

# Énoncé du Travail Pratique 2

10 mars 2017

Préparé par  
Benjamin Lemelin

## 1 Résumé du travail

L'application à développer doit permettre de crypter et de décrypter de petits messages. Ces messages seront cryptés en utilisant une simple technique de substitution de caractères. La clé de chiffrement devra être obtenue à partir d'un serveur en ligne. De plus, l'application devra permettre de crypter n'importe quelle chaîne de caractère en provenance d'une autre application, tel que l'application de messagerie SMS.

## 2 Conditions de réalisation du travail

Valeur sur la note finale	Type	Durée	Nombre de remises
15 %	En équipe	3 semaines	1

## 3 Travail à effectuer

Cette section vise à vous présenter les différentes étapes du travail à effectuer.

### 3.1 Importer et comprendre le code

Vous trouverez avec cet énoncé le projet de base à utiliser pour ce travail. Il contient :

- Un dialogue pour sélectionner une clé et sa classe de tests d'interface (Voir « KeyPickerDialog »)
- Un filtre de caractères pour les « EditText » et sa classe de tests (Voir « CharactersInputFilter »)
- Une activité et sa classe de tests d'interfaces (Voir « MainActivity »)
- Un extrait de code permettant de placer du texte dans le presse-papier (Voir « MainActivity »)

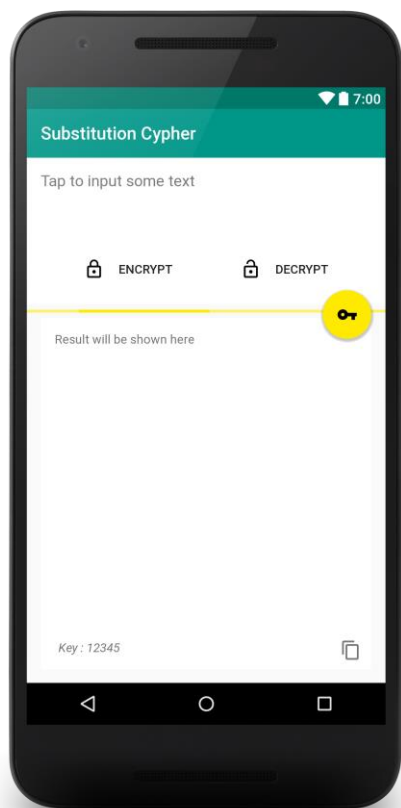
L'utilisation du « KeyPickerDialog » est très simple. En fait, cela fonctionne exactement comme un « MessageBox ». Le premier paramètre est l'activité dans lequel le dialogue doit être affiché, le deuxième est la clé à afficher au départ, le troisième est la longueur de la clé et le dernier est l'observateur à appeler lorsque le dialogue est fermé (dans l'exemple ci-dessous, c'est aussi l'activité).

```
KeyPickerDialog.show(this, 123, 3, this);
```

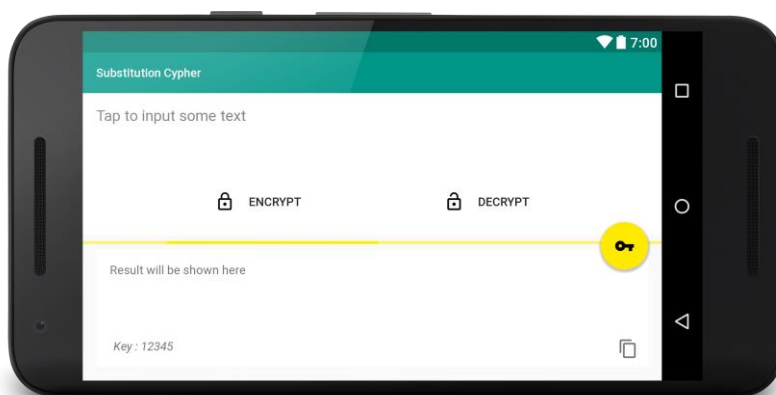
Le « CharactersInputFilter » est un filtre applicable sur les « EditText ». Il permet de limiter ce qui peut être saisi dans un « EditText » par l'utilisateur. Pour que cela fonctionne, il faut l'appliquer à un « EditText » via la méthode « [setFilters](#) ». Consultez [la documentation](#) pour les détails.

