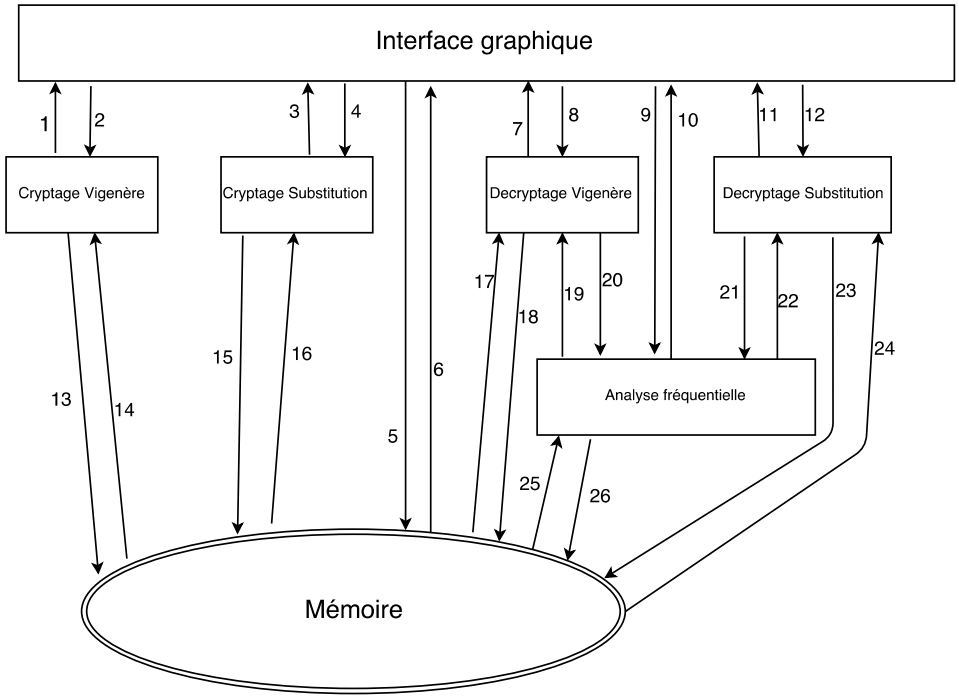
Decryptage Vigenère :
-decrypter le message

