

Native Mobile Applications

TalTech, Andres Käver, 2020-2021, Fall semester

Skype: akaver Email: akaver@itcollege.ee



Course topics - Android

2

- ▶ Android overview
- ▶ Basic Android Studio usage
- ▶ Kotlin language
- ▶ UI creation
- ▶ App lifecycle and state
- ▶ Local storage and data access (SQLite)
- ▶ Sensors (proximity, geomagnetic, motion, GPS, ...)
- ▶ Web services (REST API)

Android

3

- ▶ Operating system, devised for mobile equipment (mostly)
- ▶ Usage: phones, tablets, TV-s, watches, glasses, cars, laptops, cameras, game consoles, ...
- ▶ Market share among smartphones – ca 86% (iOS 13%)
- ▶ Open source project
- ▶ Google apps and services are closed source (mail, map, etc.)

Short history

- ▶ 2003 – founded (lead: Andy Rubin)
 - ▶ Initial idea – OS for cameras
 - ▶ New plan – Mobile OS, (others: Symbian/Nokia and Win Mobile)
- ▶ 2005 – Google acquires the whole project
- ▶ 2007 – Open Handset Alliance
 - ▶ Google, HTC, Sony, Samsung, Dell, Motorola, LG, Qualcomm, Intel, etc...
- ▶ 2008 – Android 1.0 (HTC Dream, no touchscreen)
- ▶ 2009 – Android 1.5 Cupcake (iPhone 2007, iPhone 3G 2008)
- ▶ 2010 – Android 2.2 Froyo, 2.3 Gingerbread

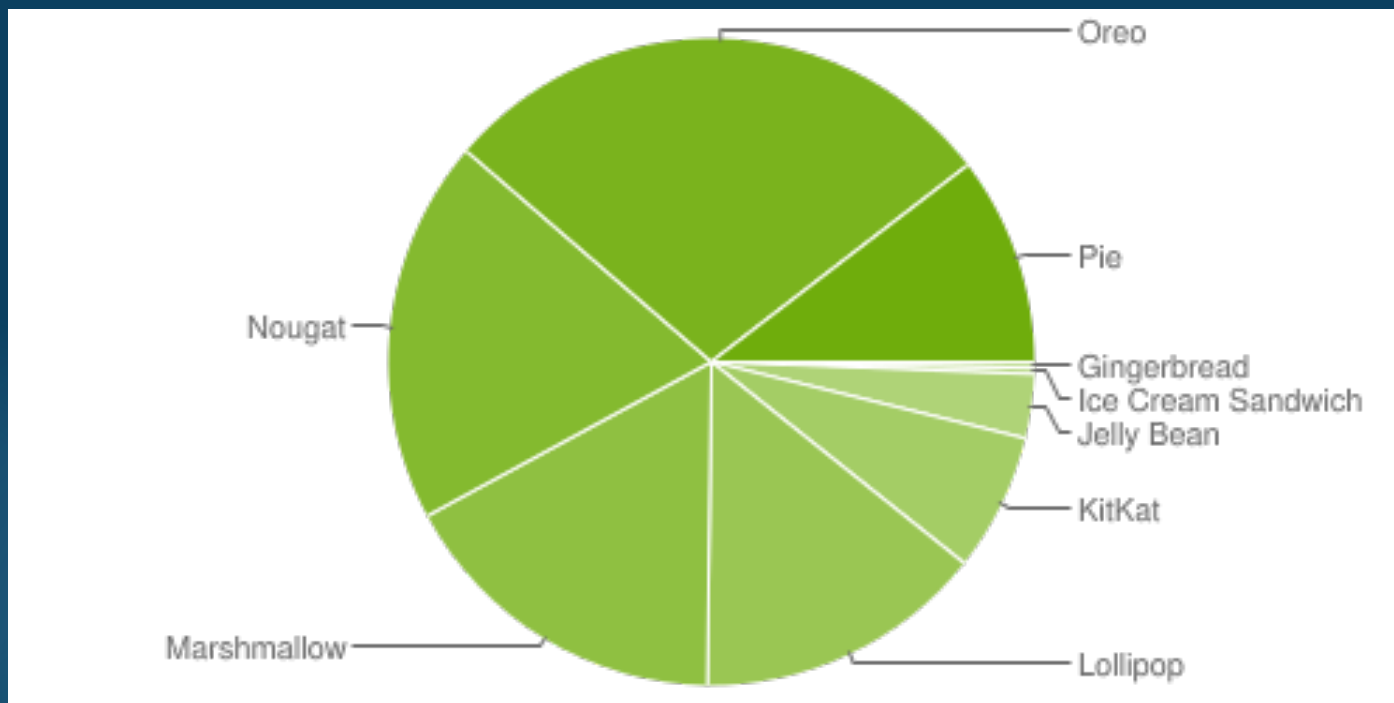
Short history

5

- ▶ 2011 – Android 3.0 Honeycomb (tablets only)
- ▶ 2011 - Android 4.0 Ice Cream Sandwich
 - ▶ HOLO UI
- ▶ 2014 - Android 5 Lollipop
 - ▶ Material design
 - ▶ Dalvik vs ART (Android Runtime) (JIT or precompile, garbage collection)
- ▶ 2015 - Android 6 Marshmallow
- ▶ 2016 – Android 7 Nougat
- ▶ 2017 – Android 8 Oreo
- ▶ 2018 – Android 9 Pie
- ▶ 2019 – Android 10 Q – moves closer to iOS, security clamped down

