



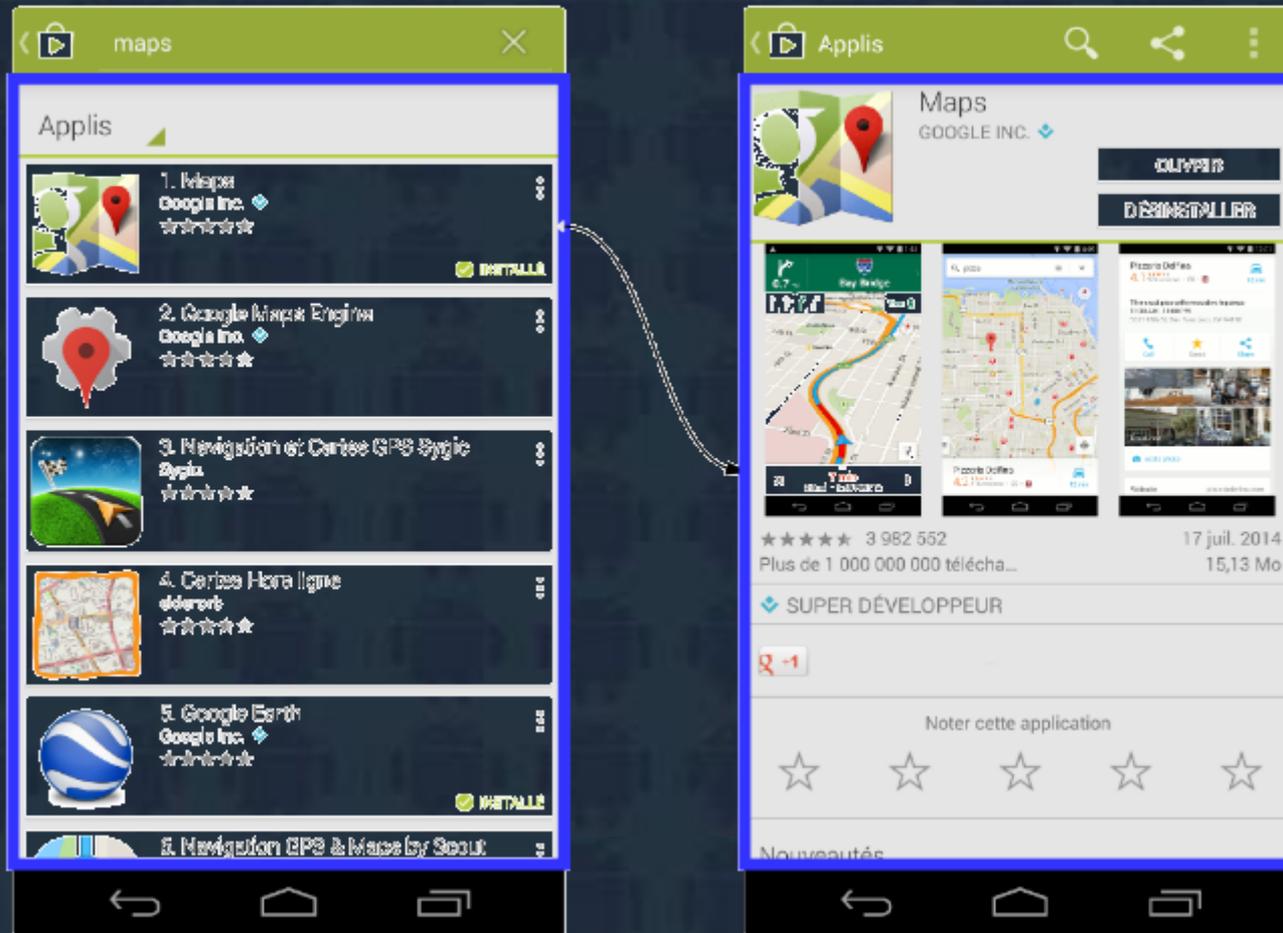
Université Mohamed Boudiaf de M'Sila
Faculté des Mathématique et de l'Informatique
Département d'Informatique

