

## TRAVAIL PRATIQUE 3 (420-254-MO)

**À REMETTRE AU PLUS TARD LE JEUDI 8 DÉCEMBRE 2016.** Aucun retard ne sera toléré pour la remise du TP3.

Le travail pratique 3 permet de mettre en pratique les notions vues jusqu'au document 15 (surtout les documents 12 à 15).

### DATES DES PROCHAINS COURS :

Jeu. 17 nov. (Théorie doc. 16 + Lab. pour TP3)  
Mar. 22 nov. (Théorie doc. 17 et 18 + Lab. pour TP3)  
Jeu. 24 nov. (Lab. pour TP3)  
Mar. 29 nov. (MONTRER LES LAYOUTS + Lab. pour TP3)  
Jeu. 1 déc. (Lab. pour TP3)  
Mar. 6 déc. (Lab. pour TP3)  
Jeu. 8 déc. (REMISE DU TP3, aucun retard ne sera toléré)  
Mar. 13 déc. (MONTRER LES EXÉCUTIONS DU TP3, de 13h00 à 15h30)  
Mar. 13 déc. (EXAMEN THÉORIQUE 2 de 16h15 à 18h00, doc. 12 à 17)

### CONSIGNES POUR LA REMISE DE CE TRAVAIL

- ▶ Travail individuel ou en équipe de deux.
- ▶ Aucun travail ne sera accepté par Courriel.
- ▶ Vous devez zipper le projet (extension .zip).
- ▶ Formater les fichiers sources selon les normes de programmation (`ctrl+Alt+L`).
- ▶ Les lignes ne doivent pas déborder. Respecter la marge d'impression dans l'éditeur.
- ▶ Vous devez me montrer les layouts de votre application d'ici le mardi 29 novembre 2016 (juste en preview). Vous pourrez les modifier par la suite si nécessaire.
- ▶ Vous devez me montrer le résultat final de l'exécution de votre application.  
Vous trouverez des jeux d'essai sur les pages 25 à 32 de l'énoncé.
- ▶ **Copier le projet zippé** sur le dépôt du collège, dans le dossier suivant (il y a déjà un dossier à votre nom). Si en équipe de deux, copier le projet zippé dans les deux dossiers.

\\Srvfich2\Depots\Kyprianou, Sotira\420-254-Android\TP3-254\VotreNom

### BARÈME DE CORRECTION

Exécution : 50 % (justesse et présentation des résultats)  
Programmation : 50 % (justesse, efficacité du code, **respect des consignes et des normes**)

## APPLICATION ANDROID À FAIRE SEUL OU EN ÉQUIPE DE DEUX

L'application doit permettre à l'utilisateur de choisir entre **trois activités** qui n'ont aucun lien entre elles :

**Calculer un prêt, Identifier un conifère et Une activité de votre cru.**

Dans l'activité qui permet de **calculer un prêt**, l'utilisateur doit entrer le montant du prêt, le taux d'intérêt annuel et la durée du prêt. L'activité calcule et affiche le montant des versements mensuels, l'intérêt total et le prêt total. Voir le détail de cette activité dans les sections suivantes.

Dans l'activité qui **identifie un conifère**, l'utilisateur observe la forme des aiguilles ou des écailles du conifère et l'activité l'aidera dans son identification. Voir le détail de cette activité dans les sections suivantes.

Dans **l'activité de votre cru**, vous devez inventer une petite activité qui met surtout en pratique les notions des documents 12 à 15. En plus du menu glissant et des actions sur la barre d'actions comme indiqués dans les sections suivantes, votre activité doit contenir au moins un composant du document 12, au moins un menu contextuel quelconque (document 13), lancer au moins une autre activité (document 14) et contient au moins un truc du document 15 (page web, boîte de dialogue, barre de progression).

Il n'est pas nécessaire que l'application fonctionne pour les deux langues. Le français peut être la langue de défaut.

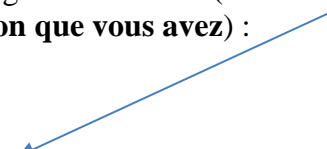
Votre application doit fonctionner pour **l'orientation portrait** (dossier `res/layout`) et **paysage** (dossier `res/layout-land`). À vous de vérifier si il y a des informations qu'il faudra sauvegarder. Utiliser les méthodes `onSaveInstanceState` et `onRestoreInstanceState` si nécessaire.

N'oubliez-pas de **commenter le code Java** aux endroits appropriés : en-tête de chaque fichier (nom du ou des auteurs, nom du fichier, date, une description du fichier). Chaque classe est commentée (avant sa définition) pour indiquer ce qu'elle représente. Chaque méthode pertinente pour le traitement est commentée (avant sa définition) pour indiquer ce qu'elle accomplit. Des commentaires pertinents (des explications) doivent être insérés dans le code, là où il est nécessaire d'en faciliter la compréhension.

## RENSEIGNEMENTS SUR LA CONCEPTION DE L'APPLICATION

- ⇒ Vous devez modifier l'icône de lancement de l'application (File/New/Image Asset). Voir page 10 du document 4.
- ⇒ Les différents textes à afficher doivent se trouver dans le fichier `strings.xml`.
- ⇒ Dans le fichier `build.gradle` (Module: `app`) dans les scripts gradle (Gradle Scripts), dans la section des dépendances (*dependencies*), vous avez déjà les lignes suivantes (**le numéro de la version peut être différent, ne pas modifier le numéro de version que vous avez**) :

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:24.2.1'  
}
```

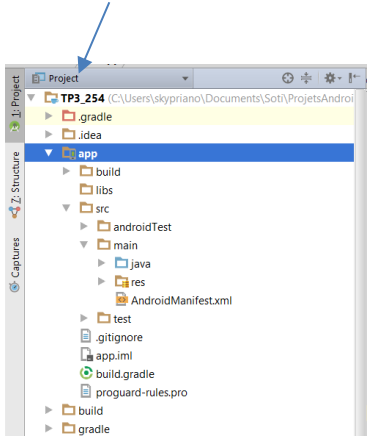


Vous devez ajouter les bibliothèques de compatibilité et différents composants pour gérer les anciennes versions d'Android. Vous devez ajouter les lignes suivantes au fichier `build.gradle` (Module: `app`) dans les scripts gradle (Gradle Scripts), dans la section des dépendances (*dependencies*). **Les numéros de version doivent correspondre à votre numéro de version à vous. Ensuite, vous devez faire Build/Rebuild Project :**

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:24.2.1'  
    compile 'com.android.support:support-v4:24.2.1'  
    compile 'com.android.support:recyclerview-v7:24.2.1'  
    compile 'com.android.support:cardview-v7:24.2.1'  
}
```

## Renseignements sur les fichiers fournis.

Vous trouverez sur le réseau public du collège et dans Col.NET, un dossier TP3-Android contenant l'énoncé et des fichiers pour votre projet. On y trouve des fichiers contenant des ressources (array, string, layout, menu et drawable) et des fichiers sources .java. À vous de les copier aux bons endroits. Il est préférable d'être dans la vue Project (au lieu de la vue Android) pour faire les copies. Voir l'utilité de ces fichiers dans les sections suivantes.



Copier les dossiers drawable dans le dossier res du projet.

Copier le fichier layout item\_liste\_conifere.xml dans le dossier res/layout du projet (même endroit que activity\_main.xml).

Copier le dossier menu dans le dossier res du projet.

Copier les fichiers values arrays.xml et strings.xml dans le dossier res/values du projet. Modifier le nom de la troisième activité dans la ressource array tab\_activites. Modifier la string app\_name, si nécessaire.

Copier les fichiers sources Pret.java, Conifere.java et ConifereAdapterRV.java dans le dossier java/nomPackage/ du projet (même endroit que MainActivity.java). Modifier le chemin du package dans ces fichiers sources, si nécessaire.

## Renseignements sur la conception des interfaces graphiques (layout).

Il y a un fichier layout par activité. Un fichier layout pour l'activité principale (premier écran de l'application), un fichier layout pour le premier écran de l'activité qui calcule un prêt, un fichier layout pour le deuxième écran de l'activité qui calcule un prêt, un fichier layout pour le premier écran de l'activité qui identifie un conifère, un fichier layout pour le deuxième écran de l'activité qui identifie un conifère, un fichier layout pour le troisième écran de l'activité qui identifie un conifère, un fichier layout pour le premier écran de l'activité de votre cru et un fichier layout pour chaque écran suivant de l'activité de votre cru.

**Voici un exemple de Preview de chaque fichier layout en mode portrait. N'oubliez pas de créer les fichiers layouts en mode paysage si nécessaire.**

Les lignes pour travailler avec le menu glissant à gauche (drawer) doivent se trouver dans chaque layout des activités (y compris ceux de l'activité de votre cru), car nous pourrions changer d'activité à tout moment.