Version distribution

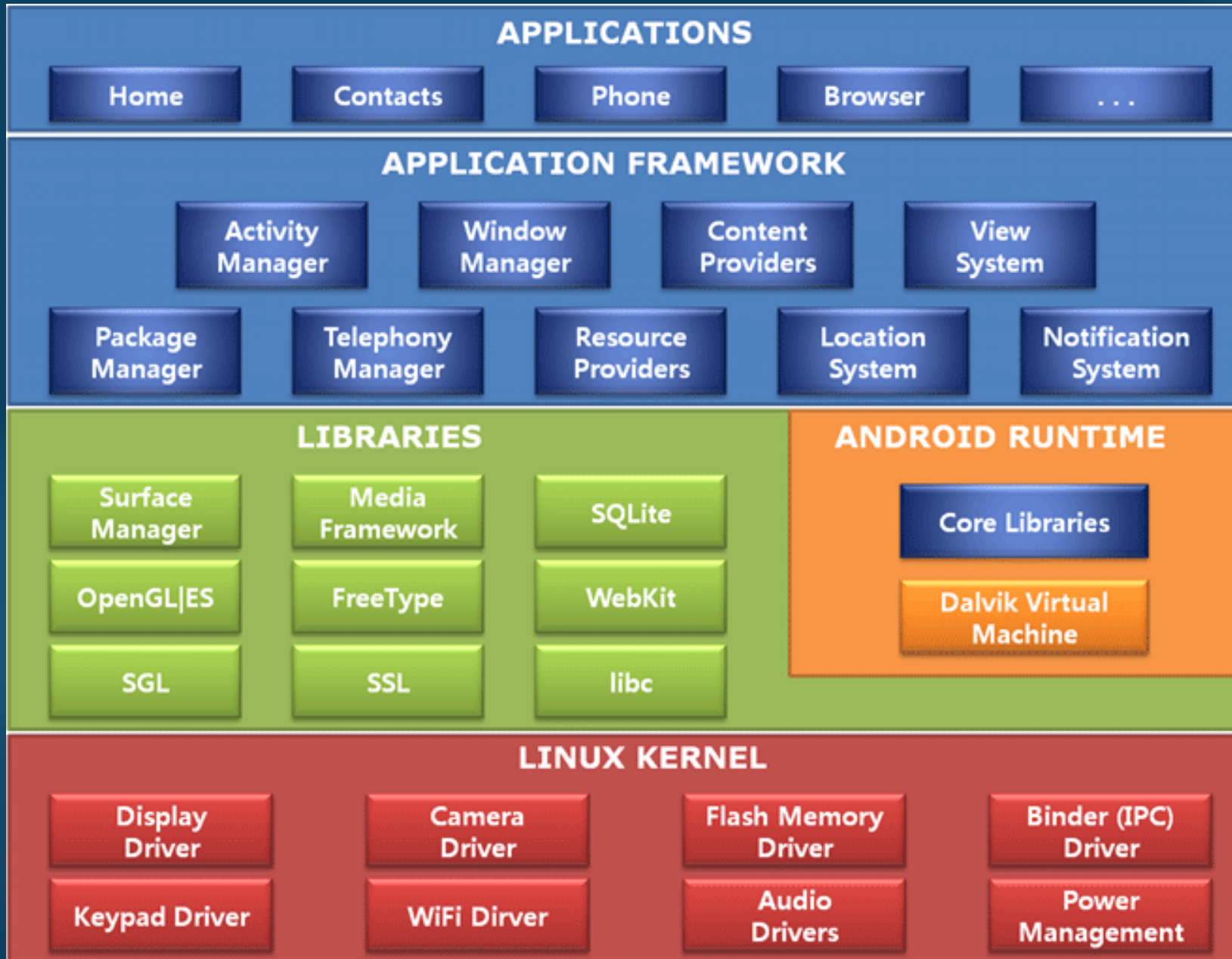
6



Pie – 9.X – 10%
Oreo – 8.X – 28%
Nougat – 7.X – 19%
Marshmallow – 6.X – 17%
Lollipop – 5.X – 14%
KitKat – 4.4 – 7%

4.4 and higher - ca 95%

► Source: <http://developer.android.com/about/dashboards/index.html>



Android architecture

Android - App types

8

- ▶ NDK - C/C++
 - ▶ Close to hardware and operating system
- ▶ SDK - Native <- this course!!!!
 - ▶ Kotlin/Java (ART/Dalvik), using system libraries
- ▶ Hybrid – React Native, Ionic, etc.
- ▶ Cross platform – Xamarin (C#), Flutter (Dart) etc.
- ▶ Html/JS – Progressive Web Apps
 - ▶ One codebase/layout for different platforms
 - ▶ Problems with UI, weak access to hardware
- ▶ There is also course on Hybrid Mobile Apps - ICD0018 (fall semester)

Android – App architecture

AndroidManifest.xml

- ▶ The manifest file presents essential information about your app to the Android system, information the system must have before it can run any of the app's code.
- ▶ Describes the components of the application — the activities, services, broadcast receivers, and content providers that the application is composed of.
- ▶ Declares which permissions the application must have in order to access protected parts of the API and interact with other applications.
- ▶ Declares the minimum level of the Android API that the application requires.
- ▶ Declares hardware requirements.

