



Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page [Préparer votre participation](#).

Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à info@trophees-nsi.fr.

NOM DU PROJET : ENIGMA

> PRÉSENTATION GÉNÉRALE :

- *Idee et objectifs*
- *Origines et intérêts du projet*
- (...)

Notre équipe est constituée de:

- Clara MONNIER, 1ère spé NSI, Lycée Simone Weil, Le Puy-En-Velay
- Ménina SAVY, 1ère spé NSI, Lycée Simone Weil, Le Puy-En-Velay

Lorsque notre professeur nous a informées de l'organisation des trophées NSI, nous avons tout de suite souhaité y participer et former une équipe toutes les deux.

Notre plus grande difficulté a été de sélectionner un sujet.

En effet, l'idée la plus évidente aurait été de faire un jeu vidéo, or n'ayant pas de grand intérêt pour cela, nous avons choisi d'axer notre projet sur les mathématiques, matière que nous adorons toutes les deux.

Notre idée initiale était de résoudre une énigme mathématique qui nous intriguait beaucoup. Sa résolution nécessitant un grand nombre de calculs, nous nous sommes penchées sur sa résolution via un programme informatique. La difficulté de l'énigme résidait dans la logique nécessaire à son déchiffrement et non dans son codage informatique. Nous nous sommes rendu compte que ce projet n'était pas suffisant. Nous avons alors choisi d'abandonner ce projet et d'en sélectionner un autre tout en restant sur le thème des mathématiques et des énigmes.

Clara a proposé de continuer un projet qu'elle avait déjà commencé dans le cadre de "contributions", qui sont des projets à faire et à présenter à l'oral dans le cadre de la spécialité NSI, à savoir coder la machine ENIGMA. Cette idée a beaucoup plu à Ménina.

Nous avons conscience de l'importance de la cryptographie dans la circulation des flux de données dans le monde et de maintenir la sécurité des informations personnelles des utilisateurs. Nous avons donc décidé de relever le défi de coder la machine ENIGMA. Par ailleurs, ce projet nous permettrait de réaliser notre rêve d'enfant de pouvoir communiquer entre nous sans que personne ne puisse nous comprendre. Au début, nous avions des doutes quant-à notre capacité de réaliser ce projet. La spécialité NSI à été pour nous deux davantage un choix stratégique pour notre orientation plutôt que pour quelque capacité en la matière. En effet, étant donné que nous voudrions être ingénieures ou chercheuses et savons que l'informatique est l'avenir. Nous avons débuté dans cette spécialités avec des bases de SNT de seconde mais ne sachant pas vraiment programmer. Le fossé entre notre niveau et celui des autres nous semblait insurmontable mais grâce à notre professeur et l'attention particulière que nous portons en cours, nous avons réussi à progresser.

Aussi, les trophées nous ont permis d'en apprendre encore plus sur le codage informatique et notamment sur la bibliothèque tkinter que l'on n'utilise pas en cours de NSI.

Nous sommes donc assez fières de notre progression en informatique et d'être parvenues à programmer ENIGMA.

> ORGANISATION DU TRAVAIL :

- *Présentation de l'équipe (prénom de chaque membre et rôle dans le projet)*
- *Répartition des tâches*
- *Organisation du travail (répartition par petits groupes, fréquence de réunions, travail en dehors de l'établissement scolaire, outils/logiciels utilisés pour la communication et le partage du code, etc.)*

Rôle de chaque membre :

- Clara : code
- Ménina : aide au code et côté administratif (création de la vidéo, du dossier technique...)

Nous nous entendons très bien , il a donc été facile de communiquer ensemble afin de s'organiser dans notre travail. Nous sommes d'ailleurs très complémentaires, ce qui a été très utile dans ce travail: Clara a pu avancer rapidement sur le projet et effectuer de nombreuses tentatives de codage de la machine ENIGMA. Ménina quant-à-elle a pu poser de nombreuses questions à Clara qui ont permis d'améliorer le code et de le rendre plus simple pour l'utilisateur. Elle a aussi fait des essais de code.

