

Rapport de projet de NSI 2021

Théo ASSELIN

**CRYPTHOR**

Application de cryptage/décryptage de texte

# Sommaire

## Sommaire :

- Présentation du projet : page 2 / 15
- Planning : page 3 / 15
- Cahier des charges : page 4 / 15
- Conception préliminaire : page 5 / 15
- Conception détaillée : page 6 / 15
- Réalisation : page 7 / 15
- Tests : page 12 / 15
- Problèmes rencontrés : page 14 / 15
- Conclusion : page 15 / 15

# Présentation de projet

## Contexte :

À la fin de l'année de terminale dans l'enseignement de spécialité de Numérique et Sciences Informatiques chaque élève doit définir un projet à réaliser portant sur un thème informatique. Puis chaque élève fera la présentation de son projet devant la classe et le professeur.

## Projet :

Réaliser une application pour Smartphone permettant le cryptage puis le décryptage de texte.

## Motivations :

J'ai décidé de réaliser ce projet principalement parce que j'aimerais devenir ingénieur informatique en cybersécurité, mon projet y est donc étroitement lié par l'aspect cryptage et décryptage car je vais donc devoir concevoir une méthode de cryptage et de décryptage de texte.

Mon projet initial était de faire seulement un algorithme de cryptage décryptage en python et de m'arrêter là, mais ça ne me plaisait pas, je trouvais cela insuffisant et incomplet. J'ai donc décidé d'en faire une application car l'idée de pouvoir crypter et décrypter du texte directement avec mon smartphone me plaisait et que cela me faisait une excellente initiation à la création d'application ainsi que ce que cela pouvait être d'être un professionnel informatique devant répondre à une demande de client.

## Utilités du projet :

Peut servir à l'échange de message de manière sécurisée par messages cryptés ou encore à la protection de données confidentielles par le cryptage, pour un site internet par exemple, afin de protéger les informations des utilisateurs.

# Planning

Date de début de projet : 11 mars 2021 :

- analyse du besoin
- définition du planning
- création du cahier des charges

12 mars :

- conception préliminaire

20 mars :

- conception détaillée

5 mai :

- réalisation
- tests
- améliorations

15 mai :

- réalisation compte rendu
- réalisation présentation

Date de fin de projet : 19 mai :

- présentation du projet

# Cahier des charges

## Forme du projet :

Le projet devra être une application pour Smartphone fonctionnant sous système d'exploitation Android.

## Objectifs de l'application :

L'objectif de cette application est de pouvoir y entrer un texte pour le crypter, puis de pouvoir décrypter ce même texte crypté avec l'application.

## Design de l'application :

L'application devra avoir un design assez sobre, classe et moderne sans fioritures, tout cela sans pour autant tomber dans trop de simplisme qui rendrait le visuel de l'application vide. Elle devra contenir une page d'accueil sur laquelle se trouvera le logo de l'application ainsi que son nom, une page de cryptage et une page de décryptage, la navigation entre ces pages devra s'effectuer par le biais d'une barre de navigation à trois boutons.

## Méthode de cryptage / décryptage :

Les méthodes de cryptage et de décryptage doivent être créées et non tirées d'internet, la méthode de cryptage ne doit pas être simpliste comme un algorithme de César mais ne doit pas non plus être extrêmement difficile et complexe. De plus une maquette de ces algorithmes de cryptage et décryptage devra être réalisée en python.

## Utilisation d'outils :

Différents outils nécessaires à la réalisation du projet devront être utilisés, ces outils sont tout d'abord des logiciels, Gimp pour la réalisation du logo, Jupyter pour la réalisation d'un notebook contenant la maquette python des algorithmes de cryptage et décryptage, et Android Studio ou Kivy ou AppInventor pour la réalisation de l'application. Enfin des langages de codages seront nécessaires, Python pour la maquette des algorithmes, Kotlin pour le fonctionnement de l'application, ainsi qu'un métalangage informatique de balisage générique, du xml.

