

Programmation des composants mobiles PROJET

1 Modalités

Le projet est à réaliser en binôme (ou, éventuellement, en monôme).

Les soutenances auront lieu soit en décembre soit en janvier, en fonction des dates disponibles (janvier semble plus probable). La date précise sera donnée ultérieurement.

Nous vous rappelons que chaque membre du binôme doit faire une partie conséquente du projet. Si, par exemple, l'un(e) d'entre vous ne fait que les layouts, il (elle) ne sera pas considéré(e) comme ayant fait une partie conséquente.

Pendant la soutenance chaque membre du binôme doit montrer la maîtrise de la totalité du code. Il est possible qu'une partie de la soutenance soit individuelle.

Par ailleurs, nous pourrions utiliser un détecteur de plagiat pour repérer les copies et autres « emprunts ».

Vous devez créer pour votre projet un dépôt git sur gaufre

`https://gaufre.informatique.univ-paris-diderot.fr/users/sign_in`

avant le 20 novembre.

Les deux enseignants de l'UE doivent être désignés comme **Reporter** de votre projet sur git. Pour stocker votre projet Android dans dépôt git on procède de façon suivante :

- (1) Vous créez un projet sur AndroidStudio.
- (2) Sur AndroidStudio vous allez vers **VCS -> Enable Version Control Integration**.
- (3) **Select a version control system to associate with the project** : choisir Git.
Les noms de fichiers à sauvegarder dans git sont désormais coloriés en rouge.

- (4) Dans le menu de AndroidStudio apparaît un onglet **Git** à la place de **VCS**. Allez vers **Git -> Manage Remotes**

Dans la fenêtre qui s'ouvre faire un clic sur **+**. Une autre fenêtre s'ouvre où vous copiez le lien `https` vers votre dépôt git.

- (5) L'étape suivante est peut-être inutile si votre nom et mail Git sont déjà renseignés globalement sur votre machine.

Vous allez vers le répertoire de votre projet et vous exécutez sur le terminal deux commandes :

```
git config user.name your_user_name
git config user.email your_email
```

`your_user_name` c'est évidemment votre nom sur Gitlab écrit entre les guillemets doubles. Vous pouvez voir ce nom en vous connectant sur le Gitlab gaufre et en allant vers l'onglet le plus à droite et en sélectionnant **Settings** (juste avant **Sign out**). D'habitude c'est votre nom et prénom avec l'espace entre les deux.

`your_mail` c'est votre mail sur Gitlab gaufre, visible également dans **Settings**.

- (6) Dans AndroidStudio vous allez vers **Git -> Commit**. Une sous-fenêtre s'ouvre à gauche avec tous les fichiers que AndroidStudio propose de sauvegarder dans le dépôt. Sélectionner tout. Écrire un **Commit message** et faire **Commit** ou directement **Commit and push**. Notez que **commit** est impossible si vous n'avez pas écrit quelque chose dans la sous-fenêtre **Commit message**.

Bien sûr votre binôme doit plutôt commencer par cloner le projet.

Dans le dépôt git vous ajoutez `README.md` où vous mettrez les noms de tous les membres de votre équipe de projet.

Le projet doit être réalisé en Kotlin.

Dans le projet, il y aura sans doute des listes. **Toutes les listes affichées doivent être implémentées par des `RecyclerViews`** avec des `Adapters` appropriés.

L'utilisation de fragments n'est pas demandée mais si vous en faites un bon usage cela vous apportera des points.

L'utilisation de `ViewModels` dans toutes les activités qui possèdent les données à préserver pendant le changement de configuration est très vivement recommandée. D'ailleurs l'application doit se comporter correctement quand on tourne l'appareil. Nous allons vérifier pendant la soutenance le comportement de votre application durant le changement de position de l'appareil. Si vous interdisez de faire tourner l'écran cela enlèvera des points à votre évaluation.

Évitez `findViewById` surtout si les layouts contiennent beaucoup de Views, view binding est bien plus adapté.