Première Application

Samir Akhrout

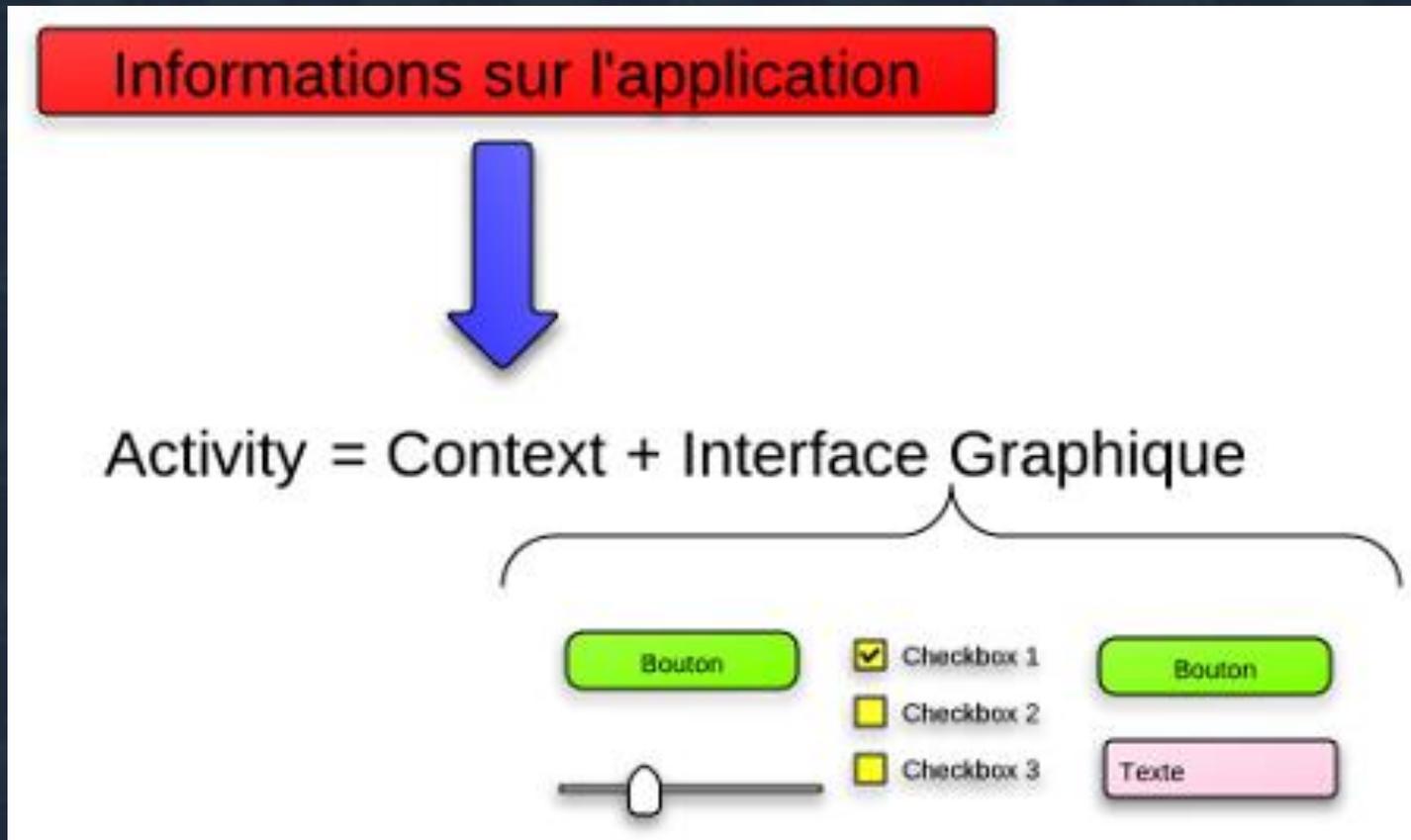


Exemple



En cliquant sur un élément de la liste à gauche, on ouvre une nouvelle activité

L'activité contient des informations sur l'état actuel de l'application : ces informations s'appellent le contexte. Ce contexte constitue un lien avec le système Android ainsi que les autres activités de l'application, comme le montre la figure suivante.



