

INFO00306 : Programmation mobile

TP1 : Prise en main d'Android Studio

Table des matières

Table des matières	1
1. Objectif.....	2
1.1. Installation d'Android Studio sous Linux.....	2
2. Prise en main	3
2.1. Première application	3
2.2. Cycle de vie	3
2.1. Internationalisation	6
2.2. Menu	6
2.3. JAVA/XML.....	7
2.3.1. TextView	7
2.3.2. Button.....	8
2.3.3. Image	8
2.4. Mise en page	9

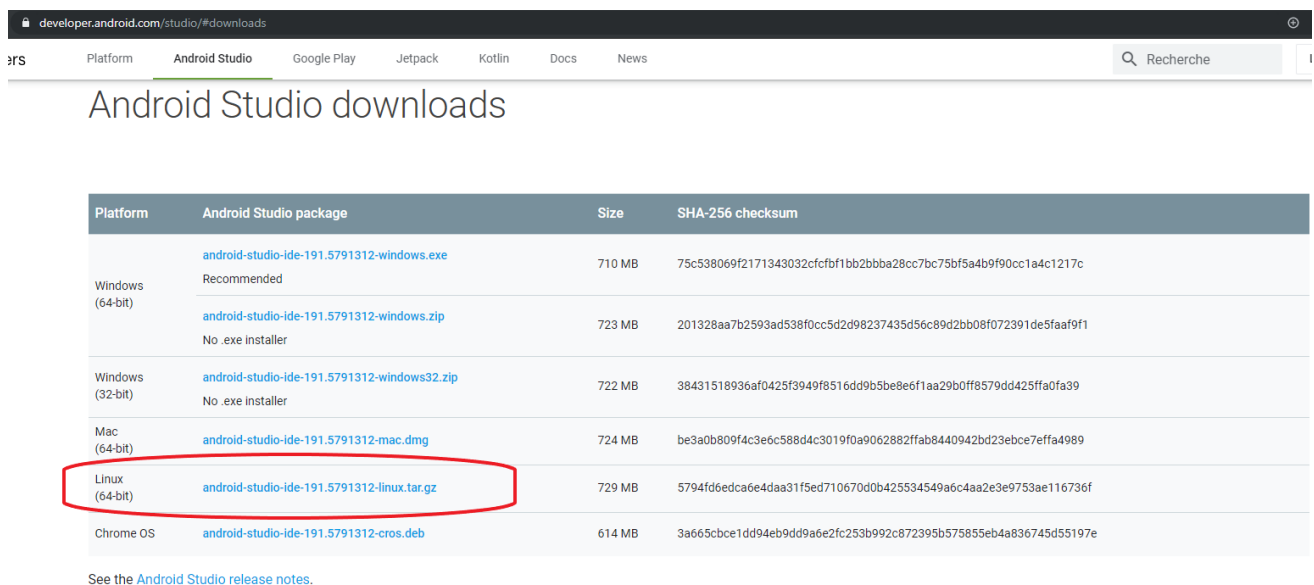
1. Objectif

Les objectifs pédagogiques de ce TP sont les suivants :

- Prise en main de l'environnement de développement
- Prise en main des émulateurs

1.1. Installation d'Android Studio sous Linux

1. Télécharger la dernière version à l'adresse : <https://developer.android.com/studio/#downloads>



Platform	Android Studio package	Size	SHA-256 checksum
Windows (64-bit)	android-studio-ide-191.5791312-windows.exe Recommended	710 MB	75c538069f2171343032cfcfbf1bb2bbba28cc7bc75bf5a4b9f90cc1a4c1217c
	android-studio-ide-191.5791312-windows.zip No .exe installer	723 MB	201328aa7b2593ad538f0cc5d2d98237435d56c89d2bb08f072391de5faaf9f1
Windows (32-bit)	android-studio-ide-191.5791312-windows32.zip No .exe installer	722 MB	38431518936af0425f3949f851dd9b5be8e6f1aa29b0ff8579dd425ffa0fa39
Mac (64-bit)	android-studio-ide-191.5791312-mac.dmg	724 MB	be3a0b809f4c3e6c588d4c3019f0a9062882ffab8440942bd23ebce7effa4989
Linux (64-bit)	android-studio-ide-191.5791312-linux.tar.gz	729 MB	5794fd6edca6e4daa31f5ed710670d0b425534549a6c4aa2e3e9753ae116736f
Chrome OS	android-studio-ide-191.5791312-cros.deb	614 MB	3a665cbce1dd94eb9dd9a6e2fc253b992c872395b575855eb4a836745d55197e