## Premier écran de l'application (fichier layout activity\_main.xml).

Vous devez entourer votre layout des lignes suivantes pour pouvoir travailler avec le menu glissant à gauche (voir pages 20 et 21 du document 13). La source de l'image est "@drawable/ic\_accueil" (vous pouvez modifier l'image). Modifier les attributs `scaleX` et `scaleY` pour qu'elle apparaisse plus grande.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!--Mettre ici votre layout.-->

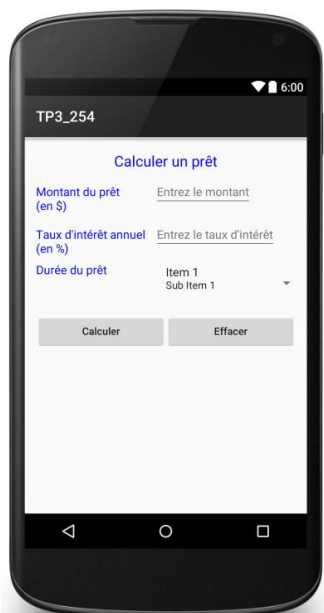
    <ListView
        android:id="@+id/list_navigation"
        android:layout_width="250dp"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:background="#FFFFFF"
        android:choiceMode="singleChoice"
        android:entries="@array/tab_activites" />

</android.support.v4.widget.DrawerLayout>
```



## Premier écran de l'activité qui calcule un prêt.

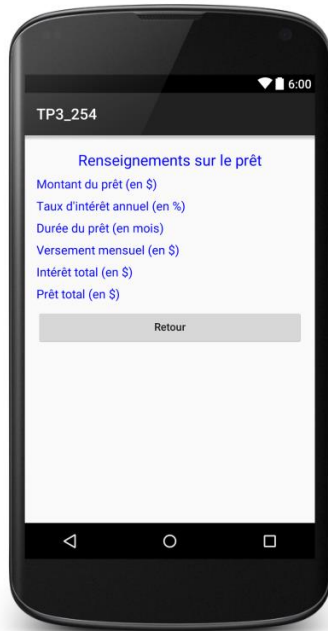
Vous devez entourer votre layout des mêmes lignes que pour le premier écran de l'application pour pouvoir travailler avec le menu glissant à gauche. L'attribut `inputType` des deux `EditText` doit être `"numberDecimal"`. N'oubliez pas l'attribut `hint` des deux `EditText`. La durée du prêt est un composant `Spinner` (voir document 12). L'attribut `entries` du spinner est `"@array/tab_duree"` et l'attribut `spinnerMode` est `"dropdown"`.



Si vous utilisez la version 2.1.3 de Android Studio et que vous avez des problèmes de rendu, modifiez la version de l'API du Preview pour 23 au lieu de 24.

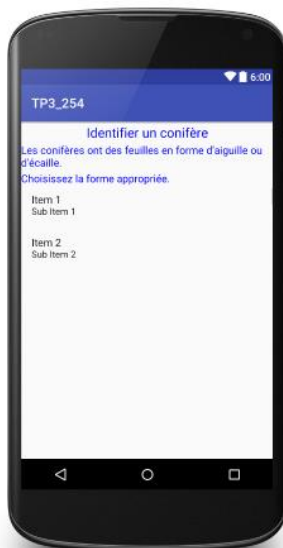
## Deuxième écran de l'activité qui calcule un prêt.

Vous devez entourer votre layout des mêmes lignes que pour le premier écran de l'application pour pouvoir travailler avec le menu glissant à gauche.



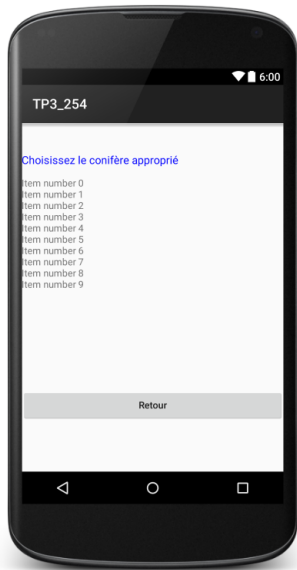
## Premier écran de l'activité qui identifie un conifère.

Vous devez entourer votre layout des mêmes lignes que pour le premier écran de l'application pour pouvoir travailler avec le menu glissant à gauche. Après les textes en haut, il y a un composant `ListView`. La hauteur de ce `ListView` est 140dp (peut être différent pour vous), l'attribut `choiceMode` est "singleChoice" et l'attribut `entries` est "@array/tab\_choix\_aiguilles". Après le `ListView`, il y a un composant `TextView` qui contiendra éventuellement le résultat de l'écran suivant. Ajouter l'attribut `android:drawablePadding="5dp"` (car contiendra texte et image).



## Deuxième écran de l'activité qui identifie un conifère.

Vous devez entourer votre layout des mêmes lignes que pour le premier écran de l'application pour pouvoir travailler avec le menu glissant à gauche. **Avant** le texte "Choisissez le conifère approprié", il y a un composant `TextView` qui affichera éventuellement un titre. Le titre sera affiché dans le code Java, car il dépend du choix de l'écran précédent. **Après** le texte "Choisissez le conifère approprié", il y a un composant `android.support.v7.widget.RecyclerView` (voir page 50 du document 12). Ici, la largeur est "match\_parent" et la hauteur est 300dp (140dp en mode paysage pour ne pas perdre l'affichage du bouton Retour sur l'écran). La hauteur peut être différente pour vous.



## Troisième écran de l'activité qui identifie un conifère.

Vous devez entourer votre layout des mêmes lignes que pour le premier écran de l'application pour pouvoir travailler avec le menu glissant à gauche. On y trouve un composant `WebView` (voir page 5 du document 15) avec une largeur et une hauteur à "match\_parent".



## Renseignements sur les actions de la barre d'actions.

La barre d'actions, doit contenir une action pour retourner au premier écran de l'application et une action pour afficher de l'aide. Ces actions seront présentes pour tous les écrans de l'application (y compris ceux de l'activité de votre cru).

Vous devez modifier le fichier `menu_barre_actions.xml` du dossier `menu` et y ajouter deux items. Voir pages 7 à 11 du document 13.

Le premier item pour l'action permet de retourner à l'écran d'accueil. Son icône est l'image `ic_accueil`, son titre est la string `action_accueil` et son comportement dans la barre d'actions est `"always|withText"`.

Le deuxième item pour l'action permet d'obtenir de l'aide. Son icône est l'image `ic_action_aide`, son titre est la string `action_aide` et son comportement dans la barre d'actions est `"always|withText"`.

Voir dans les sections suivantes, le code Java pour la barre d'actions.

## ACTIVITÉ PRINCIPALE DE L'APPLICATION

Cette activité est gérée par le fichier `MainActivity.java` et le layout `activity_main.xml`.

Dans la méthode `onCreate`, après le `setContentView`, vous devez exécuter une méthode qui fera la création du menu glissant (drawer) à gauche de l'application. Voir les pages 19 à 23 du document 13.

Voir l'exemple de la page 21 pour le code de la méthode qui fera la création du drawer.

Dans le code de la variable qui écoute la liste, vous devez lancer l'activité qui a été sélectionnée avec une intention explicite. Vous n'avez pas à faire circuler des données à ces activités et ces activités n'ont pas à vous fournir ou à vous annoncer quoi que soit à leur retour. Voir milieu de la page 8 du document 14. Fermer le `drawerLayout` après une sélection.

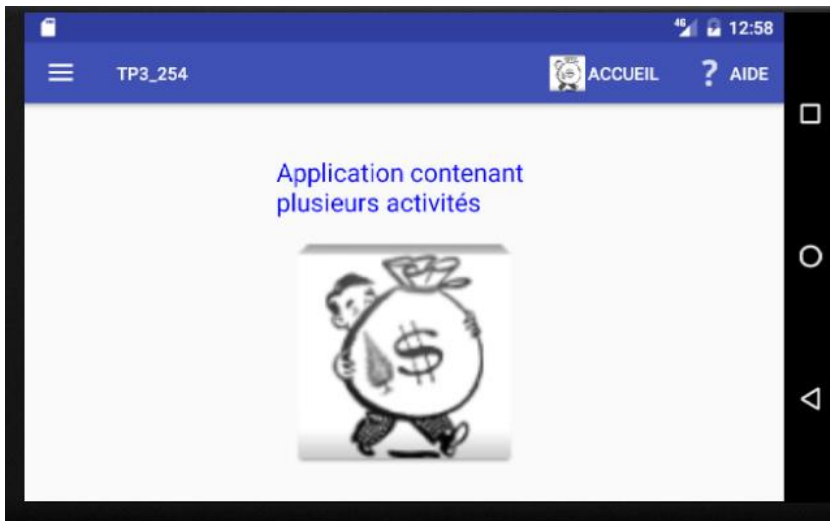
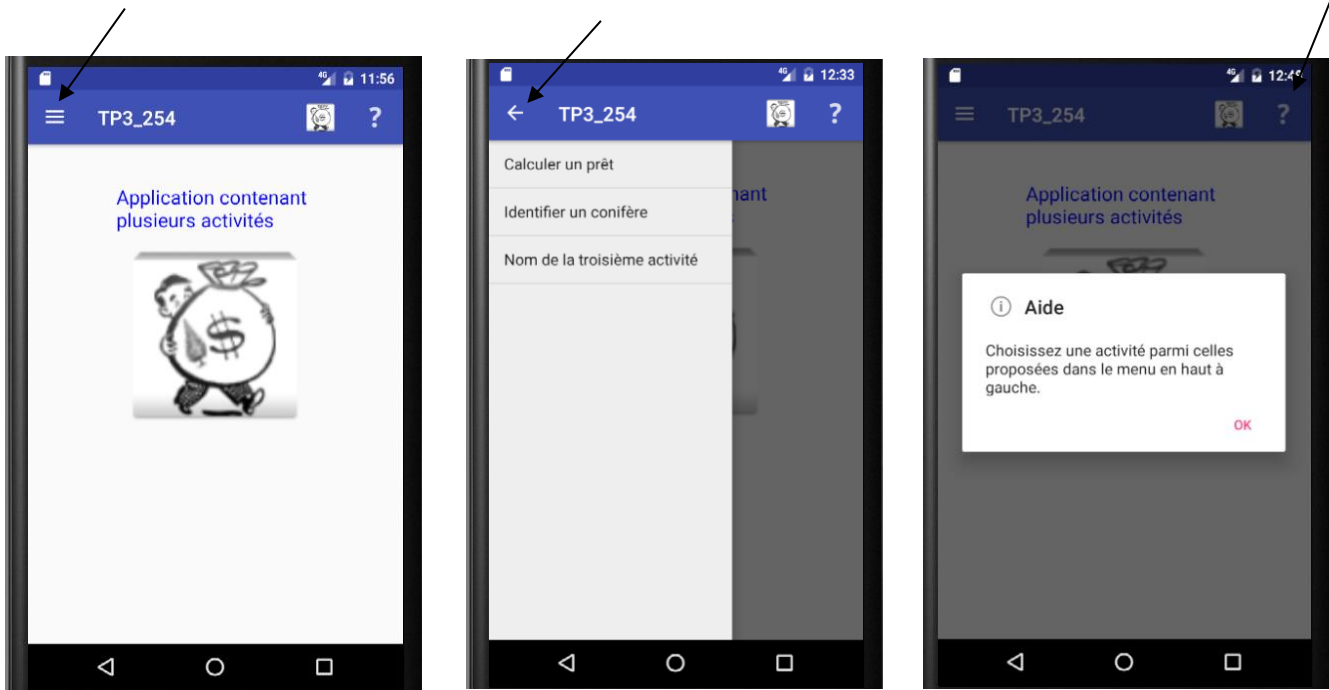
Coder la méthode `onPostCreate` pour synchroniser le `drawerToggle`. Voir page 22 du document 13.

Coder La méthode `onOptionsItemSelected` pour lier la barre d'actions à l'activité. Voir en bas de la page 8 du document 13.

Coder la méthode `onOptionsItemSelected` pour gérer les événements de la barre d'actions. Voir page 22 du document 13. Ici, le `case` dans le `switch` est pour l'action à effectuer lorsque l'utilisateur cliquera sur l'aide dans la barre d'actions. L'action est d'exécuter une méthode qui permettra d'afficher une boîte de dialogue dans laquelle on trouvera l'aide sur l'activité courante. Voir le code de cette méthode sur la page suivante.



Dans la méthode qui permet d'afficher une boîte de dialogue pour l'action aide de la barre d'actions, vous devez créer une boîte de dialogue avec la classe `AlertDialogue`. Voir pages 10 à 12 du document 15. Ici, le titre est la string `action_aide`, l'icône est l'image `ic_info_aide` et le message est la string `aide_accueil`. Programmer **seulement** l'écoute du bouton qui se trouve tout à droite, le texte du bouton est la string `txt_alertdialog_ok` et aucun traitement ne sera effectué pour ce bouton. N'oubliez pas les deux instructions en haut de la page 12 du document 15 pour créer et afficher la boîte de dialogue.



## COMPLÉTER LA CLASSE `Pret`

Dans la méthode `calculerVersement`, faire le calcul suivant seulement si le montant, l'intérêt et la durée sont supérieurs à 0. Attention ! C'est exposant -durée