Une activité est constituée du contexte de l'application et d'une seule et unique interface graphique

États d'une activité

Une activité à plusieurs états au cours de sa vie:

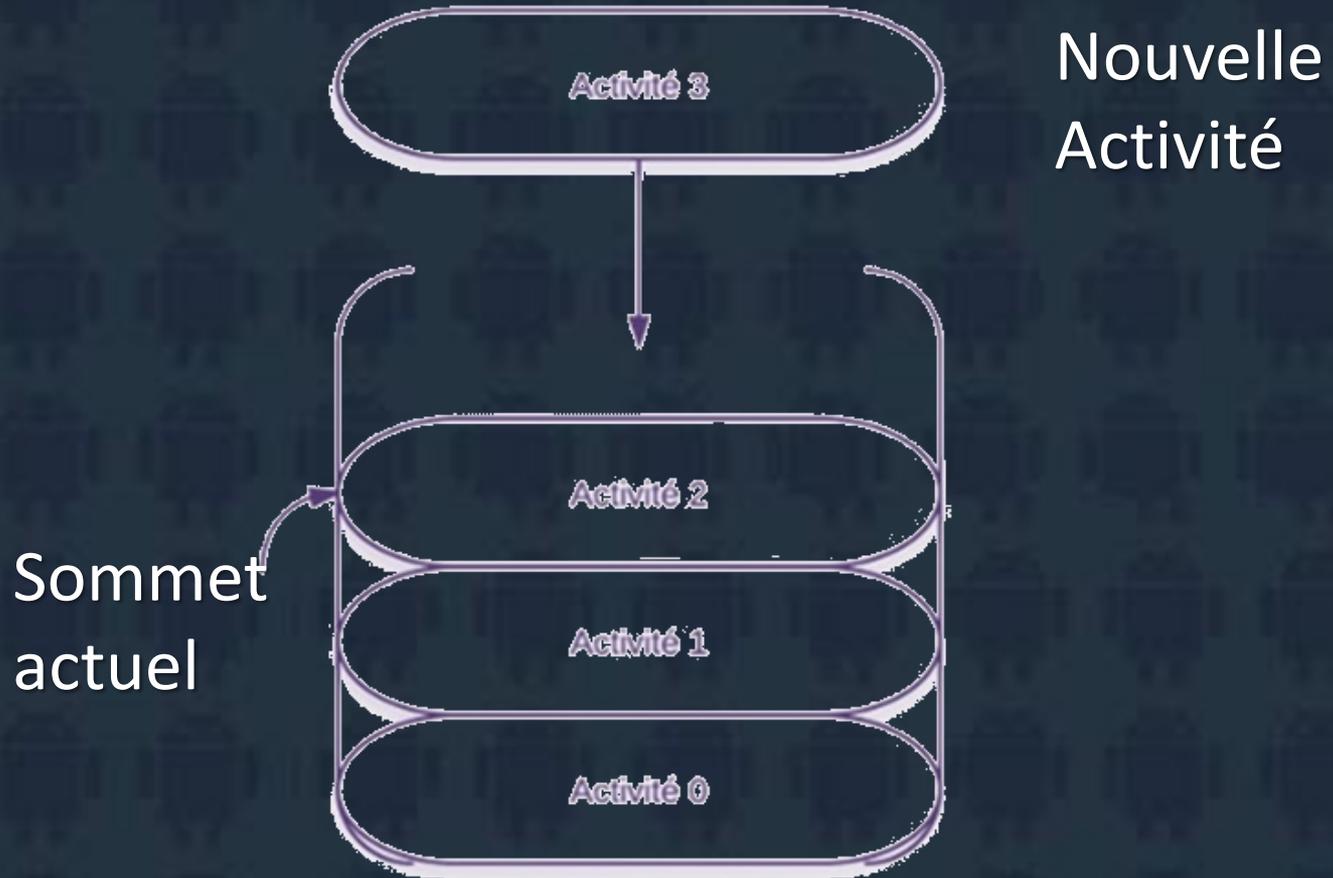
- ❑ un état actif pendant lequel l'utilisateur l'exploite,
- ❑ un état de pause quand l'utilisateur exécute une autre activité ayant une priorité élevée,
- ❑ un état activité suspendue.

Quand une application se lance, elle se met tout en haut de ce qu'on appelle la pile d'activités.