Tout au long du projet, nous nous sommes communiquées nos avancées. Nous sommes dans la même classe, donc dès que l'une de nous deux avait une question ou une idée nous en parlions pendant les interours.

Nous ne pouvons pas dire exactement combien de temps ce projet nous a pris étant donné que Clara l'avait déjà débuté en amont et que ce projet n'était, pour nous, pas une contrainte mais un divertissement sur lequel nous travaillions dès que nous en avions le temps et l'envie. Nous pouvons alors estimer notre durée de travail de 100 heures.

> LES ÉTAPES DU PROJET :

- *Présenter les différentes étapes du projet (de l'idée jusqu'à la finalisation du projet)*

Tout d'abord, Clara qui avait foncé tête baissée dans la programmation sans essayer de comprendre en détail le fonctionnement de la machine a vite été confrontée à des problèmes. Ménina a alors posé des questions à Clara sur son code ce qui nous a permis de nous rendre compte que nous avions encore des incertitudes sur le fonctionnement de la machine. Nous avons donc entrepris de nous documenter sur le fonctionnement en détail de la machine.

Lorsque nous avons compris le fonctionnement de la machine grâce à des simulations sur papier, nous nous sommes lancées dans la programmation. Nous nous sommes alors réparti le travail: Clara qui avait déjà commencé a continuer le codage de la machine et Ménina s'est occupée des parties principales du programme.

Une fois les tâches réparties, vint la programmation :

- Nous avons commencé par créer la « base du programme », pour cela nous nous sommes servi de la base créée par Clara et de Wikipédia, notamment pour le codage historique des rotors.

Pour cette étape, nous avons travaillé en commun, au CDI chacune sur notre ordinateur, et en communiquant en temps réel.

- Une fois la base du programme créée, nous avons chacune de notre côté créé les fonctions nécessaires à notre programme. Au début, nous avons décidé que l'une ferait le cryptage et

l'autre le décryptage, mais nous nous sommes en fait rendues compte que pour crypter ou décrypter un message grâce à ENIGMA, il n'y a besoin que d'un code commun. Clara, qui avait déjà bien avancé sur le codage a continué à créer les fonctions utiles au codage : "decalage", "passage_rotor", "passage_reflecteur", "passage_rotor_retour" et Ménina a repris le codage et a créé les autres fonctions : "choix_rotor", "tableau_de_fiche", "tableau_de_fiche_sortie".

Nous avons ensuite mis en commun nos fonctions et les avons améliorées.

- Une fois les fonctions créées, nous nous sommes rassemblées pour se répartir les tâches : la date du rendu des projets avançait et notre programme n'était toujours pas fonctionnel. Nous avons alors, tout un matin, travaillé ensemble à la conception de la fonction "code", puis Clara s'est occupée de finir le programme et Ménina a cherché la cause du dysfonctionnement de la fonction réflecteur grâce à l'aide de Clara puis a travaillé sur l'aspect administratif du projet, tout en surveillant ce que faisait Clara afin de la prévenir en cas d'erreur.

- Alors que la date du rendu des projets était très proche, notre programme ne marchait toujours pas, nous avons donc fait tester le programme à notre entourage, des personnes qui avaient des bases en informatique comme des personnes qui n'en avaient pas, pour qu'ils nous donnent leur avis. Cela nous a permis de nous rendre compte qu'il manquait une lettre à un de nos rotors et que par conséquent le programme ne pourrait jamais marcher.

- Une fois cette erreur identifiée, il ne nous a fallu plus qu'une petite dizaine d'heures pour terminer le projet: Ménina a alors rapidement continué le côté administratif et Clara a fini le code.

- Nous nous sommes alors retrouvées une dernière fois afin de se demander ce qu'il manquait à notre programme. C'est à ce moment-là que l'idée de rendre notre programme plus agréable pour l'utilisateur, grâce à la bibliothèque tkinter, nous est venue. Pendant que Ménina terminait le côté administratif, Clara s'est chargée de tkinter, ce qui n'était pas simple puisque c'était pour nous une découverte.