```
tauxMois = intérêt / 100 / 12

versement = montant * tauxMois
            1 - (1 + tauxMois)-durée
```

Dans la méthode `calculerPretTotal`, faire le calcul suivant seulement si le montant, l'intérêt et la durée sont supérieurs à 0.

```
pretTotal = versement mensuel * durée
```

Dans la méthode `calculerInteretTotal`, faire le calcul suivant seulement si le montant, l'intérêt et la durée sont supérieurs à 0.

```
interetTotal = prêt total - montant
```

## ACTIVITÉ QUI PERMET DE CALCULER UN PRÊT

### Premier écran de l'activité qui permet de calculer un prêt.

Cet écran est géré par un autre layout et une autre classe `.java`.

Ajouter cette activité dans le fichier manifeste `AndroidManifest.xml`. Voir page 7 du document 14.

Faire un copier/coller du contenu de la classe `MainActivity` dans cette classe et apporter les modifications nécessaires :

- Modifier le nom de la classe pour la nouvelle classe.
- Dans la méthode `onCreate`, modifier le layout du `setContentView`.
- Dans la méthode qui crée le drawer, ajouter une instruction juste avant celle qui ajoute la flèche de remontée pour ajouter le sous-titre (la string `titre_pret`) à la barre d'actions. Voir en haut de la page 5 du document 13.

- Dans le code de la variable qui écoute la liste du drawer, vous devez lancer l'activité qui a été sélectionnée **à partir de cette classe** et avec une intention explicite. Vous n'avez pas à faire circuler des données à ces activités et ces activités n'ont pas à vous fournir ou à vous annoncer quoi que soit à leur retour. Juste avant de lancer l'activité en question, ajouter seulement un flag à l'intent (`Intent.FLAG_ACTIVITY_CLEAR_TOP`) pour dépiler les activités de la pile avant de lancer l'activité. Voir page 13 du document 14 (ici, juste un flag). Fermer le `drawerLayout` après une sélection.
- Dans la méthode `onOptionsItemSelected`, ajouter un case dans le `switch` pour l'action à effectuer lorsque l'utilisateur cliquera sur l'icône accueil. L'action est de dépiler les activités de la pile jusqu'à la première. Voir page 13 du document 14.
- Dans la méthode qui permet d'afficher une boîte de dialogue pour l'action aide de la barre d'actions, vous devez apporter les modifications nécessaires : le nom de l'activité et le message est la string `aide_pret`.
- Dans la méthode `onCreate`, après l'appel à la méthode qui crée le drawer, effectuer les traitements nécessaires pour cette activité.

Récupérer les différents composants du layout.

Créer un objet de la classe `Pret` de base.

Modifier la durée du prêt par la durée qui se trouve à la position 0 du spinner. La méthode `getItemAtPosition` du spinner permet de récupérer l'item qui se trouve à une position. Vous devez convertir l'item en `String` ("12 mois") et ensuite convertir les deux premiers caractères "12" en nombre entier 12.

Définir l'écouteur du spinner dans une variable. Voir en bas de la page 5 du document 12.

Définir l'écouteur pour le bouton calculer dans une variable.

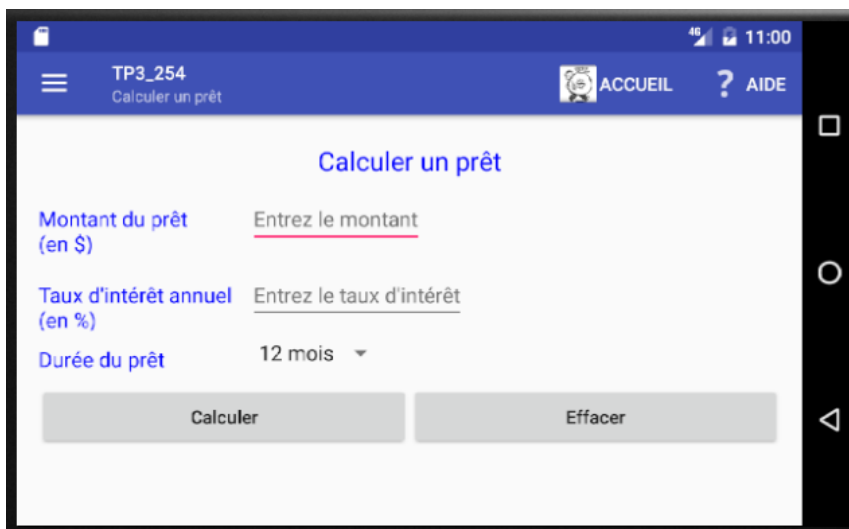
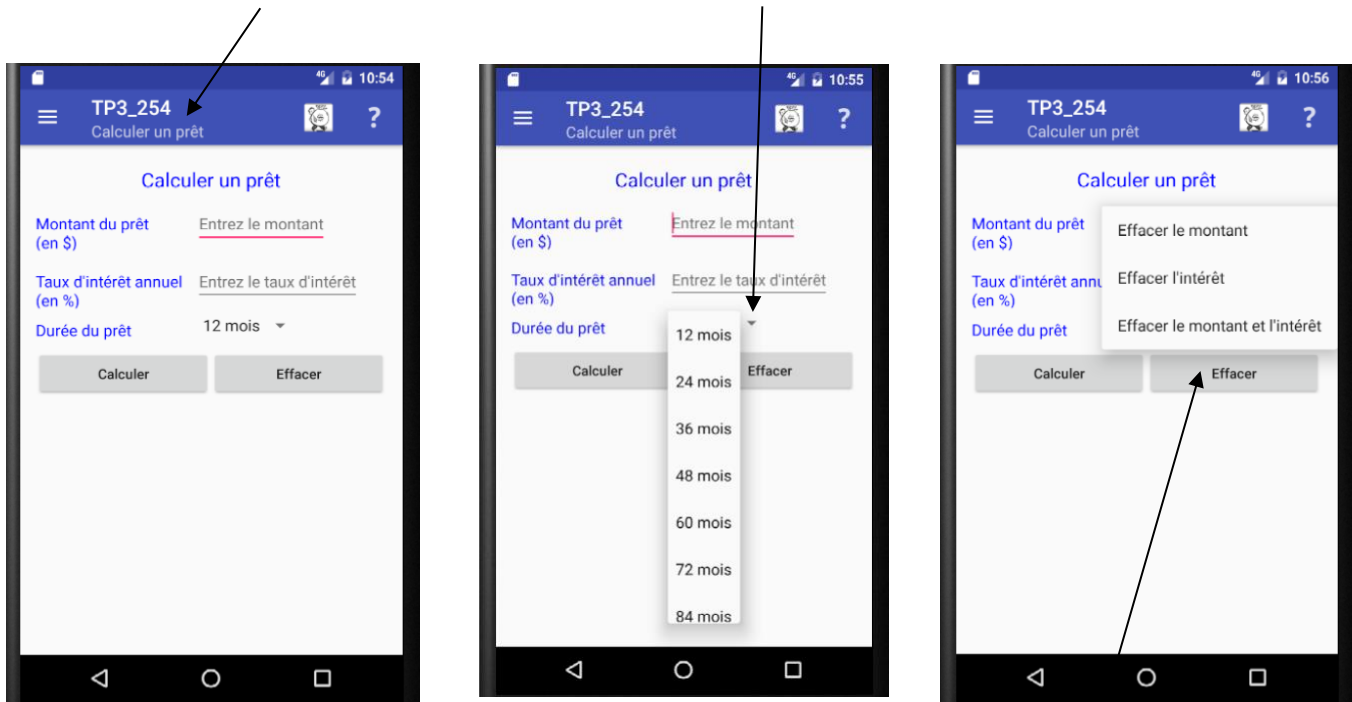
Exécuter une méthode pour traiter le menu contextuel Pop-Up pour le bouton effacer.

- Dans la variable qui écoute le spinner, vous devez récupérer la durée et modifier la durée du prêt par celle récupérée. Attention! La durée correspond au deux premiers caractères de la sélection. Par exemple, si l'item est "36 mois", alors la durée est 36. Voir page 6 du document 12.
- Dans la variable qui écoute le bouton calculer, vous devez lancer l'activité qui calculera et affichera les résultats du prêt avec une intention explicite. Vous avez à fournir à cette activité la donnée extra suivante : l'objet du prêt. Voir en bas de la page 23 du document 14 (ici, l'objet est déjà créé). L'activité lancée n'a pas à fournir ou à annoncer quoi que soit à son retour. **ATTENTION!!** L'objet du prêt doit contenir les données entrées et l'activité sera lancée seulement si toutes les informations sont entrées et que les valeurs sont supérieures à 0. Un message `Toast` sera affiché s'il n'est pas possible de calculer les résultats. Le message à afficher est la string `warning_infos_pret`.

- Dans la méthode qui permet de traiter le menu contextuel Pop-Up pour le bouton effacer, vous devez affecter un pop-up menu (menu contextuel) au bouton Effacer. Voir pages 17 à 19 du document 13.

Vous devez créer un fichier .xml dans le dossier menu contenant les items du menu contextuel (un attribut `android:id` et `android:title` par item).

Dans la méthode, vous devez utiliser la classe `PopupMenu`. Vous devez gérer l'événement `OnMenuItemClickListener` pour les entrées du menu contextuel. Le traitement à effectuer est de mettre une chaîne vide "" dans les champs appropriés.



## Deuxième écran de l'activité qui permet de calculer un prêt.

Cet écran est géré par un autre layout et une autre classe `.java`.

Ajouter cette activité dans le fichier manifeste `AndroidManifest.xml`. Voir page 7 du document 14.

Faire un copier/coller du contenu de la classe `MainActivity` dans cette classe et apporter les modifications nécessaires :

- Modifier le nom de la classe pour la nouvelle classe.
- Dans la méthode `onCreate`, modifier le layout du `setContentView`.
- Dans la méthode qui crée le drawer, ajouter une instruction juste avant celle qui ajoute la flèche de remontée pour ajouter le sous-titre (la string `sous_titre_infos_pret`) à la barre d'actions. Voir en haut de la page 5 du document 13.