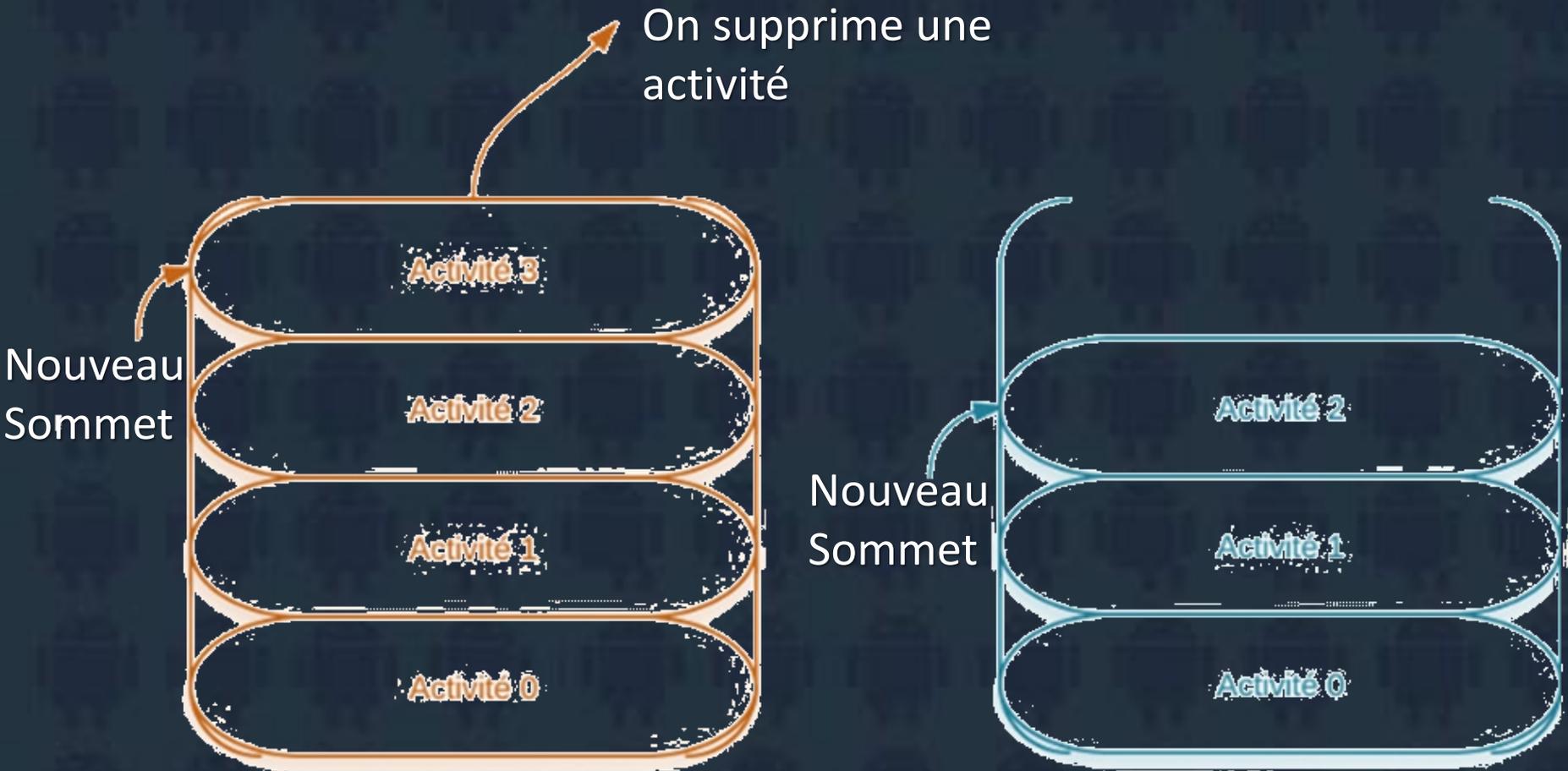
Une pile est une structure de données de type « LIFO », c'est-à-dire qu'il n'est possible d'avoir accès qu'à un seul élément de la pile, le tout premier élément, qui est appelé **sommet**.

La figure suivante illustre le mode de gestion des activités :

La pile des activités



Une activité va être ajoutée



Ce qui est visible est l'interface graphique de l'activité qui se trouve au sommet de la pile.

États d'une activité

État	Visibilité	Description
Active («active» ou «running»)	L'activité est visible en totalité.	Sommet de la pile, application ayant le focus
Suspendue («paused»)	L'activité est partiellement visible à l'écran.	L'application n'a plus le focus, c'est l'application au-dessus qui l'a.
Arrêtée («stopped»)	L'activité est tout simplement invisible pour l'utilisateur, car une autre activité prend toute la place sur l'écran.	L'application n'a évidemment plus le focus, et puisque l'utilisateur ne peut pas la voir, il ne peut pas agir dessus.

Une activité est la composante principale pour une application Android. Elle représente l'implémentation métier dans une application Android, permettant de gérer l'ensemble des vues et ressources.

Une activité peut être avec ou sans interface utilisateur. Il est possible d'avoir plusieurs activités dans le même programme. Elle doit toujours être déclarée dans le fichier **AndroidManifest.xml**.

