Paixon

Before we start…

Wer die Android-Studio-Installation noch nicht gestartet hat, jetzt starten:

➔ https://developer.android.com/studio

Code für die Übungen sowie die Lösungen + Slides erhaltet ihr über GitHub:

➔ https://github.com/sschoeb/OST-Mobile-App-Engineering-2024/

Paixon

Wer ist Java Entwickler?

Eclipse?

Intellij IDEA?

Wer hat bereits für Android entwickelt?

iOS?

Wer hat Erfahrungen mit Cross-Platform Frameworks?

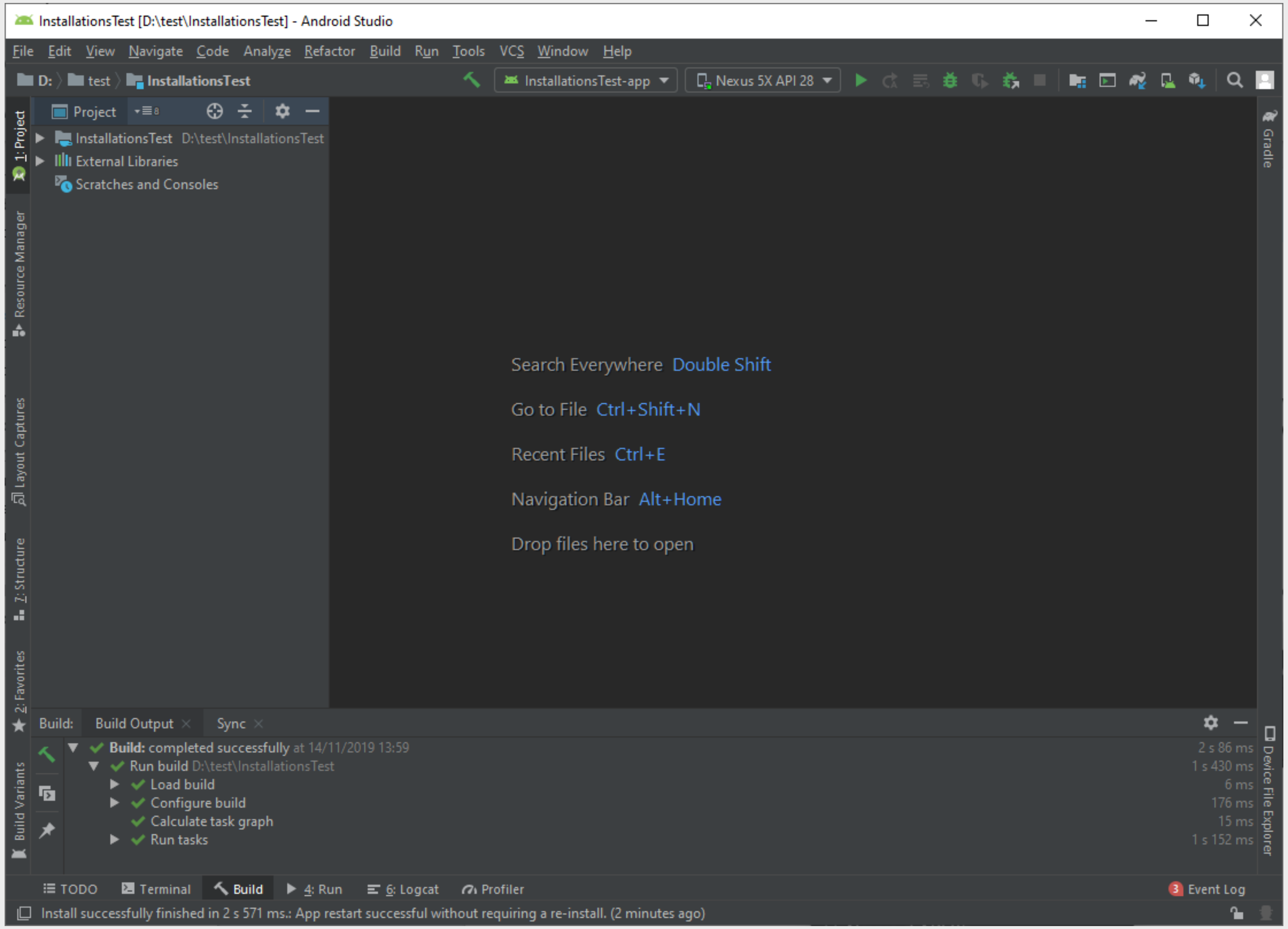
Xamarin, PhoneGap, Ionic, …

Dev-Tools

Android Studio installiert?

Android Studio

Paixon


InstallationsTest [D:\test\InstallationsTest] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
D: test InstallationsTest
InstallationsTest-app
Nexus 5X API 28
Project
InstallationsTest D:\test\InstallationsTest
External Libraries
Scratches and Consoles
1: Project
Resource Manager
Layout Captures
Z: Structure
2: Favorites
Build Variants
Gradle
Device File Explorer
Build: Build Output Sync
TODO Terminal Build 4: Run 6: Logcat Profiler
3 Event Log

Search Everywhere Double Shift

Go to File Ctrl+Shift+N

Recent Files Ctrl+E

Navigation Bar Alt+Home

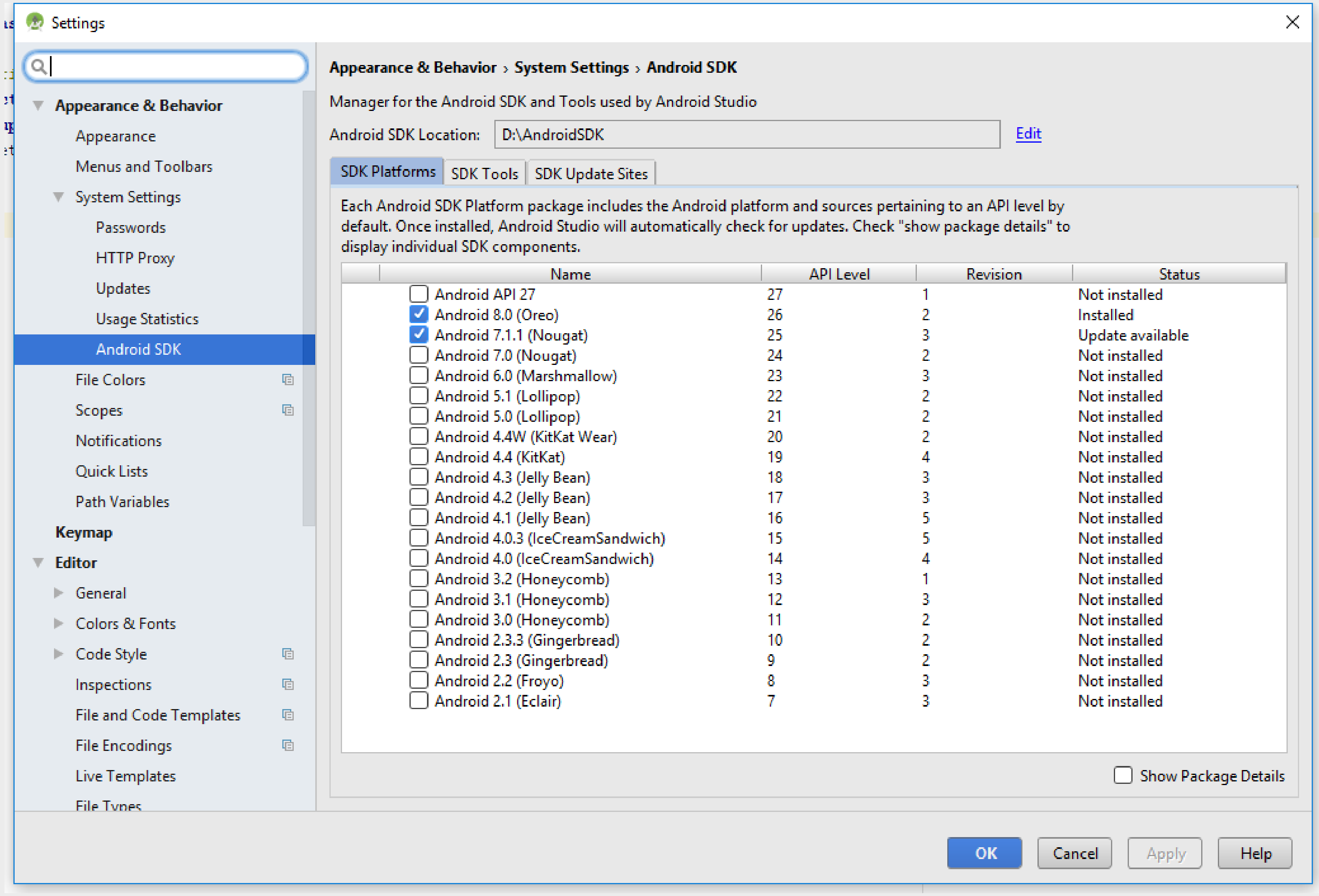
Drop files here to open

```
Build: completed successfully at 14/11/2019 13:59    2 s 86 ms
  Run build D:\test\InstallationsTest                1 s 430 ms
    Load build                                           6 ms
    Configure build                                    176 ms
    Calculate task graph                                15 ms
    Run tasks                                        1 s 152 ms
```

Install successfully finished in 2 s 571 ms.: App restart successful without requiring a re-install. (2 minutes ago)

SDK - Manager

Paixon



Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: D:\AndroidSDK Edit

SDK Platforms | SDK Tools | SDK Update Sites

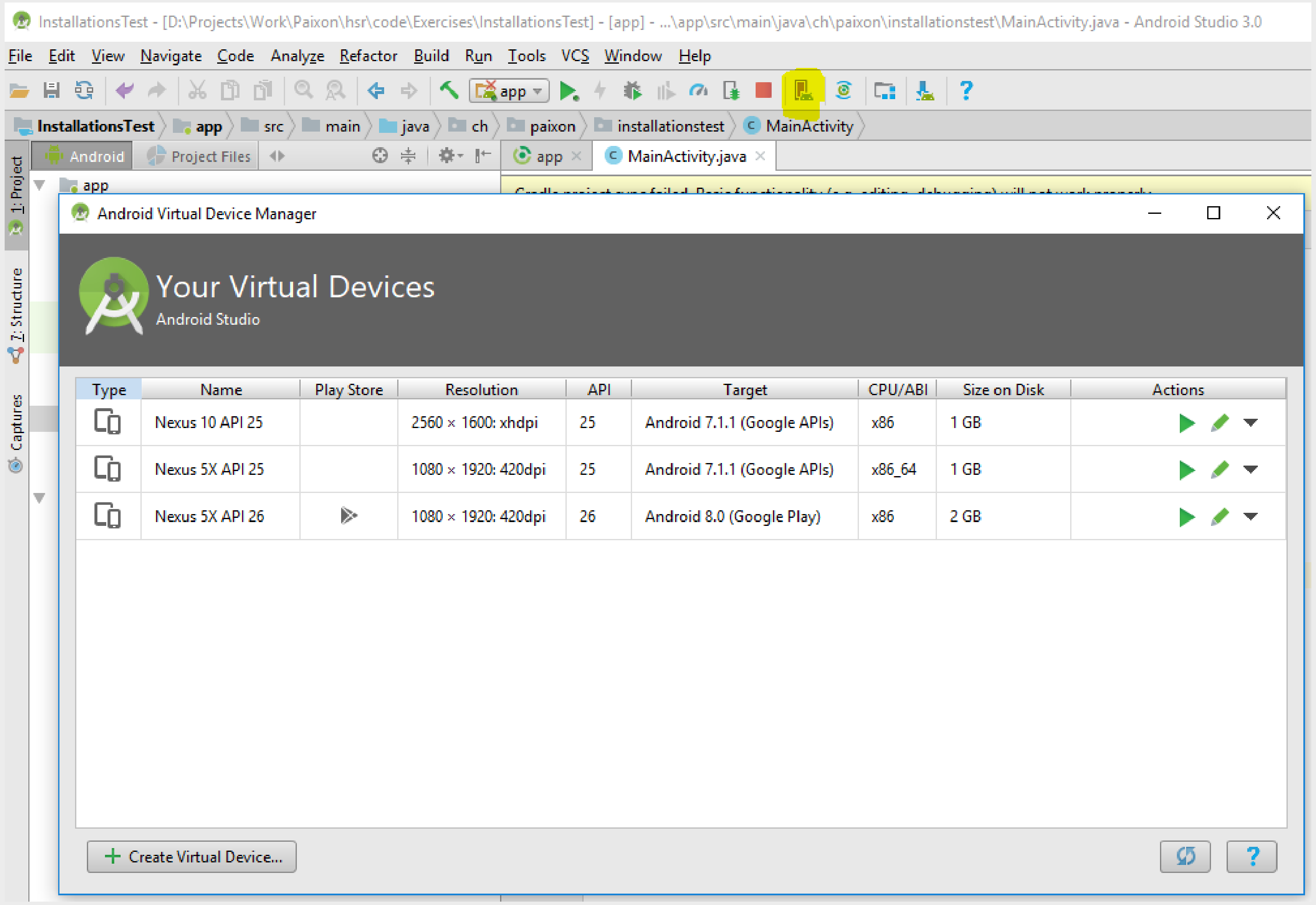
Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
☐	Android API 27	27	1	Not installed
✓	Android 8.0 (Oreo)	26	2	Installed
✓	Android 7.1.1 (Nougat)	25	3	Update available
☐	Android 7.0 (Nougat)	24	2	Not installed
☐	Android 6.0 (Marshmallow)	23	3	Not installed
☐	Android 5.1 (Lollipop)	22	2	Not installed
☐	Android 5.0 (Lollipop)	21	2	Not installed
☐	Android 4.4W (KitKat Wear)	20	2	Not installed
☐	Android 4.4 (KitKat)	19	4	Not installed
☐	Android 4.3 (Jelly Bean)	18	3	Not installed
☐	Android 4.2 (Jelly Bean)	17	3	Not installed
☐	Android 4.1 (Jelly Bean)	16	5	Not installed
☐	Android 4.0.3 (IceCreamSandwich)	15	5	Not installed
☐	Android 4.0 (IceCreamSandwich)	14	4	Not installed
☐	Android 3.2 (Honeycomb)	13	1	Not installed
☐	Android 3.1 (Honeycomb)	12	3	Not installed
☐	Android 3.0 (Honeycomb)	11	2	Not installed
☐	Android 2.3.3 (Gingerbread)	10	2	Not installed
☐	Android 2.3 (Gingerbread)	9	2	Not installed
☐	Android 2.2 (Froyo)	8	3	Not installed
☐	Android 2.1 (Eclair)	7	3	Not installed