See the [Android Studio release notes](#).

2. Décompresser l'archive :

```
$ tar -xvzf android-studio-ide-191.5791312-linux.tar.gz
```

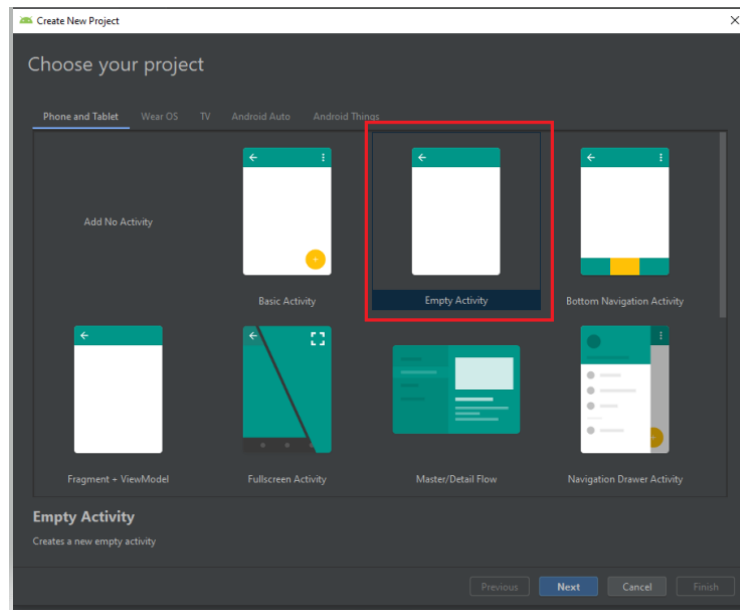
Note : Si votre machine est en 64-bit vous pourriez avoir besoin d'installer des librairies supplémentaires :

```
$ sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 libbz2-1.0:i386
```

3. Installer Android Studio en allant dans le dossier résultant de la décompression puis dans le sous-dossier `bin/` puis en exécutant `studio.sh`

2. Prise en main

2.1. Première application

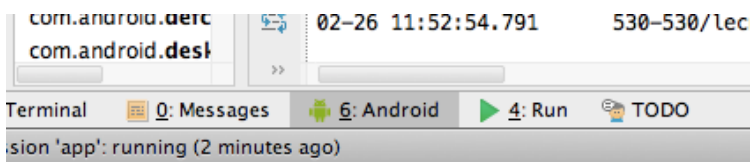


2.2. Cycle de vie

Exercice 1 : Surchargez l'ensemble des méthodes `onXXX()` du cycle de vie afin de visualiser les différents états de ce dernier à l'aide d'un affichage dans le LogCat.

