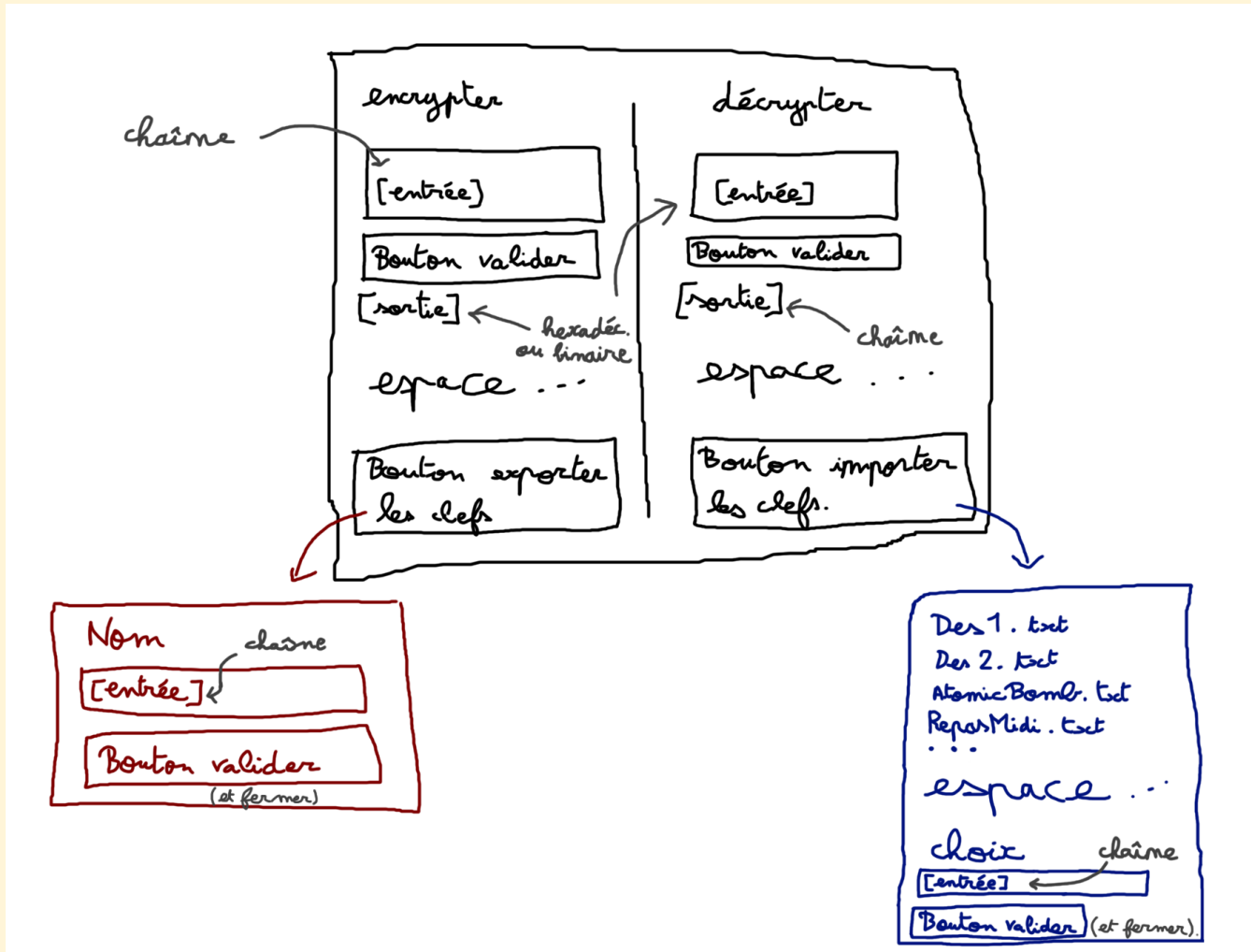


Fonctionnement

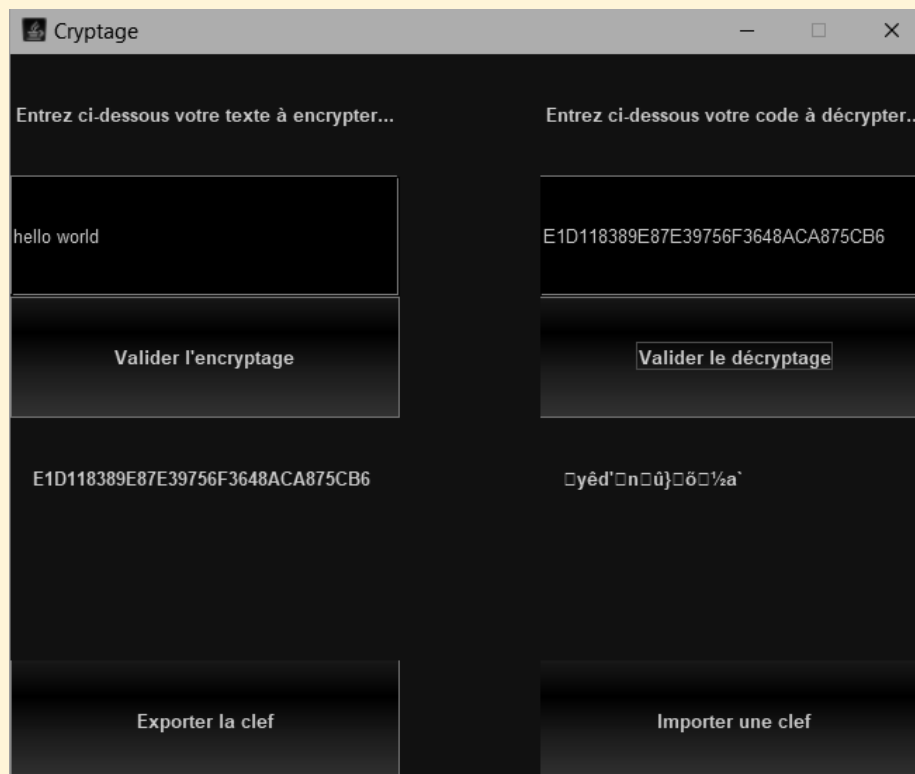
L'objectif est de créer un algorithme de **triple DES** afin de permettre à plusieurs personnes ayant des clefs de cryptage en commun de **communiquer** entre elles.



Pour cela, nous avons créé une **interface graphique** divisée en 2 sections : une pour **encrypter** (à gauche) et une autre pour **décrypter** (à droite).

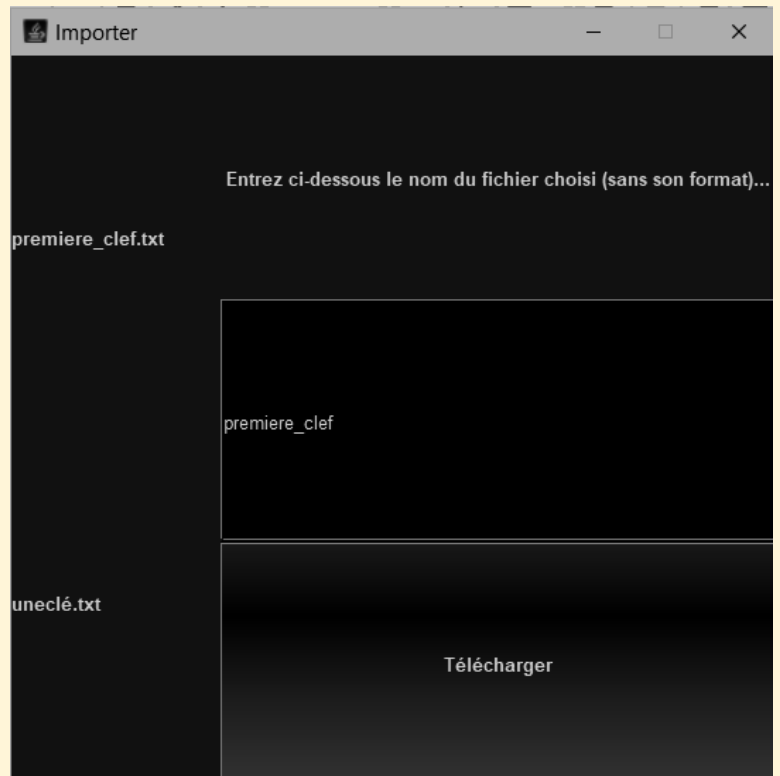
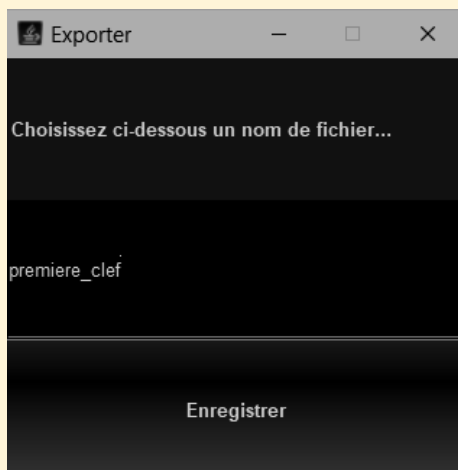
L'encryptage propose (semi-aléatoirement) **un** codage en hexadécimal du texte saisi, en l'affichant dessous et dans la console pour permettre à l'utilisateur de le copier à sa guise.