Android – AndroidManifest.xml

Example -

10

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ee.taltech.akaver.test20_01">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Android – code

MainActivity.kt

11

```
package ee.taltech.akaver.test20_01

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

Android – Layout - Resource files

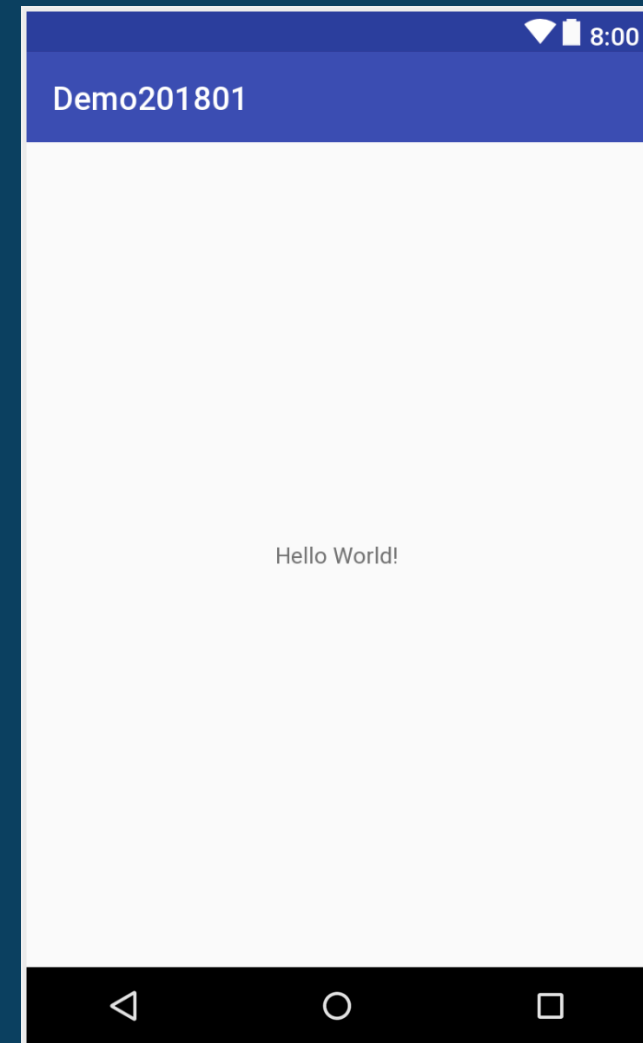
activity_main.xml

12

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



Android – Other resources

13

- ▶ Images
- ▶ Animations
- ▶ Menu
- ▶ Strings
- ▶ Misc files

Android - apk

14

- ▶ Android Application Package
- ▶ ZIP file, combines all the resources and java bytecode
- ▶ Signed with developer key
- ▶ Developer key must be the same from version to next version
- ▶ Don't lose your keys (passwords)
- ▶ Android Studio takes care of APK creation
- ▶ APK-s can be downloaded from store, using 3-rd party utilities