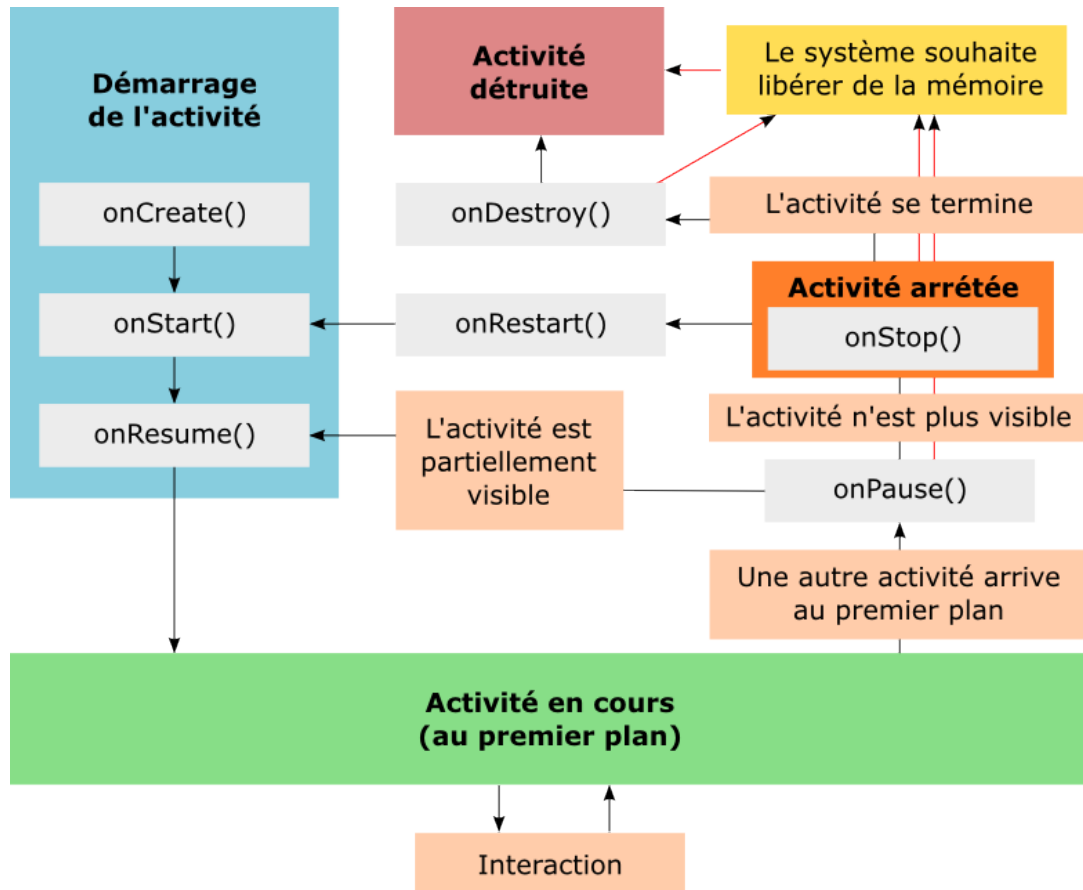
Indices :

1/AndroidStudio : Affichage LogCat

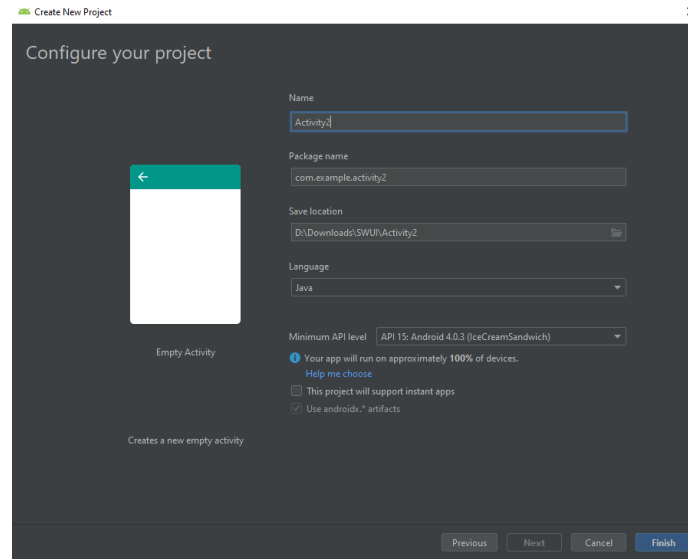


2/ Usage de `Log.i(TAG,MSG)` pour affichage dans la console.

3/ Cycle de vie (sans oublier le `onBackPressed`)



Exercice 2 : Créez une seconde activité et vérifiez sa présence dans le manifest.



Exercice 3 : Ajoutez à l'activité principale un bouton appelant la méthode « openSecondActivity » directement depuis le XML correspondant et un second bouton appelant la méthode « changeText ».