Présentation de projet

C'est un projet qui est sensé aider l'utilisateur à apprendre les langues étrangères. Il y a maintenant beaucoup de dictionnaires disponibles en ligne et trouver une traduction ne présente aucun problème. Mais le fait d'avoir vu une fois la traduction d'un mot ne signifie pas que nous l'avons mémorisé de façon définitive et la mémorisation de mots et de phrases ou expressions repose sur la répétition.

L'idée est de maintenir une base de données qui contient les mots à apprendre avec des liens vers les pages web de traduction de ce mots.

Pour chaque mot on stockera :

- la langue source (la langue du mot),
- la langue de destination (la langue de traduction), cette langue peut être la même que la langue source, les dictionnaires monolingue sont très utiles,
- l'adresse http vers la page web de traduction du mot,
- d'autre informations à déterminer.

L'application à implémenter sera composée d'au moins quatre parties :

- recherche de traduction,
- sauvegarde de résultat de recherche dans la BD,
- l'exercice de mémorisation,
- paramétrage de l'application.

2 Recherche de traduction

L'application doit maintenir dans la BD une liste de dictionnaires.

L'utilisateur écrit un mot ou une phrase dans un `EditView`, choisit un dictionnaire et lance la recherche. En pratique cela signifie que l'application lance un navigateur internet avec l'adresse http ou https du dictionnaire et le mot recherché.

Peut-être il faudra aussi mémoriser comment est formée l'adresse http avec le mot recherché.

Par exemple pour le dictionnaire français-anglais Larousse on prend l'adresse `https://www.larousse.fr/dictionnaires/francais-anglais` du dictionnaire et on concatène la mot recherché après un / :

`https://www.larousse.fr/dictionnaires/francais-anglais/maison` pour chercher la traduction de `maison` en anglais.

C'est la même chose pour le dictionnaire

`http://dictionnaire.reverso.net/francais-anglais` ou pour

`https://dictionary.cambridge.org/fr/dictionnaire/francais-anglais`. Mais est-ce que ce que c'est toujours le cas, est-ce qu'il suffit chaque fois juste concaténer après / le mot recherché à la fin de l'Url ?

Parfois on veut traduire un mot mais on n'a aucun dictionnaire adéquat enregistré (ce notamment le cas quand on lance l'application pour la première fois).

Dans ce cas l'activité de recherche lancera la recherche via google ou un autre site de recherche. Cette recherche via google doit être proposée même si on connaît déjà un dictionnaire, peut-être utilisateur veut voir d'autres dictionnaires.

Recherche gogle. En examinant l'adresse formée par google il est facile de voir que la recherche google est lancée avec l'url `http://www.google.fr/search?q=` avec le mot (ou phrase) recherché concaténé après le caractère `=` à la fin.

2.1 Comment lancer un navigateur

Pour lancer un navigateur avec l'Url particulier on procède de façon suivante.

Soit `txt` un variable **String** qui contient l'adresse url complet. Le code suivant lance votre navigateur préféré avec cette adresse :

```
val intent = Intent(Intent.ACTION_VIEW)
intent.setData( Uri.parse( txt ) )
startActivity( intent )
```

3 Sauvegarde le résultat de la recherche

Supposons que le navigateur affiche une page de traduction et nous voulons sauvegarder le lien vers cette traduction dans la BD :

Une possibilité, *surtout à ne pas suivre*, consiste à

- copier l'adresse de la page dans le presse-papier,
- aller vers l'activité de sauvegarde dans votre application,
- remplir :
 - le mot à traduire,
 - la langue source (français pour une traduction de français),
 - la langue de destination (anglais pour la traduction vers l'anglais),
 - copier l'adresse de la page depuis le presse-papier dans un `EditText`,
- cliquer sur un bouton de sauvegarde pour mettre tout dans la BD.



FIGURE 1 – La page de traduction de mot « nager » dans le dictionnaire wordreference

Cette procédure est bien trop compliquée pour être viable, en particulier l'utilisation de presse papier pour passer l'Url depuis le navigateur vers l'activité pose un problème. Pourquoi copier l'adresse dans le presse-papier quand on peut faire un transfert automatique.

Le mécanisme moderne pour faire passer un lien depuis le navigateur vers une activité utilise le partage (**partager** ou **Share**). L'utilisateur ouvre le menu de navigateur et choisit le partage. Android affiche la liste d'activités capables de recevoir le partage.

