TP2 : La première application Android

Dans ce TP nous allons prendre l'exemple d'un bon vieux « Hello World » pour étudier l'environnement de développement *AndroidStudio*.

L'objectif de ce TP est :

- 1. Création d'une application « HelloWorld » avec un assistant,
- 2. Tour du propriétaire,
- 3. Exécution de l'application,

1. Création d'un projet et d'une application "Hello World"

a. Lancez l'envirenment « Android studion » et sélectionnez « Start a new Android Studio Project ».



Figure 1: l'envirenement Android studio

b. Nous arrivons à la création d'une activité (un écran avec une interface graphique). Sélectionnez *Empty Activity* et cliquez *Next*.

>>> Les **activités** sont les éléments constitutifs principaux d'une application. Elles sont normalement associées à une vue graphique faite pour être affichée en plein écran. Pour simplifier, nous pourrons considérer qu'une activité se réduit à la notion de fenêtre.

🕐 Create New Project			×
Select a Proje	ct Template		
Phone and Tablet Wear OS T	V Automotive Android Thing	IS	
No Activity	€ : + Basic Activity	← Empty Activity	Eottom Navigation Activity
¢			
Empty Activity Creates a new empty activity			
		Previous	<u>N</u> ext <u>Cancel</u> Finish

Figure 2: project template

c. Cliquez sur Next puis remplissez les champs comme dans la figure suivante :

Create New Project	_		×		
Configure Your Project					
	<u>N</u> ame				
	MyfirstApplica	tion			
E 100 100 100 100 100 100 100 100 100 10	Package name				
· · · · · · · · · · · · · · · · · · ·	com.example.myfirstapplication				
	Save location				
	C:\Users\mg\Documents\AndroidStudioProjects\MyfirstApplication2				
	<u>L</u> anguage				
	Java	v			
	Minimum SDK	ADI 15: Android 4.0.2 (IsoCroomSandwich)			
Empty Activity	Your app will Help me cho	I run on approximately 100% of devices.			
Creates a new empty activity	Use legacy a	android.support libraries 💿			
		Previous Next Cancel Ein	ish		

Figure 3: configuration du projet

Application name : c'est le nom qui va apparaitre dans la liste des applications dans le Play Store et sur l'appareil.

Package name : il est utilisé comme identifiant de l'application, il permet de considérer différentes versions d'une application comme étant une même application. Il doit être unique parmi tous les packages installés sur le système.

Save location : vous pouvez choisir un emplacement pour votre projet.

Langage : pour choisir le langage de programmation préféré (Java dans notre cas).

Minimum SDK : c'est la version Android la plus ancienne sur laquelle l'application peut tourner. Il faut éviter de remonter trop en arrière, ça réduirait les fonctionnalités que vous pourriez donner à votre application.

>>> l'application sera compilée avec le compilateur le plus récent. Mais pour être installable sur les versions d'OS plus anciennes, elle aura besoin d'être installée avec les éléments utilisés qui n'existaient pas à l'époque.

😟 Andro	oid Platform/API Version Distribution					×
	ANDROID PLATFORM VERSION	API LEVEL		Jelly Bean		
10	Ice Cream Sandwich	15		App Components	Multimedia	
4.0		15	99.8%	Isolated services Memory management	Enhanced RenderScript functionality	
4.1	Jelly Bean	16	55,070	Content providers	Animations	
4.2		17	99,2%	App stack navigation	Activity launch animations Time animator	
43	Jellv Bean	18	98,4%	Multimedia	User Interface	
1.0		10	98.1%	Media codecs Record audio on cue	Improved notifications	
4.4	KITKAT	19		Timed text tracks	Remote views	
5.0	Lollipop	21	94,1%	Gapless playback	More font families	
51	Lollinon	22	92,3%	Camera	Input Framework	
0.1			94.00/	Camera sounds	Multiple input devices Vibrate for input controllers	
6.0		23	04,9%	Connectivity	·	
7.0		24	73,7%	Android Beam over Bluetooth		
7 4		21	66.2%	Wi-Fi P2P service discovery		
7.1		25		Detect metered networks		
8.0	Oreo	26	60,8%	Accessibility		
8.1	Oreo	27	53,5%	Accessibility service APIs Customizable app navigation More accessible widgets		
			39,5%	Copy and Paste		
			· ·	Copy and paste with intents		
9.0		28				
10			8.2%			
10.	Android 10	29	0,270	https://developer.android.com/about	/versions/android-4.1.html	
					ОКС	ancel



Help me choose : Lien d'aide au choix nous permettra de voir l'état des lieux de la fragmentation du marché et les spécificités introduites dans chaque version d'Android.

d. Lorsque vous êtes prêt à créer votre projet, cliquez sur Finish.

>>>Après avoir validé la création du projet, l'éditeur de code s'ouvre en montrant l'arborescence du projet, le code Java de l'activité créée et le XML de son layout. Voilà, la première application Android est faite. Elle est exécutable et affiche le bon vieux message « Hello world ».

>>> Attention ! Lors de la création d'un nouveau projet, le gestionnaire de dépendances (Gradle) peut mettre un certain temps à finir.