Tandis que le décryptage fait la fonction inverse, affichant ainsi **un** texte (avec accents depuis JDK 18) à partir du codage saisi plus haut ; les formats supportés ici sont le **binaire** et l'**hexadécimal**.



Mais il est également possible de synchroniser ses résultats de cryptage, à l'aide de **clefs** !

Pour ce faire, nous avons rajouté 2 boutons pour **exporter** (à gauche) et **importer** (à droite) ; lors de l'exportation, les valeurs aléatoires précédemment utilisées (appelées graines de la génération) sont alors extraites du dossier de stockage dynamique "**~/temp**", puis conservées dans un fichier texte par sérialisation, à l'intérieur d'un autre dossier "**~/datas**", contenant ainsi toutes les clefs exportées, alors que l'importation copie le fichier du nom saisi dans "**~/temp**", en vue d'une future utilisation par les fonctions de cryptage.



Résultat :

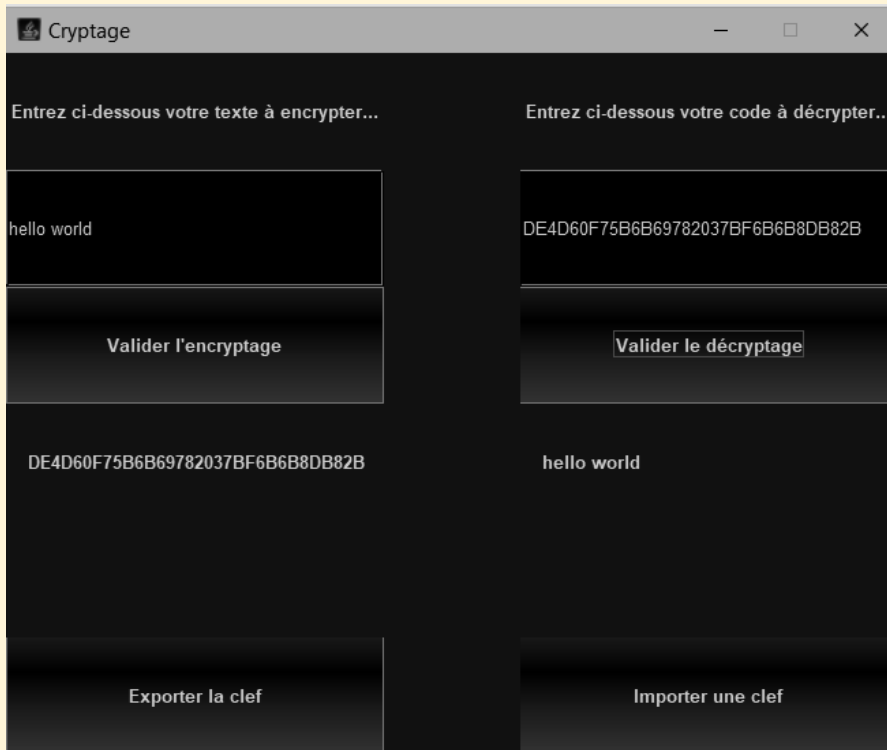


Schéma des clefs :