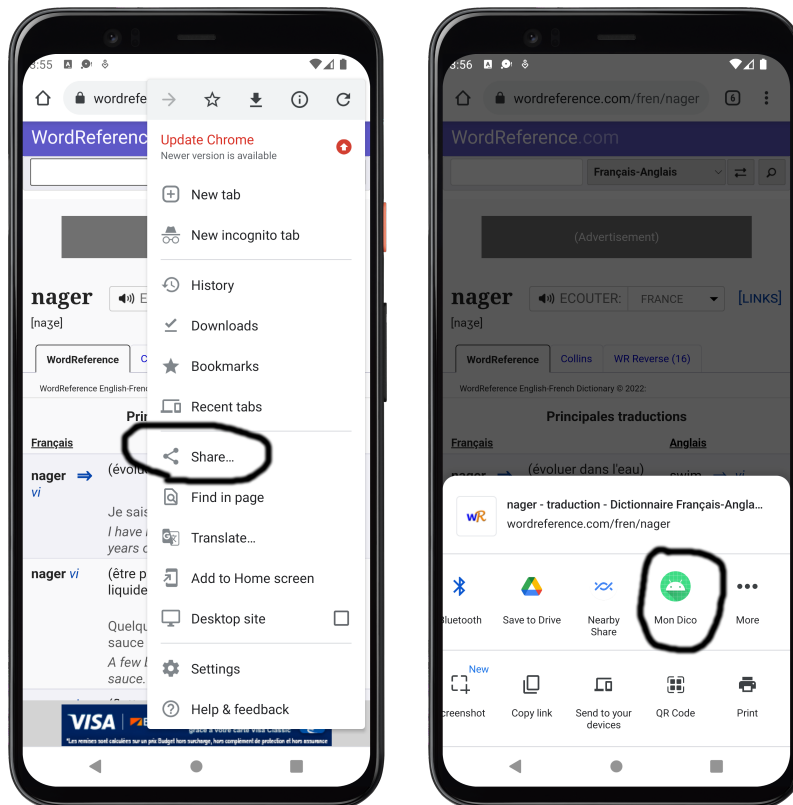
Il faut faire en sorte que votre activité de sauvegarde soit sur cette liste. Sur la figure 2 (b) c'est l'activité **Mon Dico** qui s'affiche sur le choix d'activités prêtes à recevoir le lien url.

L'utilisateur clique sur l'activité qu'il choisit et l'android ouvre l'activité sélectionnée en lui passant le lien vers la page web.

L'utilisateur doit toujours remplir les langues source et destination mais il n'a plus rien à faire avec le presse-papier.

3.1 Faire afficher votre activité dans le partage

Pour qu'une activité s'affiche dans la liste d'activités qui peuvent recevoir l'adresse http il suffit d'ajouter un filtre dans AndroidManifest.xml :



(a) Ouvrir le menu de navigateur et choisir Partager

(b) Choisir l'activité à laquelle le navigateur envoie le lien vers la page affichée

FIGURE 2 – Envoyer un lien web depuis le navigateur vers une activité.

```
<activity
    android:name=".SauvegardeActivity"
    android:label="Mon Dico"
    ...>

<intent-filter>
    <action android:name="android.intent.action.SEND" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="text/plain"/>

</activity>
```

Le filtre, installé ici pour l'activité `SauvegardeActivity`, fait que l'activité s'affiche sur la liste d'activités qui peuvent recevoir l'information depuis le navigateur. L'attribut `android:label` donne le nom qui s'affiche sur le choix d'activités. Sans doute il convient de changer aussi l'icône associée à l'activité, elle s'affiche aussi sur l'écran de partage.

3.2 Récupérer l'adresse url envoyé par le navigateur

Supposons que le navigateur affiche une page de traduction que nous voulons sauvegarder. L'utilisateur a choisi l'activité de sauvegarde dans le partage et cette activité est lancée par Android. Mais comment l'activité de sauvegarde trouvera-t-elle l'adresse de la page web envoyée par le navigateur ?