Une activité n'est pas linéaire, elle est soumise à plusieurs événements. Chaque événement est représenté dans une méthode. La figure suivante illustre le cycle de vie d'une application Android :

- onCreate() : Cette méthode est appelée à la création d'une activité. Elle permet de l'initialiser. C'est ici que l'interface graphique est spécifiée.
- onStart() :
cette méthode est appelée quand l'application est démarrée.
- onResume() : cette méthode est appelée quand l'application passe (ou repasse) en avant plan.
- onPause() : Appelée quand l'application passe en arrièreplan et qu'une autre application se met devant.
- onStop() : Appelée quand l'application n'est plus visible.
- onRestart():
Appelée quand l'application redevient visible.
- onDestroy() :
Appelée quand votre application est fermée par le système à cause d'un manque de ressources, ou par l'utilisateur à l'utilisation d'un finish().

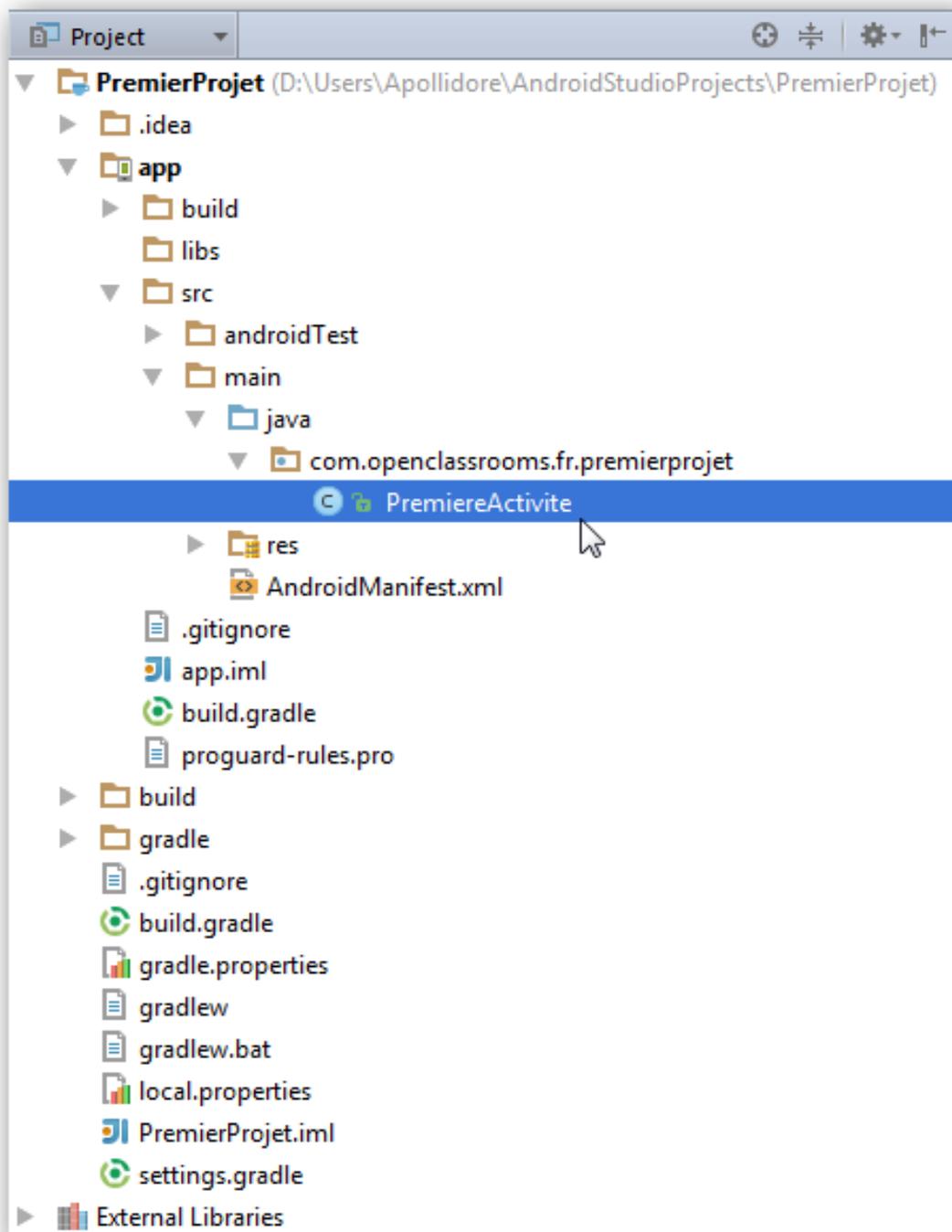
Les activités héritent de la classe [Activity](#). Or, la classe `Activity` hérite de l'interface [Context](#) dont le but est de Représenter tous les composants d'une application. On les trouve dans le package `android.app.Activity`.

