

COMPTE-RENDU DE PROJET ET ANALYSE

La cryptographie est, de nos jours, une des disciplines essentielles en informatique. C'est pour ceci que j'ai choisi le projet de cryptages, qui est assez simple mais il m'a semblé que c'était une bonne opportunité pour m'introduire à la discipline, ainsi que pour apprendre sur certains algorithmes et méthodes de cryptage. J'ai ainsi décidé d'intégrer un système de cryptage de messages complexe techniquement dans une interface simple pour l'utilisateur, afin également de mettre ce que j'ai appris au service d'un public moins avancé.

Après avoir défini techniquement le projet, nous verrons ce qui s'est passé pendant la première phase du projet, pour aboutir aux résultats et à l'analyse de ceux-ci.

La première tâche à réaliser était de définir exactement à quoi devait ressembler mon projet. Il m'était proposé de coder le cryptage de messages selon quatre méthodes de codage (ROT13, code de César, chiffre de Vigenère, carré de Polybe), et dans un niveau plus avancé, proposer des fonctions pour décoder des messages selon ces mêmes méthodes. L'idée m'a semblé intéressante, mais j'ai décidé d'ajouter une interface graphique qui ferait le pont entre l'utilisateur classique, sur son ordinateur, et mes fonctions de codage en Python, plus difficiles à utiliser, et qui surtout devaient être exécutées en console. Ce projet était plus compliqué mais plus original également, et a été approuvé lors de la première remise intermédiaire.

J'ai codé les fonctions de cryptage en Python, sans utiliser de bibliothèques tierces, sur l'éditeur VS Code et avec Python 3.10. J'ai choisi cet éditeur car c'est celui que j'utilise habituellement, et cette version de Python car c'est la dernière disponible. Pour l'interface graphique, j'ai utilisé la bibliothèque Python tkinter, qui allait faciliter énormément le développement de l'interface. Afin de tirer le plus grand parti de cette bibliothèque et de produire un code propre, j'ai dû introduire l'utilisation d'un concept de programmation plus avancé : la programmation orientée objet. Certes, ce paradigme de programmation n'est pas expliqué en première, mais j'en avais des notions que j'ai pu reprendre (cf. sources d'apprentissage, en fin de document), et son utilisation allait améliorer la lisibilité et l'organisation du code et des modules. L'organisation en modules est aussi d'un niveau un peu plus élevé mais ceci était nécessaire car un seul fichier avec tout le code aurait été illisible. Enfin, le fichier .exe a été facile à créer grâce à la bibliothèque PyInstaller, et avoir un fichier exécutable facilite l'utilisation du logiciel.



Il fallait de même établir des critères d'évaluation pour tester le projet. Puisqu'il s'agit de codages et décodages de messages, un des critères les plus efficaces est que les résultats produits par mon projet et ceux produits par les encodeurs de sites comme dcode.fr donnent des résultats identiques. Le choix du jeu de tests est également important, et j'ai choisi de tester des phrases complexes avec des caractères spéciaux, comme « J'avais l'activité de Français

(celle d'hier) », par exemple. Il fallait aussi vérifier que le décodage d'un message codé donne le message originel, et que l'interface s'affiche correctement et soit intuitive et utilisable même sans mode d'emploi, ainsi que le fonctionnement du fichier exécutable sur plusieurs ordinateurs qui n'ont pas forcément Python installé. Une fois que tout ceci fut défini, il fallait passer à la réalisation du projet.

Ce que je veux évaluer par des tests	Comment je prévois de le faire (critères d'évaluation)
<p>Par objectifs :</p> <ol style="list-style-type: none"><ol style="list-style-type: none">Le codage est identique à celui proposé sur dcode.frDes phrases avec des caractères spéciaux sont correctement encodées, le programme ne produit pas d'erreurs<ol style="list-style-type: none">Le décodage est identique à celui proposé sur dcode.frLe décodage des messages codés donne le message original, ou un message différent si la clé ou le décalage varie<ol style="list-style-type: none">L'interface apparaît sur l'écran, tous les éléments sont à leur place, même sur différents écrans avec différentes résolutions, le texte qui doit changer selon les choix de l'utilisateur le fait sans erreurDes utilisateurs témoin ne connaissant pas le fonctionnement parviennent à utiliser le logicielLes données sont sauvegardées sur des fichiers .txt dans le repertoire /output. Les données sauvegardées correspondent à cette opération.Le fichier exécutable fonctionne sous Windows sur différentes machines, même celles n'ayant pas Python installé	<p>Par objectifs :</p> <ol style="list-style-type: none"><ol style="list-style-type: none">Tester les mêmes codages sur dcode et sur mon logiciel et les comparerTester des phrases avec des caractères spéciaux, comme « J'avais l'activité de Français (celle d'hier) »<ol style="list-style-type: none">Tester les mêmes décodages sur dcode et sur mon logiciel et les comparerCréer un script python qui génère des messages aléatoires et les code et décode avec mes fonctions, retourne le taux de réussite<ol style="list-style-type: none">Lancer l'interface puis changer les réglages d'affichage du systèmeDonner le logiciel pour test à mon frèreRéaliser différentes sauvegardes et vérifier qu'elles soient correctesLancer le logiciel sur différents ordinateurs et vérifier qu'il fonctionne. Certains de ses ordinateurs doivent ne pas avoir Python installé

La première phase du projet, qui s'est étendue sur les premières semaines de réalisation, a marqué le rythme de travail et le devenir du projet. Je me suis mis en route dès la première remise intermédiaire, travaillant sur le projet 1 à 2 heures par semaine. C'est ainsi que j'ai codé les fonctions d'encodage et de décodage selon les quatre méthodes en trois semaines. Il m'a bien sûr fallu un peu plus de temps pour réaliser les premiers tests, mais j'avais déjà une certaine avance. J'ai alors abordé le développement de l'interface graphique, et j'ai dû prendre un main un nouvel outil : la bibliothèque tkinter. S'approprier de cet outil n'a pas été un problème au niveau de la compréhension mais m'a pris un certain temps, quoique l'avance que j'avais pris a permis de ne pas prendre du retard sur l'échéancier final.