<https://developer.android.com/training/sharing/receive> décrit en détail comment récupérer cette adresse mais cela peut se résumer de façon suivante :

L'activité est lancée depuis le navigateur par un `Intent` dont la référence se trouve dans la propriété `intent` de l'activité. Dans `onCreate()` de l'activité on pourra récupérer l'url de la page web grâce au code suivant :

```
if( intent.action.equals( "android.intent.action.SEND" ) ){
    val txt = intent.extras?.getString( "android.intent.extra.TEXT" )
    ...
    // si txt différent de null alors txt contient maintenant une réfé
    rence vers un String
    // avec l'adresse Url
    // de la page qui s'affichait dans le navigateur
}
```

Une fois l'adresse http de traduction récupérée par l'activité de sauvegarde il suffit que l'utilisateur spécifie le mot donc la traduction est sur la page web, les langues source et destination et l'activité de sauvegarde peut enregistrer ces informations dans la base de données.

4 Apprentissage

Chaque jour l'application envoie vers les notifications $n = 10$ mots choisis au hasard dans la base de donnée (dans n notifications distinctes).

L'utilisateur peut

- soit écarter une notification s'il connaît le mot (c'est une auto-évaluation)
- soit cliquer sur un bouton action dans la notification ce qui doit provoquer l'affichage de la page web de traduction dans un navigateur.

Dans les deux cas la notification sera supprimée.

Suggestion. Supposons que pendant l'apprentissage l'application a présenté 10 mots et l'utilisateur en a écarté 4 sans avoir demandé la page de traduction. Cela signifie qu'en pratique il a rafraîchi sa mémoire pour 6 mots au lieu de 10. Il est logique que pour chaque mot écarté l'application lui propose un nouveau mot. Donc dans notre cas, l'application proposera 4 nouveau mots dans les notifications (dont certains peuvent être écartés à nouveau).

Ne demandez pas trop. Supposons qu'un jour l'utilisateur n'a pas fait son devoir et le jour suivant il reste 10 notifications non consommées. Il vaut mieux d'éviter d'ajouter des nouvelles notifications. En général chaque jour on complète les notifications affichées pour proposer exactement n mots.

Ne pas proposer les mots maîtrisés. Si l'utilisateur a écarté deux ou trois fois un mot, on peut considérer qu'il le maîtrise définitivement. Il convient de ne plus le proposer dans la phase d'apprentissage - au moins pour quelques semaines/mois (sans le supprimer de la BD).

Notez que la phase d'apprentissage n'utilise aucune activité. Il suffit que l'alarme « réveille » chaque jour un Service qui sélectionne les mots à mettre dans les notifications.

5 Paramétrage de l'application et nettoyage de la BD

Il faut une activité qui permet de fixer certains paramètres :

- le nombre de mots proposés à chaque séance d'apprentissage,
- la fréquence : une fois par jours ou plus souvent ?
- l'utilisateur peut apprendre plusieurs langues en même temps. Mais cela a peu de sens de proposer de traductions vers les langues différentes en même temps. Par exemple l'utilisateur peut demander de lui présenter les traduction français-anglais lundi, mercredi, jeudi et les traduction français-allemand les autres jour de la semaine.

Nettoyage de BD. Il faut prévoir le nettoyage de BD – supprimer les mots appris, ou même supprimer certains langues.

Bien sûr il faut aussi prévoir la gestion de dictionnaires.

6 Aller plus loin

Il y a des extensions utiles du projet. Deux exemples.

Et si l'utilisateur souffre parfois de l'absence de connexion internet ? Il serait utile de pouvoir s'exercer hors connexion. Cela implique la possibilité de stocker localement les pages web à la place de liens. Peut-être pas de façon systématique mais donner le choix entre le stockage de lien et stockage de page web au moment d'ajout d'un nouveau mot dans la BD (avec la possibilité de modifier ce choix ensuite).

Et pour le travail en connexion donner la possibilité de choisir entre la connexion wifi et data.

Avez-vous d'autres extensions utiles ?

7 Notation

La notation prend en compte

- les fonctionnalités implémentées,
- facilité de navigation dans votre application,
- la qualité de programmation.