☐ Show Package Details

Android Virtual Device Manager

Paixon



InstallationsTest - [D:\Projects\Work\Paixon\hsr\code\Exercises\InstallationsTest] - [app] - ...\app\src\main\java\ch\paixon\installationstest\MainActivity.java - Android Studio 3.0

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
InstallationsTest app src main java ch paixon installationstest MainActivity
Android Project Files
app MainActivity.java
1: Project
7: Structure
Captures

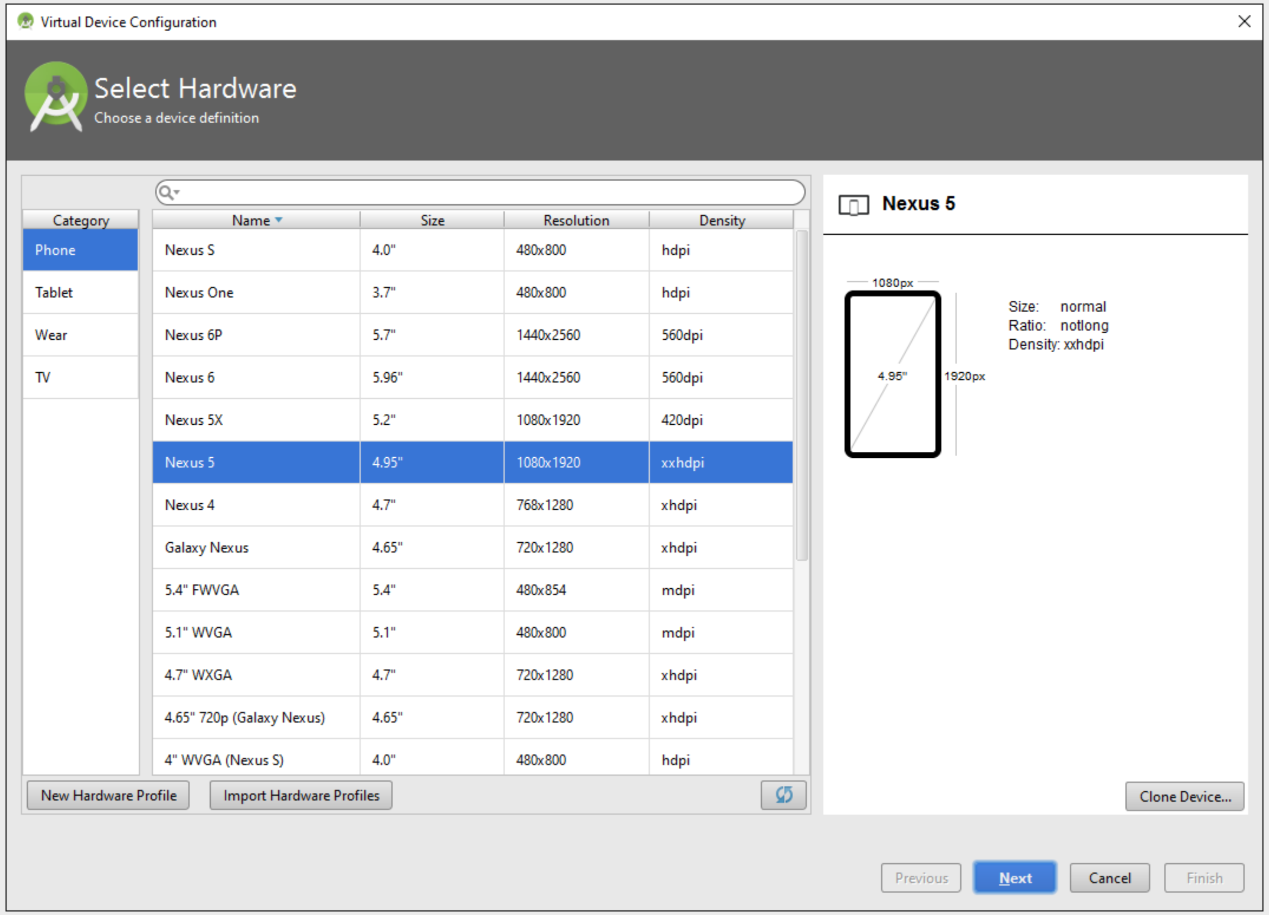
Android Virtual Device Manager

Your Virtual Devices
Android Studio

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus 10 API 25		2560 × 1600: xhdpi	25	Android 7.1.1 (Google APIs)	x86	1 GB	
	Nexus 5X API 25		1080 × 1920: 420dpi	25	Android 7.1.1 (Google APIs)	x86_64	1 GB	
	Nexus 5X API 26	▶	1080 × 1920: 420dpi	26	Android 8.0 (Google Play)	x86	2 GB	

Virtual Device Configuration

Paixon



Virtual Device Configuration

Select Hardware

Choose a device definition

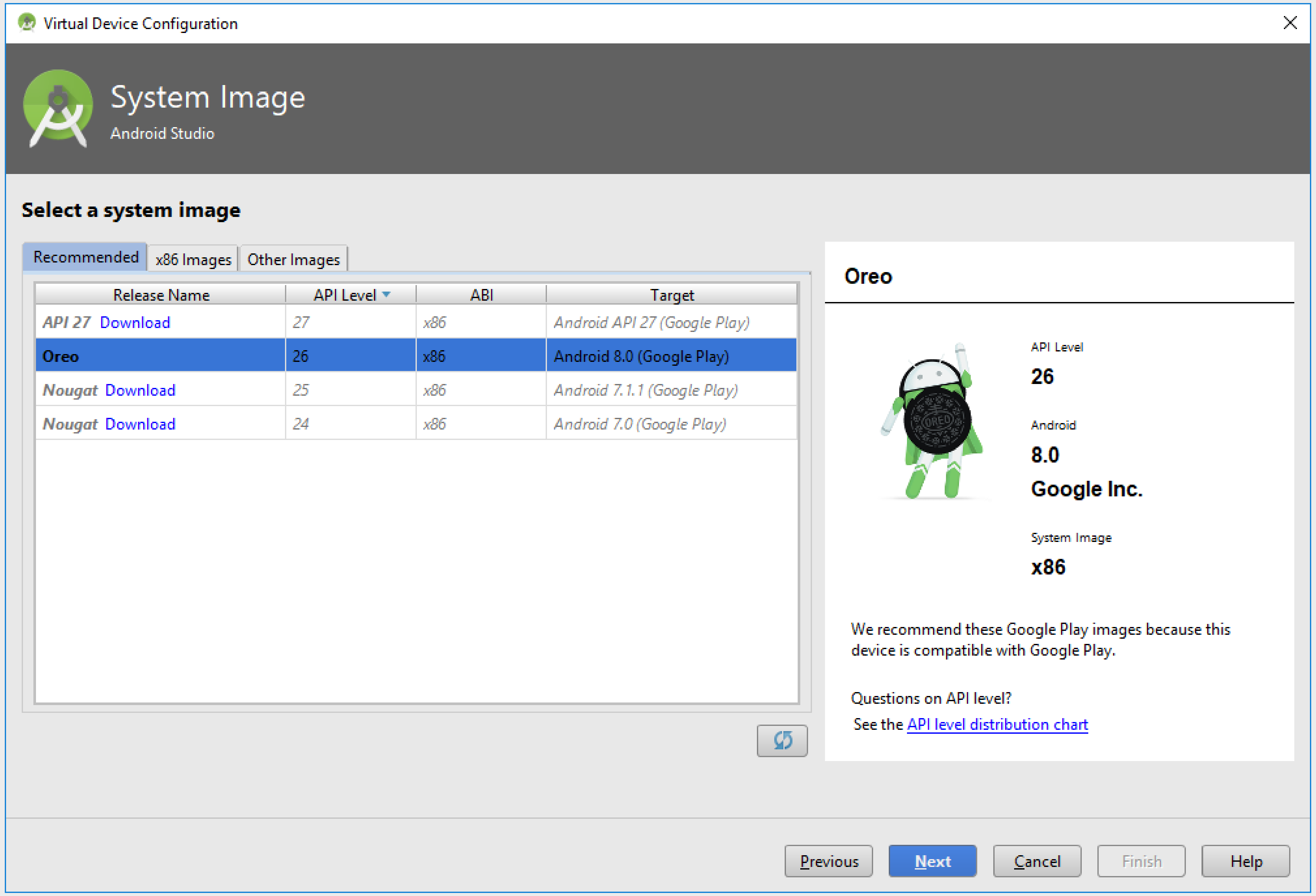
Category	Name	Size	Resolution	Density
Phone	Nexus S	4.0"	480x800	hdpi
Tablet	Nexus One	3.7"	480x800	hdpi
Wear	Nexus 6P	5.7"	1440x2560	560dpi
TV	Nexus 6	5.96"	1440x2560	560dpi
	Nexus 5X	5.2"	1080x1920	420dpi
	Nexus 5	4.95"	1080x1920	xxhdpi
	Nexus 4	4.7"	768x1280	xhdpi
	Galaxy Nexus	4.65"	720x1280	xhdpi
	5.4" FWVGA	5.4"	480x854	mdpi
	5.1" WVGA	5.1"	480x800	mdpi
	4.7" WXGA	4.7"	720x1280	xhdpi
	4.65" 720p (Galaxy Nexus)	4.65"	720x1280	xhdpi
	4" WVGA (Nexus S)	4.0"	480x800	hdpi

Nexus 5

Size: normal
Ratio: notlong
Density: xxhdpi

Virtual Device Configuration

Paixon



Select a system image

Release Name	API Level	ABI	Target
API 27 Download	*27*	*x86*	*Android API 27 (Google Play)*
Oreo	26	x86	Android 8.0 (Google Play)
Nougat Download	*25*	*x86*	*Android 7.1.1 (Google Play)*
Nougat Download	*24*	*x86*	*Android 7.0 (Google Play)*

Oreo

API Level
26

Android
8.0
Google Inc.