- Enfin, nous nous sommes amusées à tester les limites de notre programme afin de trouver une réponse à chacune d'entre elles. Nous avons aussi essayé d'optimiser notre code de façon à ce que l'on ait le moins de lignes de code possibles, ce qui est plutôt réussi puisque notre code est passé de plus de 500 lignes à moins de 200...

- Une fois le code et le côté administratif terminés nous avons filmé la vidéo de présentation que Ménina avait préparé au préalable.

Pour conclure la réalisation de ce programme n'a été possible que grâce à la grande persévérance de Clara face aux erreurs et ses nombreuses tentatives ainsi que l'entraide qui nous a permis de surmonter ces obstacles.

➤ FONCTIONNEMENT ET OPÉRATIONNALITÉ :

- *Avancement du projet (ce qui est terminé, en cours de réalisation, reste à faire)*
- *Approches mises en œuvre pour vérifier l'absence de bugs et s'assurer de la facilité d'utilisation du projet*
- *Difficultés rencontrées et solutions apportées*

Nous avons réalisé des tests de fonctionnement tout au long de notre programmation, à chaque fois que nous créions une nouvelle fonction ou une nouvelle branche au programme, nous la testions de

façon à ne pas avoir des milliers d'erreurs une fois le programme terminé. Nous avons à la fin testé notre programme et sommes allées chercher ses limites afin d'améliorer le programme.

Lors du travail individuel, nous faisons tester notre travail à l'autre pour qu'elle apporte des améliorations, des critiques...

Nous avons aussi fait utiliser le programme à nos parents, qui n'ont pas de connaissances en informatique. Cela nous a permis de nous détacher de notre regard de « scientifiques » et d'avoir un regard extérieur. Cela nous a permis d'approcher notre programme de façon plus détachée.

Notre programme est entièrement terminé et fonctionnel.

Nos difficultés principales ont été tout d'abord de comprendre le fonctionnement de la machine. Pour y faire face, nous avons fait des simulations sur papier, comme présenté dans la vidéo.

Nous avons aussi été confrontées à de nombreuses fautes de frappe qui ont bloqué notre code. L'erreur la plus embêtante a été que nous avons oublié une lettre dans un des rotors ce qui faisait que le programme renvoyait l'erreur « out of range ». Nous avons alors fait tester le programme à nos familles et amis, et l'un d'eux a trouvé notre erreur.

Nous voulions aussi faire un assert pour ne pas pouvoir utiliser 3 fois le même rotor mais en se renseignant nous avons appris qu'il était possible de le faire et avons donc abandonné l'idée.

> OUVERTURE :

- *Idées d'améliorations (nouvelles fonctionnalités)*
- *Stratégie de diffusion pour toucher un large public (faites preuve d'originalité !)*
- *Analyse critique du résultat (si c'était à refaire, que changeriez-vous dans votre organisation, les fonctionnalités du projet et les choix techniques ?)*

Nos idées d'améliorations sont :

- Il pourrait être intéressant d'insérer plus de paramètres à notre code, même si cela ne serait plus exactement ENIGMA.
- Nous pourrions améliorer notre interface graphique de façon à ce que l'utilisateur puisse comprendre le fonctionnement de la machine en voyant sa lettre se déplacer dans les rotors en temps réel.
- Nous pourrions aussi créer une possibilité au code de permettre à l'utilisateur d'en apprendre un peu plus sur notre code et la machine.
- Nous pourrions permettre à l'utilisateur de coder des chiffres, les signes de ponctuation et les caractères spéciaux. Nous pourrions même laisser à l'utilisateur le choix de la base dans laquelle il souhaite coder les chiffres.

Pour diffuser notre projet, nous allons déjà pouvoir le présenter lors des cordées de la réussite organisées entre notre lycée, le collège Jules Vallès et Clermont INP.

