Homework 1

CS 478 - Software Development for Mobile Platforms - Fall 2019

Matteo Corain 650088272

September 8, 2019

1 Question 1

In order to be able to build and run Android applications through the AVD emulator, three components are needed for each API level we want to target: the *Android SDK*, the *Android sources* and the *Android x86 system image*.

After installing the required components for both API levels 27 (Android 8.1) and 28 (Android 9), the *SDK manager* built in the Android Studio IDE shows the screen reported in figure 1.

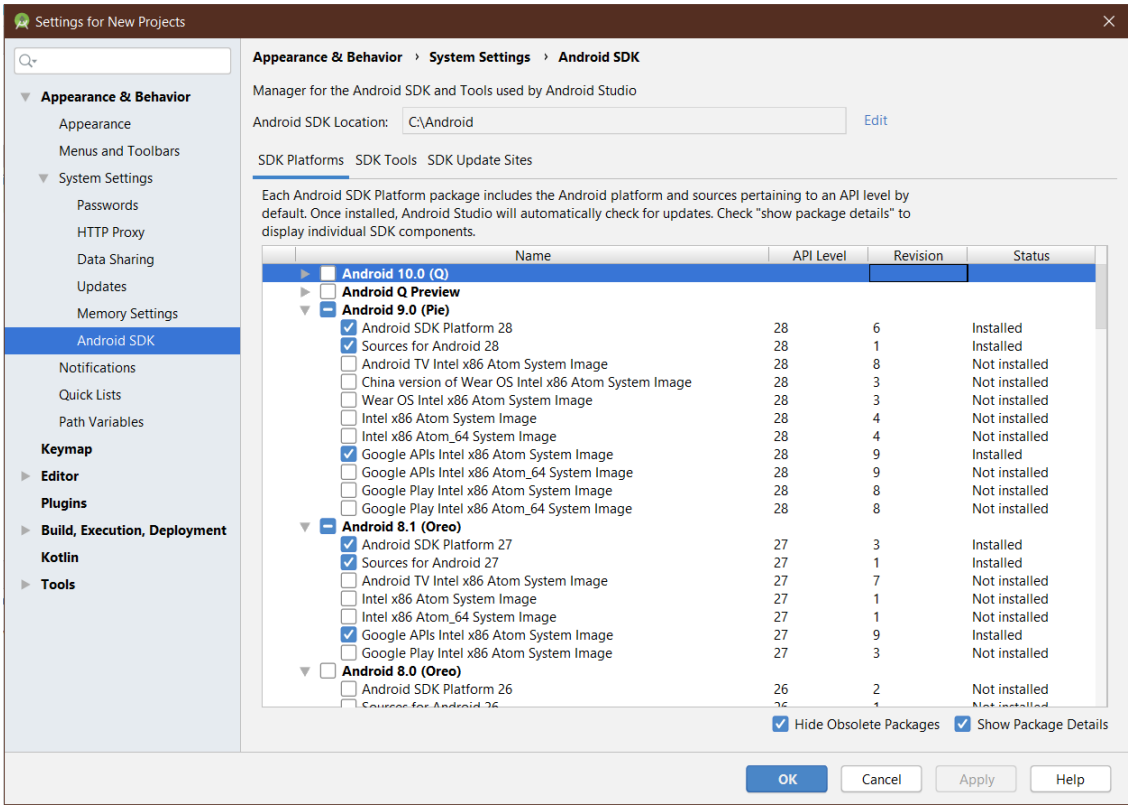

Figure 1: SDK manager after installing the required components

2 Question 2

The process of creation of a virtual device using the *AVD manager* provided by the Android Studio environment involves three steps:

- The selection of the physical device to be emulated: in this case, the *Pixel 2XL* has been used for both devices;
- The selection of the system image to be used: in this case, the Oreo (8.1) system image has been chosen for the first device and the Pie (9.0) system image for the second device;
- The selection of a name for the device and the setting of some advanced parameters, which were left at their default values.

After the creation of the two virtual devices, the *AVD manager* shows the screen reported in figure 2.