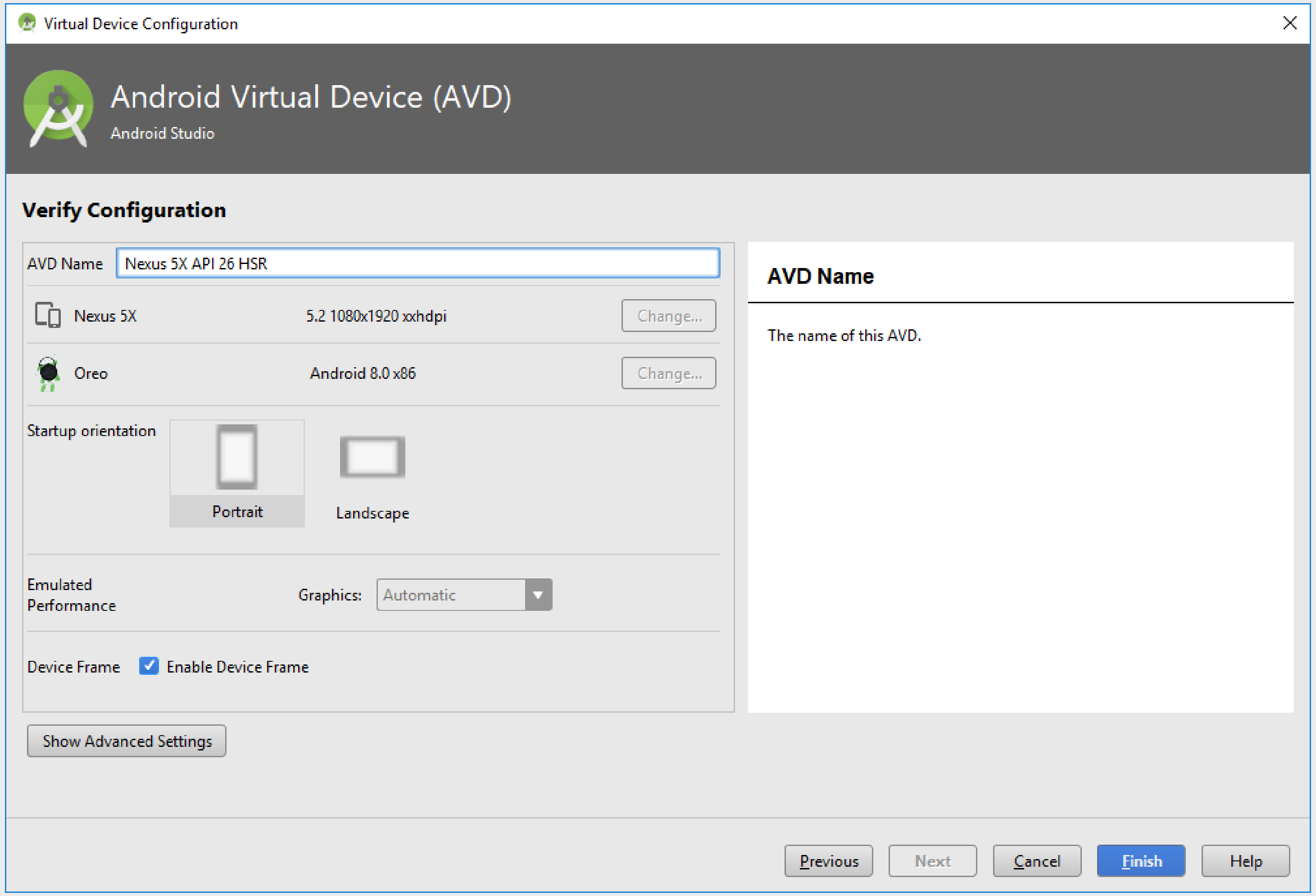
System Image
x86

We recommend these Google Play images because this device is compatible with Google Play.

Questions on API level?
See the API level distribution chart

Virtual Device Configuration

Paixon



Virtual Device Configuration

Android Virtual Device (AVD)

Android Studio

Verify Configuration

AVD Name: Nexus 5X API 26 HSR

Nexus 5X	5.2 1080x1920 xxhdpi	Change...
Oreo	Android 8.0 x86	Change...

Startup orientation: [✓ Portrait] Landscape

Emulated Performance — Graphics: Automatic

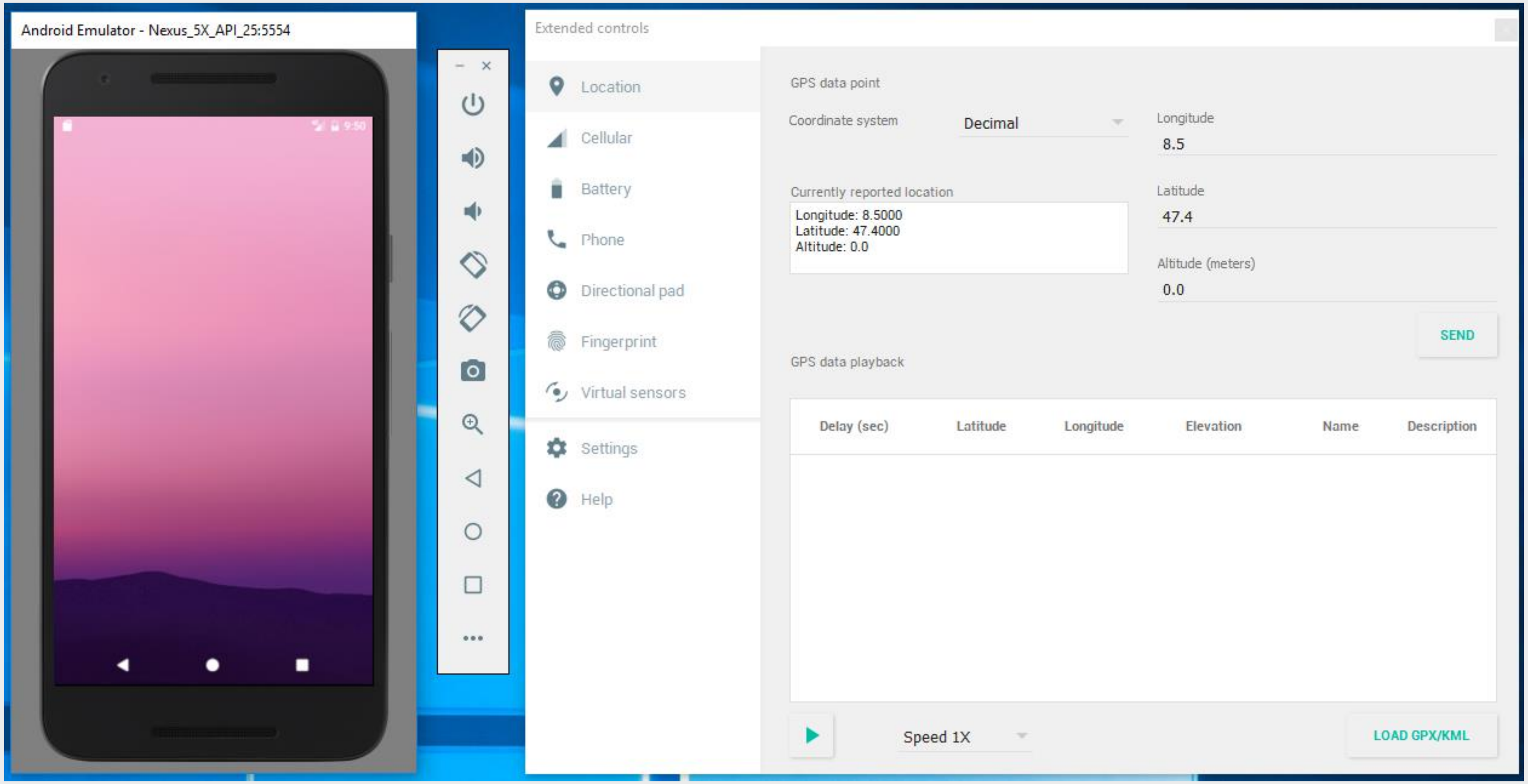
Device Frame — [✓] Enable Device Frame

AVD Name

The name of this AVD.

Android Emulator

Paixon


Android Emulator - Nexus_5X_API_25:5554

Extended controls

GPS data point

Coordinate system: Decimal

Longitude: 8.5

Latitude: 47.4

Altitude (meters): 0.0

Currently reported location

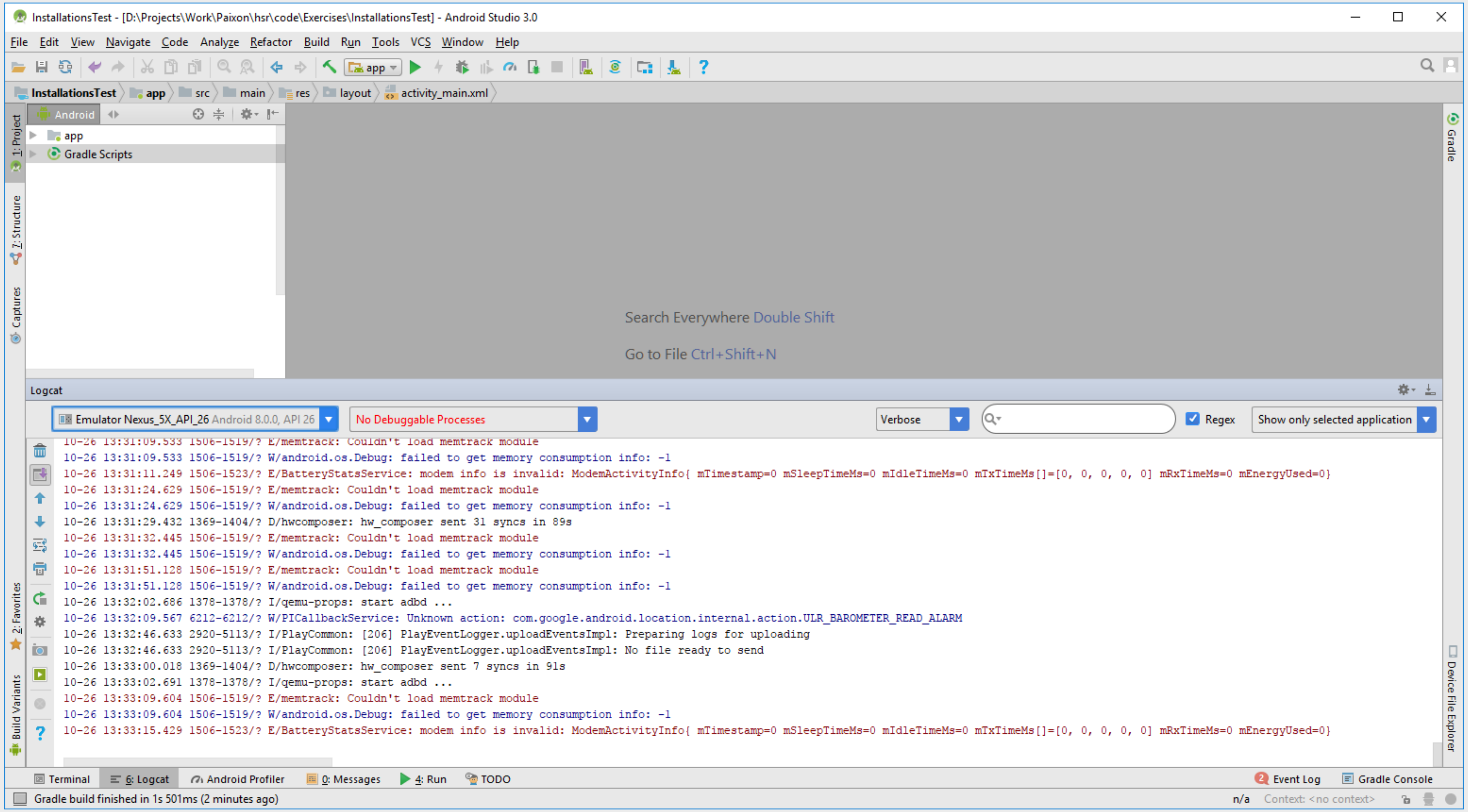
Longitude: 8.5000
Latitude: 47.4000
Altitude: 0.0