Exercice 4 : Ajoutez au code de votre activité principale la méthode « openSecondActivity » et y affichez un Log dans la console de type : Click : openSecondActivity.

Exercice 5 : Dans cette même méthode, à l'aide d'une intention (Intent), démarrez la seconde activité après le Log. Vérifiez son fonctionnement sur un appareil ou simulateur.

Exercice 6 : Ajoutez au code de votre activité principale la méthode « changeText » qui modifie la valeur du TextView.

Exercice 7 : Modifiez votre code pour que la seconde activité s'ouvre directement à l'ouverture de l'application (Dans fichier manifest). Vérifiez son fonctionnement et annulez cette modification. (Question importante pour bien comprendre le fonctionnement d'une application, ne la passez donc pas...svp).

Exercice 8 : Dans le LogCat, filtrez vos messages provenant uniquement de l'application. Puis, filtrez les ensuite à l'aide d'un TAG que vous nommerez « lifecycle ». Vérifiez le fonctionnement du filtre. Vous ne devriez voir apparaître que vos propres Logs.

Exercice 9 : Mettez en commentaire l'ensemble de vos Logs et ajoutez un Toast pour chaque Log mis en commentaire permettant d'afficher le nom de la méthode onXXX() sur une courte durée.

2.1. Internationalisation

Exercice 10 : Votre activité principale comporte, normalement, déjà un TextView « Hello World ». Recherchez comment ce texte est affecté à ce TextView. Modifiez la ressource afin d'afficher « Bonjour » à la place.

Exercice 11 : Effectuez les démarches pour internationaliser le texte en fonction de la langue de l'appareil. Vérifiez son fonctionnement.

FR : « Bonjour »
EN : « Hello »

Vous pouvez changer la langue de l'appareil ou du simulateur directement dans les préférences.

Indice : <http://developer.android.com/training/basics/supporting-devices/languages.html>

Exercice 12 : Effectuez la même opération pour le nom de l'application sur le bureau (dit Home). Vérifiez que le nom de l'application change sur le bureau quand vous changez la langue.

FR : « INFO0306_TP1 »
EN : « TP1 »

Exercice 13 : Nous allons maintenant internationaliser les Toasts en ajoutant les ressources dans le fichier string.xml. Faite en sorte d'obtenir un affichage pouvant ressembler à ceci :

FR : « Activité créée », « Activité en pause », « Activité reprise » etc.
EN : « Activity created », « Activity on pause », « Activity on resume », etc.

Indice : <http://developer.android.com/training/basics/supporting-devices/languages.html>
Indice : R.string.xxx

2.2. Menu

Exercice 14 : Dans l'activité principale, ajoutez un item au menu principal en ajoutant une balise XML dans le fichier. Vérifiez l'affichage de votre nouvel item.

Exercice 15 : Dans l'activité principale, repérez les deux méthodes permettant :

- De charger le menu
- De répondre au clic sur un item

Constatez qu'un clic sur le nouvel item n'engendre aucune action.

Exercice 16 : Modifiez le code pour que le clic sur ce nouvel item déclenche un Toast internationalisé de courte durée.

Exemple :
FR : « Item cliqué »
EN : « Item clicked »

Exercice 17 : Modifiez le fichier XML pour que les 2 items apparaissent dans la barre d'action de votre activité principale.

Indice : showAsAction

Exercice 18 : Pour chaque item, modifiez le fichier XML pour y faire apparaître une image à la place du texte

Indice : icon

Exercice 19 : Modifiez le fichier XML pour que les 2 items soient inversés.

Indices :

item 2, item1 : au lieu de item 1, item 2

orderInCategory

2.3. JAVA/XML

La mise en page est totalement libre. Positionnez donc librement vos vues.

2.3.1. TextView

Exercice 20 : Dans le XML, ajoutez dans une activité (principale ou secondaire) un *TextView* puis associez son texte à une ressource de type string « mon_texte » (internationalisation à prévoir).