Cryptage Substitution :
-créer une clé aleatoirement
-crypter le message

Cryptage Vigenère :
-crypter le message

Decryptage Substitution :
-decrypter le message

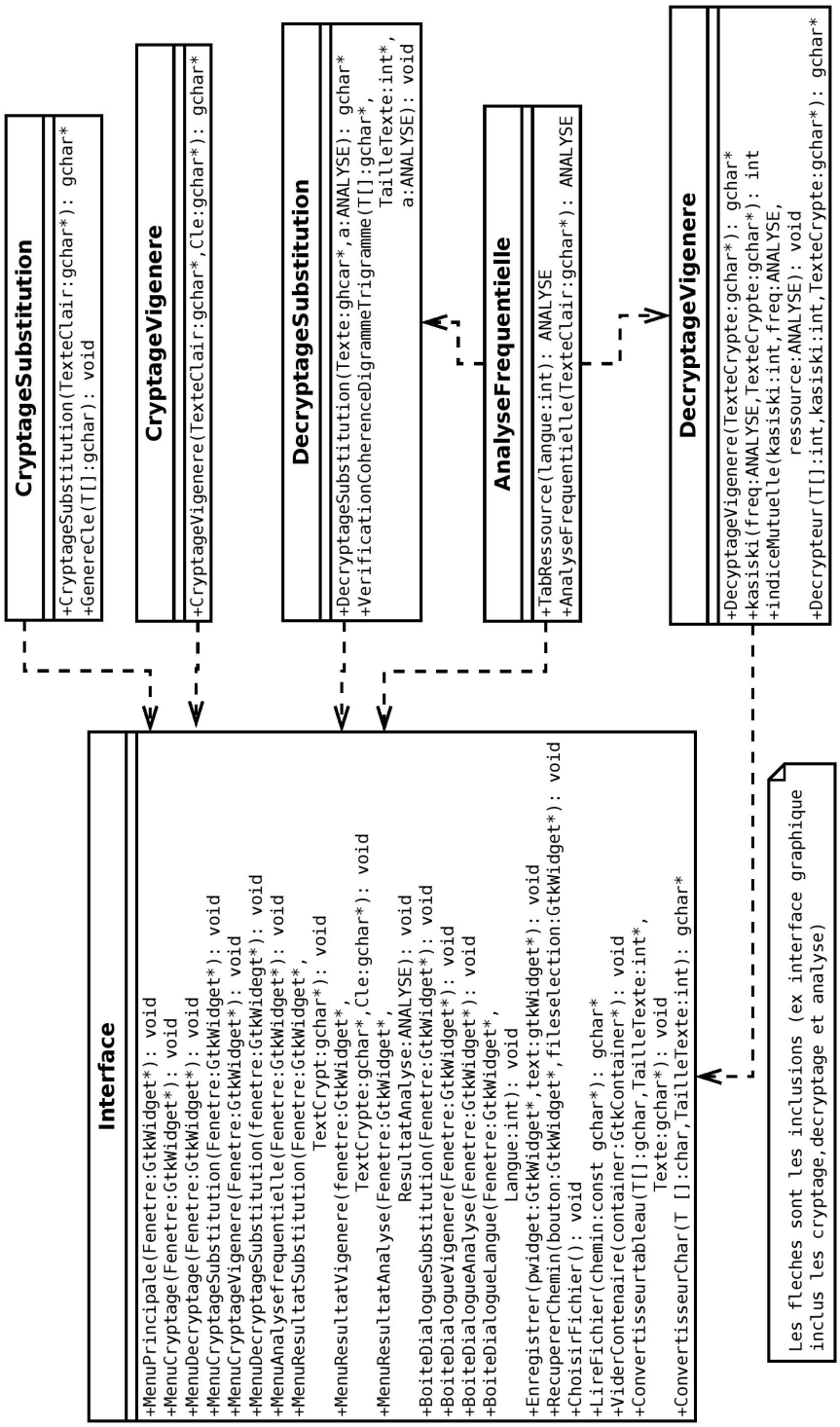
Analyse fréquentielle :
-analyse frequentielle sur texte donné



1) Un ensemble de données contenant le texte crypté
2) Un nom de fichier
3) Un ensemble de données contenant le texte crypté
4) Un nom de fichier
5) Un nom de fichier
6) Fichier texte : clair, crypter, décrypter ou analyse fréquentielle
7) Un ensemble de données contenant le texte en clair
8) Un nom de fichier
9) Un nom de fichier
10) Un ensemble de données contenant l'analyse fréquentielle
11) Un ensemble de données contenant le texte décrypté ou le texte clair
12) Un nom de fichier
13) Un nom de fichier
14) Fichier texte clair
15) Un nom de fichier
16) Fichier texte clair
17) Un fichier texte crypté
18) Un nom de fichier
19) Un ensemble de données contenant l'analyse fréquentielle
20) Un nom de fichier
21) Un nom de fichier
22) Un ensemble de données contenant l'analyse fréquentielle
23) Un nom de fichier
24) Un fichier texte crypté
25) Un fichier texte en crypté ou en clair
26) Un nom de fichier

3 Signatures et Explications des methodes

3.1 Diagramme des fonctions des differents modules



3.2 Structures et nouveaux types

Le gchar* est une chaîne de caractères, on utilise pas char* car dans la bibliothèque GTK+ l'affichage d'un texte se fait avec gchar*.

les widgets (GTKWidget) sont les boutons, les zones de texte, les menus, enfin.. à peu près tout ce qui constitue une interface.

la seule variable globale utilisée dans notre programme est Fenetre. Elle nous permet de conserver la fenêtre principale de notre application :
GtkWidget *Fenetre;

struct phoneme{
int frequence;
gchar* nom;
}PHONEME;
Cette structure represente les digrammes ou trigrammes et leurs frequences.