GPS data playback

Delay (sec)	Latitude	Longitude	Elevation	Name	Description

Speed 1X

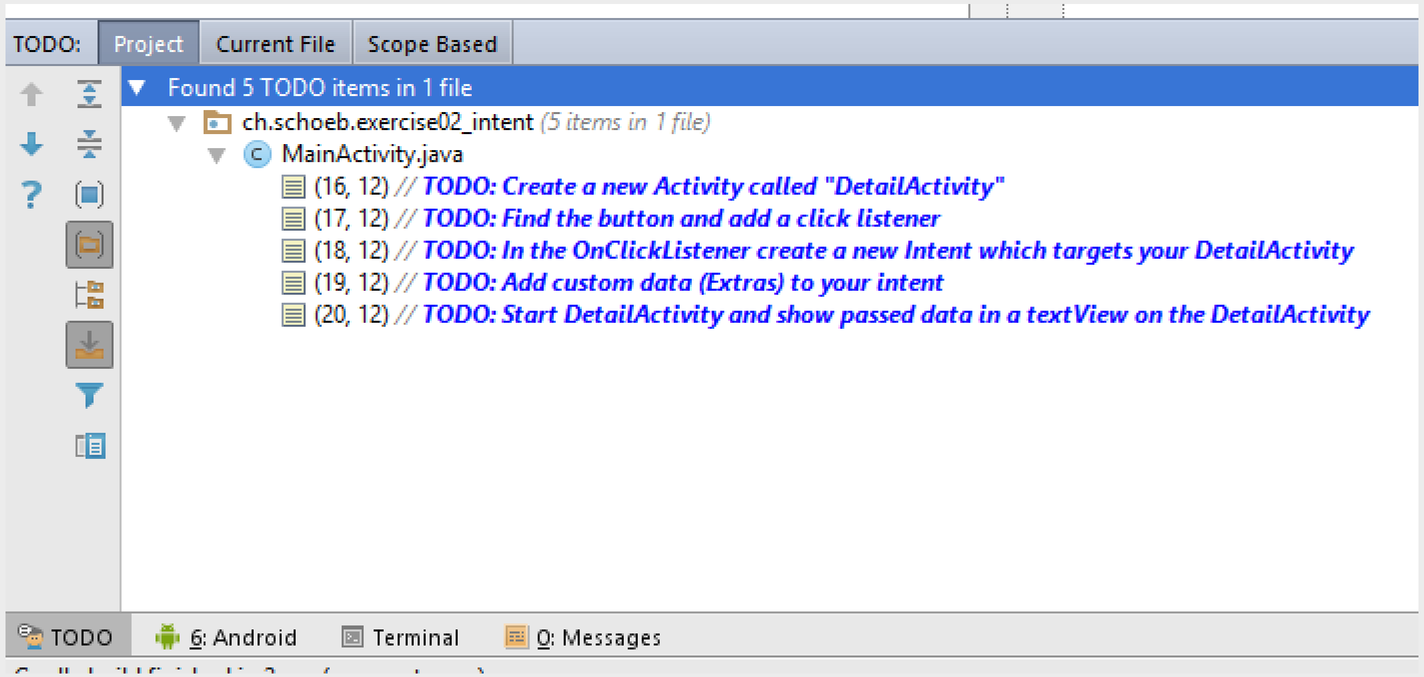
Logcat


InstallationsTest - [D:\Projects\Work\Paixon\hsr\code\Exercises\InstallationsTest] - Android Studio 3.0
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
InstallationsTest > app > src > main > res > layout > activity_main.xml
Android
app
Gradle Scripts
Search Everywhere Double Shift
Go to File Ctrl+Shift+N
Logcat
Emulator Nexus_5X_API_26 Android 8.0.0, API 26
No Debuggable Processes
Verbose
Regex
Show only selected application
Terminal
6: Logcat
Android Profiler
0: Messages
4: Run
TODO
Event Log
Gradle Console
Gradle build finished in 1s 501ms (2 minutes ago)
n/a
Context: <no context>

```
10-26 13:31:09.533 1506-1519/? E/memtrack: Couldn't load memtrack module
10-26 13:31:09.533 1506-1519/? W/android.os.Debug: failed to get memory consumption info: -1
10-26 13:31:11.249 1506-1523/? E/BatteryStatsService: modem info is invalid: ModemActivityInfo{ mTimestamp=0 mSleepTimeMs=0 mIdleTimeMs=0 mTxTimeMs[]=[0, 0, 0, 0, 0] mRxTimeMs=0 mEnergyUsed=0}
10-26 13:31:24.629 1506-1519/? E/memtrack: Couldn't load memtrack module
10-26 13:31:24.629 1506-1519/? W/android.os.Debug: failed to get memory consumption info: -1
10-26 13:31:29.432 1369-1404/? D/hwcomposer: hw_composer sent 31 syncs in 89s
10-26 13:31:32.445 1506-1519/? E/memtrack: Couldn't load memtrack module
10-26 13:31:32.445 1506-1519/? W/android.os.Debug: failed to get memory consumption info: -1
10-26 13:31:51.128 1506-1519/? E/memtrack: Couldn't load memtrack module
10-26 13:31:51.128 1506-1519/? W/android.os.Debug: failed to get memory consumption info: -1
10-26 13:32:02.686 1378-1378/? I/qemu-props: start adbd ...
10-26 13:32:09.567 6212-6212/? W/PICallbackService: Unknown action: com.google.android.location.internal.action.ULR_BAROMETER_READ_ALARM
10-26 13:32:46.633 2920-5113/? I/PlayCommon: [206] PlayEventLogger.uploadEventsImpl: Preparing logs for uploading
10-26 13:32:46.633 2920-5113/? I/PlayCommon: [206] PlayEventLogger.uploadEventsImpl: No file ready to send
10-26 13:33:00.018 1369-1404/? D/hwcomposer: hw_composer sent 7 syncs in 91s
10-26 13:33:02.691 1378-1378/? I/qemu-props: start adbd ...
10-26 13:33:09.604 1506-1519/? E/memtrack: Couldn't load memtrack module
10-26 13:33:09.604 1506-1519/? W/android.os.Debug: failed to get memory consumption info: -1
10-26 13:33:15.429 1506-1523/? E/BatteryStatsService: modem info is invalid: ModemActivityInfo{ mTimestamp=0 mSleepTimeMs=0 mIdleTimeMs=0 mTxTimeMs[]=[0, 0, 0, 0, 0] mRxTimeMs=0 mEnergyUsed=0}
```

Android Studio – TODOs

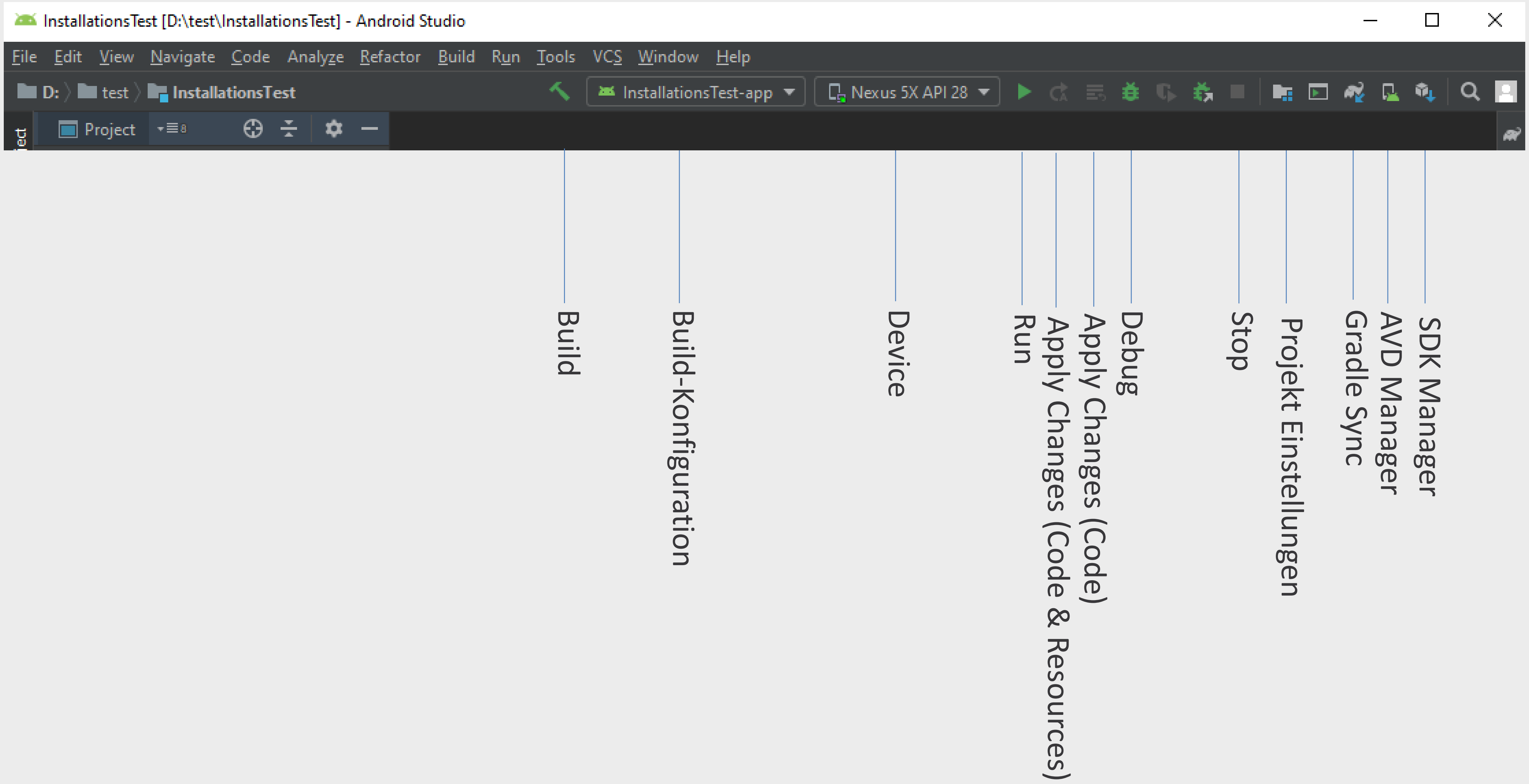
Paixon



```
Found 5 TODO items in 1 file
  ch.schoeb.exercise02_intent (5 items in 1 file)
    MainActivity.java
      (16, 12) // TODO: Create a new Activity called "DetailActivity"
      (17, 12) // TODO: Find the button and add a click listener
      (18, 12) // TODO: In the OnClickListener create a new Intent which targets your DetailActivity
      (19, 12) // TODO: Add custom data (Extras) to your intent
      (20, 12) // TODO: Start DetailActivity and show passed data in a textView on the DetailActivity
```