## 3.2 Créer l'interface utilisateur

Recréez l'interface ci-dessous. Pour simplifier, il vous est conseillé d'utiliser un « ConstraintLayout ».



*Note :* La grande barre jaune est une barre de [progression à durée indéterminée](#).



Notez le gros bouton jaune avec une clé. C'est un « FloatingActionButton ». Cette « View » fait partie de la librairie « design » de Google qu'il faudra ajouter dans votre « build.gradle », si ce n'est pas déjà fait :

```
dependencies {  
    compile 'com.android.support:design:25.1.0'  
}
```

Vous pouvez alors l'ajouter dans votre vue en XML ainsi :

```
<android.support.design.widget.FloatingActionButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    app:fabSize="normal"  
    app:srcCompat="@drawable/ic_key"/>
```

Il agit exactement comme n'importe quel ImageButton. Consultez [la documentation](#) pour plus de détails.

Testez le fonctionnement de votre application en créant des tests d'interface avec Espresso. *Indice : il devrait y avoir au moins (mais probablement plus) de tests d'interface Espresso qu'il y a de « User Stories ».*

Notez que ce projet vient avec des « ViewAction » spécialement conçus pour le « KeyPickerDialog ». Utilisez-les ainsi dans vos tests Espresso :

```
onView(withId(R.id.keyPickerDialog)).perform(setKey(123)); //Changer la clé  
onView(withId(R.id.keyPickerDialog)).perform(ok()); //Faire OK  
onView(withId(R.id.keyPickerDialog)).perform(cancel()); //Faire Annuler
```

### 3.3 Créer le module de chiffrement

Votre application devra utiliser un [chiffrement par substitution](#). Le principe est simple : chaque lettre est remplacée par une autre. Créez donc votre couche « Modèle » pour répondre à ce besoin. Vous avez carte blanche : faites autant de classes, de fonctions et de packages que nécessaire. Par contre, vous devrez avoir une classe capable de représenter une clé de chiffrement, exactement ainsi.

SubstitutionCypherKey
<div>-key : int</div> <div>-inputCharacters : char[]</div> <div>-outputCharacters : char[]</div>
<div>+SubstitutionCypherKey(key : int, inputCharacters : char[], outputCharacters : char[])</div> <div>+getKey() : int</div> <div>+getInputCharacters() : char[]</div> <div>+getOutputCharacters() : char[]</div>

La clé de chiffrement est simple : il s'agit de deux tableaux dont vous vous servirez pour faire correspondre les caractères en fonction de leur position. Par exemple :

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 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 .
x T H I a q Z O X V z v S M R L h m B l d e s K N n Q r p D o I g W b . G U E w C f P k A J t F c y u j Y

Notez que rien ne vous empêche d'y ajouter des fonctions. Aussi, n'oubliez pas de tester toutes les classes de votre couche « Modèle » unitairement avec JUnit. Enfin, documentez correctement votre code.

### 3.4 Créer la tâche asynchrone d'obtention de clé

*(À faire dès que la matière sur les appels réseau avec la plateforme Android est vue et comprise).*

Les clés de chiffrement peuvent être obtenues en effectuant un appel à ce service web :

<code>http://cypherkeys-acodebreak.rhcloud.com/substitution/key/01977.json</code>
---

Essayez-le dans votre navigateur plusieurs fois en remplaçant la partie en jaune par n'importe quel entier sur 32 bits. Notez que chaque clé possède un identifiant unique. Par exemple, avec « 01977 », vous aurez :

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 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 .
x T H I a q Z O X V z v S M R L h m B l d e s K N n Q r p D o I g W b . G U E w C f P k A J t F c y u j Y

Implémentez une tâche asynchrone qui effectue un appel à ce service web, tel que montré dans l'Exercice 5. Aussi, utilisez « Jackson » pour effectuer la désérialisation, tel que montré dans l'Exercice 6.