# Conception préliminaire

## Réflexion sur les méthodes de cryptage et de décryptage :

Le but étant de concevoir une méthode de cryptage propre à l'application et qui sera uniquement utilisée par l'application, j'ai décidé de partir sur une méthode de cryptage par échantillons sélectionnés au hasard contenus dans une base de données intégrée à l'algorithme, le tout mis en place à l'aide de comparaison. De manière plus concrète, une base de donnée sera définie dans l'algorithme de cryptage, pour chaque lettre de l'alphabet ainsi que pour l'espace, une grande liste est attribuée, chacune de ces grandes listes contiendra 5 petites listes, qu'on appellera échantillon, contenant chacune 4 lettres. Chaque échantillon de la base de données est unique. Lorsque l'on donne un texte à l'algorithme il prend le premier caractère, l'identifie, puis va chercher la grande liste d'échantillons qui lui est attribué, puis renvoie au hasard un des échantillons, ensuite il passe au caractère suivant du texte donné et ainsi de suite jusqu'à avoir traité tout le texte. À noter que cet algorithme n'accepte pas les caractères spéciaux, les majuscules et les chiffres. L'algorithme de cryptage fonctionne de la même manière mais dans le sens inverse, il connaît chaque échantillon et sait à quel caractère il correspond. Il vient alors chercher dans le texte crypté qu'on lui donne les caractères par groupes de 4 puis les compare à sa base de données pour trouver le caractère décrypté puis le renvoi.

## Choix du logiciel pour la création de l'application :

J'ai dû choisir entre trois logiciels disponibles pour créer mon application lequel serait le plus propice au bon aboutissement de mon projet. Mon choix retiendra finalement Android Studio pour diverses raisons. Tout d'abord c'est un logiciel conçu et détenu par Google, tout comme le système d'exploitation Android, ce qui fait que c'est un outils parfaitement conçu et adapté pour le développement d'applications sous Android. De plus c'est un logiciel qui ne pose aucune limite, on est libre de créer absolument ce que l'on souhaite car il n'y a aucune limitation technique. Enfin, c'est un logiciel très intuitif, bien construit et bien organisé qui de plus propose des outils très pratiques et utiles.

## Réflexion sur le design de l'application :

J'ai ensuite réfléchi plus précisément sur des détails du design de l'application. En premier les couleurs, j'ai donc décidé d'utiliser 4 couleurs, du noir, deux teintes de gris assez foncées, et du blanc. Ensuite il fallait que je trouve un nom pour mon application, j'ai décidé de l'appeler Crypthor, crypt en référence au cryptage et Thor en référence au héros Marvel pour inspirer la puissance et l'efficacité de l'application. Puis je devais imaginer un logo, j'ai donc décidé de faire un logo très épuré et simple étant un mélange entre le marteau de Thor et un clé ancienne. Enfin il fallait que je trouve une police d'écriture adaptée pour écrire le nom de l'application sur la page d'accueil, j'ai décidé qu'il me fallait une police qui soit classe moderne et design.

# Conception détaillée

Réalisation de la maquette des algorithmes en python :

Voir sur Github : <https://github.com/TheoAsselin/algorithme-maquette-de-cryptage-d-cryptage-en-python>

Découverte du logiciel Android Studio et des langages xml et kotlin :