Un package est un répertoire qui permet d'organiser notre code source, un récipient dans lequel nous allons mettre nos classes de façon à pouvoir trier votre code et différencier des classes qui auraient le même nom.

Les **vues** (que nos amis anglais appellent **view**), sont ces fameux composants qui viendront se greffer sur notre échafaudage, il s'agit de l'unité de base de l'interface graphique. Leur rôle est de fournir du contenu visuel avec lequel il est éventuellement possible d'interagir.

Les vues héritent de la classe `View`. On les trouve dans le package `android.view.View`.

Vous trouvez
(voir figure sur
classe à laque



CTRL + N.

1. Programmation d'applications

Une application Android est composée de :

- ❑ **Sources Java (ou Kotlin) compilés pour une machine virtuelle**
- ❑ Fichiers appelés **ressources** :
 - ✓ **format XML** : interface, textes. . .
 - ✓ **format PNG** : icônes, images. .
- ❑ **Manifeste = description du contenu du logiciel**
 - ✓ fichiers présents dans l'archive
 - ✓ demandes d'autorisations
 - ✓ signature des fichiers, durée de validité, etc.

1.1. SDK Android et Android Studio

1.1.1. SDK et Studio

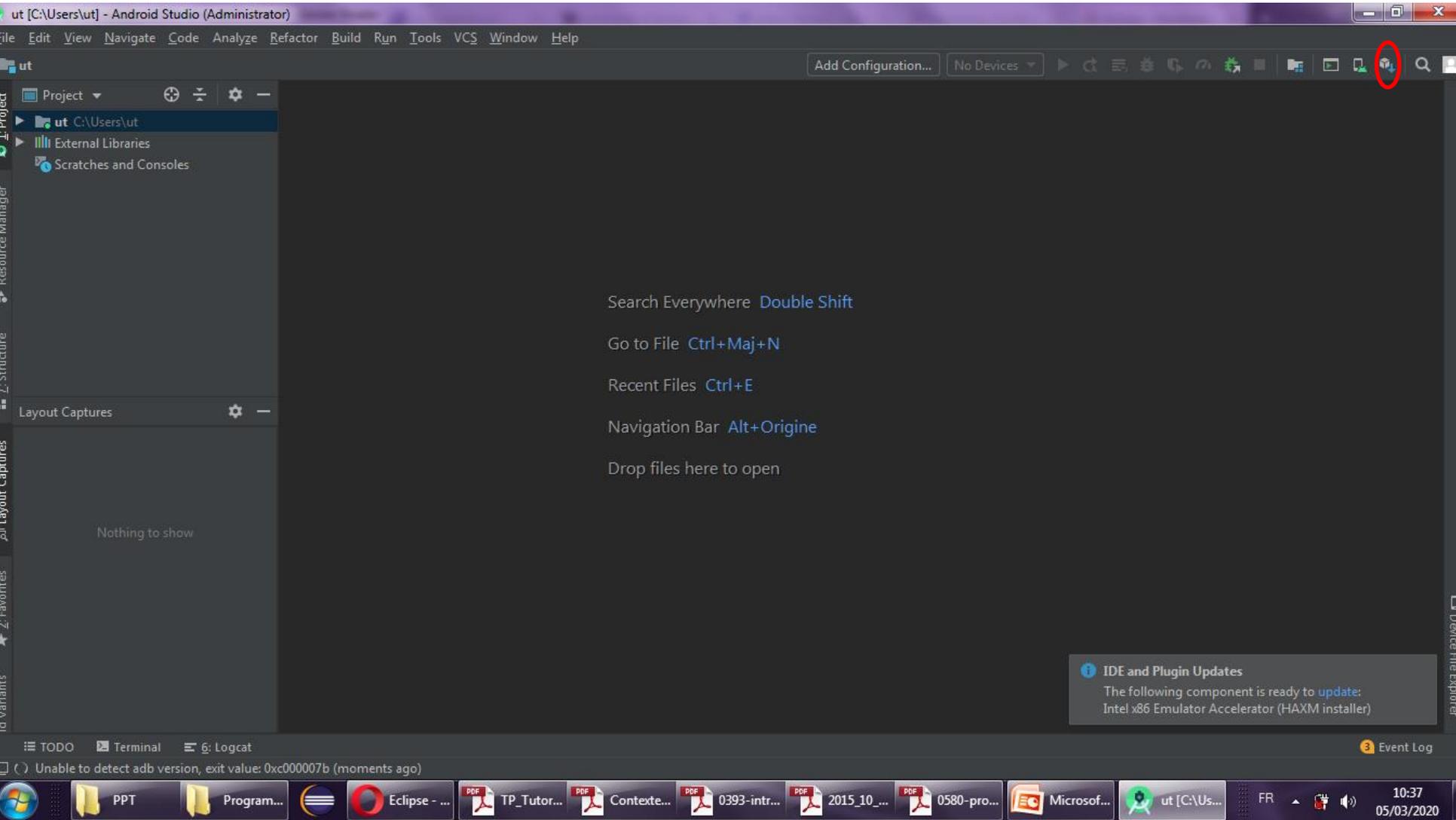
Le SDK contient :