struct analyse{
int nb //nbre de lettre
float occ[25];
PHONEME di[];
PHONEME tr[];
gchar* pgor;
}ANALYSE;
Cette structure correspond aux caracteristiques d'un texte. Elle sera remplie lors de l'appel de la fonction "AnalyseFrequentielle". Ainsi, on connaitra le nombre de caracteres dans celui-ci, les digrammes/trigrammes, la plus grande occurence rencontrée et la frequence d'apparition de chaque lettre.

struct ressourceslangue{
float occ[25];
PHONEME di[];
PHONEME tr[];
}RESSOURCESLANGUE;
Cette structure correspond aux caracteristiques d'une langue. On connaitra les digrammes/trigrammes et la frequence d'apparition de chaque lettre dans la langue choisie.

3.3 Signatures

3.3.1 Analyse Frequentielle

ANALYSE TabRessource(int langue);
la fonction prend en entrée un entier correspondant à la langue(francais ou anglais) et va ainsi remplir la structure "ANALYSE" et ses attributs suivants : la frequence d'apparition de chaque lettre dans la langue choisie ainsi que les digrammes et les trigrammes.

ANALYSE AnalyseFrequentielle(gchar* TexteClair)
la fonction prend en entrée le texte clair a analyser et va ainsi remplir la structure "ANALYSE" et ses attributs suivants : le nombre de lettres, la fréquence d'apparition de chaque lettre, les digrammes, les trigrammes et la plus grande occurence rencontrée.

3.3.2 Cryptage Substitution

gchar* CryptageSubstitution(gchar* TexteClair);
Cette fonction permet de crypter un texte avec la methode de Substitution

Pour cela, elle prend en entrée une chaine de caractères qui correspond au texte clair.
En sortie, le texte crypté sera sous forme de chaine de caractères.
Le texte a crypter se trouve dans un tableau que je vais parcourir. Et pour chaque caractere nous allons aller lire dans un autre tableau [2][26] ou la premiere ligne correspond a l'alphabet de base et la deuxieme celui de substitution(le nouveau). Ainsi, nous regardons la lettre dans le 1e tableau (du texte a crypter), on bascule dans le 2e tableau et trouve ce caractere dans la 1e ligne, puis on regarde sa lettre de substitution, et enfin on remplace dans le 1e tableau la lettre de base par ce caractere de substitution.

void GenereCle(gchar T[]) ;
genere une clé(alphabet) aléatoirement

3.3.3 Decryptage Substitution

gchar* DecryptageSubstitution(gchar* texte, ANALYSE a)
Prend en argument un texte crypté.
Aprés avoir appelé la fonction d'analyse fréquentielle, cette fonction va, dans une boucle, comparer les résultats de l'analyse frequentielle à l'alphabet correspondant à la langue choisie et remplacer une lettre du message crypté par la lettre qu'elle semble crypter grace à une conjecture basée sur les fréquences des lettres les plus utilsés et complétée par l'analyse des digrammes et trigrammes. Cette fonction appelera ensuite la fonction VerificationCoherenceDigrammeTrigramme.
Retourne en sortie le texte décrypté (gchar*).

void VerificationCoherenceDigrammeTrigramme(gchar* T[], int *TailleTexte, ANALYSE a)
Prend en argument le texte crypté en cours de décryptage.
Cette fonction va parcourir le texte en cours de décryptage pour vérifier que les lettres trouvées ne forment pas de digramme ou trigramme inabituels. Si c'est le cas elle proposera à l'utilisateur de corriger le décryptage du texte ou le corrigera d'elle-même.
Retourne en sortie le texte crypté en cours de décryptage(gchar*).