- Dans le code de la variable qui écoute la liste du drawer, vous devez lancer l'activité qui a été sélectionnée **à partir de cette classe** et avec une intention explicite. Vous n'avez pas à faire circuler des données à ces activités et ces activités n'ont pas à vous fournir ou à vous annoncer quoi que soit à leur retour. Juste avant de lancer l'activité en question, ajouter seulement un flag à l'intent (`Intent.FLAG_ACTIVITY_CLEAR_TOP`) pour dépiler les activités de la pile avant de lancer l'activité. Voir page 13 du document 14 (ici, juste un flag). Fermer le `drawerLayout` après une sélection.
- Dans la méthode `onOptionsItemSelected`, ajouter un case dans le switch pour l'action à effectuer lorsque l'utilisateur cliquera sur l'icône accueil. L'action est de dépiler les activités de la pile jusqu'à la première. Voir page 13 du document 14.
- Dans la méthode qui permet d'afficher une boîte de dialogue pour l'action aide de la barre d'actions, vous devez apporter les modifications nécessaires : le nom de l'activité et le message est la string `aide_infos_pret`.
- Dans la méthode `onCreate`, après l'appel à la méthode qui crée le drawer, effectuer les traitements nécessaires pour cette activité.

Récupérer les différents composants du layout.

Exécuter une méthode qui permet d'afficher les résultats.

Définir l'écouteur pour le bouton Retour.

- Dans la méthode qui affiche les résultats, vous devez récupérer l'objet `Intent` et la donnée extra de l'activité précédente. Voir en haut de la page 27 du document 14. Vous devez afficher les différents résultats dans les champs appropriés.

Pour convertir une valeur double en `String` avec 2 chiffres après la virgule, vous pouvez utiliser la manière suivante :

```
String chaine = String.format( Locale.getDefault(), "%1$.2f", valeur );
```

- Dans la variable qui écoute le bouton Retour, le traitement a effectué est de tout simplement terminer l'activité courante avec l'instruction suivante : `finish()` ;.



## ACTIVITÉ QUI PERMET D'IDENTIFIER UN CONIFÈRE

### Premier écran de l'activité qui permet d'identifier un conifère.

Cet écran est géré par un autre layout et une autre classe `.java`.

Ajouter cette activité dans le fichier manifeste `AndroidManifest.xml`. Voir page 7 du document 14.

Faire un copier/coller du contenu de la classe `MainActivity` dans cette classe et apporter les modifications nécessaires :

- Modifier le nom de la classe pour la nouvelle classe.
- Dans la méthode `onCreate`, modifier le layout du `setContentView`.
- Dans la méthode qui crée le drawer, ajouter une instruction juste avant celle qui ajoute la flèche de remontée pour ajouter le sous-titre (la string `titre_conifere`) à la barre d'actions. Voir en haut de la page 5 du document 13.
- Dans le code de la variable qui écoute la liste du drawer, vous devez lancer l'activité qui a été sélectionnée **à partir de cette classe** et avec une intention explicite. Vous n'avez pas à faire circuler des données à ces activités et ces activités n'ont pas à vous fournir ou à vous annoncer quoi que soit à leur retour. Juste avant de lancer l'activité en question, ajouter seulement un flag à l'intent (`Intent.FLAG_ACTIVITY_CLEAR_TOP`) pour dépiler les activités de la pile avant de lancer l'activité. Voir page 13 du document 14 (ici, juste un flag). Fermer le `drawerLayout` après une sélection.
- Dans la méthode `onOptionsItemSelected`, ajouter un case dans le switch pour l'action à effectuer lorsque l'utilisateur cliquera sur l'icône accueil. L'action est de dépiler les activités de la pile jusqu'à la première. Voir page 13 du document 14.
- Dans la méthode qui permet d'afficher une boîte de dialogue pour l'action aide de la barre d'actions, vous devez apporter les modifications nécessaires : le nom de l'activité et le message est la string `aide_conifere`.
- Dans la méthode `onCreate`, après l'appel à la méthode qui crée le drawer, effectuer les traitements nécessaires pour cette activité.

Récupérer les différents composants du layout : la `ListView` pour les choix des aiguilles et le `TextView` pour afficher le résultat.

Définir l'écouteur de la liste des choix des aiguilles dans une variable. Voir page 30 du document 14.

- Dans la variable qui écoute la liste des choix des aiguilles, vous devez lancer l'activité qui permettra à l'utilisateur de choisir le conifère avec une intention explicite. Voir page 30 du document 14. Vous avez à fournir à cette activité la donnée extra suivante : la position (`int`).

L'activité lancée aura à fournir des données et à annoncer si elle s'est terminée normalement à son retour.

- Vous devez redéfinir la méthode `onActivityResult` pour gérer les traitements à faire lorsque l'activité lancée sera terminée. Voir page 31 du document 14.

Vous devez vous assurer que c'est la bonne activité qui s'est terminée.

Vous devez vérifier si l'activité lancée s'est terminée normalement (l'utilisateur a fait un choix).

Si l'activité lancée ne s'est pas terminée normalement (l'utilisateur n'a pas fait de choix), le composant `TextView` qui affiche le résultat, ne doit contenir aucun résultat.

Si l'activité lancée s'est terminée normalement (l'utilisateur a fait un choix), le composant `TextView` doit contenir le résultat à afficher. Vous devez récupérer les données extras que l'activité lancée vous a fournies : un nom de conifère (`String`), un identifiant d'une image drawable du conifère (`int`) et une page web (`String`). Les données extras se trouvent dans l'objet `Intent` (paramètre `data`).

Le composant `TextView` contiendra une image et un texte. Le début du texte est dans la string `mes_res_arbre`. L'attribut `android:drawablePadding` dans le fichier layout permet de spécifier l'espace à laisser entre l'image et le texte. Voici comment mettre l'image dans le champ `TextView` (à gauche du texte) :