- les librairies Java pour créer des logiciels
- les outils de mise en boîte des logiciels
- *AVD : un émulateur de tablettes pour tester les applications*
- *ADB : un outil de communication avec les vraies tablettes*

Android Studio offre :

- un éditeur de sources et de ressources
- des outils de compilation : *gradle*
- des outils de test et de mise au point

SDK Manager



Gestionnaire de paquets Android

Settings for New Projects

Appearance & Behavior > System Settings > Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: [Edit](#) [Optimize disk space](#)

SDK Platforms SDK Tools SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
<input type="checkbox"/>	Android R Preview	R	1	Not installed
<input checked="" type="checkbox"/>	Android 10.0 (Q)	29	4	Installed
<input type="checkbox"/>	Android 9.0 (Pie)	28	6	Not installed
<input type="checkbox"/>	Android 8.1 (Oreo)	27	3	Not installed
<input type="checkbox"/>	Android 8.0 (Oreo)	26	2	Not installed
<input type="checkbox"/>	Android 7.1.1 (Nougat)	25	3	Partially installed
<input type="checkbox"/>	Android 7.0 (Nougat)	24	2	Not installed
<input type="checkbox"/>	Android 6.0 (Marshmallow)	23	3	Partially installed
<input checked="" type="checkbox"/>	Android 5.1 (Lollipop)	22	2	Installed
<input type="checkbox"/>	Android 5.0 (Lollipop)	21	2	Not installed
<input type="checkbox"/>	Android 4.4W (KitKat Wear)	20	2	Partially installed
<input type="checkbox"/>	Android 4.4 (KitKat)	19	4	Not installed
<input type="checkbox"/>	Android 4.3 (Jelly Bean)	18	3	Partially installed
<input type="checkbox"/>	Android 4.2 (Jelly Bean)	17	3	Partially installed
<input type="checkbox"/>	Android 4.1 (Jelly Bean)	16	5	Partially installed
<input type="checkbox"/>	Android 4.0.3 (IceCreamSand)	15	5	Partially installed
<input type="checkbox"/>	Android 4.0 (IceCreamSand)	14	4	Not installed
<input type="checkbox"/>	Android 3.2 (Honeycomb)	13	1	Not installed
<input type="checkbox"/>	Android 3.1 (Honeycomb)	12	3	Not installed
<input type="checkbox"/>	Android 3.0 (Honeycomb)	11	2	Not installed
<input checked="" type="checkbox"/>	Android 2.3.3 (Gingerbread)	10	2	Installed
<input type="checkbox"/>	Android 2.3 (Gingerbread)	9	2	Not installed
<input checked="" type="checkbox"/>	Android 2.2 (Froyo)	8	3	Installed
<input type="checkbox"/>	Android 2.1 (Eclair)	7	3	Not installed

Looking for updates...

Hide Obsolete Packages Show Package Details

OK Cancel **Apply**

1.2. Choix des éléments du SDK

Le gestionnaire permet de choisir les versions à installer, ex. :

- Android 8.1 (API 27)
- Android 8.0 (API 26)
- Android 7.0 (API 24)
- Android 6 (API 23)
-

Choisir celles qui vous correspondent. Il faut cocher “Show Package Details” dans la figure précédente, puis choisir élément par élément.