### 3.5 Finaliser l'application

Placez votre application dans un dépôt Git ([GitHub](#), [GitLab](#) ou [BitBucket](#)). Ajoutez chaque « User Story » dans les « Issues » de votre dépôt. Servez-vous en pour vous tenir à jour sur la progression du travail en tant qu'équipe.

#### 3.5.1 User Stories

No	Description	OK
1	En tant qu'utilisateur, je veux pouvoir choisir l'identifiant de ma clé de chiffrement afin de sécuriser convenablement un message.	
<del>2</del>	<del>En tant qu'utilisateur, je veux qu'une clé aléatoire soit choisie au lancement de l'application pour ne pas utiliser la même clé que d'autres utilisateurs.</del>	
3	En tant qu'utilisateur, je veux pouvoir crypter un message avec la clé de chiffrement que j'aurai choisi préalablement.	
4	En tant qu'utilisateur, je veux pouvoir décrypter un message avec la clé de chiffrement que j'aurai sélectionné préalablement.	
5	En tant qu'utilisateur, je veux être en mesure de voir, en un clin d'œil, l'identifiant de ma clé de chiffrement afin de pouvoir la partager avec le destinataire du message.	
6	En tant qu'utilisateur, je veux pouvoir copier le message crypté ou décrypté afin de m'en servir dans d'autres applications.	
7	En tant qu'utilisateur, je veux pouvoir envoyer du texte vers cette application pour le décrypter facilement.	
8	En tant qu'utilisateur, je veux que l'on me demande de choisir ma clé de chiffrement lorsque j'envoie du texte vers cette application afin d'être plus efficace.	
9	En tant qu'utilisateur, je veux que l'application utilise un chiffrement par substitution afin que les messages soient facilement chiffrés et déchiffrés.	
10	En tant qu'utilisateur, je veux que l'on m'empêche de saisir des caractères qui ne seraient pas supportés par la méthode de chiffrement.	
<del>11</del>	<del>En tant qu'utilisateur, je veux que l'on m'avertisse de toute erreur qui s'est produite et que l'on me propose une solution.</del>	
<del>12</del>	<del>En tant que francophone, je veux que l'application soit en français.</del>	
<del>13</del>	<del>As an english speaker, I want my application to be in english.</del>	

### 3.5.2 Critères d'acceptabilité de la conception

No	Description	OK
1	La clé de chiffrement doit être obtenue à partir du service web à cette adresse : <a href="http://cypherkeys-acodebreak.rhcloud.com/substitution/key/#####.json">http://cypherkeys-acodebreak.rhcloud.com/substitution/key/#####.json</a>	
2	Aucune donnée ne doit être perdue pendant un changement d'orientation. Il n'est cependant pas nécessaire de conserver l'état du « KeyPickerDialog ».	
3	Les images utilisées doivent toutes être des images vectorielles, sauf l'icône de l'application.	
4	Affichez une « <a href="#">ProgressBar</a> » durant tout appel à un service réseau.	
5	Lors d'une erreur de communication avec le serveur, affichez un « Snackbar » et proposez à l'utilisateur d'activer le Wifi. Pour ce faire, créez un « <a href="#">Intent</a> » avec <a href="#">l'action correspondante</a> .	
6	Lors d'une erreur de la part du serveur (tel qu'une <a href="#">erreur HTTP 500</a> ), affichez un « Snackbar » et indiquez à l'utilisateur de réessayer plus tard. Pour simuler une erreur, utilisez l'URL <a href="http://cypherkeys-acodebreak.rhcloud.com/error">http://cypherkeys-acodebreak.rhcloud.com/error</a> .	