```
idTextView.setCompoundDrawablesWithIntrinsicBounds( idImageConifere, 0, 0, 0 );
```

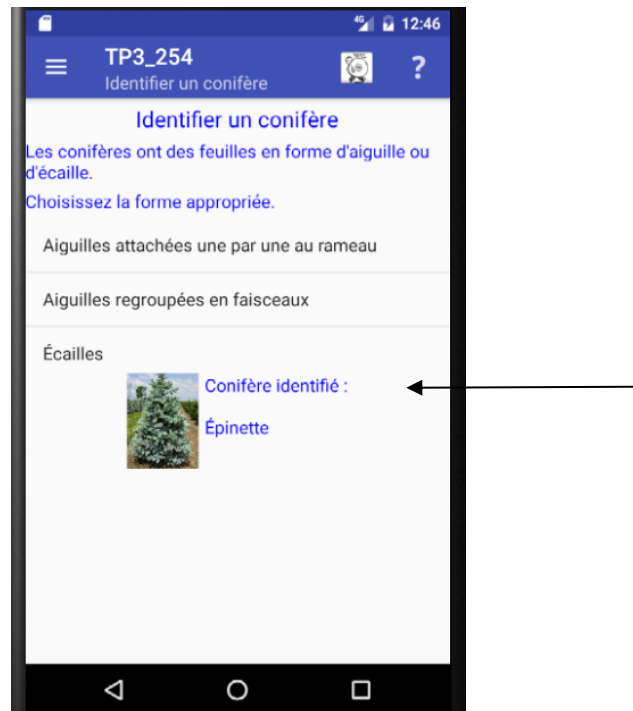
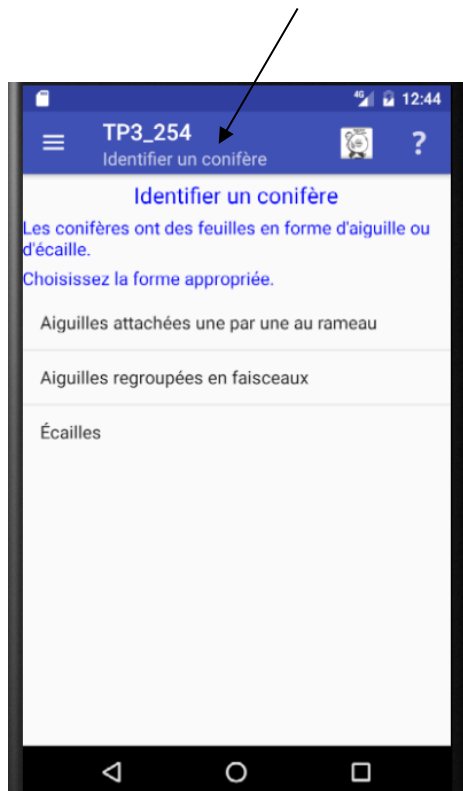
Rappel : L'id `android.R.color.transparent` représente aucune image.

Exécuter une méthode qui permet d'écouter le composant `TextView`.

- Dans la méthode qui permet d'écouter le composant `TextView`, définir un écouteur pour le composant `TextView` et le traitement à effectuer est de lancer l'activité qui permet d'afficher la page web avec une intention explicite. Vous avez à fournir à cette activité les données extras suivantes : le nom du conifère (`String`) et la page web (`String`). L'activité lancée n'a pas à fournir ou à annoncer quoi que soit à son retour. Voir pages 22 et 23 du document 14.
- Vous devez redéfinir les méthodes de sauvegarde (`onSaveInstanceState`) et de récupération (`onRestoreInstanceState`) du cycle de vie de l'activité pour ne pas perdre le conifère identifié lors du changement dans l'orientation de l'écran. Voir page 7 du document 9.

Vous devez sauvegarder les informations qui ont été mises (texte et image) dans le champ `TextView`, le nom du conifère et la page web. Vous devez récupérer toutes les informations sauvegardées, mettre à jour le champ `TextView` et si le texte du composant `TextView` n'est pas vide, exécuter la méthode qui permet d'écouter le composant `TextView`.





## Deuxième écran de l'activité qui permet d'identifier un conifère.

Cet écran est géré par un autre layout et une autre classe `.java`.

Ajouter cette activité dans le fichier manifeste `AndroidManifest.xml`. Voir page 7 du document 14.

Faire un copier/coller du contenu de la classe `MainActivity` dans cette classe et apporter les modifications nécessaires :

- Modifier le nom de la classe pour la nouvelle classe.
- Dans la méthode `onCreate`, modifier le layout du `setContentView`.
- Dans le code de la variable qui écoute la liste du drawer, vous devez lancer l'activité qui a été sélectionnée **à partir de cette classe** et avec une intention explicite. Vous n'avez pas à faire circuler des données à ces activités et ces activités n'ont pas à vous fournir ou à vous annoncer quoi que soit à leur retour. Juste avant de lancer l'activité en question, ajouter seulement un flag à l'intent (`Intent.FLAG_ACTIVITY_CLEAR_TOP`) pour dépiler les activités de la pile avant de lancer l'activité. Voir page 13 du document 14 (ici, juste un flag). Fermer le `drawerLayout` après une sélection.
- Dans la méthode `onOptionsItemSelected`, ajouter un case dans le switch pour l'action à effectuer lorsque l'utilisateur cliquera sur l'icône accueil. L'action est de dépiler les activités de la pile jusqu'à la première. Voir page 13 du document 14.
- Dans la méthode qui permet d'afficher une boîte de dialogue pour l'action aide de la barre d'actions, vous devez apporter les modifications nécessaires : le nom de l'activité et le message est la string `aide_choix_conifere`.

- Vous devez définir les constantes suivantes dans le fichier source de cette activité. Ces constantes vous seront très utiles :

```
// Ressources array des différents textes et images des items de la liste
// pour aider à l'identification.

private static final int[] TAB_AIGUILLES_CONIFERES = {
    R.array.tab_aiguilles_une,
    R.array.tab_aiguilles_faisceaux,
    R.array.tab_ecailles };

private static final int[] TAB_IMAGES_AIGUILLES_CONIFERES = {
    R.array.tab_images_aiguilles_une,
    R.array.tab_images_aiguilles_faisceaux,
    R.array.tab_images_ecailles };

// Ressources array des différents noms, images et pages web des conifères.