Figure 5: l'environnement de l'IDE



Figure 6: Androidmanifest.xml



Figure 7: mainactivity.java



Figure 8: activity main.xml (text view)



Figure 9: activity_main (graph view)

File Edit View Navigate Code Analyze Refactor B	Ald Run Lools VCS Window Help My First Application [C:\Users\m g\Documents\Android (layout)	dStudioProjects/MyFirstApplication]\app\src\main\res\Jayo	buti,activity_main.xml [app] — □
Android * 💮 ÷ 🏚 –	activity main.xml × @ build.gradle (app) × @ gradle.properties × @ MainActivity.a	ava X 4 AndroidManifest.xml X	
instance instance	Palette Q ¢ - & Q □ Pixel × ¥ 29 × @AppTheme ×	© Default (en-us) ∨	Attributes Q ¢ -
 Java com.example.myfirstapplication 	Common Ab TextView 💿 🔯 , 0dp , 🎜 🎘 🚼 🔳 I		Ab <unnamed> TextView</unnamed>
 Com.example.myfirstapplication (androidTest) Com.example.myfirstapplication (test) 	Test Button Buttons ImageView		id
res	Widgets C cfragments		T I mout
V Dilayout difference d	Layout Scoliveer Containers Google Legary Component Tree Component Tree Ato TextView Hello Works*	Þ •••••••	Constraint Widget
			▼ Common Attributes
		+	Example text Knemis university If text 0
	÷	3 //	contentDescription
E 6: Logcat ≔ TODO ≺ Build 团 Terminal			C Event Log
daemon started successfully (13 minutes ago)			1:1 CRLF UTF-8 4 spaces V 🐪

Figure 10 : activity_main (changement du message)

On peut changer le message affiché par l'application dans « common attributes » en bas a droite.



Figure 11: mipmap

Dans « mipmap » on trouve les icones utilisés dans l'application.

WyfirstApplication Image <	Q a frade
Market	stion ✓
We we personal state Open editor Hide notification We we personal state Deconceample myfintapplication (indications) Improve personal state Open editor We we personal state Deconceample myfintapplication (indications) Improve personal state Improve personal state Improve personal state We we personal state Deconceample myfintapplication (indications) Improve personal state Improve personal state Improve personal state We we personal state Deconceample myfintapplication (indications) Improve personal state Improve personal state Improve personal state We we personal state Deconceample myfintapplication (indications) Improve personal state Improve personal state Improve personal state We we personal state Deconceample myfintapplication (indications) Improve personal state Improve personal state Improve personal state We we personal state Deconceample myfintapplication (indications) Improve personal state Improve personal state Improve personal state We we personal state Deconceample myfintapplication (indications) Improve personal state Improve personal state Improve personal state We we personal state Deconceample myfintapplication (indications) Improve personal stat	stion
Image: Specific S	Device Re
*	
resources	splorer
In Strongest in London & Banko an Leininian	splorer

Figure 12: Strings.xml

On trouve ici (par exemple) toutes les chaines de caractères utilisées dans l'application.

Creation d'un « virtual device » :

Pour exécuter une application Android, il est nécessaire d'avoir une machine Android. Elle peut être virtuelle ou physique. Si pour diverses raisons, vous préférez exécuter votre programme dans une machine virtuelle Android : avant de cliquer sur « exécuter », choisissez l'option « **Open AVD manager** » tel qu'il est indiquer dans la figure 12.



Figure 13: Android Virtual Device Manager

Cliquez sur « create virtual device »

lect a system imag	je		_	
commended x86 Images	Other Images	ARI	Tarnet	R
R Download	R	x86	Android API R (Google Play)	
0 Download	29	x86	Android 10.0 (Google Play)	R
Pie Download	28	x86	Android 9.0 (Google Play)	
Dreo Download	27	x86	Android 8.1 (Google Play)	Android
Dreo Download	26	x86	Android 8.0 (Google Play)	Google Inc.
Nougat Download	25	x86	Android 7.1.1 (Google Play)	System Image
Nougat Download	24	x86	Android 7.0 (Google Play)	x86
				We recommend these Google Play images because this device is compatible with Google Play.
			G	Questions on API level? See the API level distribution chart
			G	

Figure 14: Virtual device configuration

Cliquez « download » devant la machine que vous voulez utiliser dans la simulation d'execution de vos projets.

Apres son installation, on doit lancer la machine virtuelle à partir du « AVD manager ».

🧕 Andro	id Virtual Device Manag	er						_	0	כ	×
2	Your Virtua Android Studio	l Device	es								
Туре	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions			
Co	Pixel 2 API 29		1080 × 1920: 420dpi	29	Android 10.0 (Googl	x86	8,9 GB				•
Co	Pixel 3a API 29		1080 × 2220: 440dpi	29	Android 10.0 (Googl	x86	8,8 GB			N .	•
Co	Pixel API 22		1080 × 1920: 420dpi	22	Android 5.1 (Google	x86	2,5 GB				~
+ c	reate Virtual Device							G		?	

Figure 15: AVD manager (installed machines)



Figure 16: Android emulator

En revenant à l'Android IDE, on peut choisir la machine virtuelle lancée et cliquer sur l'exécution.



Et l'application sera téléchargé dans l'émulateur.



C'est fait !