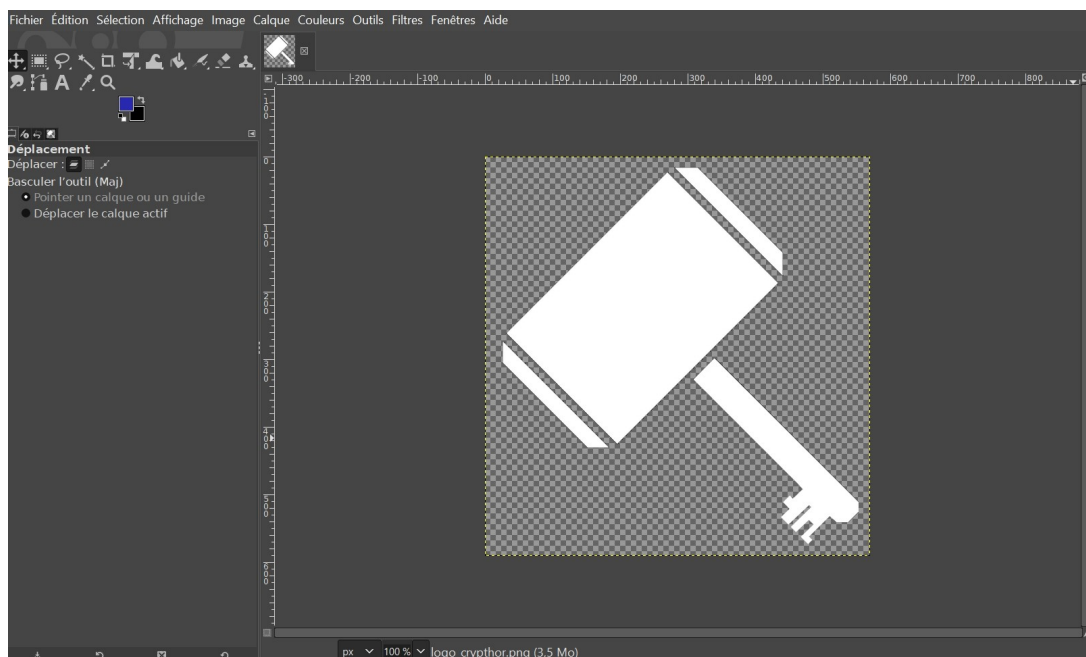
Je découvre pour la première fois ces outils, j'ai donc dû faire quelques recherches et regarder quelques tutoriels afin de pouvoir m'en servir. À noter que le langage xml n'est pas considéré comme langage informatique mais comme langage de balisage, il ne sert qu'à afficher des choses et ne peut faire aucune action. Également le langage Kotlin lui est un langage informatique, il a d'ailleurs été créé par Google en 2017 et est destiné à remplacer le langage Java, c'est un langage principalement orienté objet mais alliant tout de même une partie fonctionnelle.

# Réalisation

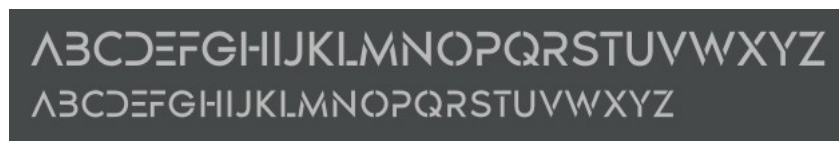
## Création des ressources :

Pour concevoir cette application différentes ressources sont nécessaires, le logo, la police d'écriture, les codes couleurs ainsi que les icônes (pour les boutons de la barre de navigation).

Pour le logo j'ai utilisé le logiciel Gimp qui est un logiciel de traitement d'image gratuit très connu et performant. À préciser que j'ai créé mon logo de A à Z, je ne l'ai pas pris sur internet. J'ai donc créé un arrière plan de type png sur lequel j'ai appliqué différentes formes afin de créer mon logo afin d'obtenir uniquement le logo sans aucun arrière plan.



Je suis ensuite allé chercher un pack de police d'écriture correspondant à mes critères sur internet, j'ai donc trouvé une police nommée Designer se présentant comme ceci :



Puis, également sur internet j'ai récupéré les codes en hexadécimal des couleurs dont j'avais besoin :

- noir : FF000 000
- gris foncé : FF292 929
- gris plus clair : FF404 040
- blanc : FFFFFFFF

Enfin pour les différentes icônes elles se trouvent directement dans le logiciel, j'ai donc choisi une icône de maison, une de cadenas ouvert et une de cadenas fermé.



## Implémentation des ressources :

Pour les intégrer dans Android Studio il suffit de les glisser dans le logiciel et de leur donner un identifiant qui permettra d'y accéder et de les utiliser.

## Création de la partie visuelle :

La partie visuelle d'une application est composée de layouts, un layout est une page que l'on vient afficher par dessus une base qui sert de fondation à l'application. Il faut donc créer 3 layouts, une page d'accueil où on affiche le logo et le nom de l'application :



Dont voici le code sur Github :

[https://github.com/TheoAsselin/Crypthor/blob/main/src/main/res/layout/fragment\\_home.xml](https://github.com/TheoAsselin/Crypthor/blob/main/src/main/res/layout/fragment_home.xml)

Une page de cryptage avec une entrée de texte pour donner un texte à crypter à l'application, un bouton crypter pour activer le cryptage et enfin une sortie de texte pour afficher le résultat :



Dont voici le code sur Github :

[https://github.com/TheoAsselin/Crypthor/blob/main/src/main/res/layout/fragment\\_cryptage.xml](https://github.com/TheoAsselin/Crypthor/blob/main/src/main/res/layout/fragment_cryptage.xml)