private static final int[] TAB_CONIFERES = {
    R.array.tab_coniferes_aiguilles_une,
    R.array.tab_coniferes_aiguilles_faisceaux,
    R.array.tab_coniferes_ecailles };

private static final int[] TAB_IMAGES_CONIFERES = {
    R.array.tab_images_coniferes_aiguilles_une,
    R.array.tab_images_coniferes_aiguilles_faisceaux,
    R.array.tab_images_coniferes_ecailles };

private static final int[] TAB_WEB_CONIFERES = {
    R.array.tab_web_coniferes_aiguilles_une,
    R.array.tab_web_coniferes_aiguilles_faisceaux,
    R.array.tab_web_coniferes_ecailles };
```

- Dans la méthode onCreate, après l'appel à la méthode setContentView et avant l'appel qui crée le drawer, effectuer les traitements nécessaires pour cette activité.

Vous devez accéder aux ressources. Voir page 1 du document 7.

Vous devez récupérer l'objet Intent de l'activité précédente. Vous devez récupérer la donnée extra de l'activité précédente (int, position dans la liste précédente). Cette position vous sera utile. Voir en haut de la page 34 du document 14.

- Dans la méthode `onCreate`, après l'appel à la méthode qui crée le drawer, effectuer les traitements nécessaires pour cette activité.

Récupérer les composants du layout pour le titre et le bouton Retour.

Récupérer le tableau des choix de l'activité précédente (tableau `tab_choix_aiguilles` du fichier `arrays.xml`). Voir en bas de la page 4 du document 7.

Modifier le texte du titre pour celui qui se trouve dans le tableau des choix de l'activité précédente à la position reçue de l'activité précédente.

Vous devez gérer l'écoute du bouton Retour. Le traitement à effectuer est de tout simplement terminer l'activité courante.

Exécuter une méthode qui permet d'afficher et d'écouter la liste des choix.

- Dans la méthode qui crée le drawer, ajouter des instructions pour ajouter le sous-titre juste avant celle qui ajoute la flèche de remontée.

Récupérer le tableau des choix des sous-titres de l'activité précédente (tableau `tab_choix_sous_titres` du fichier `arrays.xml`). Voir en bas de la page 4 du document 7.

Ajouter le sous-titre pour celui qui se trouve dans le tableau des choix des sous-titres de l'activité précédente à la position reçue de l'activité précédente.

- Dans la méthode qui affiche et écoute la liste des choix, vous allez faire la gestion du composant `RecyclerView` (pages 48 à 56 du document 12). Lisez ce qui suit avant d'écrire le code de la méthode.

Ici, le layout de chaque item de la liste est un `TextView` qui contiendra éventuellement une image et un texte. Ce fichier vous est fourni et se nomme `item_liste_conifere.xml`.

Un objet de type `Conifere` contiendra les informations d'un item de la liste. Le fichier `Conifere.java` contenant la classe `Conifere` vous est fourni. Vous aurez probablement besoin de modifier le chemin du package.

L'adaptateur à fournir à la liste sera un tableau d'objets `Conifere`. Le fichier `ConifereAdapterRV.java` contenant la classe `ConiferAdapterRV` pour l'adaptateur qui gère les items de la liste vous est fourni. Vous aurez probablement besoin de modifier le chemin du package.

Pour le code de la méthode, inspirez-vous de l'exemple de la page 54 du document 12. Attention ! Ne pas exécuter la méthode `setContentView` (déjà fait avant).

- Vous devez définir une méthode qui permet de créer la liste des conifères que l'adaptateur a besoin. Inspirez-vous de l'exemple de la page 41 du document 12.

Les deux tableaux de `String` pour les conifères sont récupérés ici (texte et image de chaque item de la liste).

Récupérer le tableau qui contient le texte de chaque item de la liste. Voir en bas de la page 4 du document 7. L'identifiant à utiliser se trouve dans le tableau `TAB_AIGUILLES_CONIFERES` à la position reçue de l'activité précédente.

Récupérer le tableau qui contient l'image de chaque item de la liste. Voir en bas de la page 4 du document 7. L'identifiant à utiliser se trouve dans le tableau `TAB_IMAGES_AIGUILLES_CONIFERES` à la position reçue de l'activité précédente.

- Dans la méthode qui écoute la sélection d'un item de la liste, il ne faut pas confondre la position de l'item sélectionné de cette liste avec la position reçue de l'activité précédente (celle de la liste de l'écran précédent).

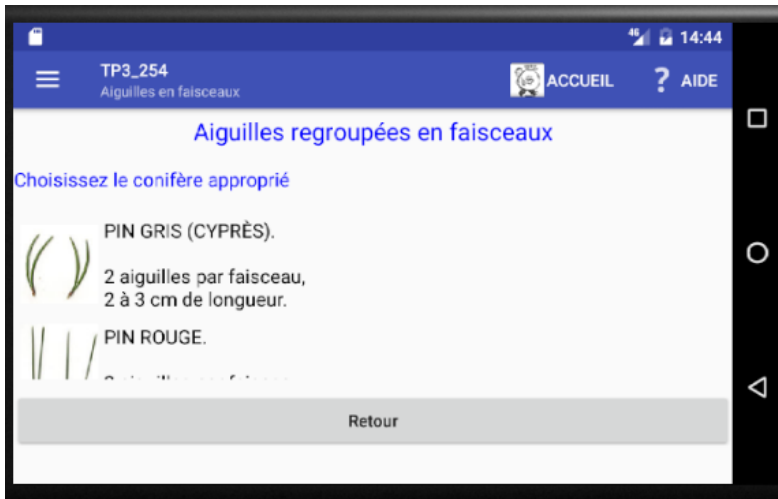
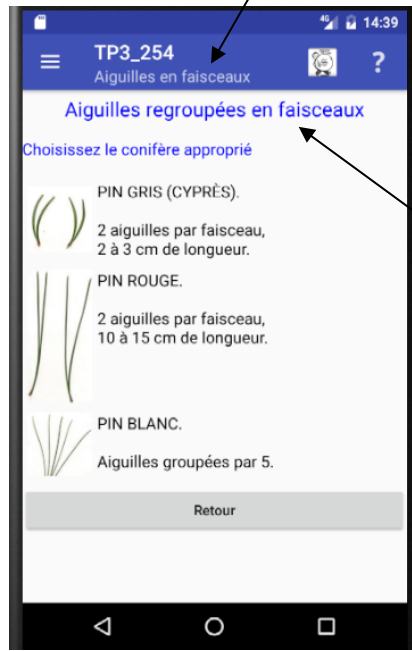
Les trois tableaux de `String` pour les noms, les images et les pages web des conifères sont récupérés ici.

Récupérer le tableau qui contient le nom du conifère de chaque item de la liste. Voir en bas de la page 4 du document 7. L'identifiant à utiliser se trouve dans le tableau `TAB_CONIFERES` à la position reçue de l'activité précédente.

Récupérer le tableau qui contient l'image du conifère de chaque item de la liste. Voir en bas de la page 4 du document 7. L'identifiant à utiliser se trouve dans le tableau `TAB_IMAGES_CONIFERES` à la position reçue de l'activité précédente.

Récupérer le tableau qui contient la page web du conifère de chaque item de la liste. Voir en bas de la page 4 du document 7. L'identifiant à utiliser se trouve dans le tableau `TAB_WEB_CONIFERES` à la position reçue de l'activité précédente.

Vous devez rendre la main à l'activité précédente avec le résultat. Voir en haut de la page 35 du document 14. Ici, vous devez ajouter des données extras à l'objet `intent` avant de rendre la main à l'activité précédente. Les données extras à ajouter sont le nom du conifère (celui à la position reçue en paramètre), l'image du conifère (celle à la position reçue en paramètre convertit en `drawable`) et la page web du conifère (celle à la position reçue en paramètre).



**Troisième écran de l'activité qui permet d'identifier un conifère. Cet écran apparaît lorsque l'utilisateur clique sur le résultat du conifère du premier écran qui permet d'identifier un conifère.**

Cet écran est géré par un autre layout et une autre classe `.java`.

Ajouter cette activité dans le fichier manifeste `AndroidManifest.xml`. Voir page 7 du document 14. Ajouter également dans le fichier manifeste, la permission pour accéder à Internet avant la balise `application`. Voir en haut de la page 4 du document 15.

Faire un copier/coller du contenu de la classe `MainActivity` dans cette classe et apporter les modifications nécessaires :

- Modifier le nom de la classe pour la nouvelle classe.
- Dans la méthode `onCreate`, modifier le layout du `setContentView`.
- Dans le code de la variable qui écoute la liste du drawer, vous devez lancer l'activité qui a été sélectionnée **à partir de cette classe** et avec une intention explicite. Vous n'avez pas à faire circuler des données à ces activités et ces activités n'ont pas à vous fournir ou à vous annoncer quoi que soit à leur retour. Juste avant de lancer l'activité en question, ajouter seulement un flag à l'intent (`Intent.FLAG_ACTIVITY_CLEAR_TOP`) pour dépiler les activités de la pile avant de lancer l'activité. Voir page 13 du document 14 (ici, juste un flag). Fermer le `drawerLayout` après une sélection.
- Dans la méthode `onOptionsItemSelected`, ajouter un case dans le switch pour l'action à effectuer lorsque l'utilisateur cliquera sur l'icône accueil. L'action est de dépiler les activités de la pile jusqu'à la première. Voir page 13 du document 14.
- Dans la méthode qui permet d'afficher une boîte de dialogue pour l'action aide de la barre d'actions, vous devez apporter les modifications nécessaires : le nom de l'activité et le message est la string `aide_web_conifere`.