- ▶ Resources can be used as is
- ▶ Most elements/code can be decompiled/recompiled

Android – Google Play - appstore

15

- ▶ Almost no review process
- ▶ Problems are dealt with afterwards
- ▶ App hijacking, etc. are real problems

Android – App security

16

- ▶ Every app works in its own private virtual machine (Zygote)
- ▶ Need permission for system resources/hardware (confirmed on app install)
- ▶ Data is private, no other app can access directly other app data
- ▶ Everything is possible on rooted device
- ▶ End user is the weakest link

Android – dev problems

17

- ▶ Gazillion different hardware devices and capabilities
- ▶ Lots of different Android implementations
 - ▶ Samsung TouchWiz
 - ▶ HTC Sense
 - ▶
- ▶ Migration to newer versions very slow (or not done at all)
- ▶ Rooted phones
- ▶ Ca 2X time spent on development compared to iOS
- ▶ Ca 60% better income on iOS

Android – testing on devices

18



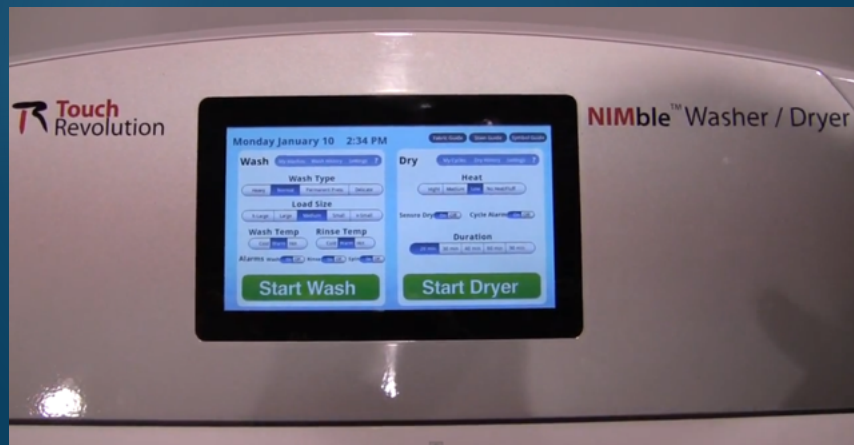
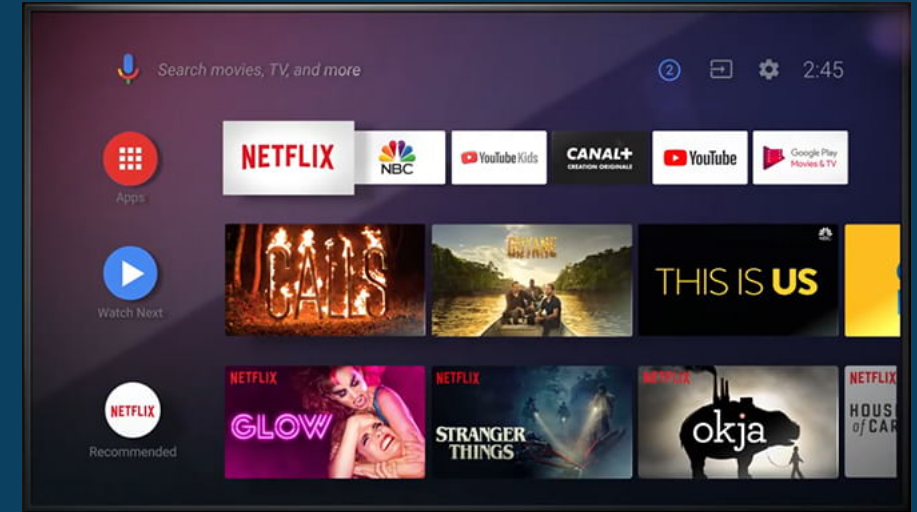
Android - devices

19



Android - devices

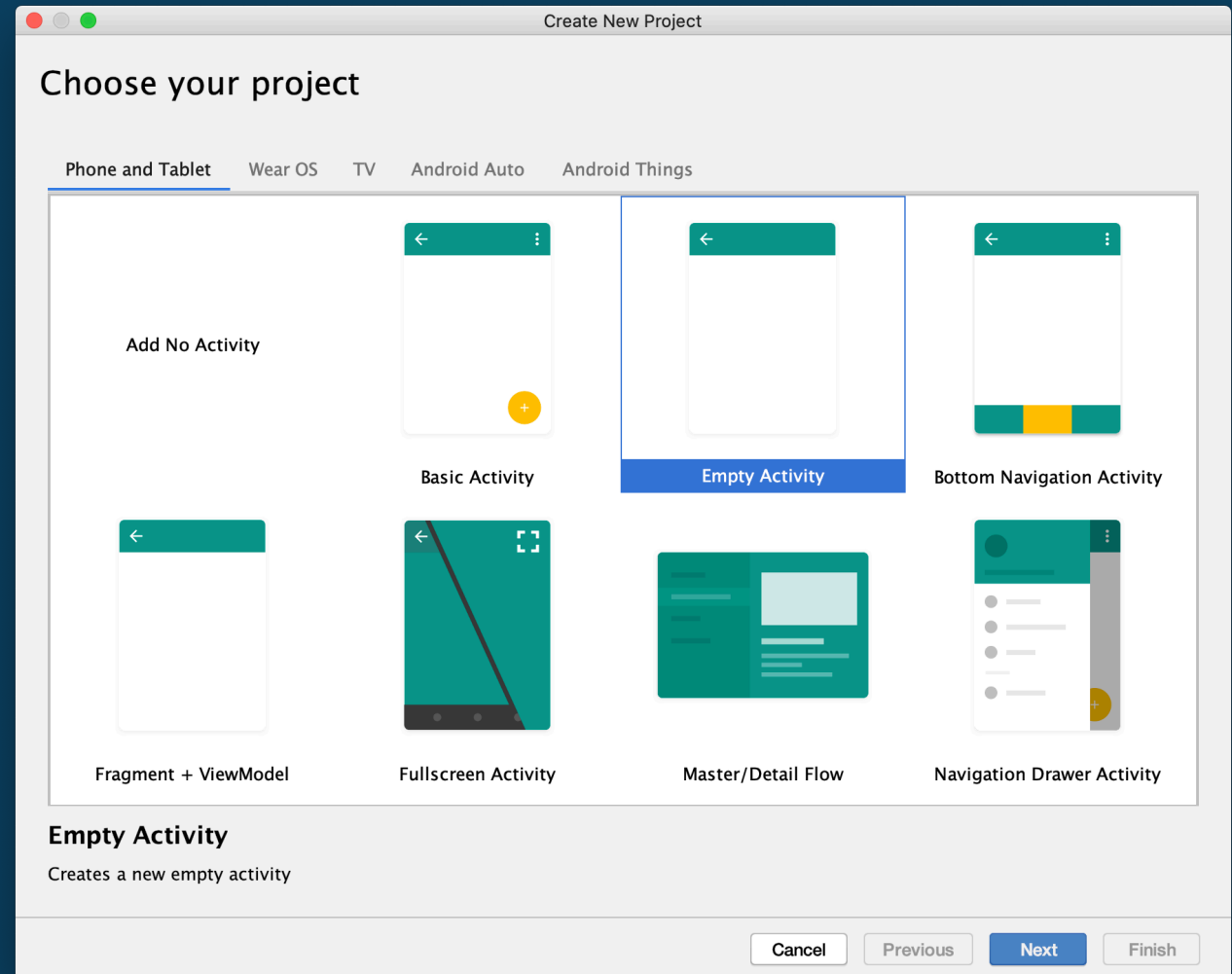
20



Android – Hello World

21

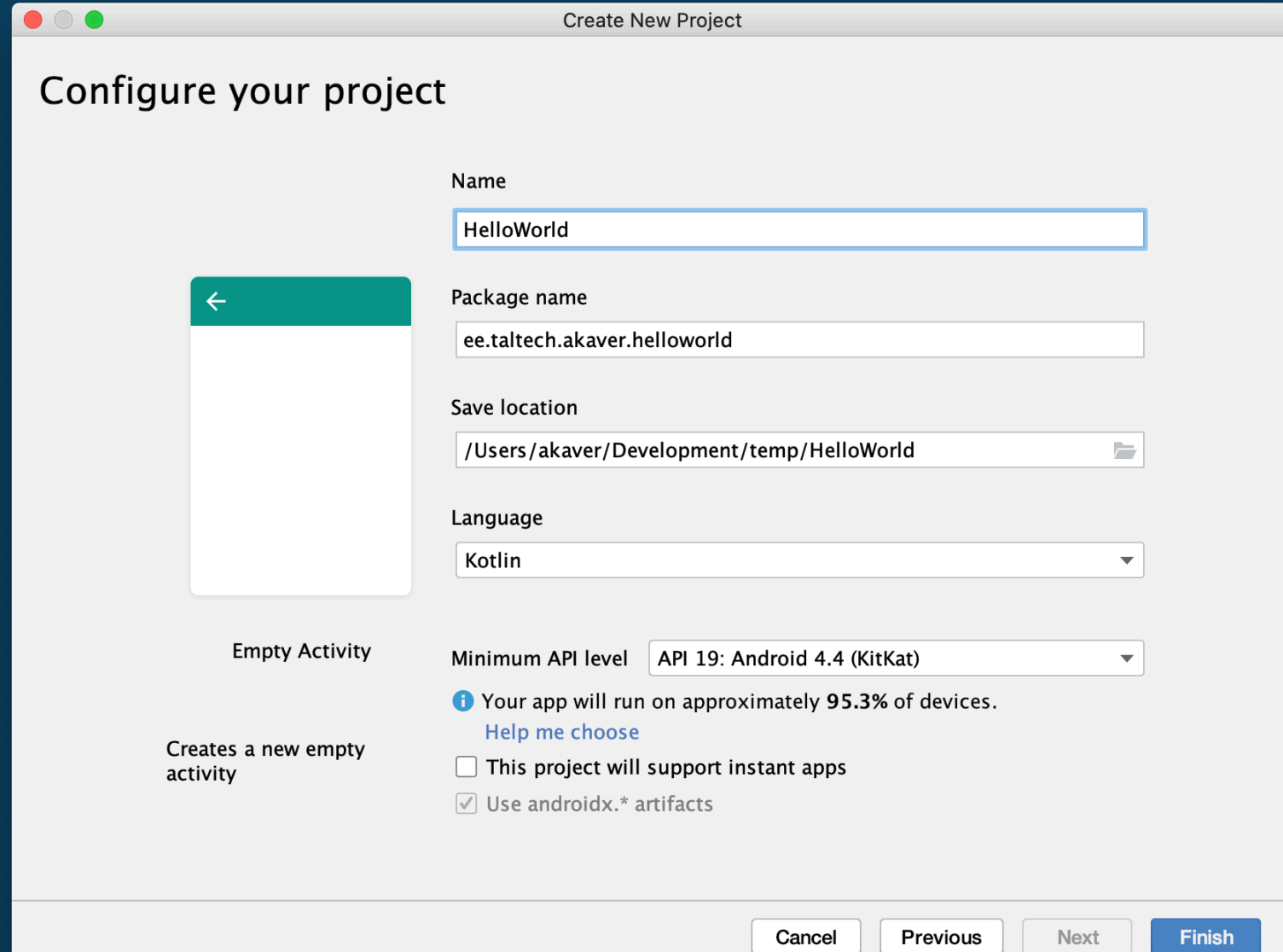
- ▶ Start a new project
- ▶ Typically start from Empty Activity



Android – Hello World

22

- ▶ Careful with naming – very hard to change afterwards. Especially when app is published.
- ▶ Choose min API level
- ▶ What language?
 - ▶ Kotlin or Java



Create New Project

Configure your project

Name: HelloWorld

Package name: ee.taltech.akaver.helloworld

Save location: /Users/akaver/Development/temp/HelloWorld

Language: Kotlin

Minimum API level: API 19: Android 4.4 (KitKat)

Your app will run on approximately 95.3% of devices.
[Help me choose](#)

☐ This project will support instant apps

☒ Use androidx.* artifacts