### 3.5.3 Directives techniques

No	Description	OK
1	L'application doit utiliser le style architectural « MVC ».	
2	Le package de l'application doit être « ca.csf.mobile1.tp2 ».	
3	Pour empêcher qu'un caractère invalide soit saisi dans un « EditText », utilisez le « InputFilter » fourni avec le projet, c'est-à-dire « CharactersInputFilter ». Ce dernier hérite de « <a href="#">InputFilter</a> ». Consultez <a href="#">la documentation</a> pour les détails.	
4	Le code de la couche « Modèle » de l'application doit être testé unitairement avec JUnit 4.	
5	L'interface utilisateur, c'est-à-dire la couche « Contrôleur » et « Vue », doivent être testés avec des tests d'interface automatisés Espresso.	
6	Vous ne savez pas encore comment tester l'aléatoire ni comment tester les erreurs de connexion réseau. Vous n'avez donc pas à écrire de tests pour les « User Story » 2 et 11.	
7	Vous devez utiliser un dépôt de code sous Git et en donner l'accès à votre professeur.	
8	Vous devez placer les « User Stories » dans la section « Issues » de votre dépôt Git.	

## 4 Modalités de remise

Remettez sur LÉA à l'heure et à la date indiquée par votre professeur un fichier texte comportant l'adresse vers votre dépôt Git. Cette adresse sera utilisée pour effectuer un « git clone ». Ajoutez à ce fichier le nom et le matricule de tous les membres de votre équipe ainsi que tout commentaire pertinent à la correction. Enfin, le nom du fichier texte doit être vos matricules, séparés par « \_ », suivis de « .txt ». Une seule journée de retard est tolérée entraînant alors une pénalité de 15 % de la note. Au-delà de ce délai, le travail est refusé et la note « 0 » est attribuée.

## 5 Évaluation

Ce travail sera évalué avec une correction normale et une correction négative. Les points seront retirés en fonction de la gravité de la faute, de la qualité globale du travail et de l'effort mis par l'étudiant. Le tableau suivant vise à résumer les éléments pouvant réduire la note, jusqu'à atteindre la note de 0 %.

Éléments	Pondération
<b>Fonctionnalités :</b> <ul style="list-style-type: none"> <li>Clé choisie aléatoirement au lancement de l'application. (0176-1)</li> <li>Possibilité de choisir précisément une clé de chiffrement. (0176-1, 0176-2)</li> <li>Utilisation d'un chiffrement par substitution. (0176-1)</li> <li>Texte saisi crypté correctement avec la clé de chiffrement actuelle. (0176-1)</li> <li>Texte saisi décrypté correctement avec la clé de chiffrement actuelle. (0176-1)</li> <li>Possibilité de copier le résultat dans le presse-papier. (0176-1, 0176-2)</li> <li>Caractères non supportés par le chiffrement refusés. (0176-1, 0176-2)</li> <li>Messages d'erreurs se produisant lorsqu'une erreur de connexion se produit. (0176-1)</li> <li>Possibilité d'envoyer du texte vers cette application pour le déchiffrer. (0176-1)</li> <li>Demande d'une clé lors de l'envoi de texte à déchiffrer vers cette application. (0176-1)</li> </ul>	5 % Correction normale
<b>Serveur de données</b> (incluant, mais sans s'y limiter) : <ul style="list-style-type: none"> <li>Données de l'application non obtenues à partir d'un service REST. (017D-6)</li> <li>Obtention des données non faite en tâche de fond. (017D-5)</li> <li>Aucune gestion des erreurs de réseau et d'obtention des données. (017D-4, 017D-5)</li> </ul>	20 % Correction négative
<b>Interface utilisateur</b> (incluant, mais sans s'y limiter) : <ul style="list-style-type: none"> <li>Non-respect de l'interface utilisateur imposée. (017D-4)</li> <li>Langue française non supportée. (017D-4)</li> <li>Langue anglaise non supportée. (017D-4)</li> <li>Non-respect de la grille de 8dp. (017D-4)</li> <li>Éléments d'interface mal alignés. (017D-4)</li> <li>Non utilisation d'images vectorielles. (017D-4)</li> <li>Non-utilisation de « Snackbar » pour les messages. (017D-4)</li> <li>Fichier XML de « Layout » Android malpropre. (017D-4)</li> <li>« Layouts » Android instables sous différentes tailles et densités d'écran. (017D-4)</li> <li>Expérience utilisateur désagréable. (017D-4)</li> <li>Perte de données lors d'un changement d'orientation. (017D-3, 017D-4)</li> </ul>	10 % Correction négative
<b>Documentation du code</b> (incluant, mais sans s'y limiter) : <ul style="list-style-type: none"> <li>Classe non documentée ou documentée de manière insuffisante. (017D-7)</li> <li>Méthode non documentée ou documentée de manière insuffisante. (017D-7)</li> <li>Paramètres non documenté ou documenté de manière insuffisante. (017D-7)</li> <li>Valeur de retour non documenté ou documenté de manière insuffisante. (017D-7)</li> <li>Lien non présent vers une classe référencé dans la documentation. (017D-7)</li> <li>Documentation malpropre. (017D-7)</li> <li>Indentation incorrecte de la documentation. (017D-7)</li> <li>Documentation qui n'est pas en français. (017D-7)</li> <li>Standard de documentation « JavaDoc » non respecté. (017D-7)</li> </ul>	5 % Correction négative

