RAPPORT SUR LE TRAVAIL DU SYSTEME DE CODAGE ENIGMA

OBJECTIFS

L'objectif de ce projet est de coder un système/programme qui pourra répliquer le système de codage de la machine allemande enigma. Pour ce fait nous allons utiliser le langage python plus précisément la 3eme version. Le principe étant de fournir en sortie de l'algorithme un message codé, situé dans un fichier texte. Le message en question doit être traiter a l'entrée de sorte à ce que les lettres avec accents soit remplacé par les mêmes sans accent, les blancs doivent être supprimer et le message final doit être transformé en Majuscule avant d'être injecter dans le système qui fera le codage. Nous utiliserons la librairie « unicodedata ».

Nous disposerons d'un fichier json « rotor.init » qui contiendra 5 rotors au total et deux réflecteurs , nous utiliserons juste 3 rotors sur les 5 , un réflecteur et une clé « K » pour réaliser le cryptage et le décryptage . Avec les paramètres initiaux le premier « A » sera code par un « C »

Stratégie de résolution – Méthodologie

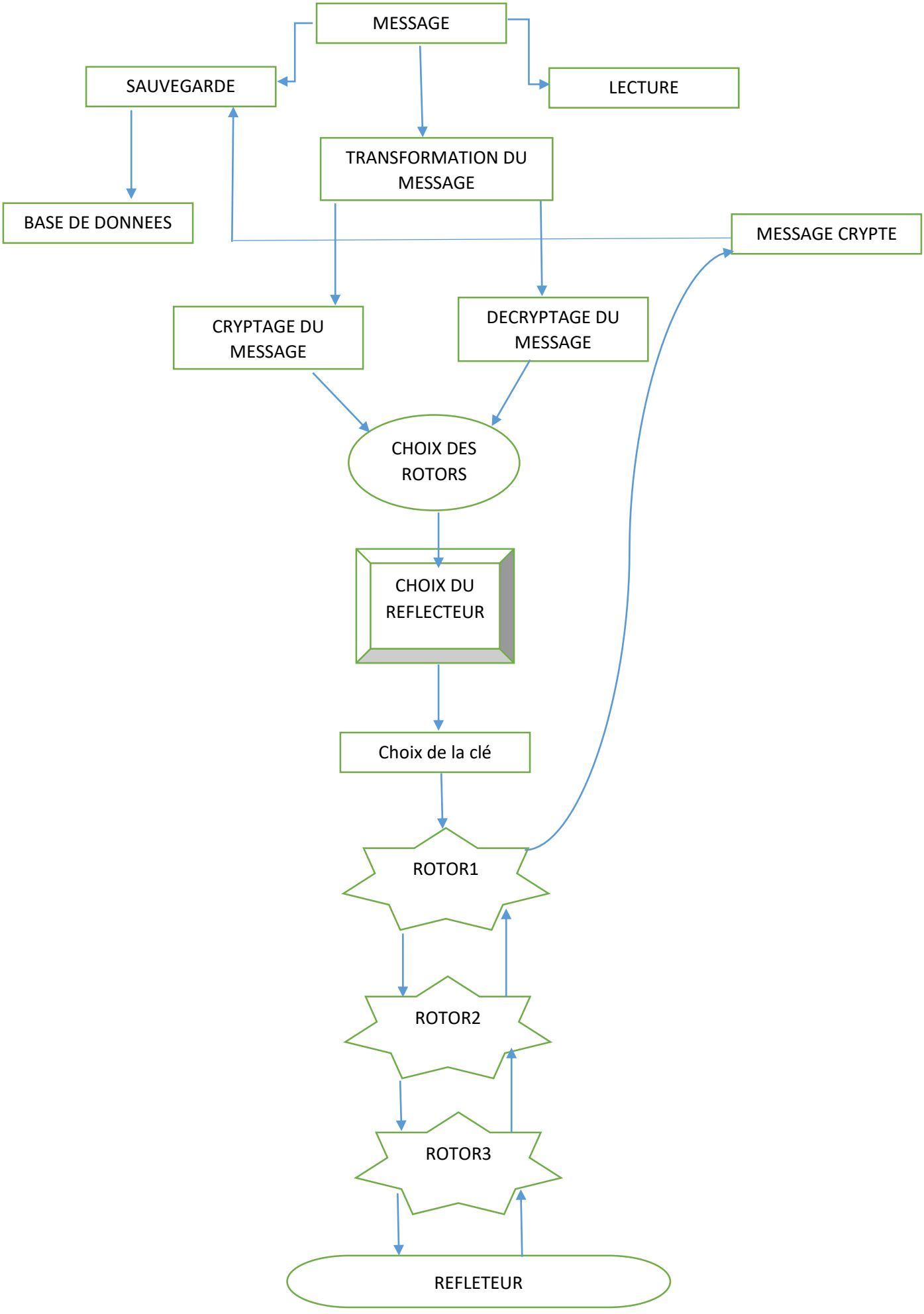
Ici il est question d'expliquer la méthode utiliser pour réaliser le problème, néanmoins avant d'y arriver nous allons donner une explication brève du fonctionnement de la machine Enigma enfin nous donnerons la solution que nous avons utiliser pour réaliser le nôtre. Le codage enigma est à la fois simple et ingénieux, chaque lettre est remplacée par une autre, il Ya un principe de substitution qui change d'une lettre à une autre. La machine enigma de base est alimente par une pile électrique, la pression sur une touche du clavier actionne une fermeture d'un circuit électriques qui allume une lampe qui indique qu'elle lettre codée l'on substitue.