Nous aimerions bien, par la suite, faire que notre code soit accessible par tous, sur un site internet, pour que n'importe qui puisse coder ou décoder un message à l'aide de notre programme.

Il pourrait être amusant, que nous créions une application grâce à laquelle des personnes puissent communiquer, code un réseau social, et que leur messages soient codés par notre code, de façon à ce que les hackers ne puissent comprendre les messages envoyés.

DOCUMENTATION

- Spécifications fonctionnelles (guide d'utilisation, déroulé des étapes d'exécution, description des fonctionnalités et des paramètres)
- Spécifications techniques (architecture, langages et bibliothèques utilisés, matériel, choix techniques, format de stockage des données, etc)
- Illustrations, captures d'écran, etc

Le code a été entièrement conçu et écrit par Clara MONNIER et Menina SAVY à travers de nombreuses tentatives et améliorations.

Code source :

ENIGMA.py

Documents externes :

Logo ENIGMA.ico

Documents utilisés :

Pour comprendre le fonctionnement de la machine ENIGMA nous avons utilisé les sites suivants :

- [https://fr.wikipedia.org/wiki/Enigma_\(machine\)](https://fr.wikipedia.org/wiki/Enigma_(machine))
- <https://www.bibmath.net/crypto/index.php?action=affiche&quoi=debvingt/enigmaquerre>
- <https://www.bibmath.net/crypto/index.php?action=affiche&quoi=debvingt/enigmafonc>
- <https://ingeniumcanada.org/fr/le-reseau/articles/le-fonctionnement-de-lenigma-revele>
- <https://web.maths.unsw.edu.au/~lafaye/CCM/crypto/enigma.htm>
- [Enigma machine / Tom MacWright | Observable \(observablehq.com\)](https://observablehq.com/@tommacwright/enigma-machine)

Prérequis :

Posséder un logiciel de programmation Python.

Bibliothèques utilisées :

Tkinter

Protocoles d'utilisation :

- Ouvrir le document ENIGMA.py grâce à un logiciel de programmation Python.
- Lancer le programme en cliquant sur la flèche verte.
- Entrer dans l'interface graphique le message à coder (celui-ci peut être écrit en majuscule ou en minuscule mais ne doit pas comporter de chiffres, de caractères spéciaux, d'espaces ou de signes de ponctuation).
- Choisir 3 rotors parmi les 5 possibles qui seront utilisés pour coder le message. Les numéros attribués aux rotors sont respectivement : 1, 2, 3, 4, 5.
- Cliquer sur le bouton "CODER" situé en bas de l'interface graphique.

- Le message codé s'affiche ensuite sur l'interface graphique.
- Pour quitter, appuyer sur le bouton "Quitter".

Simulation :

Faisons une simulation du fonctionnement d'ENIGMA avec la lettre G. Nous choisissons les rotors 1, 2 et 3 pour cette simulation.

1) Passage dans le tableau de fiche :

Le F devient G.

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

2) Détermination de la place de la lettre dans l'alphabet :

G est à la place 6.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

3) Décalage du rotor :

Le K prend la position 0, le M la position 1, ..., le E la position 25.

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

4) Passage de la lettre dans le premier rotor :

G qui est à la place 6 devient Q qui est à la place 16.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
K	M	F	L	G	D	Q	V	Z	N	T	O	W	Y	H	X	U	S	P	A	I	B	R	C	J	E
10	12	5	11	6	3	16	2	2	1	1	1	2	2	7	2	2	1	1	0	8	1	1	2	9	4

5) Passage de la lettre dans le deuxième rotor :

Le Q qui est à la place 16 devient Q qui est à la place 16.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	J	D	K	S	I	R	U	X	B	L	H	W	T	M	C	Q	G	Z	N	P	Y	F	V	O	E
0	9	3	10	18	8	17	20	22	11	7	22	19	12	2	16	6	25	13	15	24	5	21	14	4	