<p><b>Tests unitaires</b> (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> <li>• Fonctionnalité d'un contrôleur non testée par un test d'interface. (0177-1, 0177-2, 0177-3)</li> <li>• Fonctionnalité d'une vue non testée par un test d'interface. (0177-1, 0177-2, 0177-3)</li> <li>• Fonctionnalité de la couche modèle non testée unitairement. (0177-1, 0177-2, 0177-3)</li> <li>• Test pas suffisamment robuste. (0177-2)</li> <li>• Test vérifiant plus qu'une fonctionnalité. (0177-1)</li> <li>• Vérification échouant dans un test unitaire. (0177-1, 0177-2, 0177-3)</li> <li>• Vérification manquante dans un test. (0176-4, 0177-1)</li> <li>• Utilisation de la mauvaise vérification « JUnit ». (0177-2)</li> <li>• Utilisation de la mauvaise vérification « Espresso ». (0177-2)</li> </ul> <p><i>Sont exclus de la correction :</i></p> <ul style="list-style-type: none"> <li>• Test des appels réseau.</li> <li>• Test de la génération aléatoire de la clé de chiffrement.</li> </ul>	<p>10 % Correction négative</p>
<p><b>Qualité générale de l'architecture</b> (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> <li>• Contrôleur effectuant de la logique réservée à la couche modèle. (017D-3, 017D-5)</li> <li>• Séparation incorrecte entre la vue et le modèle. (017D-3, 017D-5)</li> <li>• Abstractions non utilisées où cela aurait été nécessaire. (016T-1)</li> <li>• Référence concrète où une référence abstraite aurait été suffisante. (016T-3)</li> <li>• Pas d'usage de l'héritage où cela aurait été nécessaire. (016T-1, 016T-3)</li> <li>• Non-respect du diagramme de classe. (016T-2)</li> <li>• Méthode ayant trop de responsabilités pas découpées convenablement. (016T-3)</li> <li>• Classe ayant trop de responsabilités pas découpées convenablement. (016T-3)</li> <li>• Découpage en « package » insuffisant. (016T-2)</li> <li>• Nom d'un « package » imprécis. (016T-3)</li> <li>• « dot.case » non utilisé pour un nom de « package ». (016T-3)</li> </ul>	<p>20 % Correction négative</p>
<p><b>Qualité générale du code</b> (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> <li>• Logique applicative « patchée ». (016T-3)</li> <li>• Nom d'une variable imprécis. (016T-3)</li> <li>• Nom d'une classe imprécis. (016T-3)</li> <li>• Nom d'une interface imprécis. (016T-3)</li> <li>• Nom d'une constante imprécis. (016T-3)</li> <li>• Nom d'une énumération imprécis. (016T-3)</li> <li>• Nom d'une méthode imprécis. (016T-3)</li> <li>• « camelCase » non utilisé pour un nom de variable ou de méthode. (016T-3)</li> <li>• « PascalCase » non utilisé pour un nom d'interface, de classe ou d'énumération. (016T-3)</li> <li>• « UPPER_CASE » non utilisé pour un nom de constante. (016T-3)</li> <li>• Non-respect des standards d'indentation Java. (016T-3)</li> <li>• Manque d'une constante ou cela aurait été nécessaire. (016T-3)</li> <li>• Attribut ou méthode sans modificateur de visibilité (ou avec le mauvais). (016T-3)</li> <li>• Méthode publique qui devrait être privée. (016T-3)</li> <li>• Méthode statique qui ne devrait pas l'être. (016T-3)</li> <li>• Méthode protégée qui ne devrait pas l'être. (016T-3)</li> <li>• Attribut de classe qui n'est utilisé que dans une seule méthode. (016T-3)</li> <li>• Commentaire inutile ne compensant pas le manque d'expressivité du code. (016T-3)</li> <li>• Erreur à la compilation. (016T-3, 016T-5)</li> <li>• Avertissement à la compilation. (016T-3, 016T-5)</li> </ul>	<p>20 % Correction négative</p>