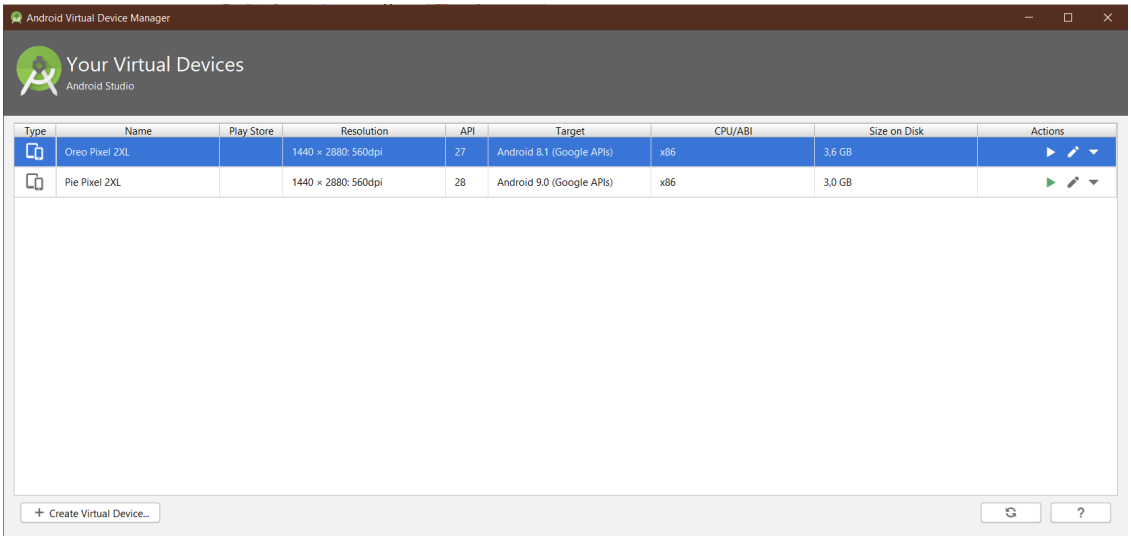

Figure 2: AVD manager after the creation of the two devices

The two devices may be started using the opportune action provided by the *AVD manager*; after the boot of the device, the typical Android home screen is shown and it is possible to use the emulator as if it was a real device. The home screen of the two devices running Android 8.1 and 9.0 are shown in figure 3.

Figure 3: The two created devices running in the emulator

3 Question 3

In order to obtain the requested message to be printed on screen, the only parameter that has to be changed in the default application as initialized by Android Studio is the `text` attribute of the `TextView` component in the `MainActivity` activity. This can be done in two ways:

- By manually modifying the layout file associated to the activity containing the `TextView` component (in this case, it is called `activity_main.xml`): it is sufficient to change the value associated to the `android:text` key inside the `TextView` block to the desired message;
- By using the graphical interface provided by Android Studio: after selecting the `TextView` component, the value of the `text` attribute can be viewed and edited in the right column, under the *Common Attributes* section, as shown in figure 4.

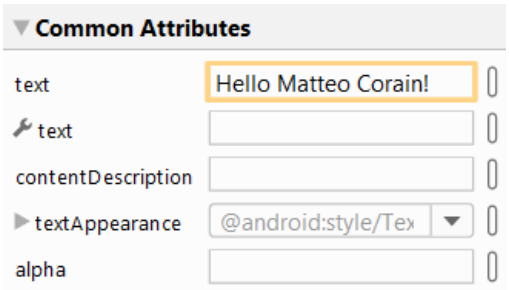

Figure 4: Common Attributes section in the Android Studio GUI

At this point, the application is ready to be compiled and sent to a device. In order to do so, it is necessary to:

- Enable the *USB Debug* option on the device through the *Developer Options*;
- Connect the device to the PC and accept the request of connection;
- Select the name of the physical device to be used in the navigation bar of Android Studio, as shown in figure 5;

Figure 5: Device selection for application running

- Issue the build and run commands.

Once the build process has terminated, the application is installed on the connected device and launched, showing the screen in figure 6.

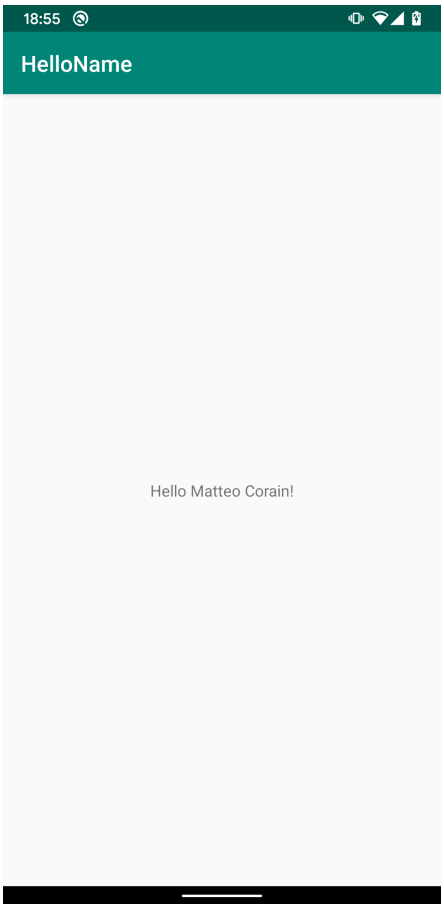

Figure 6: Application running on the selected device

4 Question 4

In order to allow the application to output a *Logcat* statement with a `INFO` priority level, the static method `i()` of the `Log` class has been used. Specifically, the following line has been added to the `onCreate()` method of the `MainActivity` class provided by Android Studio:

```
Log.i(TAG, "Matteo Corain was here!");
```

Where `TAG` is a string constant defined as:

```
private final static String TAG = "MainActivity";
```

As an effect, when the application is started on a device, the corresponding line is printed on the *Logcat* tab of Android Studio, as shown in figure 7.

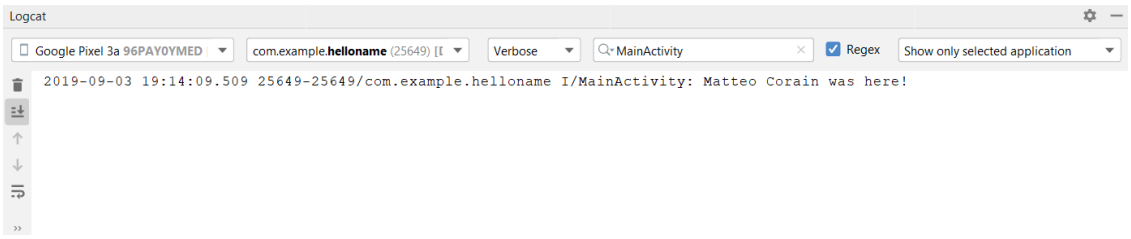

Figure 7: Logcat output in Android Studio