Et une page de décryptage avec une entrée de texte pour donner un message à décrypter à l'application, un bouton décrypter pour activer le décryptage et enfin une sortie de texte pour afficher le résultat :



Dont voici le code sur Github :

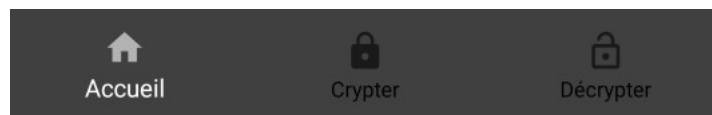
[https://github.com/TheoAsselin/Crypthor/blob/main/src/main/res/layout/fragment\\_decryptage.xml](https://github.com/TheoAsselin/Crypthor/blob/main/src/main/res/layout/fragment_decryptage.xml)

Il faut aussi créer la barre de navigation, la définir et définir où elle doit s'afficher à savoir sur la fondation de l'application pour qu'elle soit tout le temps affichée et ne bouge pas, son code sur Github : (troisième et dernier paragraphe du code)

[https://github.com/TheoAsselin/Crypthor/blob/main/src/main/res/layout/activity\\_main.xml](https://github.com/TheoAsselin/Crypthor/blob/main/src/main/res/layout/activity_main.xml)

Puis il faut définir les différents éléments qui vont s'y trouver, les trois boutons vers les différentes pages de l'application ainsi que leur légende, son code sur Github :

[https://github.com/TheoAsselin/Crypthor/blob/main/src/main/res/menu/navigation\\_bar.xml](https://github.com/TheoAsselin/Crypthor/blob/main/src/main/res/menu/navigation_bar.xml)



Mise en fonctionnement des différents layouts et de la barre de navigation :

Il faut d'abord définir pour chaque layout sa fonctionnalité, un conteneur d'informations, ce qui permettra également plus tard de les afficher. Pour cela on attribue à

chaque layout un « Kotlin class file », qui est tout simplement l'endroit où on écrit le code en Kotlin des layouts.

Code Kotlin de la page d'accueil sur Github :

<https://github.com/TheoAsselin/Crypthor/blob/main/src/main/java/fr/samay/crypthor/fragments/HomeFragment.kt>

Code Kotlin de la page de cryptage sur Github :

<https://github.com/TheoAsselin/Crypthor/blob/main/src/main/java/fr/samay/crypthor/fragments/CryptageFragment.kt>

Code Kotlin de la page de décryptage sur Github :

<https://github.com/TheoAsselin/Crypthor/blob/main/src/main/java/fr/samay/crypthor/fragments/DecryptageFragment.kt>

Il faut ensuite s'occuper de d'indiquer le rôle et le fonctionnement de la fondation de l'application. On lui indique qu'elle sert d'afficheur et on définit qu'elle page doit s'ouvrir lors du démarrage de l'application, la page d'accueil, puis on y implémente la barre de navigation ainsi que son fonctionnement, lorsque tel ou tel bouton est sélectionné l'application doit afficher la page qui correspond.

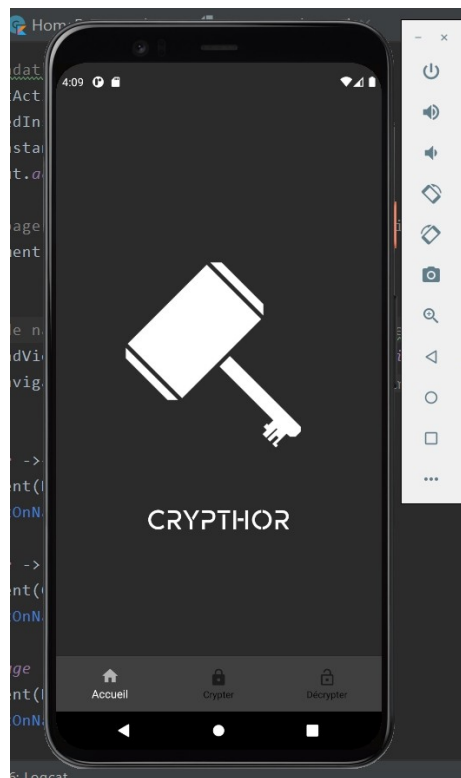
Son code sur Github :