L'ensemble du projet s'est déroulé sans trop de problèmes, mais déterminer les étapes à suivre pour encoder et décoder les messages s'est présenté comme la première difficulté. En effet, les algorithmes peuvent être un peu abstraits, et c'est en les schématisant que j'ai réussi à bien définir comment ils devaient fonctionner. Un autre problème que j'ai rencontré est le fait qu'il faut bien s'assurer que les paramètres indiqués aux fonctions soient du type désiré. La solution que j'ai trouvée pour ce problème est de tester que les paramètres conviennent au début de la fonction et de générer une erreur, qui sera affichée sur l'interface, si un problème survient. La solution à ce problème a eu une très grande importance dans mon projet car j'ai réalisé que je devais faire ces vérifications pour toutes mes fonctions au long de mon code. Le reste des problèmes rencontrés étaient minimaux je les ai résolus rapidement.

Enfin, le projet terminé comporte deux éléments principaux : l'interface, disponible pour l'utilisation d'un utilisateur avec des connaissances basique et que l'on peut lancer à travers le fichier .exe mais aussi en exécutant le fichier .py principal ; et les fonctions de codage, qui peuvent être importées et utilisées depuis un autre script en important le module « algorithmes ». L'interface est intuitive et permet de choisir si on veut encoder ou décoder le message, choisir la méthode, éventuellement une clé, et d'introduire le message. Ensuite, un bouton permet de coder le message et un autre de sauvegarder les métadonnées de l'opération dans un fichier texte (cf. mode d'emploi). Les fonctions de codage sont organisées par fichiers selon la méthode, et prennent des paramètres pour retourner un dictionnaire avec les résultats de l'opération.

The screenshot shows a Windows application window titled 'Encodeur - Décodeur'. The interface is in French and has a light gray background. At the top, it says 'Choisissez la méthode, le mode et la clé de codage'. Below this, there are two radio buttons: 'Encoder' (selected) and 'Décodeur'. Underneath the radio buttons is a dropdown menu currently showing 'CODE DE VIGENÈRE'. Below the dropdown is a text input field labeled 'Entrez le message:' containing the text 'J'AI REUSSI'. Below that is another text input field labeled 'Entrez la clé de codage:' containing the text 'PROJET'. Below the key field is a label 'Le résultat est:' followed by a text output field containing 'Y'RW AINHJW'. At the bottom of the window, there are two buttons: 'Coder' and 'Sauvegarder'.

Je suis très satisfait avec le résultat atteint dans ce projet. En effet, tous les tests ont réussi et les critères ont tous été remplis (cf. carnet de bord). Particulièrement, le fait d'avoir réussi à présenter les codages sur une interface facile à utiliser et sans erreurs est très satisfaisant. Les fonctions de cryptage encodent et décodent les messages selon les règles désirées, l'interface les présente correctement, sauvegarde bien les données quand ceci est demandé. Il y a quand même quelques aspects que je voudrais améliorer si j'avais plus de temps. L'un d'eux est l'aspect de l'interface, qui est simple et intuitive mais pourrait être plus esthétique ; et l'autre est d'améliorer les adressages relatifs dans mon code pour les importations et les sauvegardes, de façon à les rendre plus lisibles et faciles à comprendre.

Personnellement, j'ai bien aimé ce projet. Je suis passionné par la programmation et j'ai été très content de passer des heures de cours à développer cet outil, qui m'a permis de m'améliorer moi aussi. D'ailleurs, outre les connaissances techniques que ce projet m'a apporté telles que la connaissance d'algorithmes de cryptage, reprendre la POO ou l'organisation de modules ; j'ai acquis des connaissances méthodologiques par rapport à l'organisation, la tenue d'un carnet de bord, et la production de documentation qui sont essentielles pour un projet de longue durée.

En conclusion, cette expérience a été très enrichissante en tant que complément de la formation en cours de nsi, car elle m'a permis de m'améliorer comme programmeur et d'appliquer tous les concepts vus en cours. J'ai réalisé un projet ambitieux mais réaliste, ce qui a fait que je maintienne la motivation jusqu'au bout. L'ouverture vers des techniques plus avancées a été également très bénéfique car elle me dirige vers de nouveaux objectifs et de plus grandes perspectives, en apprenant un peu plus à chaque projet, personnel ou scolaire, que je réalise.

Bibliographie / Sitographie / Logithèque

Sources d'inspiration:

- [Dcode.fr](https://www.dcode.fr)

Tutoriels et cours pour la prise en main d'outils et apprentissage de concepts avancés de Python, POO notamment :

- "2022 Complete Python Bootcamp From Zero to Hero in Python", sur [udemy.com](https://www.udemy.com)
- stackoverflow.com
- [geeksforgeeks.org](https://www.geeksforgeeks.org)

Documentations officielles consultées :

- Python (docs.python.org)
- Tkinter (tkdocs.com)
- PyInstaller (pyinstaller.readthedocs.io)

Logiciels utilisés :

- Python 3.10
- VS Code