Empty Activity

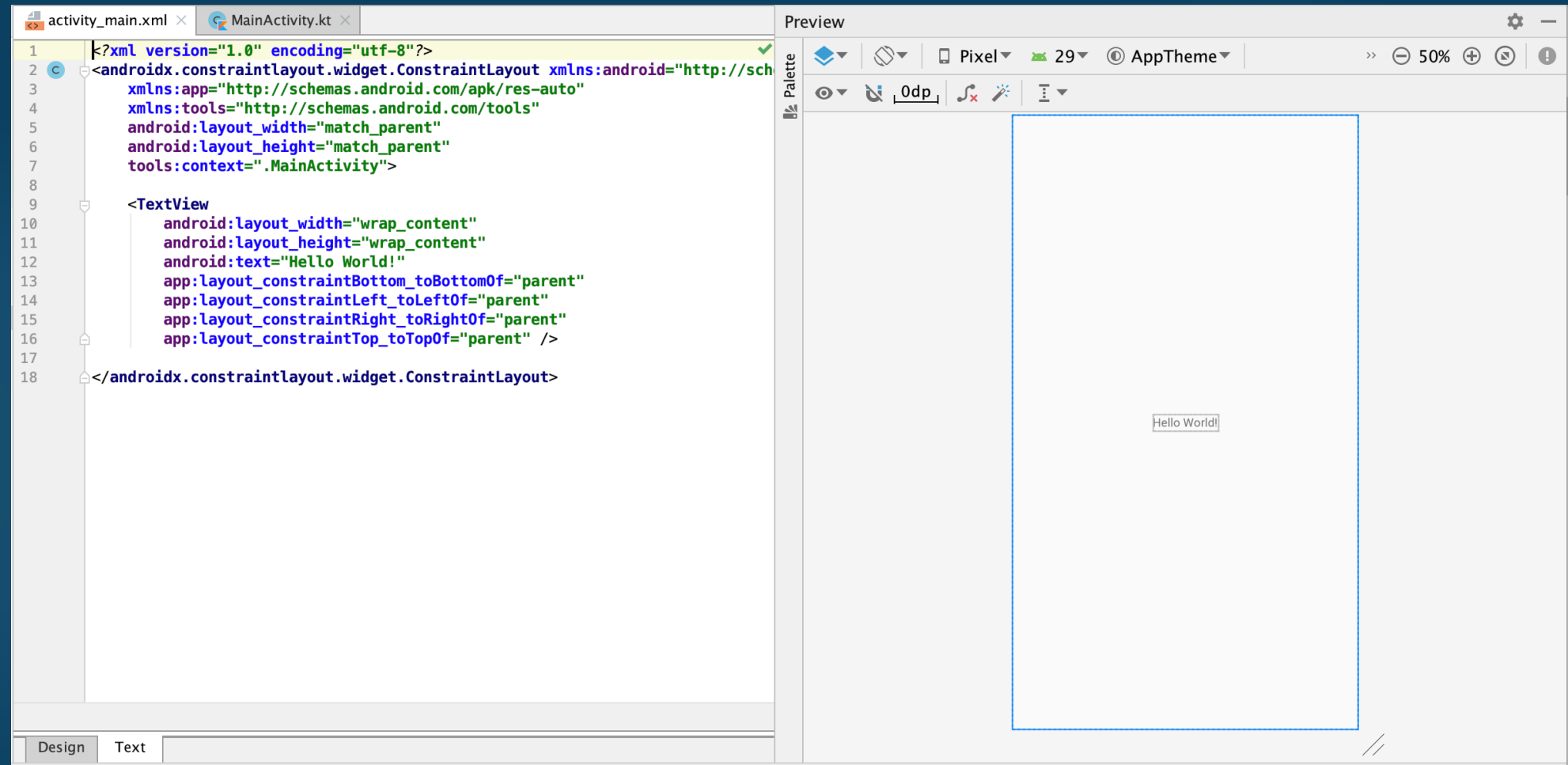
Creates a new empty activity

Cancel Previous Next Finish

Android – Hello World

23

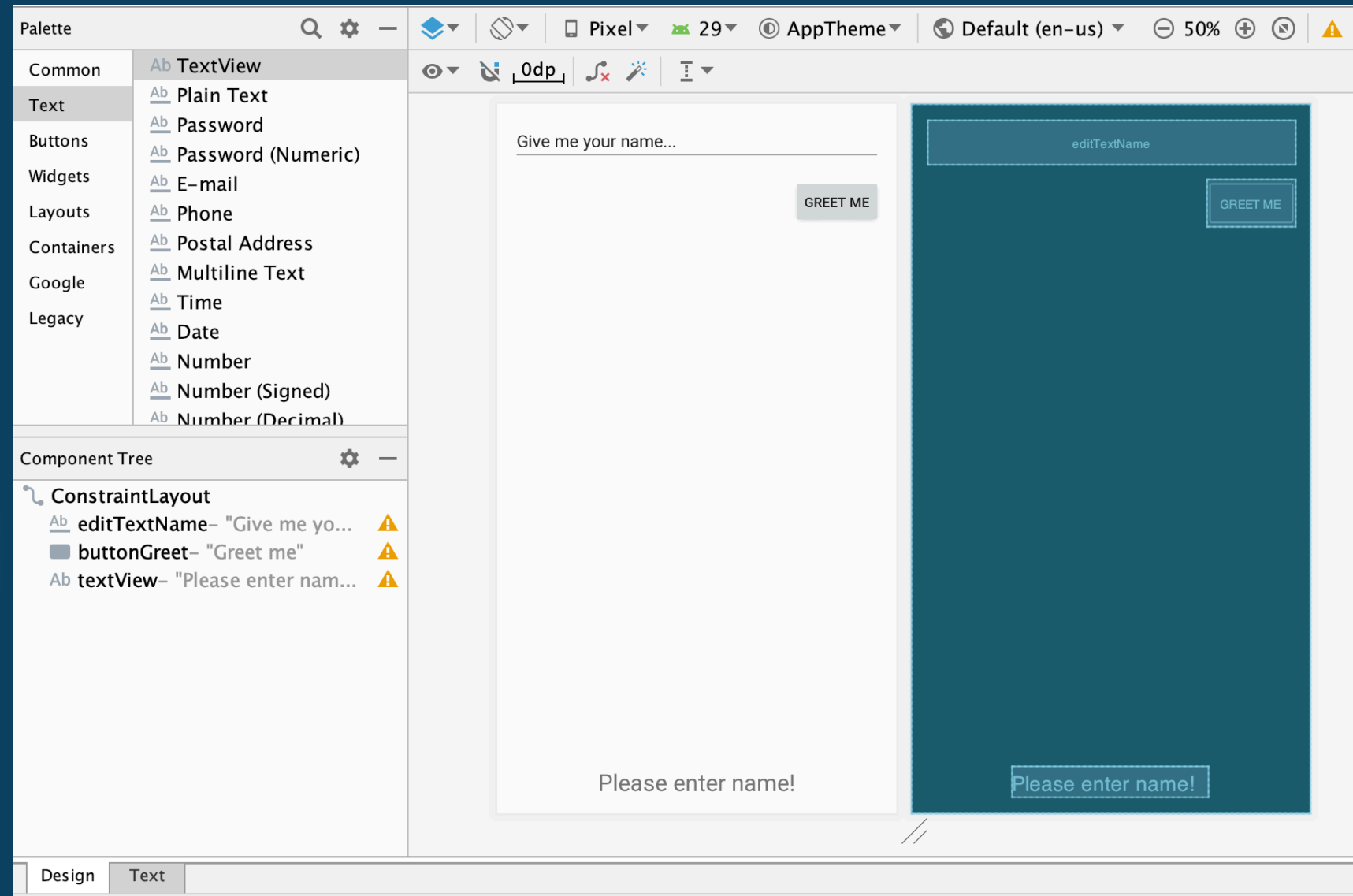
- Design the UI, ConstraintLayout



Android – Hello World

24

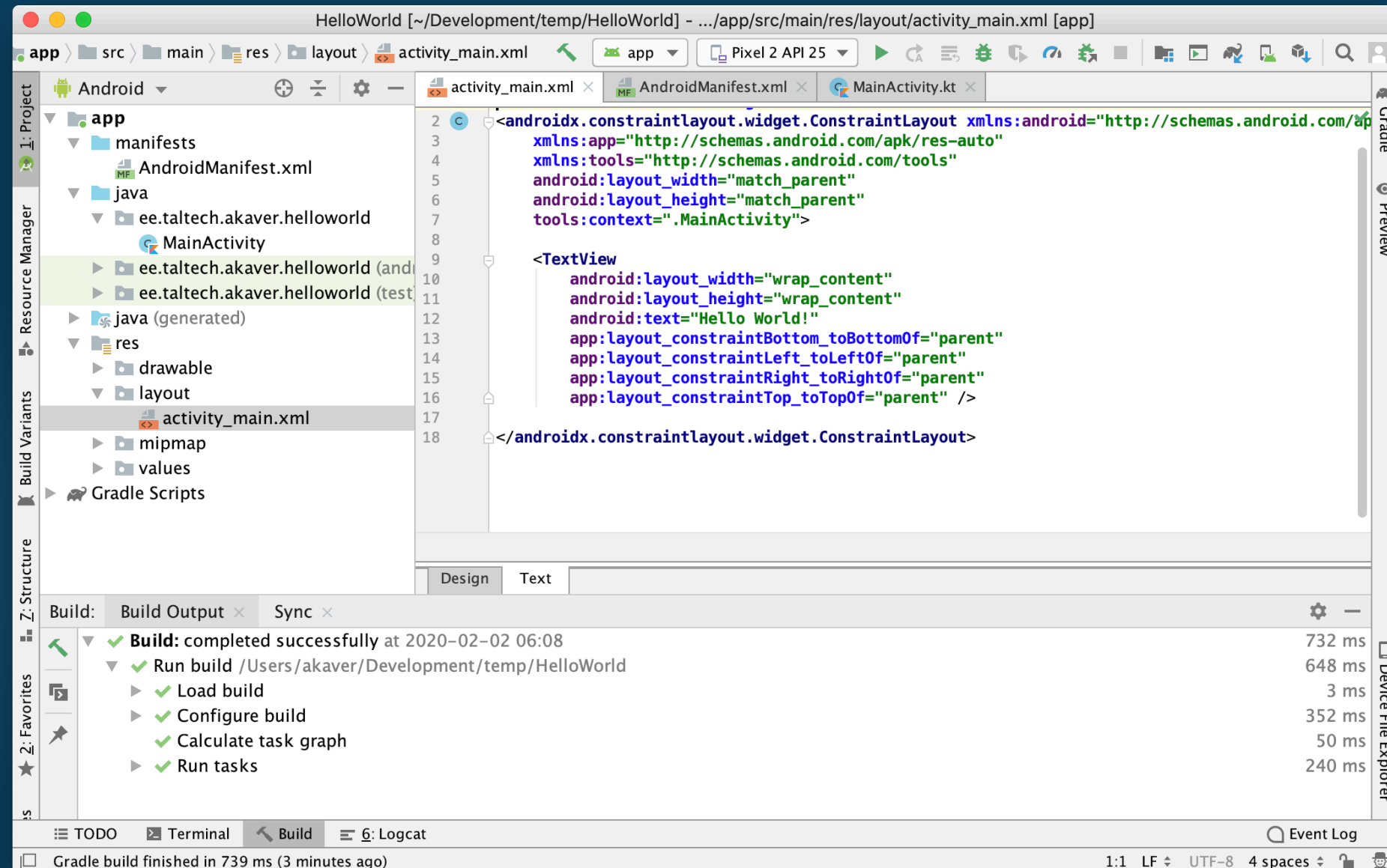
- ▶ TextView
- ▶ Button
- ▶ EditText



Android – Hello World

25

- ▶ Android Studio
- ▶ Based on IntelliJ



Android – Hello World, Layout XML

26

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
```

```
<EditText
    android:id="@+id/editTextName"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:autofillHints=""
    android:ems="10"
    android:hint="Please enter name"
    android:inputType="textPersonName"
    android:text="Give me your name..."
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/buttonGreet"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:text="Greet me"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextName" />