<https://github.com/TheoAsselin/Crypthor/blob/main/src/main/java/fr/samay/crypthor/MainActivity.kt>

# Tests

Comment sont fait les tests :

C'est l'un des avantages du logiciel Android Studio, il est très pratique et efficace pour se rendre compte de l'avancée de ses travaux et de leur bon fonctionnement au fur et à mesure que l'on travaille. En effet il permet, même si l'application n'est pas finie, de la lancer pour voir si tout fonctionne correctement et correspond à ce que l'on souhaite. Il y a pour cela deux manières de faire, la première est d'utiliser un émulateur, directement intégré au logiciel, qui va tout simplement créer un smartphone virtuel fonctionnant comme un vrai sur lequel l'application va se lancer. À noter que l'émulateur est assez lourd et demande donc à l'ordinateur d'avoir une certaine puissance.



L'autre moyen de tester son application est de directement lancer l'application sur son smartphone réel en le connectant avec un câble à l'ordinateur pendant que Android Studio est activé puis de lancer la simulation en sélectionnant son smartphone à la place de l'émulateur, l'avantage de ce système est qu'il demande peu de ressources à l'ordinateur.

Quand sont fait les tests :

Des tests sont fait régulièrement, après l'implémentation des différents layouts, de la barre de navigation ainsi que des différentes actions afin de vérifier que tout fonctionne.

Qu'est-ce que l'on vérifie :

On vérifie que les bonnes couleurs et les bons objets (boutons, entrées de texte, barre de navigation, logo, textes) sont aux bons endroits. On vérifie le bon fonctionnement de chaque objet et si les bonnes pages s'affichent aux bons moments. Et enfin on vérifie que lorsque l'application se lance elle ouvre bien par défaut la page d'accueil et que la console du logiciel Android Studio n'affiche pas d'erreurs.

# Problèmes rencontrés

Lors de la création de la page d'accueil :

- Mauvais type de layout créé empêchant la mise en forme de la page d'accueil
- Erreur dans le code Kotlin empêchant son affichage
- Logo importé avec le mauvais format l'affichant complètement flou
- Police d'écriture importée avec une erreur dans son identifiant empêchant son utilisation pour afficher le nom de l'application

Lors de la création de la page de cryptage :

- Mauvaise implémentation de l'entrée de texte empêchant de voir le texte écrit
- Mauvaise couleur utilisée pour le bouton de cryptage

Lors de la création de la page de décryptage :

- Mauvaise implémentation des contraintes de placement des différents objets rendant leur placements incorrects

Lors de la création de la barre de navigation :

- Mauvaise implémentation de la barre de navigation sur la fondation de l'application empêchant son affichage
- Boutons affichés dans le mauvais ordre
- Barre de navigation trop grande
- Mauvaise implémentation des boutons empêchant leur changement de couleurs lors de leur utilisation
- Légende des boutons pas reliées aux boutons et à la mauvaise taille

Lors de la mise en fonction de la barre de navigation et des layouts :

- Mauvaise implémentation du code Kotlin de la fondation de l'application empêchant l'affichage des layouts
- Mauvaise implémentation du code Kotlin des layouts empêchant leur affichage
- Mauvaise implémentation de la barre de navigation empêchant son affichage par dessus les layouts
- Mauvaise implémentation dans le fonctionnement de la barre de navigation empêchant le changement des layouts
- Oubli de préciser quel layout doit être affiché par défaut affichant la mauvaise page au démarrage de l'application

# Conclusion

Je n'ai malheureusement pas réussi à finir mon projet, j'ai réussi à terminer la partie visuelle de l'application mais je n'ai pas eu le temps de commencer la partie fonctionnelle, je pense que pour cela il m'aurait fallu un mois de plus environ. Je compte continuer petit à petit ce projet chez moi en fonction de mon temps libre car je souhaite essayer de le finaliser pour mon plaisir personnel.

J'ai beaucoup apprécié réaliser ce projet car j'ai aimé découvrir de nouvelles choses, de nouveaux outils, de nouvelles méthodes et faire quelque chose de nouveaux. J'ai trouvé que c'était une expérience très enrichissante. De plus j'ai beaucoup apprécié que l'on soit libre de choisir ce que l'on souhaitait faire comme projet, cela m'a permis de m'épanouir dans cet exercice.