FR : « Coucou »

EN : « Hi »

Exercice 21 : Dans le code, ajoutez à la fin du onCreate() de votre activité, le code permettant de récupérer le TextView.

Indices :

findViewById

cast

Exercice 22 : Ajoutez une ressource de type string « nouveau_texte » (internationalisation) sans pour le moment l'attacher à quoi que ce soit.

FR : « Chargé »

EN : « Loaded »

Exercice 23 : Dans le code, à la fin du onCreate() modifiez le texte de votre TextView de sorte qu'il affiche la chaîne correspondant à « nouveau_texte ». Vérifiez le bon affichage.

Indice : <http://developer.android.com/training/basics/supporting-devices/languages.html>

Indice : setText

Exercice 24 : Dans le code et plus précisément dans le onCreate(), utilisez le Bundle « savedInstanceState » pour conserver la dernière valeur du TextView (qui reprend sinon sa valeur initiale si vous réorientez l'écran après l'avoir modifié). Vérifiez le bon affichage.

Indice : ajouter la fonction override onSaveInstanceState

Indice : savedInstanceState != null

2.3.2. Button

Activité principale

Exercice 25 : Dans le XML, mettez en commentaire l'attribut onclick.

Indice : comme en HTML

Exercice 26 : Si ce n'est pas déjà le cas, ajoutez un identifiant à votre Button.

Exemple : android:id=@+id/button1

Info : le + est indispensable si usage dans le code java ou référence dans le XML.

Exercice 27 : Dans le code, à la fin du onCreate(), récupérez le Button.

Indice : idem que pour TextView.

Exercice 28 : Ajoutez une classe de type View.OnClickListener comme « écouteur » de votre bouton.

Indice : setOnClickListener(new ...)

Exercice 29 : Dans la surcharge de la méthode onClick() de votre écouteur, démarrez la seconde activité.

Indice : idem que pour TextView.

2.3.3. Image

Activité principale

Objectif : Modifiez une image lors d'un clic sur un bouton.

Exercice 30 : Ajoutez une image (libre) à votre projet. (ne prendre en compte les résolution pour le moment).

Exercice 31 : Dans le XML, disposez une ImageView comme bon vous semble ainsi qu'un nouveau Button. Appliquez dès maintenant une image (libre).

Exercice 32 : A la fin du onCreate(), récupérer l'ImageView et le Button.

Attention : Le bouton devra être déclaré comme membre privé de votre classe MainActivity.

Exercice 33 : De la manière qu'il vous plaira (Listener ou onClick), faites en sorte que votre bouton puisse appeler une méthode (vide pour le moment).

Exercice 34 : Ajoutez une seconde image (libre) à votre projet.

Exercice 35 : Dans la méthode appelé par votre bouton, faites en sorte de modifier l'image par cette nouvelle image.

Indice : setImageResource

Indice : R.drawable.xxx

Exercice 36 : A l'aide d'un booléen, faites en sorte que chaque clic puisse inverser l'image affiché.

2.4. Mise en page

Exercice 37 : Dans une nouvelle activité, disposez une mise en page de type LinearLayout verticale et faites en sorte que cette activité soit démarrée depuis la première activité.

Indice : Orientation

Indice : startActivity

Exercice 38 : A ce LinearLayout, ajoutez :

- une couleur de fond
- une marge externe de 5dp
- une marge interne de 5dp

Indice : margin, padding, background

Exercice 39 : Dans le LinearLayout, disposez 3 RelativeLayout disposant chacun d'une couleur de fond différente (+ différente du LinearLayout).

Exercice 40 : Mettez en place les attributs de largeur afin d'obtenir 3 zones occupant toute la largeur du LinearLayout.

Indice : match_parent

Exercice 41 : Mettez l'attribut de la hauteur à 0dp.

Exercice 42 : Ajoutez l'attribut « weight » afin d'obtenir une répartition des zones de la sorte « 20%, 50%, 30% ».

Exercice 43 : Modifiez les différents attributs afin de vous familiariser avec.