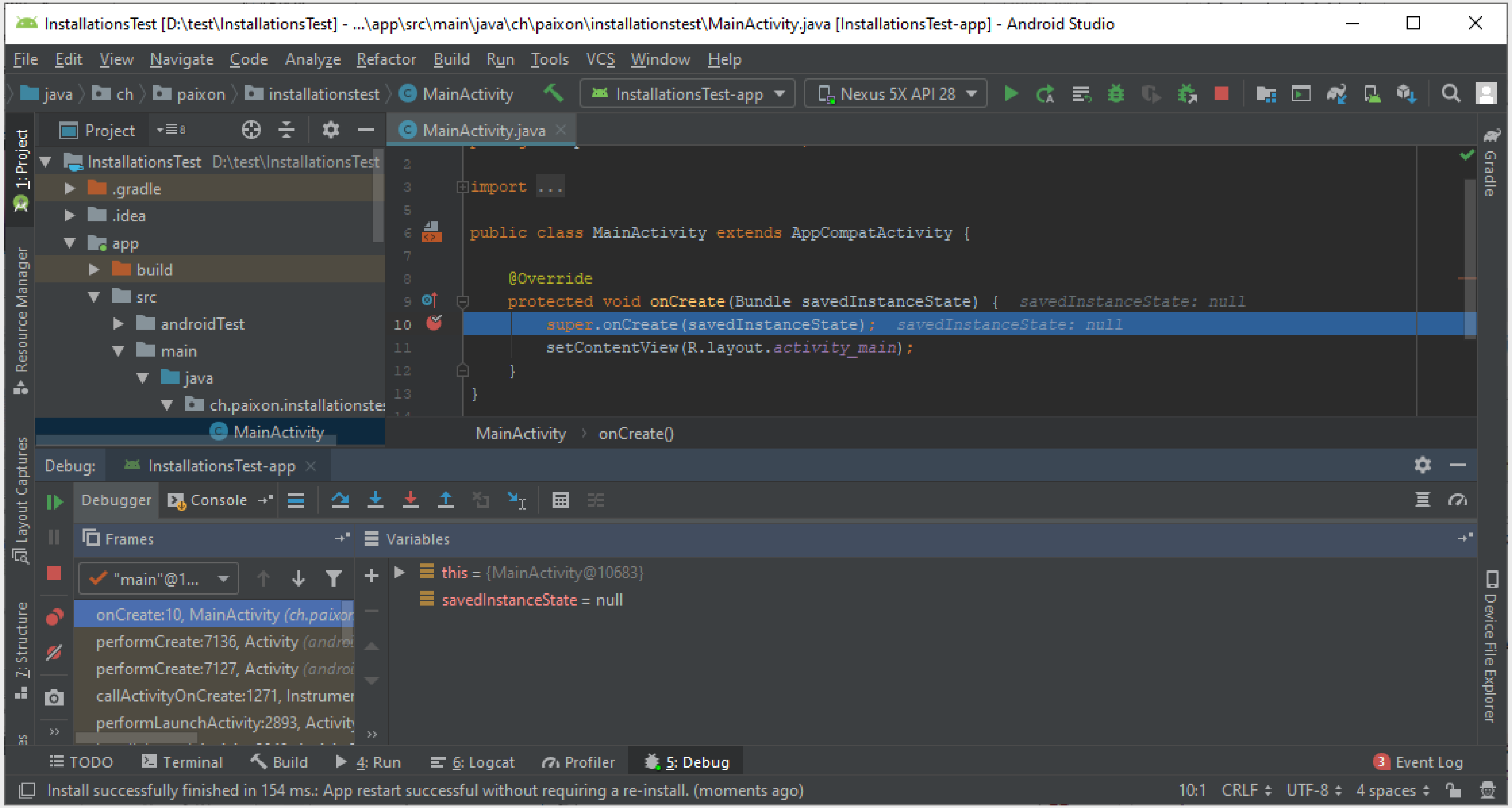
Android Studio – Toolbar

Paixon


InstallationsTest [D:\test\InstallationsTest] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
D: test InstallationsTest
InstallationsTest-app
Nexus 5X API 28
Project
Build
Build-Konfiguration
Device
Run
Apply Changes (Code & Resources)
Apply Changes (Code)
Debug
Stop
Projekt Einstellungen
Gradle Sync
AVD Manager
SDK Manager

Android Studio – Debugging

Paixon


InstallationsTest [D:\test\InstallationsTest] - ...\app\src\main\java\ch\paixon\installationstest\MainActivity.java [InstallationsTest-app] - Android Studio

```
import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {  savedInstanceState: null
        super.onCreate(savedInstanceState);  savedInstanceState: null
        setContentView(R.layout.activity_main);
    }
}
```

Frames:

```
onCreate:10, MainActivity (ch.paixor
performCreate:7136, Activity (andro
performCreate:7127, Activity (andro
callActivityOnCreate:1271, Instrumer
performLaunchActivity:2893, Activity
```

Variables:

```
this = {MainActivity@10683}
savedInstanceState = null
```

Install successfully finished in 154 ms.: App restart successful without requiring a re-install. (moments ago)

Paixon

Let’s start with a simple App

Übung – Erste Android App

Gemeinsamer Teil

- Neues Android Projekt erstellen
- Projekt auf dem Emulator starten

Aufgabe

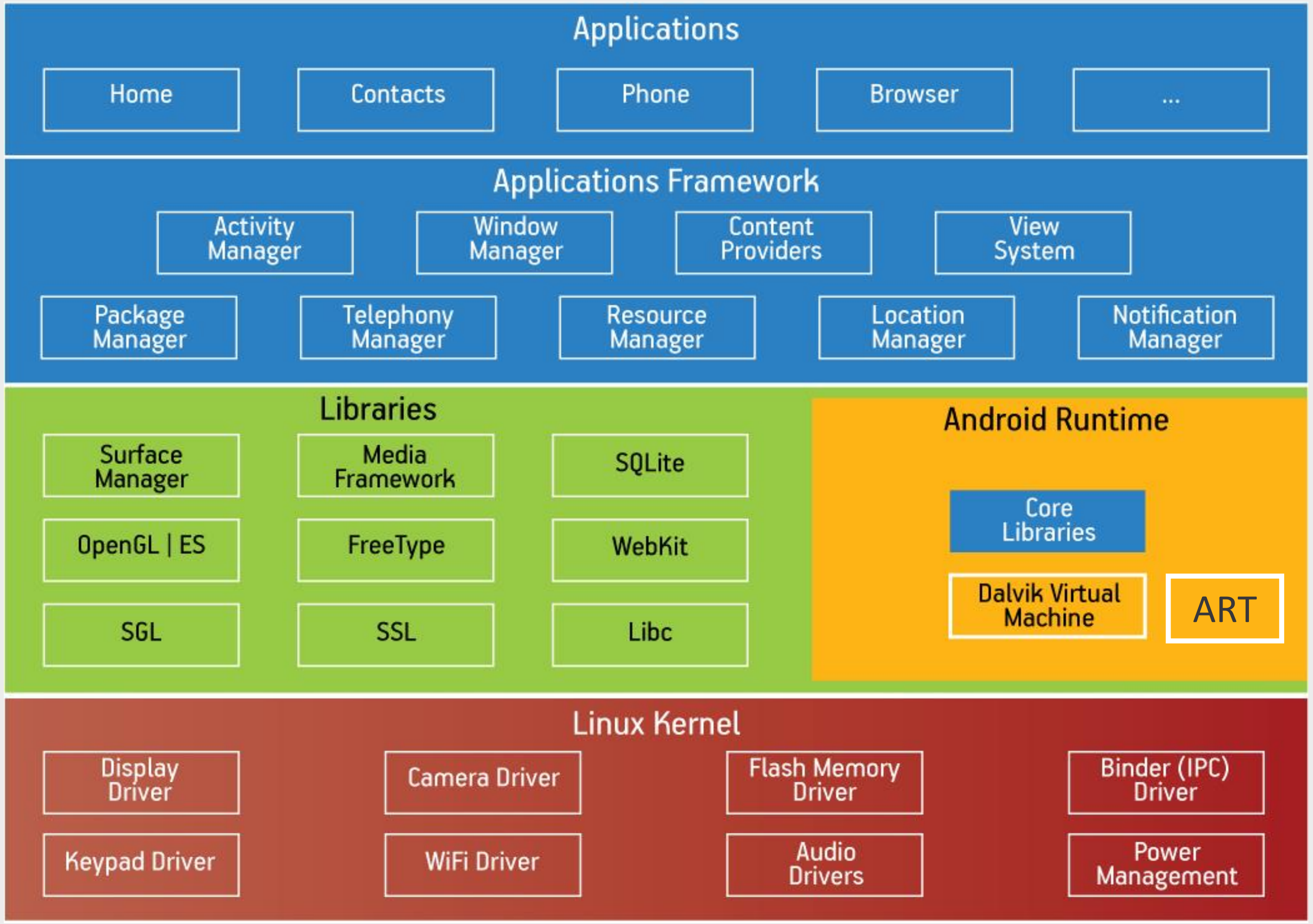
- Verschiedene Projekt-Ansichten analysieren
- Einen eigenen Filter im Android Monitor erstellen
- App im Debug-Modus starten

Paixon

Android System / App Basics

Android System

Paixon



App Basics

- App besteht aus
  - Code (Kotlin, Java oder C++)
  - Resourcen
  - Manifest-File
- Android SDK Tools kompilieren alles in eine APK-Datei (*.apk) / AAB-Datei (*aab)
  - Android package → zip-Datei
  - Beinhaltet alles um die App auf einem Android-Gerät zu installieren

- App läuft in einem eigenen Prozess
- App hat eigenen Linux User → Filesystem-Sandbox

App Basics – Resources

- Alles was ihr benötigt aber nicht Sourcecode ist
  - Texte, Bilder, Layoutdefinitionen, …
- Abgelegt im «res»-Ordner
- Bestehen hauptsächlich aus XML-Files

```
Declaration in Strings.xml:
<string name="app_name">Services</string>
```

```
Usage in any other XML-File:
android:label="@string/app_name"
```

App Basics – Android Manifest

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="ch.schoeb.services“
           android:versionCode="1“
           android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="8“ android:targetSdkVersion="17" />

    <application android:icon="@drawable/ic_launcher“
                                 android:label="@string/app_name"
                     android:theme="@style/AppTheme" >
        <activity android:name="ch.schoeb.services.MainActivity"
                  android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                  <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

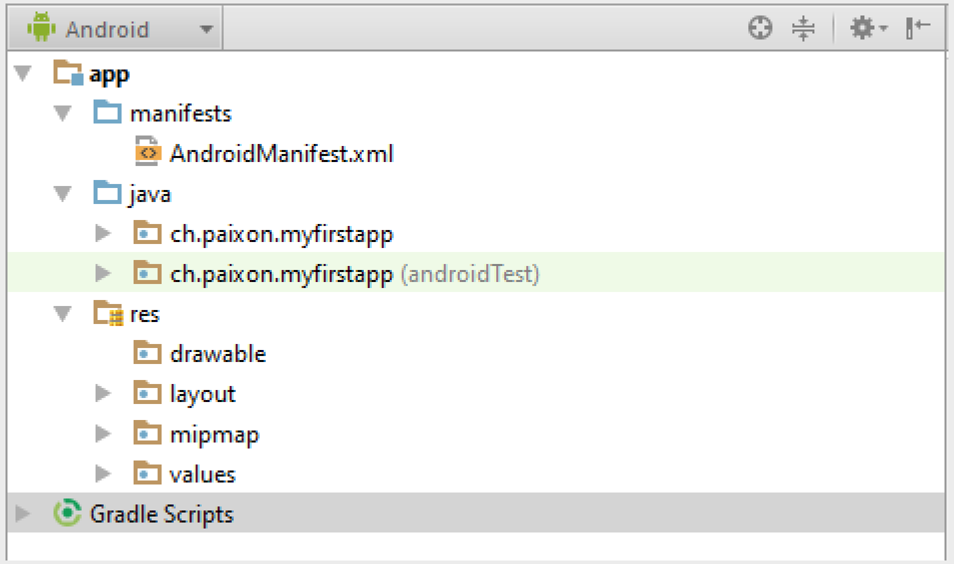
Jede Applikation hat genau ein AndroidManifest.xml

Liefert Informationen über die App

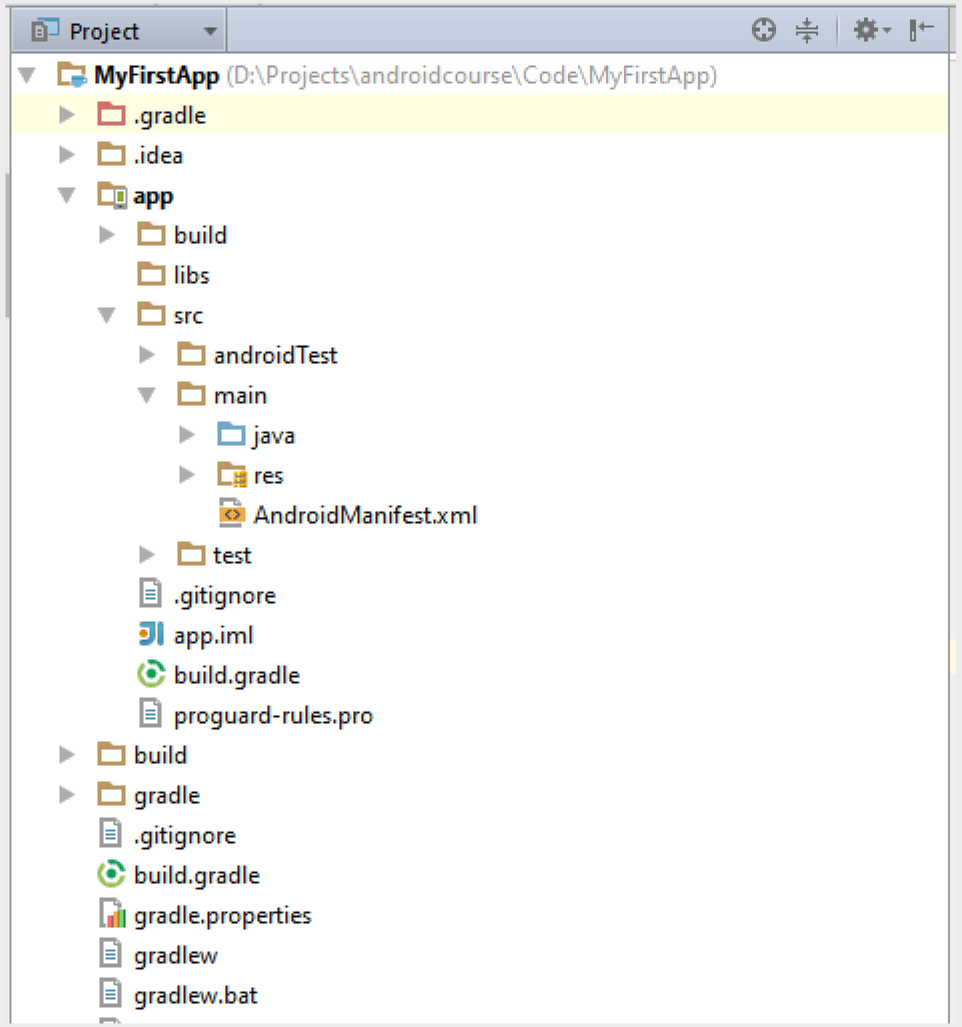
- Names the Java-Package
- Describes the components (Activities, Services, ...)
- Permissions
- Android API requirements
- List of linked libraries