```

```
<TextView
    android:id="@+id/textViewGreeting"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:text="Please enter name!"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Android – Hello World

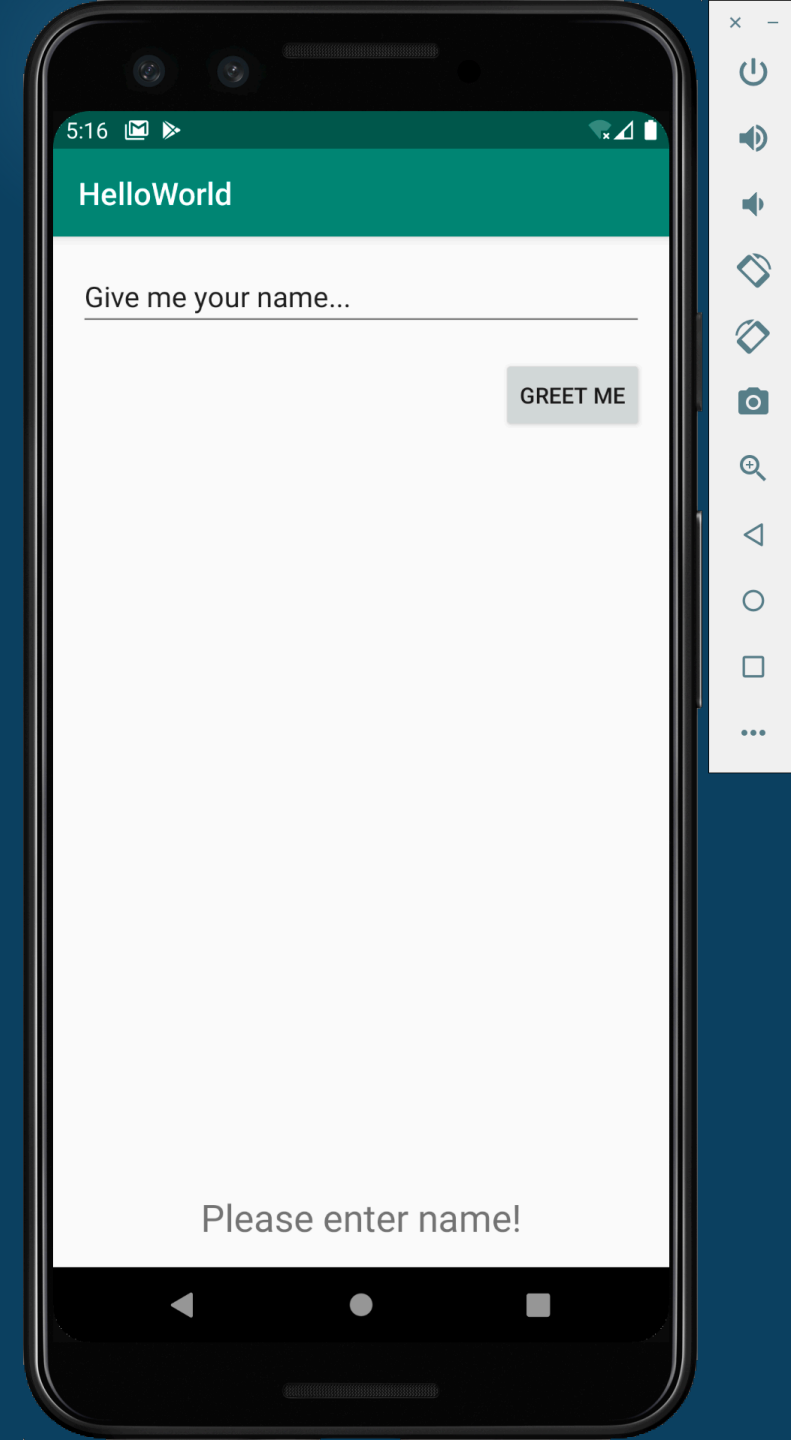
27

- ▶ Install/Start emulator from AVD Manager (choose one with Play Store/Google Services)



Android – Hello World

- ▶ Launch your app (Click play icon)



Android – Hello World

29

- ▶ Write minimal code to get value from text input on button click and copy it over to textview.
- ▶ Add this attribute to button in XML
 - ▶ **`android:onClick="bottonGreetClicked"`**

Android – Hello World

30

- ▶ Code in MainActivity.kt
- ▶ Compare code with Java version on next slide.

```
package ee.taltech.akaver.helloworld

import ...

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    fun buttonGreetClicked(view: View) {
        textViewGreeting.text = "Hello, " + editTextName.text + "!"
        val imm = getSystemService(INPUT_METHOD_SERVICE) as InputMethodManager
        imm.hideSoftInputFromWindow(getCurrentFocus()?.windowToken, 0);
    }
}
```

Android – HelloWorld – MainActivity.java

31

```
package com.akaver.helloworld;

import ...

public class MainActivity extends AppCompatActivity {

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

    public void buttonClicked(View view) {
        TextView textView = (TextView) findViewById(R.id.textView);
        EditText editText = (EditText) findViewById(R.id.editText);
        textView.setText("Hello " + editText.getText() + "!");
        editText.setText("");

        InputMethodManager imm = (InputMethodManager) getSystemService(INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(), 0);
    }
}
```

Android

32

► The end!

Android

33