- Tableau de connexion : sont rôle est de faire l'échange des paires de l'alphabet 2 à 2 au moyen de fiches, il y'a en tout 6 fiches qui permettent l'échange de 12 lettres
- Les rotors : ils jouent aussi le rôle de permutations mais cette fois dans un sens quelconque à chaque lettre saisie. Les rotors ont en tout 26 positions possibles correspondant aux lettre de l'alphabet. Il y avait en tous 5 ou 6 rotors (selon les document lus) dans la machine qui pouvaient être monte ensemble pur augmenter la complexité, l'on utilisait généralement 3 pour réaliser le codage
- Le réflecteur : il était placé après 3 rotors sont but était de faire une permutation supplémentaire enfin de complexifier plus le système

Pour le nôtre nous avons effectivement utiliser des rotors (3 précisément) sur lequel nous avons utilisé algorithme de César pour permuter les lettres dans le rotor en plus de la rotation de ceux-ci, ensuite nous avons utiliser un réflecteur (modifier). La méthode était la suivante,

nous avons fait un menu qui demande à l'utilisateur de faire un choix entre la lecture, sauvegarde ou le cryptage et décryptage du message se situant dans le fichier texte sauvegarde. En effet pour ce qui est du cryptage/décryptage (enigma proprement dit) nous avons pris en entrer un message que nous avons au préalable transformer afin d'éviter des caractères avec accent et les blanc , message que nous avons ensuite concaténer puis nous l'avons transformer en majuscule ;ensuite nous avons envoyer le message dans les rotors (nous avons utilisé les rotors 1 ,2 ,3 sur les 5 disponible). A l'entrée des rotors nous avons effectué l'algorithme de césar pour permuter les lettres ensuite nous avons applique une rotation du rotor avec les lettres permutées, cette manipulation a été faite sur les 3 rotors ensuite nous avons passé le message dans un réflecteur pour un autre changement afin de renforcer le niveau de complexité de l'algorithme. Cette méthode s'applique à la fois pour le cryptage et le décryptage.

Diagramme de flux fonctionnel de la machine enigma



Interface de programmation

Application s	def Menu()	Ici est celle qui sera visible par l'utilisateur pour afin de faire son choix
Logique	def Egnima(rotors_file,rotors, messageIN)	Ici il est question de la fonction qui fait l'essentiel de la fonction de cryptage
	def caesarShift(str, key)	Partie responsable du premier codage a l'entrée de chaque rotor
	def RemoveAcents(text)	Permet de retirer les accents sur les lettres y disposant
Data	def SauvegardeMessage(file, message)	Permet de sauvegarder le message codé dans un fichier
	def LectureMessage(file)	Permet de lire le message contenu dans le fichier

Tests

Le code fonctionne effectivement pour les exemples que nous avons pris

Voici le test réalisé du message que nous avons codé

```
1 -- LectureMessage
2 -- EnregistreMessage
3 -- Cryptage et decryptage
4 -- Quitter
entrer votre choix ...3

 message IN :   ARRIVAGEDEBONBONSDANSLEMAGASINPOURLEPLAISIRDESSOIREES

 message OUT:
 CLLGSCIFMFYHXYHXVMCXVRFDCICVGXZHKLRFZRCGVGLMFVVHGLFFV
```

```
 message IN :   CLLGSCIFMFYHXYHXVMCXVRFDCICVGXZHKLRFZRCGVGLMFVVHGLFFV

 message OUT:
 ARRIVAGEDEBONBONSDANSLEMAGASINPOURLEPLAISIRDESSOIREES
1 -- LectureMessage
2 -- EnregistreMessage
3 -- Cryptage et decryptage
4 -- Quitter
entrer votre choix ...
```

Conclusion

Au terme de ce projet nous pouvons être satisfait du résultat obtenu, néanmoins il n'a pas été facile à réaliser. Nous avons fait face à plusieurs difficultés à savoir la compréhension du fonctionnement du programme enigma ensuite de sa réalisation, mais grâce à nos cours et la documentation que nous avons recherché, nous avons pu arriver à bout de ce projet

Bibliographie

https://docs.google.com/document/d/1-Y-UrBUSEIjIR530ESdlTxgbxcFdnDWU8FaSir3vGhQ/edit

http://mathweb.free.fr/crypto/debvingt/enigmafonc.php3#:~:text=La%20machine%20Enigma%20est%20aliment%C3%A9e,lettre%20cod%C3%A9e%20l'on%20substitue.

https://ingeniumcanada.org/fr/le-reseau/articles/le-fonctionnement-de-lenigma-revele

https://www.commentcamarche.net/contents/205-cryptographie-enigma

http://www.sciencesalecole.org/wp-content/uploads/2019/10/LYC70.pdf

https://www.youtube.com/watch?v=YuZA4spMjtU

https://www.youtube.com/watch?v=2dKG21u2aSo