App Basics – Projekt Aufbau

Paixon

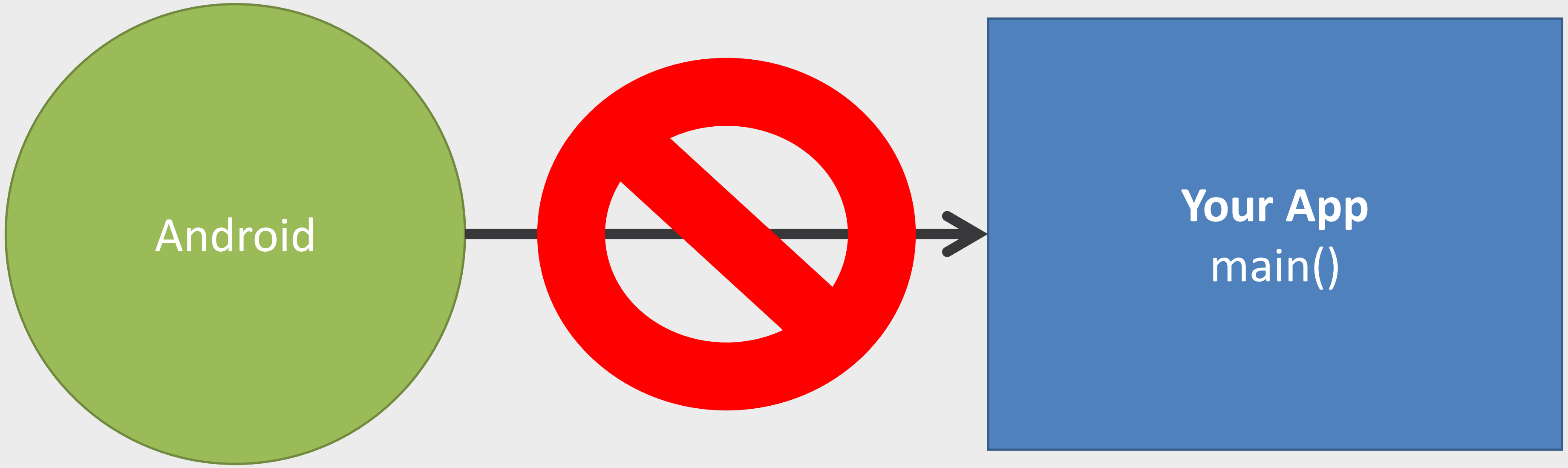


```
app
  manifests
    AndroidManifest.xml
  java
    ch.paixon.myfirstapp
    ch.paixon.myfirstapp (androidTest)
  res
    drawable
    layout
    mipmap
    values
Gradle Scripts
```



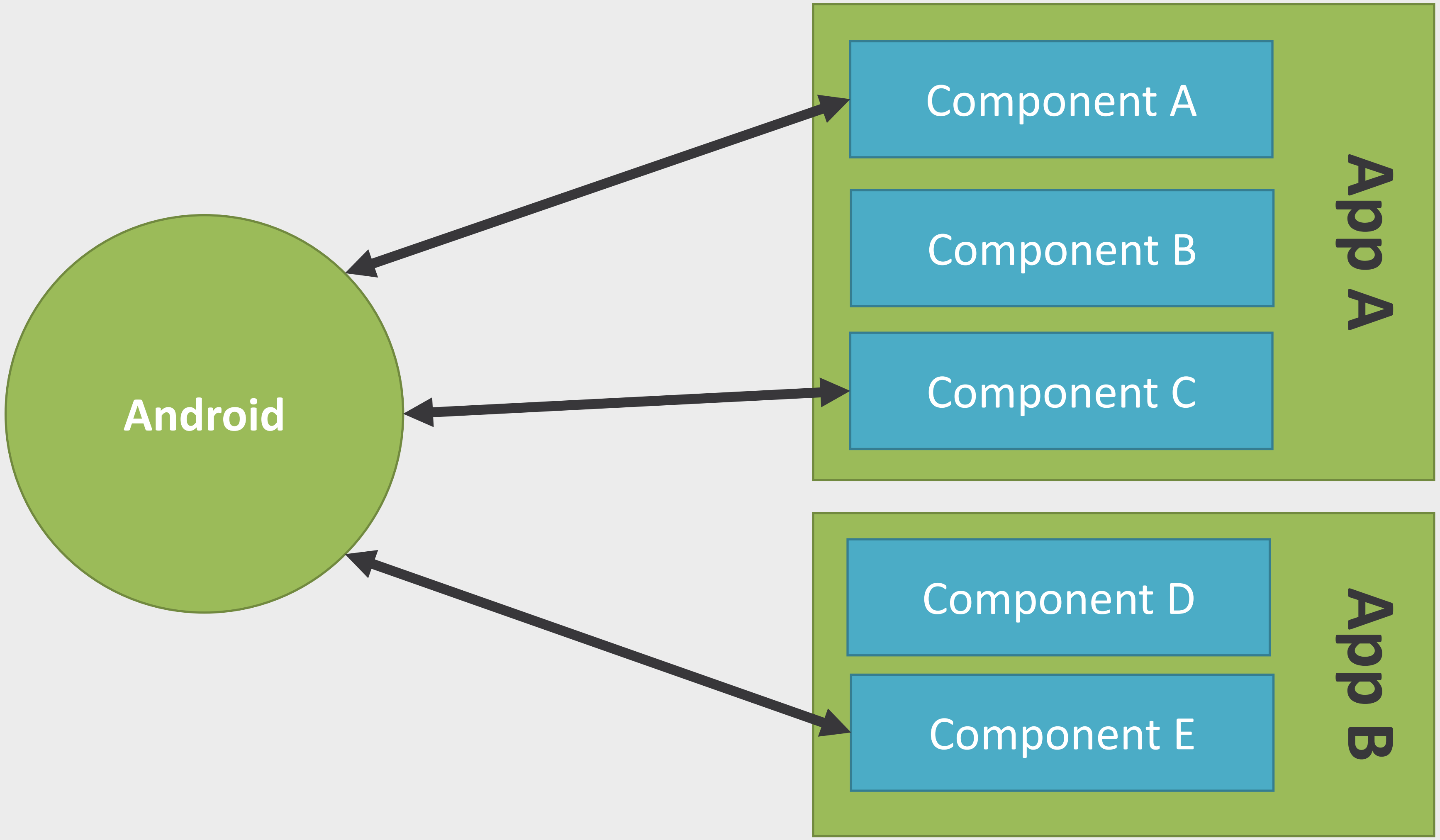
```
MyFirstApp (D:\Projects\androidcourse\Code\MyFirstApp)
  .gradle
  .idea
  app
    build
    libs
    src
      androidTest
      main
        java
        res
        AndroidManifest.xml
      test
    .gitignore
    app.iml
    build.gradle
    proguard-rules.pro
  build
  gradle
  .gitignore
  build.gradle
  gradle.properties
  gradlew
  gradlew.bat
```

Component driven



Component driven

Paixon



Paixon

Components

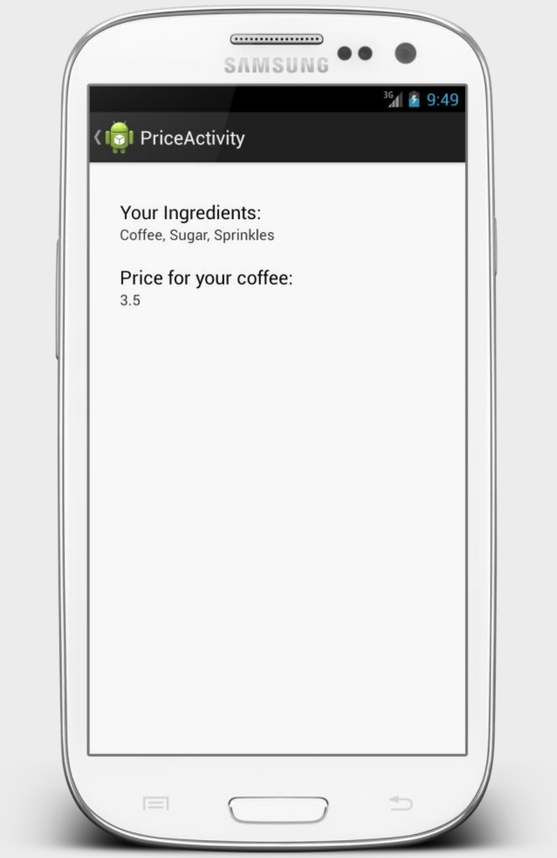
Activity	Service	ContentProvider	BroadcastReceiver
UI	Background	Provide data	Get notified about events

Activity

Activity

Paixon

- Represents a single screen a user can interact with
- Userinterface defined in a xml file or direct in code
- Application normaly has multiple activities
- Activity lifecycle
- Class extending Activity (AppCompatActivity)

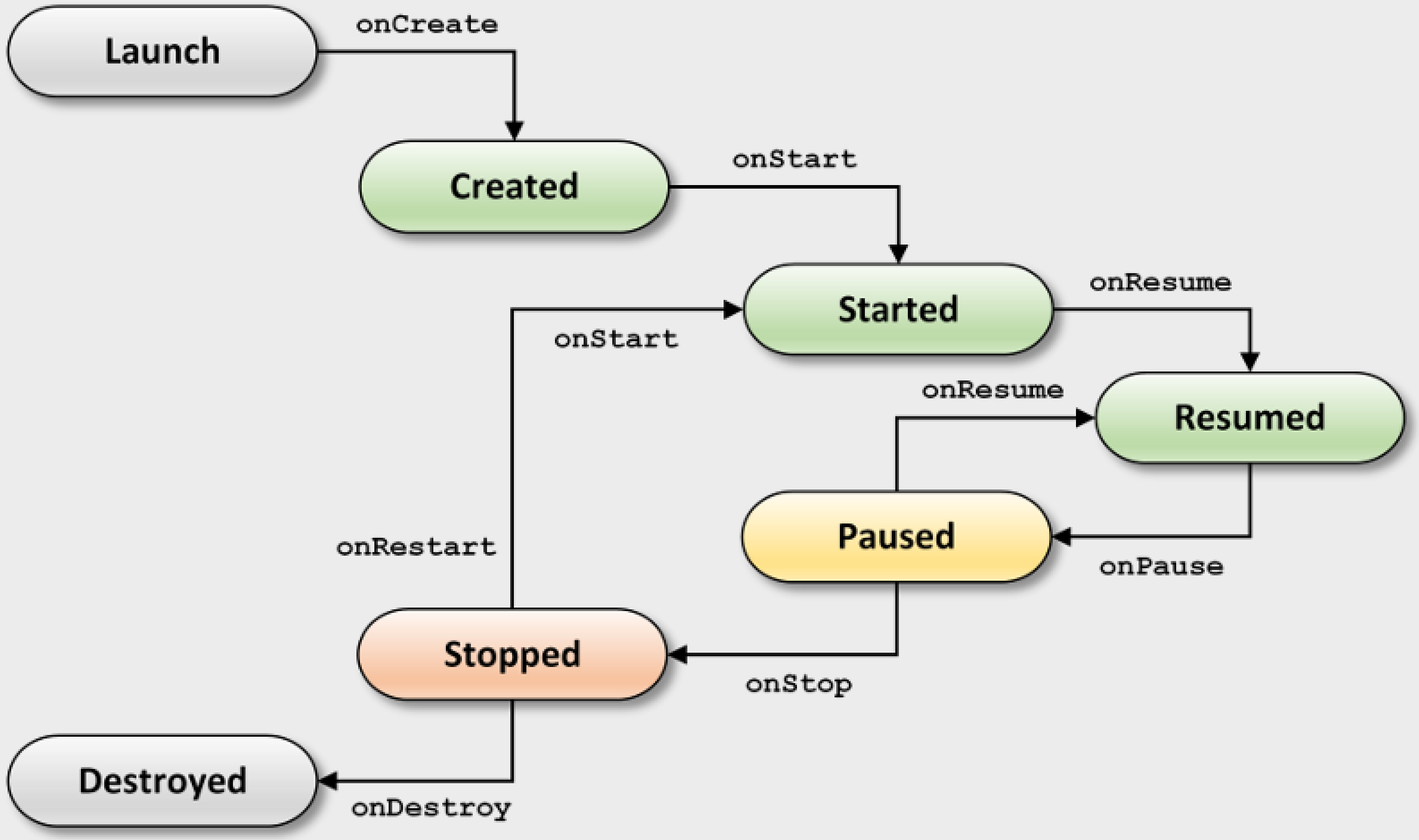


Your Ingredients:
Coffee, Sugar, Sprinkles

Price for your coffee:
3.5

Activity – Lifecycle methods

Paixon



Activity – Lifecycle methods

Paixon

onCreate() Called when the activity is first created. This is where you should do all of your normal static set up: create views, bind data to lists, etc.