- Dans la méthode `onCreate`, après l'appel à la méthode `setContentView` et avant l'appel qui crée le drawer, effectuer les traitements nécessaires pour cette activité.

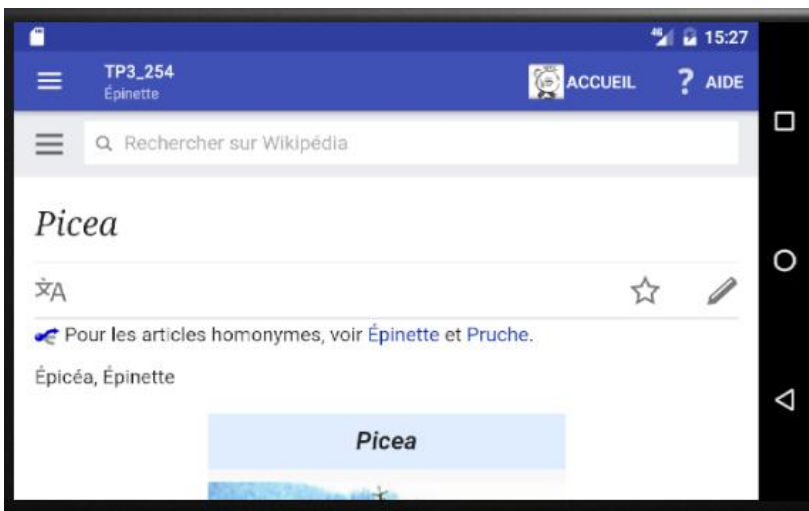
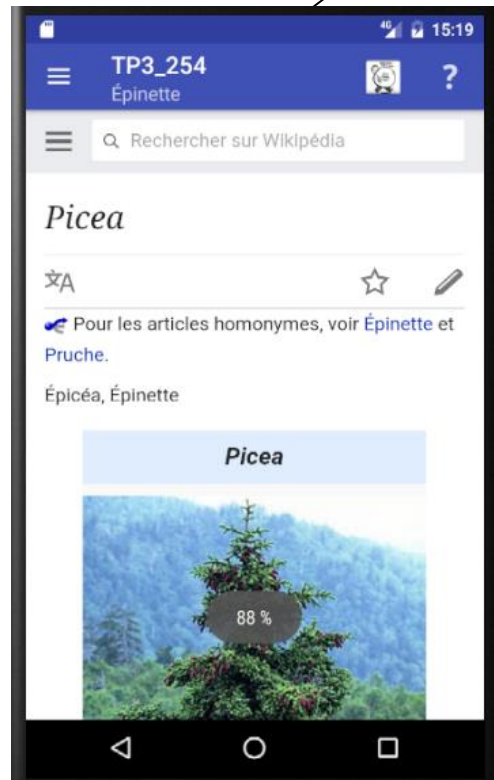
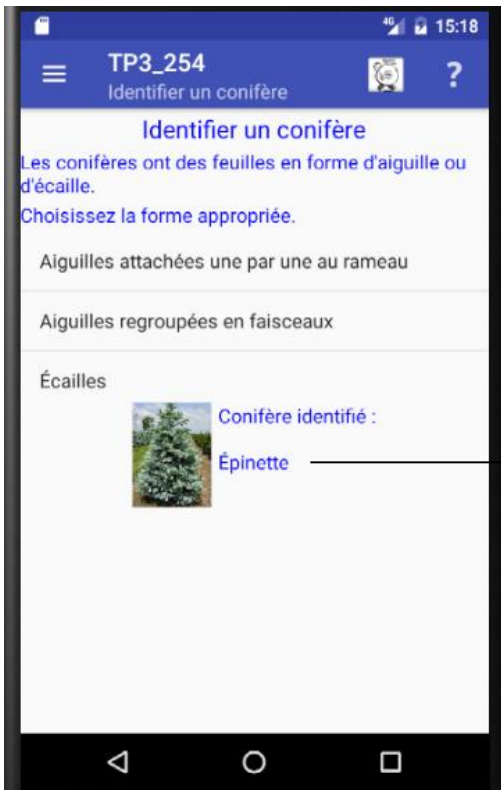
Vous devez accéder aux ressources. Voir page 1 du document 7.

Vous devez récupérer l'objet `Intent` de l'activité précédente. Vous devez récupérer les deux données extras de type `String` de l'activité précédente (nom et page web du conifère). Voir en haut de la page 34 du document 14.

- Dans la méthode qui crée le drawer, ajouter une instruction juste avant celle qui ajoute la flèche de remontée pour ajouter le sous-titre (le nom du conifère) à la barre d'actions. Voir en haut de la page 5 du document 13.

- Dans la méthode `onCreate`, après l'appel à la méthode qui crée le drawer, effectuer les traitements nécessaires pour cette activité.

Récupérer le composant `WebView` du layout et gérer l'affichage de la page web comme indiqué sur les pages 5 et 6 du document 15.





# JEUX D'ESSAI DU TRAVAIL PRATIQUE 3 (420-254-MO)

## ÉCRAN D'ACCUEIL DE L'APPLICATION

- Vérifiez que l'écran de départ contient ce qu'il faut.
- Changez l'orientation de l'écran pour paysage et vérifiez l'affichage. Changez à nouveau l'orientation de l'écran pour portrait et vérifiez l'affichage.
- Vérifier l'action pour afficher l'aide.
- Cliquez sur le choix **Calculer un prêt** et vérifiez que l'écran change pour le **premier** écran de cette activité. Cliquez sur l'icône pour revenir à l'écran d'accueil.
- Cliquez sur le choix **Identifier un conifère** et vérifiez que l'écran change pour le **premier** écran de cette activité. Cliquez sur l'icône pour revenir à l'écran d'accueil.
- Cliquez à nouveau sur le choix **Calculer un prêt** pour tester cette activité.

## ACTIVITÉ 1 : CALCULER UN PRÊT

### Premier écran de l'activité qui permet de calculer un prêt.

- Vérifiez que le premier écran de cette activité contient ce qu'il faut.
- Vérifiez le sous-titre de la barre d'actions.
- Vérifier l'action pour afficher l'aide.
- Vérifier le menu glissant à gauche. Revenir à cette activité après.
- N'entrez aucune donnée et cliquez sur le bouton Calculer. Le message d'avertissement devrait apparaître.
- Entrez un montant, mais n'entrez pas de taux d'intérêt et cliquez sur le bouton Calculer. Le message d'avertissement devrait apparaître. Entrez un taux d'intérêt, mais n'entrez pas de montant et cliquez sur le bouton Calculer. Le message d'avertissement devrait apparaître.
- Entrez la valeur 0 pour le montant et la valeur 0 pour le taux d'intérêt et cliquez sur le bouton Calculer. Le message d'avertissement devrait apparaître.
- Entrez la valeur 0 pour le montant et la valeur 5 pour le taux d'intérêt et cliquez sur le bouton Calculer. Le message d'avertissement devrait apparaître.
- Entrez la valeur 5 pour le montant et la valeur 0 pour le taux d'intérêt et cliquez sur le bouton Calculer. Le message d'avertissement devrait apparaître.

- Vérifiez que la liste déroulante pour la durée fonctionne. Choisir une valeur dans la liste, autre que la première, et vérifiez que celle-ci apparaît.
- Entrez une valeur pour le montant et aussi pour le taux d'intérêt. Cliquez sur le bouton Effacer et vérifiez que le menu contextuel s'affiche. Choisir l'option qui permet d'effacer le montant et vérifiez que seulement le montant est effacé.
- Entrez une valeur pour le montant. Cliquez sur le bouton Effacer et vérifiez que le menu contextuel s'affiche. Choisir l'option qui permet d'effacer le l'intérêt et vérifiez que seulement le taux d'intérêt est effacé.
- Entrez une valeur pour le taux d'intérêt. Cliquez sur le bouton Effacer et vérifiez que le menu contextuel s'affiche. Choisir l'option qui permet d'effacer le montant et l'intérêt et vérifiez que le montant et que le taux d'intérêt sont effacés.
- Entrez la valeur 10000 pour le montant, la valeur 5 pour le taux d'intérêt et la valeur 36 mois pour la durée.
- Changer l'orientation de l'écran pour paysage et vérifiez l'affichage. Changez à nouveau l'orientation de l'écran pour portrait et vérifiez l'affichage.
- Cliquez sur le bouton Calculer et vérifiez que l'écran change pour le **deuxième** écran de cette activité.

### Deuxième écran de l'activité qui permet de calculer un prêt.

- Vérifiez que le deuxième écran de cette activité contient ce qu'il faut.
- Vérifiez le sous-titre de la barre d'actions.
- Vérifier l'action pour afficher l'aide.
- Vérifier le menu glissant à gauche. Revenir à cette activité après, avec les mêmes valeurs.
- Vérifier l'action pour revenir à l'écran d'accueil. Revenir à cette activité après, avec les mêmes valeurs.
- Vérifiez les résultats. Vérifiez 2 chiffres après le point, sauf pour la durée.



- Changer l'orientation de l'écran pour paysage et vérifiez l'affichage. Le bouton Retour ne doit pas disparaître de l'écran. Changez à nouveau l'orientation de l'écran pour portrait et vérifiez l'affichage.
- Cliquez sur le bouton Retour et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité.
- Entrez la valeur 5000 pour le montant, la valeur 14.5 pour le taux d'intérêt et la valeur 12 mois pour la durée. Cliquez sur le bouton Calculer et vérifiez que l'écran change pour le deuxième écran de cette activité. Vérifiez les résultats.



- Cliquez sur le bouton Retour et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité.
- Entrez la valeur 55000 pour le montant, la valeur 4.5 pour le taux d'intérêt et la valeur 84 mois pour la durée. Cliquez sur le bouton Calculer et vérifiez que l'écran change pour le deuxième écran de cette activité. Vérifiez les résultats.



- Cliquez sur le choix **Identifier un conifère** pour tester cette activité.

## ACTIVITÉ 2 : IDENTIFIER UN CONIFÈRE

### Premier écran de l'activité qui permet d'identifier un conifère.

- Vérifiez que le premier écran de cette activité contient ce qu'il faut.
- Vérifiez le sous-titre de la barre d'actions.
- Vérifier l'action pour afficher l'aide.
- Vérifier le menu glissant à gauche. Revenir à cette activité après.
- Vérifier l'action pour revenir à l'écran d'accueil. Revenir à cette activité après.
- Changer l'orientation de l'écran pour paysage et vérifiez l'affichage. Changez à nouveau l'orientation de l'écran pour portrait et vérifiez l'affichage.
- Cliquez sur le choix **Aiguilles attachées une par une au rameau** et vérifiez que l'écran change pour le **deuxième** écran de cette activité.

### Deuxième et troisième écran de l'activité qui permet d'identifier un conifère.

- Vérifiez que le deuxième écran de cette activité contient ce qu'il faut.
- Vérifiez le sous-titre de la barre d'actions.
- Vérifiez que vous avez les bons renseignements dans la liste des choix pour le choix du conifère.
- Vérifier l'action pour afficher l'aide.
- Vérifier le menu glissant à gauche. Revenir à cette activité après, avec les mêmes valeurs.
- Vérifier l'action pour revenir à l'écran d'accueil. Revenir à cette activité après, avec les mêmes valeurs.
- Cliquez sur le bouton Retour et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité. Aucun résultat ne devrait apparaître, car on n'a fait aucun choix.
- Cliquez sur le choix **Aiguilles regroupées en faisceaux** et vérifiez que l'écran change pour le deuxième écran de cette activité.
- Vérifiez le sous-titre de la barre d'actions et le titre de cet écran.
- Vérifiez que vous avez les bons renseignements dans la liste des choix pour le choix du conifère.