3.3.4 Decryptage Vignere

gchar* DecryptageVigenere(gchar* TexteCrypte) ;
Cette fonction prend un texte crypté par le chiffrement de Vigenere
et renvoie un texte clair et lisible par l'utilisateur.

int Kasiski(ANALYSE freq, gchar* TexteCrypte) ;
Cette fonction cherche les différentes distances de PGOR(Plus Grande Occurence Rencontrée) et effectue le PGCD de ces distances. Elle renvoie ce résultat sous forme d'un entier (qui représente la taille du mot clé cherché).

void indiceMutuelle(int kasiski, ANALYSE freq, ANALYSE ressource) ;
Cette fonction calcule les indices de coïncidences mutuelles et les stocke sous forme d'un tableau tab[25][kasiski].
Elle parcourt et compare les 26 valeurs de chaque ligne afin de reperer celles qui sont proches de 0,065 (une par ligne).
Elle renvoie l'indice de chacunes des valeurs valides dans un tableau a une dimension.

gchar* decrypteur(int T[], int kasiski, gchar* TexteCrypte) ;
Cette fonction soustrait la valeur du mot clef (tableau renvoyé par indiceMutuelle) au texte crypté.
Elle va repeter la suite de nombre composant le mot clef jusqu'a la fin du texte afin de toujours pouvoir soustraire une valeur du mot clé a une valeur du texte crypté.
Elle renvoie le texte claire.

3.3.5 Cryptage Vignere

gchar* CryptageVigenere(gchar* TexteClair, gchar* Cle) ;
Cette fonction permet de crypter un texte avec la methode de Vigenere
Elle prend en argument le texte clair et la clé qui sont tout les deux des chaines de caracteres.
La fonction renvoiera le texte crypté
On construit un tableau [3][taille texte]. En premiere ligne on trouvera le texte a crypter, en 2e la clé (qu'on repetera de facon a couvrir toute la 1e ligne(message a chiffrer)) et enfin en 3e ligne le resultat de l'addition des caracteres des deux premieres lignes le tout modulo 26(c'est a dire le texte maintenant crypté).

3.3.6 Interface Graphique

Dans ce module nous allons utiliser plusieurs menus pour afficher notre application et naviguer entre les menus pour cela nous allons creer plusieurs procedures qui vont afficher les menus
Les Menus

void MenuPrincipal(GtkWidget *Fenetre) ;
Cette procedure permet d'afficher le menu principal.
Elle prend en argument la fenetre qui sera le contenaire(des boutons et labels).
Cette procedure ne renvoie rien .

void MenuCryptage(GtkWidget *Fenetre) ;
Cette procedure permet d'afficher le menu de cryptage.
Elle prend en argument la fenetre qui sera le contenaire(des boutons et labels).
Cette procedure ne renvoie rien .

void MenuDecryptage(GtkWidget *Fenetre) ;
Cette procedure permet d'afficher le menu de decryptage.
Elle prend en argument la fenetre qui sera le contenaire(des boutons et labels).
Cette procedure ne renvoie rien .

void MenuCryptageSubstitution(GtkWidget *Fenetre) ;
Cette procedure permet d'afficher le menu de cryptage par la méthode de substitution.
Elle prend en argument la fenetre qui sera le contenaire(des boutons et labels).
Cette procedure ne renvoie rien .

void MenuCryptageVigenere(GtkWidget *Fenetre) ;
Cette procedure permet d'afficher le menu de cryptage par la méthode de vigenere.
Elle prend en argument la fenetre qui sera le contenaire(des boutons et labels).
Cette procedure ne renvoie rien .

void MenuDecryptageSubstitution(GtkWidget *Fenetre) ;
Cette procedure permet d'afficher le menu de decryptage par la méthode de substitution.
Elle prend en argument la fenetre qui sera le contenaire(des boutons et labels).
Cette procedure ne renvoie rien .

void MenuDecryptageVigenere(GtkWidget *Fenetre) ;
Cette procedure permet d'afficher le menu de decryptage par la méthode de vigenere.
Elle prend en argument la fenetre qui sera le contenaire(des boutons et labels).
Cette procedure ne renvoie rien .

void MenuAnalyseFrequentielle(GtkWidget *Fenetre) ;
Cette procedure permet d'afficher le menu d'analyse frequentielle.
Elle prend en argument la fenetre qui sera le contenaire(des boutons et labels).