onRestart() Called after your activity has been stopped, prior to it being started again.

onStart() Called when the activity is becoming visible to the user.

onResume() Called when the activity will start interacting with the user. At this point your activity is at the top of the activity stack, with user input going to it.

onPause() Called when the system is about to start resuming a previous activity. This is typically used to commit unsaved changes to persistent data, stop animations and other things that may be consuming CPU, etc. Implementations of this method must be very quick because the next activity will not be resumed until this method returns.

onStop() Called when the activity is no longer visible to the user, because another activity has been resumed and is covering this one. This may happen either because a new activity is being started, an existing one is being brought in front of this one, or this one is being destroyed.

onDestroy() The final call you receive before your activity is destroyed. This can happen either because the activity is finishing (someone called finish() on it, or because the system is temporarily destroying this instance of the activity to save space.

DEMO

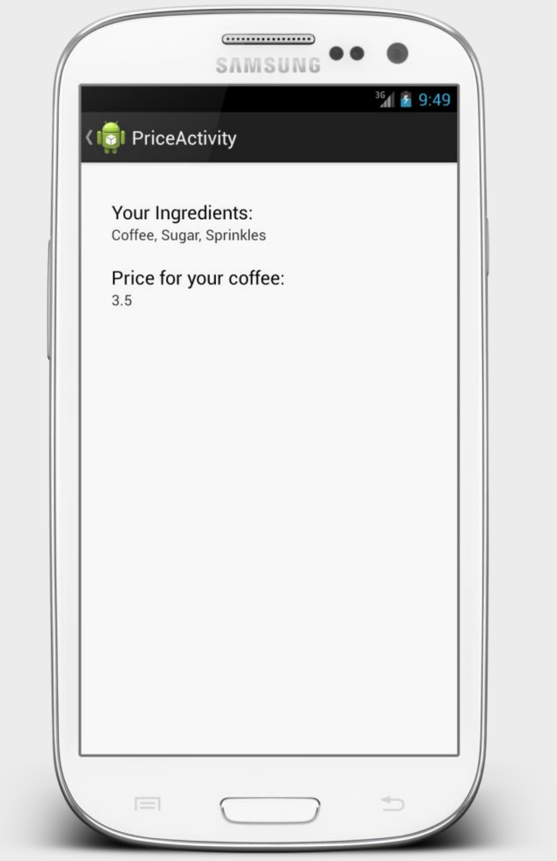
Lifecycle methods

Activity – Connect Java and XML

Paixon

PriceActivity.java
- Extend Activity class
- Use setContentView(…) to connect
- Use findViewById(…) to access view

activity_price.xml
- Declarative xml to define UI
- Define ID’s for every view



PriceActivity

Your Ingredients:
Coffee, Sugar, Sprinkles

Price for your coffee:
3.5

Activity - Layout

Paixon

activity_price.xml

```
<RelativeLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

    <Button
        android:id="@+id/demoButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="MyButton" />

</RelativeLayout>
```

Define id using @+id/yourId

Activity – Connect Java and XML

Paixon

Set layout for activity:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_price);
}
```

Access views in activity:

```
Button demoButton = (Button) findViewById(R.id.demoButton);
```

Generated R-Class

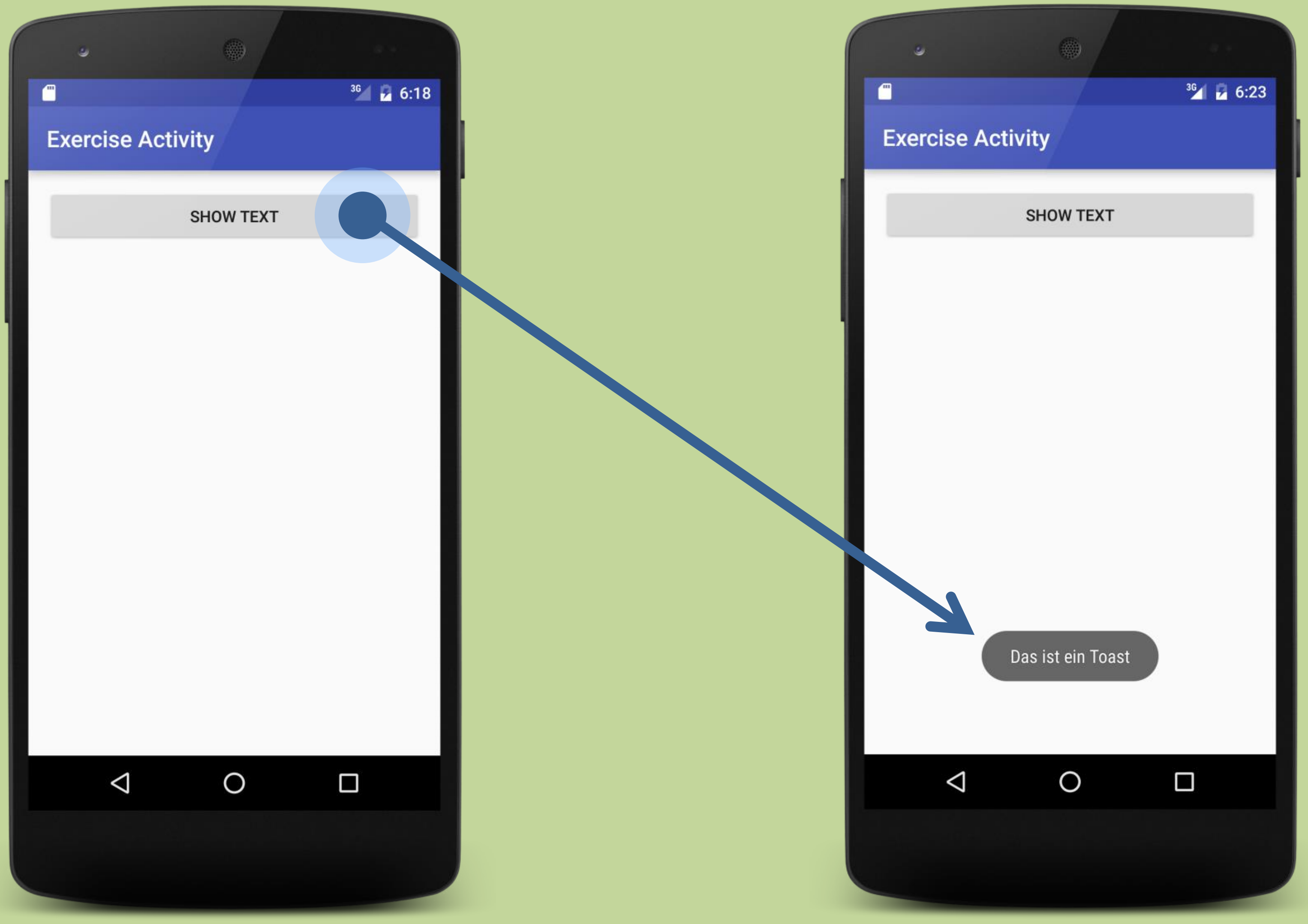
Paixon

- Located in your “build/…” folder
- Automatically generated
- Contains all resources ID’s from your res-folder as public constants
- Subclasses for all Resources-Types:
  - R.drawable / R.layouts / R.id / R.string / …
- Used to access resources in code