- Changer l'orientation de l'écran pour paysage et vérifiez l'affichage. Le bouton Retour ne doit pas disparaître de l'écran. Changez à nouveau l'orientation de l'écran pour portrait et vérifiez l'affichage.
- Cliquez sur le bouton Retour et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité. Aucun résultat ne devrait apparaître, car on n'a fait aucun choix.
- Cliquez sur le choix **Écailles** et vérifiez que l'écran change pour le deuxième écran de cette activité.
- Vérifiez le sous-titre de la barre d'actions et le titre de cet écran.
- Vérifiez que vous avez les bons renseignements dans la liste des choix pour le choix du conifère.
- Cliquez sur le bouton Retour et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité. Aucun résultat ne devrait apparaître, car on n'a fait aucun choix.
- Cliquez sur le choix **Aiguilles attachées une par une au rameau** et vérifiez que l'écran change pour le deuxième écran de cette activité.
- Choisissez l'**épinette** et vérifiez qu'il revient sur l'écran précédent avec le résultat.



- Changer l'orientation de l'écran pour paysage et vérifiez l'affichage. Le résultat devrait rester sur l'écran. Changez à nouveau l'orientation de l'écran pour portrait et vérifiez l'affichage.
- Cliquez sur le résultat **Épinette** et vérifiez que l'écran change pour le troisième écran de cette activité.
- Vérifiez que le troisième écran de cette activité charge la page web pour l'épinette.
- Vérifiez que le sous-titre de la barre d'actions contient le nom du conifère.

- Vérifier l'action pour afficher l'aide.
- Vérifier le menu glissant à gauche. Revenir à cette activité après, avec les mêmes valeurs.
- Vérifier l'action pour revenir à l'écran d'accueil. Revenir à cette activité après, avec les mêmes valeurs.
- Cliquez sur le bouton Back et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité avec le même résultat.
- Cliquez sur le choix **Aiguilles attachées une par une au rameau** et vérifiez que l'écran change pour le deuxième écran de cette activité.
- Choisissez le **sapin baumier** et vérifiez qu'il revient sur l'écran précédent avec le résultat.
- Cliquez sur le résultat **Sapin baumier** et vérifiez que l'écran change pour le troisième écran de cette activité. Vérifiez que le troisième écran de cette activité charge la bonne page web et que le sous-titre de la barre d'actions contient le bon nom du conifère.
- Cliquez sur le bouton Back et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité avec le même résultat.
- Cliquez sur le choix **Aiguilles regroupées en faisceaux** et vérifiez que l'écran change pour le deuxième écran de cette activité.
- Choisissez le **pin gris** et vérifiez qu'il revient sur l'écran précédent avec le résultat.
- Cliquez sur le résultat **Pin gris (cyprès)** et vérifiez que l'écran change pour le troisième écran de cette activité. Vérifiez que le troisième écran de cette activité charge la bonne page web et que le sous-titre de la barre d'actions contient le bon nom du conifère.
- Cliquez sur le bouton Back et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité avec le même résultat.
- Cliquez sur le choix **Aiguilles regroupées en faisceaux** et vérifiez que l'écran change pour le deuxième écran de cette activité.
- Choisissez le **pin rouge** et vérifiez qu'il revient sur l'écran précédent avec le résultat.
- Cliquez sur le résultat **Pin rouge** et vérifiez que l'écran change pour le troisième écran de cette activité. Vérifiez que le troisième écran de cette activité charge la bonne page web et que le sous-titre de la barre d'actions contient le bon nom du conifère.
- Cliquez sur le bouton Back et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité avec le même résultat.

- Cliquez sur le choix **Aiguilles regroupées en faisceaux** et vérifiez que l'écran change pour le deuxième écran de cette activité.
- Choisissez le **pin blanc** et vérifiez qu'il revient sur l'écran précédent avec le résultat.
- Cliquez sur le résultat **Pin blanc** et vérifiez que l'écran change pour le troisième écran de cette activité. Vérifiez que le troisième écran de cette activité charge la bonne page web et que le sous-titre de la barre d'actions contient le bon nom du conifère.
- Cliquez sur le bouton Back et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité avec le même résultat.
- Cliquez sur le choix **Aiguilles regroupées en faisceaux** et vérifiez que l'écran change pour le deuxième écran de cette activité.
- Choisissez le **mélèze** et vérifiez qu'il revient sur l'écran précédent avec le résultat.
- Cliquez sur le résultat **Mélèze** et vérifiez que l'écran change pour le troisième écran de cette activité. Vérifiez que le troisième écran de cette activité charge la bonne page web et que le sous-titre de la barre d'actions contient le bon nom du conifère.
- Cliquez sur le bouton Back et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité avec le même résultat.
- Cliquez sur le choix **Écailles** et vérifiez que l'écran change pour le deuxième écran de cette activité.
- Choisissez le **cèdre** et vérifiez qu'il revient sur l'écran précédent avec le résultat.
- Cliquez sur le résultat **Cèdre (Thuya)** et vérifiez que l'écran change pour le troisième écran de cette activité. Vérifiez que le troisième écran de cette activité charge la bonne page web et que le sous-titre de la barre d'actions contient le bon nom du conifère.
- Cliquez sur le bouton Back et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité avec le même résultat.
- Cliquez sur le choix **Écailles** et vérifiez que l'écran change pour le deuxième écran de cette activité.
- Choisissez le **genévrier** et vérifiez qu'il revient sur l'écran précédent avec le résultat.
- Cliquez sur le résultat **Genévrier** et vérifiez que l'écran change pour le troisième écran de cette activité. Vérifiez que le troisième écran de cette activité charge la bonne page web et que le sous-titre de la barre d'actions contient le bon nom du conifère.
- Cliquez sur le bouton Back et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité avec le même résultat.

- Cliquez sur le choix **Aiguilles regroupées en faisceaux** et vérifiez que l'écran change pour le deuxième écran de cette activité.
- Cliquez sur le bouton Retour et vérifiez qu'il revient sur l'écran précédent, soit le premier écran de cette activité. Aucun résultat ne devrait apparaître, car on n'a fait aucun choix.
- Changez l'orientation de l'écran pour paysage et vérifiez l'affichage. Aucun résultat ne devrait apparaître. Changez à nouveau l'orientation de l'écran pour portrait et vérifiez l'affichage.

### **ACTIVITÉ 3 : NOM DE L'ACTIVITÉ DE VOTRE CRU**

- Faites les tests appropriés pour vérifier les différents résultats. N'oubliez pas l'orientation de l'écran.
- Vérifiez que les écrans contiennent ce qu'il faut.
- Vérifiez le sous-titre de la barre d'actions.
- Vérifier l'action pour afficher l'aide.
- Vérifier le menu glissant à gauche. Revenir à cette activité après.
- Vérifier l'action pour revenir à l'écran d'accueil. Revenir à cette activité après.
- Vérifiez que votre activité contient au moins un composant du document 12, au moins un menu contextuel quelconque (document 13), lance au moins une autre activité (document 14) et contient au moins un truc du document 15 (page web, boîte de dialogue, barre de progression).

**BRAVO !! Vous avez réussi la programmation de votre application Android. 😊**

**BON TRAVAIL ! Soti 😊**