6) Passage de la lettre dans le troisième rotor :

Le Q qui est à la place 16 devient I qui est à la place 8.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
B	D	F	H	J	L	C	P	R	T	X	V	Z	N	Y	E	I	W	G	A	K	M	U	S	Q	O
1	3	5	7	9	11	2	15	17	19	23	21	22	13	24	4	8	22	6	0	10	12	20	18	16	14

7) Passage de la lettre dans le réflecteur :

Le I est associé au P. Le I qui est à la place 8 devient P qui est à la place 8.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Y	R	U	H	Q	S	L	D	P	X	N	G	O	K	M	I	E	B	F	Z	C	W	V	J	A	T

8) Passage retour de la lettre dans le troisième rotor :

Le P prend la place 7 qui est la lettre H.

B	D	F	H	J	L	C	P	R	T	X	V	Z	N	Y	E	I	W	G	A	K	M	U	S	Q	O
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

9) Passage retour de la lettre dans le deuxième rotor :

Le H prend la place 11 qui est la lettre L.

A	J	D	K	S	I	R	U	X	B	L	H	W	T	M	C	Q	G	Z	N	P	Y	F	V	O	E
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

10) Passage retour de la lettre dans le premier rotor :

Le L prend la place 3 qui est la lettre D.

K	M	F	L	G	D	Q	V	Z	N	T	O	W	Y	H	X	U	S	P	A	I	B	R	C	J	E
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

11) Passage retour dans le tableau de fiche :

Le D devient T.

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

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

12) Codage du message :

Ici notre message est F il devient alors T.

Si on code T on obtient alors F.

Vérifions cela à l'aide de notre programme :



Et à l'inverse :



Pour illustrer cela nous allons refaire une simulation :

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

On peut remarquer une symétrie entre les deux lettres. En effet, une lettre donne une lettre qui redonne la lettre initiale. C'est pourquoi dans notre programme nous n'avons qu'une fonction code qui permet de crypter et décrypter un message.

Explication du code :

```
#appel des bibliothèques nécessaires à notre code:
from tkinter import*

#définition de l'alphabet:
alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

#définition des rotors:
rotor1 = 'EKMFLGDQVZNTOWYHXUSPAIBRCJ'
rotor2 = 'AJDKSIRUXBLHWTMCQGZNPYFVOE'
rotor3 = 'BDFHJLCPRTXVZNYEIWGAKMUSQO'
rotor4 = 'ESOVZPJAYQUIRHXNLFTGKDCMWB'
rotor5 = 'VZBRGITYUPSDNHLXAWMJQOFECK'

tab_fiche={ 'A': 'I', 'B': 'M', 'C': 'E', 'D': 'T', 'E': 'C', 'F': 'G',
            'G': 'F', 'H': 'R', 'I': 'A', 'J': 'Y', 'K': 'S', 'L': 'Q',
            'M': 'B', 'N': 'Z', 'O': 'X', 'P': 'W', 'Q': 'L', 'R': 'H',
            'S': 'K', 'T': 'D', 'U': 'V', 'V': 'U', 'W': 'P', 'X': 'O',
            'Y': 'J', 'Z': 'N'}

#définition du réflecteur:
reflecteur = {0: "Y", 1: "R", 2: "U", 3: "H", 4: "Q", 5: "S", 6: "L", 7: "D", 8: "P",
              9: "X", 10: "N", 11: "G", 12: "O", 13: "K", 14: "M", 15: "I", 16: "E",
              17: "B", 18: "F", 19: "Z", 20: "C", 21: "W", 22: "V", 23: "J", 24: "A",
              25: "T"}
```

Cette partie du programme permet d'appeler la bibliothèque tkinter qui sera utile à la conception de l'affichage graphique de notre code.