Les suivants sont indispensables :

- Android SDK Platform
- Intel x86 Atom_64 System Image

1.3. Dossiers du SDK

Le gestionnaire installe environ 1 Go de fichiers :

- SDK Tools : indispensable, contient le gestionnaire,
- SDK Platform-tools : indispensable, contient adb,
- SDK Platform : indispensable, contient les librairies,
- System images : pour créer des AVD,
- Android Support : divers outils pour créer des applications,
- Exemples et sources.

1.4. Première application

1.4.1. Objectif du TP

ceci est seulement un aperçu rapide des possibilités :

- Création d'une application « *HelloWorld* » avec un *assistant*,
- Tour du propriétaire,
- Exécution de l'application,
- Mise sous forme d'un paquet.

1.4.2. Assistant de création d'application

Android Studio contient un assistant de création d'applications : Voir la figure, diapositive suivante.



Select a Project Template

Phone and Tablet

Wear OS

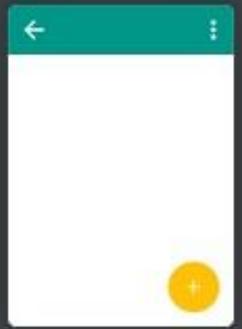
TV

Automotive

Android Things



No Activity



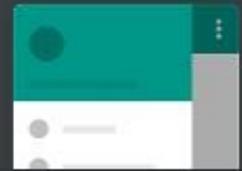
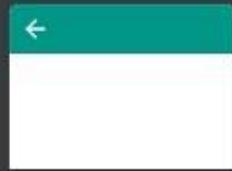
Basic Activity



Empty Activity



Bottom Navigation Activity



Empty Activity

Creates a new empty activity

Previous

Next

Cancel

Finish



Configure Your Project



Empty Activity

Creates a new empty activity

Name

My Application

Package name

com.dam.myapplication

Save location

C:\Users\ut\AndroidStudioProjects\MyApplication3

Language

Java

Minimum SDK

API 16: Android 4.1 (Jelly Bean)

i Your app will run on approximately **99,6%** of devices.
[Help me choose](#)

Use legacy android.support libraries [?](#)

Previous

Next

Cancel

Finish

1.4.3. Choix de la version

Chaque version d'Android, dénotée par son *API level*, ex: 25, apporte des améliorations et supprime des dispositifs obsolètes.

Toute application exige un certain niveau d'API :

- Minimum SDK : il faut au moins cette API car on utilise certaines classes et méthodes absentes des précédentes APIs,

Il y a aussi deux notions à connaître :

- Target SDK : l'application sera testée et marchera correctement jusqu'à ce niveau d'API,

- Compile With : c'est le niveau maximal de fonctionnalités qu'on se limite à employer. Si on fait appel à quelque chose de plus récent que ce niveau, le logiciel ne se compilera pas.

Build your first app

The screenshot shows an IDE window titled "MyFirstApp" with the following structure:

- Project: MyFirstApp
- Package: com.example.myfirstapp
- File: MainActivity.kt

```
1 package com.example.myfirstapp
2
3 import ...
4
5
6 class MainActivity : AppCompatActivity() {
7
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        setContentView(R.layout.activity_main)
11    }
12
13 }
```

The Build Output panel shows the following tasks:

- Transform legacy-support-core-annotations (androidx.legacy:legacy-support-core-annotations:1.0.0-beta01) with AarTransform
- Transform constraintlayout.aar (androidx.constraintlayout:constraintlayout:1.1.3) with AarTransform
- Transform appcompat.aar (androidx.appcompat:appcompat:1.0.0-beta01) with AarTransform
- Transform core-ktx.aar (androidx.core:core-ktx:1.1.0-alpha04) with AarTransform
- Transform vectordrawable-animated.aar (androidx.vectordrawable:vectordrawable-animated:1.0.0-beta01) with AarTransform
- Transform fragment.aar (androidx.fragment:fragment:1.0.0-beta01) with AarTransform
- Task :app:checkDebugManifest
- Task :app:compileDebugRenderscript
- Task :app:generateDebugBuildConfig
- Task :app:prepareLintJar
- Task :app:generateDebugSources

BUILD SUCCESSFUL in 0s
5 actionable tasks: 5 executed

At the bottom, the status bar shows: "Gradle build finished in 618 ms (2 minutes ago)" and "Event Log" with "1:1 LF UTF-8 Context: <no context>".

Le reste du TP est décrit dans le document :
TD2 Création d'un premier projet