**PROJET SUR LES CRYPTAGES : MODE D’EMPLOI**

1. **STRUCTURE DES FICHIERS**

Pour utilisation :

cryptage /

… // reste des fichiers accompagnant l’exécutable

output/ // contiendra les fichiers.txt des métadonnées

cryptage.exe // fichier exécutable

icone.ico // icône du logiciel

…

Pour développement :

src/

modules/

algorithmes/

\_\_init\_\_.py

cesar.py

Ces fichiers d’extension “.py” contiennent les fonctions de codage et décodage selon la méthode du nom du fichier. Ils sont importés et utilisés dans le fichier principal pour réaliser les cryptages et les montrer sur l’interface, mais ils peuvent être utilisés séparément.

polybe.py

rot13.py

vigenere.py

interface/

\_\_init\_\_.py

interface.py

DE-LA-CUEVA\_Daniel\_code.py // fichier principal, en l’exécutant l’interface apparaît

1. **PRÉREQUIS**

Langage :

* Python 3, doit être installé si on veut exécuter le fichier « .py ».

Version utilisée : Python 3.10

Bibliothèques tierces utilisées :

* Tkinter (pip install tk) : à installer si on exécute le code depuis le fichier « .py », non nécessaire si on exécute le fichier « .exe ».
* PyInstaller : utilisée pour produire le fichier « .exe », elle n’est pas nécessaire au fonctionnement du code.

1. **FONCTIONS DE CODAGE**

Les fonctions utilisées au sein du projet pour réaliser l’action de crypter et décrypter en soi sont contenues dans des fichiers indiquant le nom de l’algorithme utilisé. Chaque fichier contient une fonction «\_c » (codage) et une fonction « \_d » (décodage).

Exemple : Le fichier « cesar.py » contient les fonctions « cesar\_c » et cesar\_d ».

Les différentes fonctions prennent différents arguments, mais retournent toutes un dictionnaire à forme similaire :

|  |  |  |
| --- | --- | --- |
| FONCTION | ARGUMENTS | RETOURNE |
| cesar\_c | * message\_clair (str) * decalage (int), 3 par défaut | Un dictionnaire contenant :   * méthode * alphabet utilisé * chiffres utilisés * clé (si elle existe) * message clair * message codé   Types de données possibles :   * str * int * NoneType |
| cesar\_d | * message\_code (str) * decalage (int), 3 par défaut |
| rot13\_c | * message\_clair (str) |
| rot13\_d | * message\_code (str) |
| vigenere\_c | * message\_clair (str) * cle\_orig (str), clé de chiffrement * alphabet (str), alphabet latin par défaut |
| vigenere\_d | * message\_code (str) * cle\_orig (str), clé de chiffrement * alphabet (str), alphabet latin par défaut |
| polybe\_c | * message\_clair (str) * alphabet (str), alphabet latin par défaut |
| polybe\_d | * message\_code (str) * alphabet (str), alphabet latin par défaut |

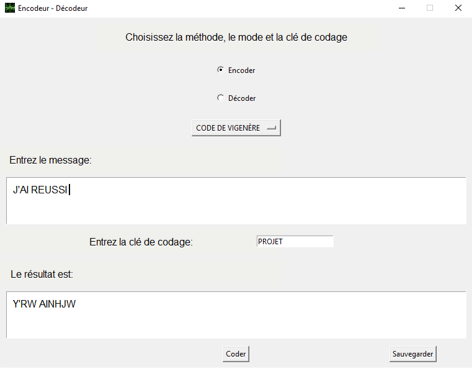
Si on exécute les fichiers contenant les fonctions, un message nous indiquera le fichier que nous avons choisi mais il n’y aura possibilité de faire aucune opération. Il faut donc importer ces modules pour les utiliser.

Note : Dû aux restrictions de la programmation à notre niveau, et pour éviter de compliquer les fonctions excessivement, certaines règles sont appliquées :

* Le message traité sera toujours renvoyé en lettres majuscules, quelle que soit la case à l’entrée
* L’alphabet par défaut est l’alphabet latin et les chiffres de 0 à 9, il ne peut être changé sur l’interface mais l’utilisateur plus avancé peut utiliser les fonctions indépendamment et utiliser un autre alphabet.
* Les autres caractères (espaces, virgules, lettres accentuées) ne sont pas codés, ils conservent leur position dans le message.
* Les chiffres ne sont pas codés avec Vigenère, ils restent tels qu’ils sont. La clé de codage ne peut contenir des nombres sous cette méthode également.
* Les chiffres ne peuvent pas êtres codés avec Polybe, car ils seraient confondus avec le message chiffré.
* L’alphabet doit comporter 25 caractères (5x5) pour le chiffre de Polybe.

1. **INTERFACE**

Voici l’interface graphique du logiciel :

****

BOUTON DE SAUVEGARDE:

Un fichier avec la date et les données de l’opération sera créé dans le directoire /output

BOUTON DE CODAGE

MÉSSAGE TRAITÉ

INTRODUCTION DE LA CLÉ

INTRODUCTION DU MESSAGE

CHOIX DE LA MÉTHODE

CHOIX DU MODE

Plus de détails : <https://github.com/DanieldelaCueva/NSI_FILES_2122>