Nous créons les variables « alphabet », « rotor1 », « rotor2 », « rotor3 », « rotor4 », « rotor5 », « reflecteur » et « tab_fiche » qui nous seront utiles lors du codage.

```
#fenetre
fenetre = Tk()
fenetre.geometry("1000x700")
fenetre.title("Trophées NSI : ENIGMA")
fenetre.iconbitmap(" Logo ENIGMA.ico")
fenetre['bg'] = "#F5f5dc"
fenetre.resizable(height = False, width = False)
```

Cette partie nous permet de créer la fenêtre graphique grâce à l'utilisation de la bibliothèque tkinter appelée précédemment.

```

def decalage(rotor):
    lettre_enleve = rotor[0]
    rotor_apres_lettre_enleve = rotor.lstrip(rotor[0])
    rotor_apres_decalage = rotor_apres_lettre_enleve + lettre_enleve
    return rotor_apres_decalage

def passage_rotor(rotor, valeur):
    nouveau=rotor[valeur]
    nouveau=alphabet.index(nouveau)
    return nouveau

def passage_reflecteur(reflecteur, valeur):
    nouvaleur=reflecteur.get(valeur)
    return nouvaleur

def passage_rotor_retour(rotor, valeur):
    nouvelle_lettre = rotor.index(valeur)
    nouvelle_lettre = alphabet[nouvelle_lettre]
    return nouvelle_lettre

```

Ici, nous créons les fonctions « decalage », « passage_rotor », « passage_reflecteur » et « passage_rotor_retour ». Elles nous servent à décaler le rotor d'un cran vers la gauche et de faire modifier la lettre lors du passage dans les rotors et le réflecteur.

```

def tableau_de_fiche(lettre):
    return tab_fiche[lettre]

#fonction tableau_de_fiche_sortie qui
def tableau_de_fiche_sortie(lettre):
    for i in tab_fiche:
        if tab_fiche[i]==lettre:
            return i

```

Ces fonctions « tableau_de_fiche » et « tableau_de_fiche_sortie » nous permettent de modifier la lettre en passant par le tableau d'affichage.

```

def obtenir():
    message = ma_variable.get()

def rotor_1():
    rotors1 = ma_variable2.get()

def rotor_2():
    rotors2 = ma_variable3.get()

def rotor_3():
    rotors3 = ma_variable4.get()

```

Ces fonctions nous permettent d'obtenir les informations entrées dans les entrées de notre interface graphique.

```

def code():
    message = ma_variable.get()
    message = message.upper()
    message_code=''
    rotorss = [None, None, None]
    rotorss[0] = ma_variable2.get()
    if rotorss[0] == '1':
        rotorss[0] = rotor1
    elif rotorss[0] == '2':
        rotorss[0] = rotor2
    elif rotorss[0] == '3':
        rotorss[0] = rotor3
    elif rotorss[0] == '4':
        rotorss[0] = rotor4
    elif rotorss[0] == '5':
        rotorss[0] = rotor5

    rotorss[1] = ma_variable3.get()
    if rotorss[1] == '1':
        rotorss[1] = rotor1
    elif rotorss[1] == '2':
        rotorss[1] = rotor2
    elif rotorss[1] == '3':
        rotorss[1] = rotor3
    elif rotorss[1] == '4':
        rotorss[1] = rotor4
    elif rotorss[1] == '5':
        rotorss[1] = rotor5

    rotorss[2] = ma_variable4.get()
    if rotorss[2] == '1':
        rotorss[2] = rotor1
    elif rotorss[2] == '2':
        rotorss[2] = rotor2
    elif rotorss[2] == '3':
        rotorss[2] = rotor3
    elif rotorss[2] == '4':
        rotorss[2] = rotor4
    elif rotorss[2] == '5':
        rotorss[2] = rotor5

    indice_initial = 1
    for i in message:
        rotorss[0] = decalage(rotorss[0])
        if indice_initial % 26 == 0:
            rotorss[1] = decalage(rotorss[1])
            if (indice_initial // 26) % 26 == 0:
                rotorss[2] = decalage(rotorss[2])
        lettre = tableau_de_fiche(i)
        lettre=alphabet.index(lettre)
        for i in range(3):
            lettre =passage_rotor(rotorss[i], lettre)
        lettre = passage_reflecteur(reflecteur, lettre)
        for i in range(3):
            lettre = passage_rotor_retour(rotorss[2-i], lettre)
        lettre = tableau_de_fiche_sortie(lettre)
        message_code = message_code + lettre
        indice_initial = indice_initial + 1
    return message_code