```
public final class R {
    public static final class string {
        public static final int action_settings=0x7f050001;
        public static final int app_name=0x7f050000;
        public static final int hello_world=0x7f050002;
    }
```

Übung Activity

Paixon



Übung Activity

Aufgaben:

1. Verstehe den Aufbau der Android App
   - Wo werden die Layout-files gespeichert?
   - Wie wird eine Activity definiert?
   - Für was ist das AndroidManifest.xml?
   - Wie können Layout und Activity-Klasse verbunden werden?
   - Wie können die Activity-Lifecycle-Methoden verwendet werden?

2. Stelle sicher, dass wenn der Benutzer auf den “Show text”-Button klickt der Text “Button clicked” als “Toast” angezeigt wird.
   - Auf dem Button kann ein OnClickListener gesetzt werden
   - Ein Toast kann mit Hilfer der «Toast»-Klasse erstellt werden

Projekt: Exercise_Activity

Connect multiple Activities

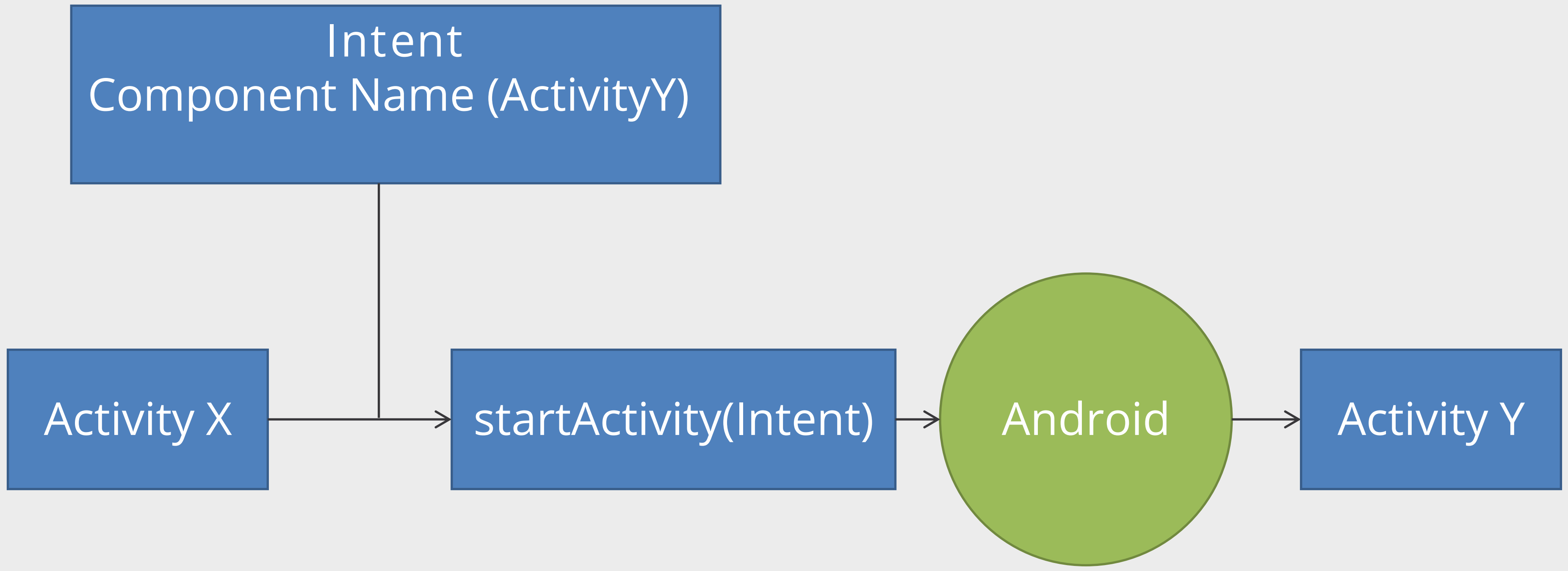
Connect multiple Activities – Intent

- Message to start another component
- Two types of intents available
  - Explicit Intents
  - Implicit Intents
- Contains data for the target Component
  - Component name, action, category, extras, flags

Target component can be defined in a different application!

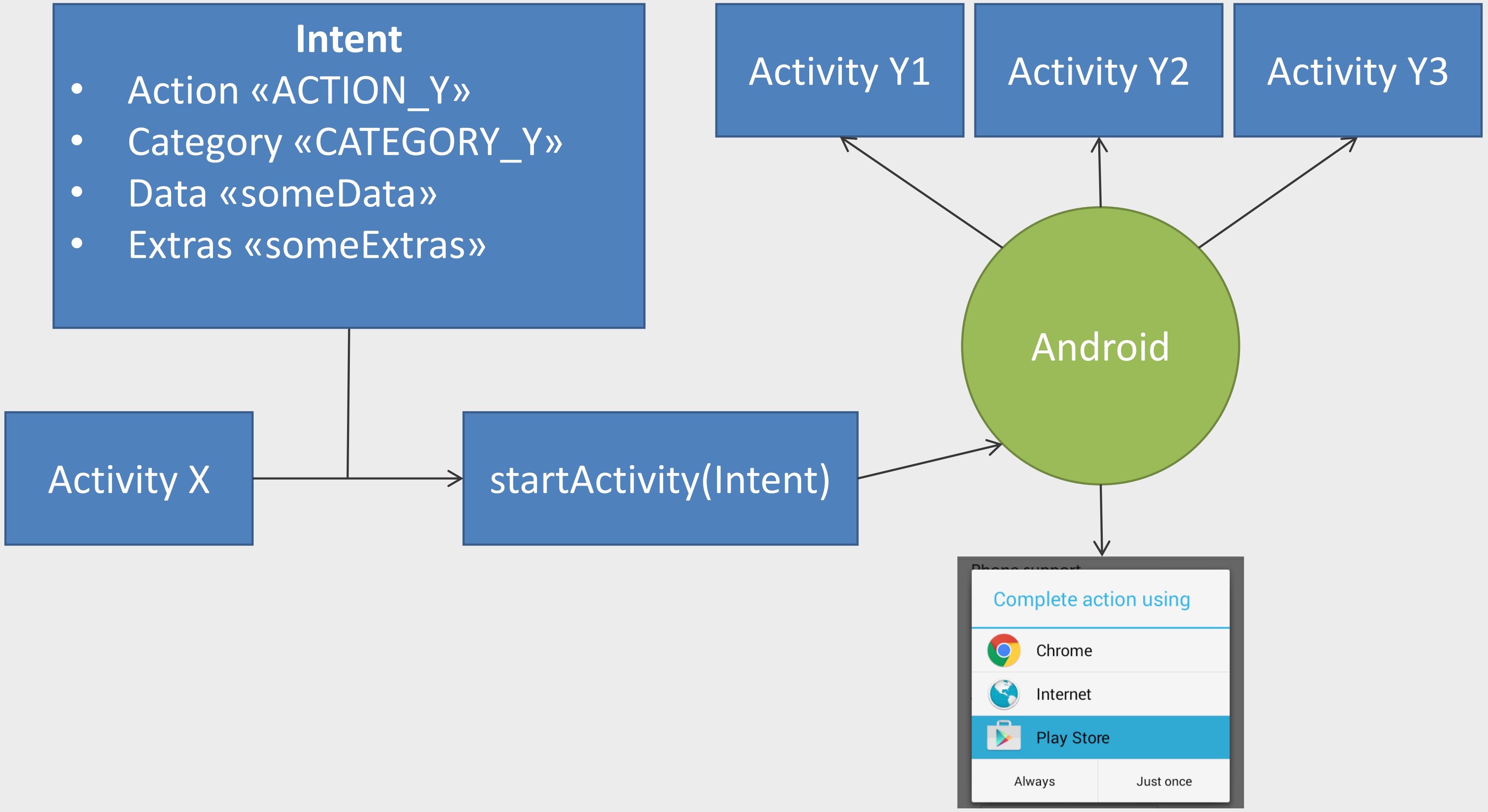
Connect multiple Activities – Explicit Intent

Paixon



Connect multiple Activities – Implicit Intent

Paixon



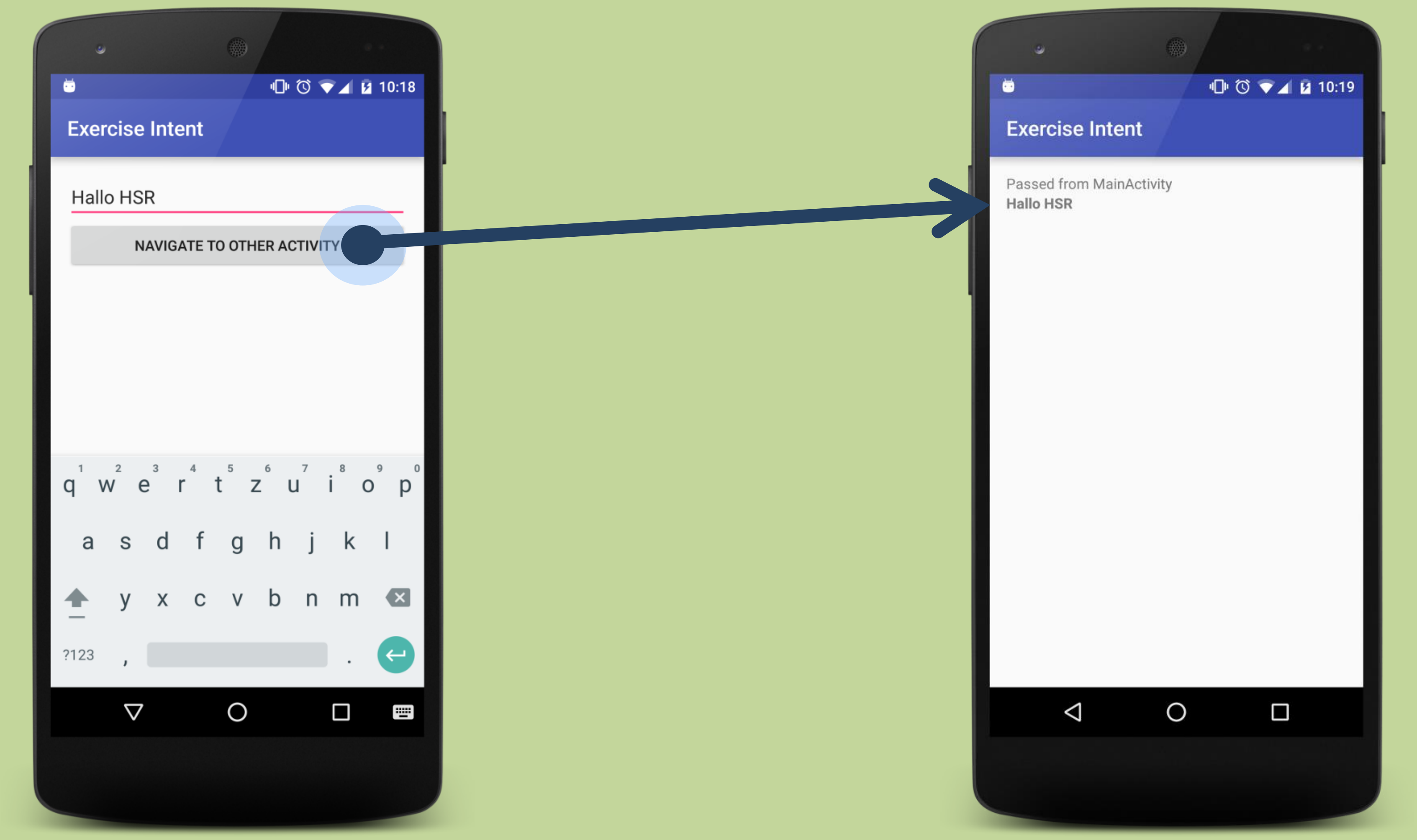
Intent example

```
// Intent erstellen (this = Context)
Intent intent = new Intent(this, TargetActivity.class);

// Daten in den Intent packen für die neue Activity
// Es können nur Value Types (int, string, long, ... ) übergeben werden
intent.putExtra("MyKey", "Die Daten");

// Neue Activity durch Android starten
startActivity(intent);
```

```
// In der onCreate()-Methode der TargetActivity:
// Intent in TargetActivity abfragen
Intent intent = getIntent();

// Extra Daten abfragen
String data = intent.getStringExtra("MyKey");
```

Übung Intent

Paixon



Übung Intent

Aufgaben:

1. Verstehe das Konzept hinter den loose gekoppelten Komponenten

2. Verstehe das Konzept eines Intents
   - Wie wird ein Intent erstellt?
   - Wie kann ich einem Intent daten mitgeben? (Extras)

3. Stelle sicher, dass wenn auf den “Navigate”-Button geklickt wird der Text aus dem EditText an eine neue Activity mitgegeben und dort angezeigt wird.

Projekt: Exercise_Intent

Paixon

Task and Back Stack

Task and Back Stack

Task

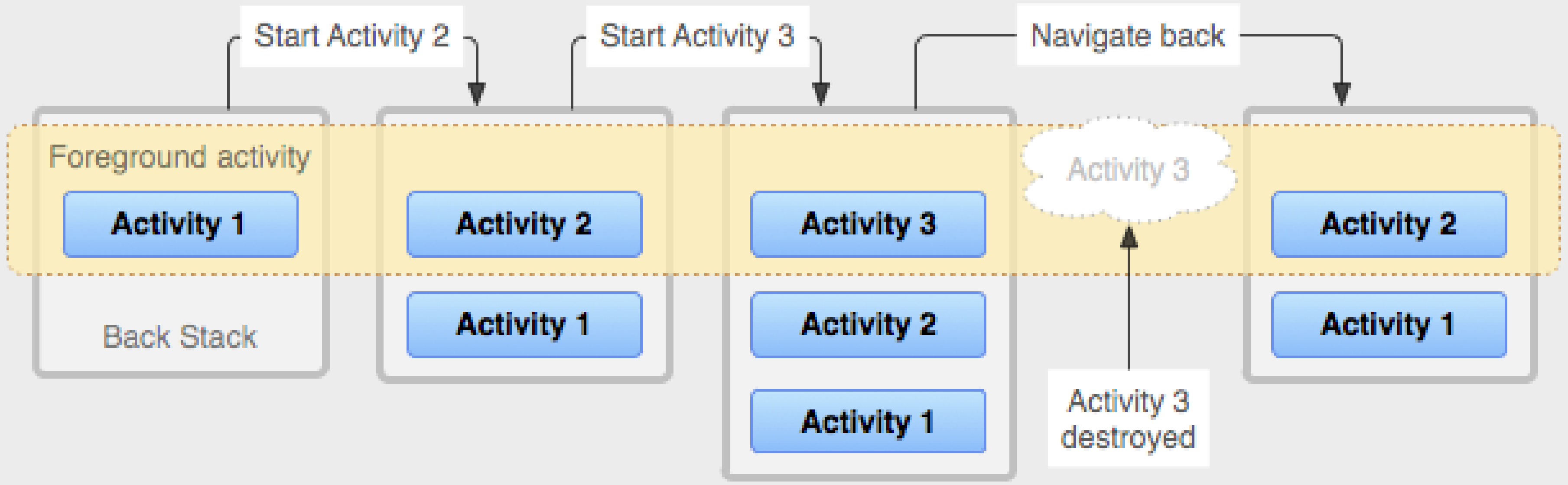
A task is a collection of activities that users interact with when performing a certain job.

Back Stack

The activities are arranged in a stack (the "back stack"), in the order in which each activity is opened.

Task and Back Stack

Paixon


Start Activity 2
Start Activity 3
Navigate back
Foreground activity
Activity 1
Back Stack
Activity 2
Activity 1
Activity 3
Activity 2
Activity 1
Activity 3
Activity 3 destroyed
Activity 2
Activity 1