<b>Qualité générale du travail</b> (incluant, mais sans s'y limiter) :		
• Corruption de la configuration Gradle des projets.	(016T-5)	10 % Correction négative
• Non-respect des consignes de remise.	(016T-5)	
• Fichiers inutiles remis avec les projets.	(016T-5)	
• Travail non remis via l'outil gestion de code source Git.	(0176-3)	

Notez que la qualité écrite de la langue française fait partie intégrante de l'évaluation de ce travail. Étant donné que l'accès aux ressources de la langue française est permis, chaque faute de français retirera donc 0,5 % à la note finale jusqu'à concurrence de 20 % retirés. Sont corrigées les fautes d'orthographe, les fautes de grammaire, les fautes syntaxiques et les fautes de ponctuation.

## 6 Compétences reliées au travail pratique

Voici la liste des compétences évaluées, évaluées partiellement ou non-évaluées par ce travail. Vous trouverez les explications sur ces choix en dessous de cette liste.

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>• 016T-1 Créer un modèle objet : Évalué.</li> <li>• 016T-2 Perfectionner le modèle objet : Non évalué.</li> <li>• 016T-3 Procéder à la codification d'une classe : Évalué.</li> <li>• 016T-5 Générer la version exécutable du programme : Évalué.</li> <li>• 0176-1 Analyser les caractéristiques de l'application : Évalué.</li> <li>• 0176-2 Analyser la nature des améliorations à apporter à l'application : Évalué.</li> <li>• 0176-3 Ajouter et modifier des fonctionnalités à une application : Évalué.</li> <li>• 0176-4 Valider le fonctionnement de l'application</li> </ul> | <ul style="list-style-type: none"> <li>• 0177-1 Planifier les tests : Évalué.</li> <li>• 0177-2 Procéder à l'exécution des différents tests : Évalué.</li> <li>• 0177-3 Valider la qualité de l'application : Évalué.</li> <li>• 017D-2 Établir le cadre technologique : Évalué.</li> <li>• 017D-3 Préparer le travail de développement : Évalué.</li> <li>• 017D-4 Produire le prototype de présentation : Évalué.</li> <li>• 017D-5 Produire le prototype de communication : Évalué.</li> <li>• 017D-6 Développer l'application : Évalué.</li> <li>• 017D-7 Produire la documentation relative à l'application : Évalué.</li> </ul> |
|---|---|

La compétence 016T-2 sera évaluée au travail pratique 3.