```

La fonction code nous permet de crypter et décrypter un message.

```
def code():
    message = ma_variable.get()
    message = message.upper()
    message_code = ''
```

Ici, nous obtenons le message que l'utilisateur a entré que nous codons ensuite.

```
rotorss = [None, None, None]
rotorss[0] = ma_variable2.get()
if rotorss[0] == '1':
    rotorss[0] = rotor1
elif rotorss[0] == '2':
    rotorss[0] = rotor2
elif rotorss[0] == '3':
    rotorss[0] = rotor3
elif rotorss[0] == '4':
    rotorss[0] = rotor4
elif rotorss[0] == '5':
    rotorss[0] = rotor5

rotorss[1] = ma_variable3.get()
if rotorss[1] == '1':
    rotorss[1] = rotor1
elif rotorss[1] == '2':
    rotorss[1] = rotor2
elif rotorss[1] == '3':
    rotorss[1] = rotor3
elif rotorss[1] == '4':
    rotorss[1] = rotor4
elif rotorss[1] == '5':
    rotorss[1] = rotor5

rotorss[2] = ma_variable4.get()
if rotorss[2] == '1':
    rotorss[2] = rotor1
elif rotorss[2] == '2':
    rotorss[2] = rotor2
elif rotorss[2] == '3':
    rotorss[2] = rotor3
elif rotorss[2] == '4':
    rotorss[2] = rotor4
elif rotorss[2] == '5':
    rotorss[2] = rotor5
```

Cette partie nous permet de faire choisir 3 rotors parmi les 5 possibles à l'utilisateur et de les récupérer pour ensuite pouvoir coder le message à l'aide de ces 3 rotors.

```
indice_initial = 1
for i in message:
    rotorss[0] = decalage(rotorss[0])
    if indice_initial % 26 == 0:
        rotorss[1] = decalage(rotorss[1])
        if (indice_initial // 26) % 26 == 0:
            rotorss[2] = decalage(rotorss[2])
```

Ici, nous faisons décaler le rotor d'un cran vers la gauche à chaque nouvelle lettre.

```
lettre = tableau_de_fiche(i)
lettre = alphabet.index(lettre)
for i in range(3):
    lettre = passage_rotor(rotorss[i], lettre)
lettre = passage_reflecteur(reflecteur, lettre)
for i in range(3):
    lettre = passage_rotor_retour(rotorss[2-i], lettre)
lettre = tableau_de_fiche_sortie(lettre)
message_code = message_code + lettre
indice_initial = indice_initial + 1
return message_code
```