Cette procedure ne renvoie rien .

void MenuResultatSubstitution(GtkWidget *Fenetre, gchar* Textcrypt) ;
Cette procedure permet d'afficher le menu du resultat du decryptage par la methode de substitution.
Elle prend en argument la fenetre qui sera le contenaire(des boutons et labels).
Cette procedure ne renvoie rien .

void MenuResultatVigenere(GtkWidget *Fenetre, gchar* Textcrypt, gchar* cle) ;
Cette procedure permet d'afficher le menu du resultat du decryptage par la methode de vigenere.
Elle prend en argument la fenetre qui sera le contenaire(des boutons et labels).
Cette procedure ne renvoie rien .

void MenuResultatAnalyse(GtkWidget *Fenetre) ;//rajouer le texte
Cette procedure permet d'afficher le menu du resultat de l'analyse frequentielle.
Elle prend en argument la fenetre qui sera le contenaire(des boutons et labels).
Cette procedure ne renvoie rien .

void BoiteDialogueSubstitution(GtkWidget *Fenetre) ;
Cette procedure affiche une zone de texte qui nous permet de rentrer le texte à crypter par substitution.
Elle prend en argument la fenêtre.

void BoiteDialogueVigenere(GtkWidget *Fenetre) ;
Cette procedure affiche deux zones de texte qui permettent de rentrer le texte à crypter et la cle qui permet d'effectuer le chiffrement.
Elle prend en argument une fenêtre.

void BoiteDialogueAnalyse(GtkWidget *Fenetre) ;
Cette procedure affiche une zone de texte qui permet de rentrer le texte sur lequel effectuer une analyse fréquentielle.
Elle prend en argument une fenêtre.

void BoiteDialogueLangue(GtkWidget *Fenetre,int langue) ;
Cette procedure nous permet de choisir la langue (0 :francais , 1 :anglais).
Elle prend en argument une fenêtre.

nous aurons peut-être besoin d'autres menus au cours de notre application(pour un meilleur affichage ou ameliorations) //menu pour la langue
Les Enregistrement/Chargements

void Enregistrer (GtkWidget *pwidget, GtkWidget *text) ;
Cette procedure permet de sauvegarder le texte obtenue.
Elle prend en argument une fenetre et le texte à sauvegarder.
Cette procedure ne renvoie rien.

void RecupererChemin(GtkWidget *bouton, GtkWidget *fileselection) ;
Cette procedure permet de recuperer le chemin d'accés d'un fichier.
Elle prend en argument une fenetre et une deuxieme fenetre (pour les fenetres de selections de fichier)
Cette procedure appel

void ChoisirFichier() ;
Cette procedure nous permet de selectionner un fichier à charger (fichier qui contient le texte clair ou crypter)

gchar* LireFichier(const gchar* chemin) ;
Cette fonction permet de recuperer le texte qui est dans un fichier, pour cela nous donnons en argument un

chemin de fichier.

Autres

void ViderContenaire(GtkContainer * container) ;
Cette procedure nous permet de vider le contenu de la fenetre, cele nous permet de travailler avec une seule fenêtre.
Cette procedure prend en argument la fenêtre à vider.
void ConvertisseurTableau(gchar T[],int *TailleTexte,gchar* Texte) ;
enlever accent et espace, et mettre dans un tableau

gchar* ConvertisseurChar(char T[],int TailleTexte) ;
Parcours d'un tableau donné et le retourne en chaine de caracteres

4 Conclusion

Pour conclure, la rédaction de ce cahier des spécifications est une étape importante pour la réalisation du projet et plus précisément permet d'indiquer comment réaliser le besoin.

Elle se situe après l'étape de conception (cahier des charges) et avant l'étape de réalisation (codage).

Ce cahier permet donc de répondre de manière spécifique aux différentes reponses apportées, notamment les données en entrée ou en sortie des fonctions/procédures qui composent nos modules.

Le diagramme résume les fonctions/procédures présentes dans chaque module ainsi que les liens d'inclusion.

Maintenant que tout ces paramètres ont été définis et que l'équipe est d'accord sur l'architecture de l'application, nous allons passer à l'implémentation en respectant tout les règles fixées par le cahier des charges et des spécifications.