Ici, nous codons chaque lettre du message pour renvoyer le message codé (crypté ou décrypté).

```
def apparition():
    texte_message_code['text'] = code()

#création des boîtes, textes, boutons...
cadre = Frame(fenetre, bg = "#F5F5dc", bd = 1, relief = SUNKEN)
titre = Label(cadre, text = "ENIGMA", font = ("Courier", 56, "bold"), bg = "#F5F5dc", fg = 'black')
sous_titre = Label(cadre, text = "Ce projet a été réalisé par Clara MONNIER et Méline SAVY, élèves de 1ère spé NSI au lycée Simone WEIL au Puy-En-Velay.", font = ("Courier", 12), bg = "#F5F5dc", fg = 'black')
texte1 = Label(cadre, text = "Grâce à notre code vous allez pouvoir coder un message à l'aide de la machine ENIGMA", font = ("Courier", 12), bg = "#F5F5dc", fg = 'black')
texte_message = Label(cadre, text = "Veuillez entrer le message que vous souhaitez coder : (sans espace ni ponctuation)", font = ("Courier", 12), bg = "#F5F5dc", fg = 'black')
ma_variable = StringVar()
entree_message = Entry(cadre, textvariable = ma_variable, bg = "#F5F5dc", fg = 'black')
texte_rotors = Label(cadre, text = "Veuillez choisir 3 rotors qui serviront au codage de votre message :", font = ("Courier", 12), bg = "#F5F5dc", fg = 'black')
texte_rotor1 = Label(cadre, text = "Entrez le numéro du premier rotor, compris entre 1 et 5 :", font = ("Courier", 12), bg = "#F5F5dc", fg = 'black')
ma_variable2 = StringVar()
entree_rotor1 = Entry(cadre, textvariable = ma_variable2, bg = "#F5F5dc", fg = 'black')
texte_rotor2 = Label(cadre, text = "Entrez le numéro du deuxième rotor, compris entre 1 et 5 :", font = ("Courier", 12), bg = "#F5F5dc", fg = 'black')
ma_variable3 = StringVar()
entree_rotor2 = Entry(cadre, textvariable = ma_variable3, bg = "#F5F5dc", fg = 'black')
texte_rotor3 = Label(cadre, text = "Entrez le numéro du troisième rotor, compris entre 1 et 5 :", font = ("Courier", 12), bg = "#F5F5dc", fg = 'black')
ma_variable4 = StringVar()
entree_rotor3 = Entry(cadre, textvariable = ma_variable4, bg = "#F5F5dc", fg = 'black')
texte_code = Label(cadre, text = "Cliquez ici pour coder le message :", font = ("Courier", 12), bg = "#F5F5dc", fg = 'black')
texte_decode = Label(cadre, text = "Votre message codé est :", font = ("Courier", 12), bg = "#F5F5dc", fg = 'black')
message_code = code()
texte_message_code = Label(cadre, text = "", font = ("Courier", 15), bg = "#F5F5dc", fg = 'black')
bouton_coder = Button(cadre, text = "CODER", font = ("Arial", 15), bg = 'black', fg = "#F5F5dc", command = apparition)
bouton_quitter = Button(cadre, text = "Quitter", font = ("Courier", 12), bd = 1, bg = 'black', fg = "#F5F5dc", command = fenetre.destroy)

#execution
cadre.place(x = 55, y = 50)
titre.pack()
sous_titre.pack(pady = 5)
texte1.pack()
texte_message.pack(pady = 5)
entree_message.pack()
texte_rotors.pack(pady = 5)
texte_rotor1.pack()
entree_rotor1.pack(pady = 5)
texte_rotor2.pack()
entree_rotor2.pack(pady = 5)
texte_rotor3.pack()
entree_rotor3.pack(pady = 5)
texte_code.pack()
texte_decode.pack(pady = 5)
texte_message_code.pack()
bouton_coder.pack(pady = 5)
bouton_quitter.pack(pady = 5)

fenetre.mainloop()
```

Cette partie, nous permet de créer l'interface graphique à l'aide d'entrées, boutons, cadres... en tkinter.

Note finale :

Nous avons beaucoup aimé travailler sur ce projet, cela nous a permis de mieux comprendre la manière dont la cryptographie fonctionne et de confirmer notre désir de poursuivre dans le domaine scientifique. Nous nous sommes rendu compte que même si un projet semble impossible, en équipe, ce projet devient plus simple.

Nous avons aussi pu voir que notre programme pouvait avoir d'autres utilités que simplement communiquer à l'aide de messages codés. Par exemple, la petite sœur de Clara s'est rendue compte en s'amusant sur le code, que nous pouvions l'utiliser comme le jeu de mastermind. En effet, une personne code un message à l'aide de notre code puis laisse le message codé à l'autre personne. Cette personne doit donc s'amuser à trouver quels ont été les rotors sélectionnés pour obtenir le message codé initialement.