



Programowanie dla Androida

Ubiquitous

Programowanie

- Kod źródłowy w Javie/Kotlinie jest tłumaczony do postaci bytecode'u (DEX)
- W momencie instalacji na urządzeniu DEX jest kompilowany do instrukcji procesora jako Executable and Linkable Format (ELF).
- W momencie uruchomienia ładowany jest kod ELF, co znacznie przyspiesza jego działanie i oszczędza baterię.

Biblioteki

- Zawierają w sobie wsparcie dla funkcji oferowanych przez urządzenie.
- Często są to wrapery dla bibliotek napisanych w C/C++

Przykładowe biblioteki

- android.app - model aplikacji
- android.database - dostęp do baz danych
- android.graphics - podstawowa grafika 2D
- android.opengl - grafika 3D
- android.net - obsługa sieci
- android.webkit - obsługa przeglądarki www

Application Framework

- Zawiera usługi umożliwiające uruchamianie i działanie aplikacji.
- Zawiera:
 - Activity Manager - steruje cyklem życia aplikacji.
 - Content Providers - umożliwia współdzielenie danych między modułami.
 - Resource Manager - zarządza zasobami spoza aplikacji.

Application Framework

- Notifications Manager - umożliwia wyświetlanie powiadomień.
- View System - widoki składające się na interfejs użytkownika.
- Package Manager - umożliwia zarządzanie zainstalowanymi aplikacjami.
- Telephony Manager - obsługuje funkcjonalność telefonu.
- Location Manager - dostęp do usług lokalizacji.

Programowanie

- Twórcy Androida byli mocno nastawieni na wielokrotne wykorzystanie komponentów.
- Stąd program na Androida składa się z szeregu niezależnych od siebie modułów nazywanych Aktywnościami (Activities), które mogą być wielokrotnie wykorzystywane.
- Aktywność nie wie kiedy zostanie wywołana i nie może bezpośrednio zależeć od innych aktywności (mieć dostępu do jej metod lub pól).
- Aktywność dziedziczy po klasie Activity.

- Aby aktywności mogły się ze sobą komunikować, muszą wykorzystywać mechanizm Intencji (Intent).
- Intencje zawierają opis operacji i opcjonalnie dane z nią związane.
- Intencja może wprost wskazać klasę, która ma zostać wywołana (explicite) albo zostawić wybór klasy ART (implicit).
- Intencje nadawcze (Broadcast Intents) - rozsyłają komunikaty do wszystkich aplikacji w systemie, które zapiszą się na ich nasłuchiwanie (Broadcast Receivers).

Usługi

- Usługa (Service) to proces, który jest pozbawiony interfejsu użytkownika i może wykonywać zadania w tle.
- Może zostać uruchomiony i zarządzany przez aktywność.
- Usługi mogą też wyświetlać wprost powiadomienia.

Manifest aplikacji

- Jest to plik XML opisujący aplikację. Zawiera informacje o:
 - Aktywnościach,
 - Usługach (Services),
 - Odbiorcach (Broadcast Receivers),
 - Dostawcach danych,
 - Uprawnieniach aplikacji.

Zasoby

- W pakiecie aplikacji znajdują się też różne zasoby, takie jak:
 - Łańcuchy tekstowe,
 - Grafiki,
 - Czcionki,
 - ...

Czas życia aplikacji

- Mimo niezłych specyfikacji technicznych współczesnych telefonów są one nadal uważane za urządzenia o ograniczonych zasobach.
- Dlatego system zarządza aplikacjami inaczej niż biurkowy system operacyjny.
- Gdy zaczyna brakować zasobów, procesy są zabijane, a ich zasoby zwalniane.
- O kolejności zabijania decyduje priorytet procesu.

Czas życia aplikacji

Proces pierwszoplanowy

Najwyższy priorytet

Proces widoczny

Proces - usługa

Proces w tle

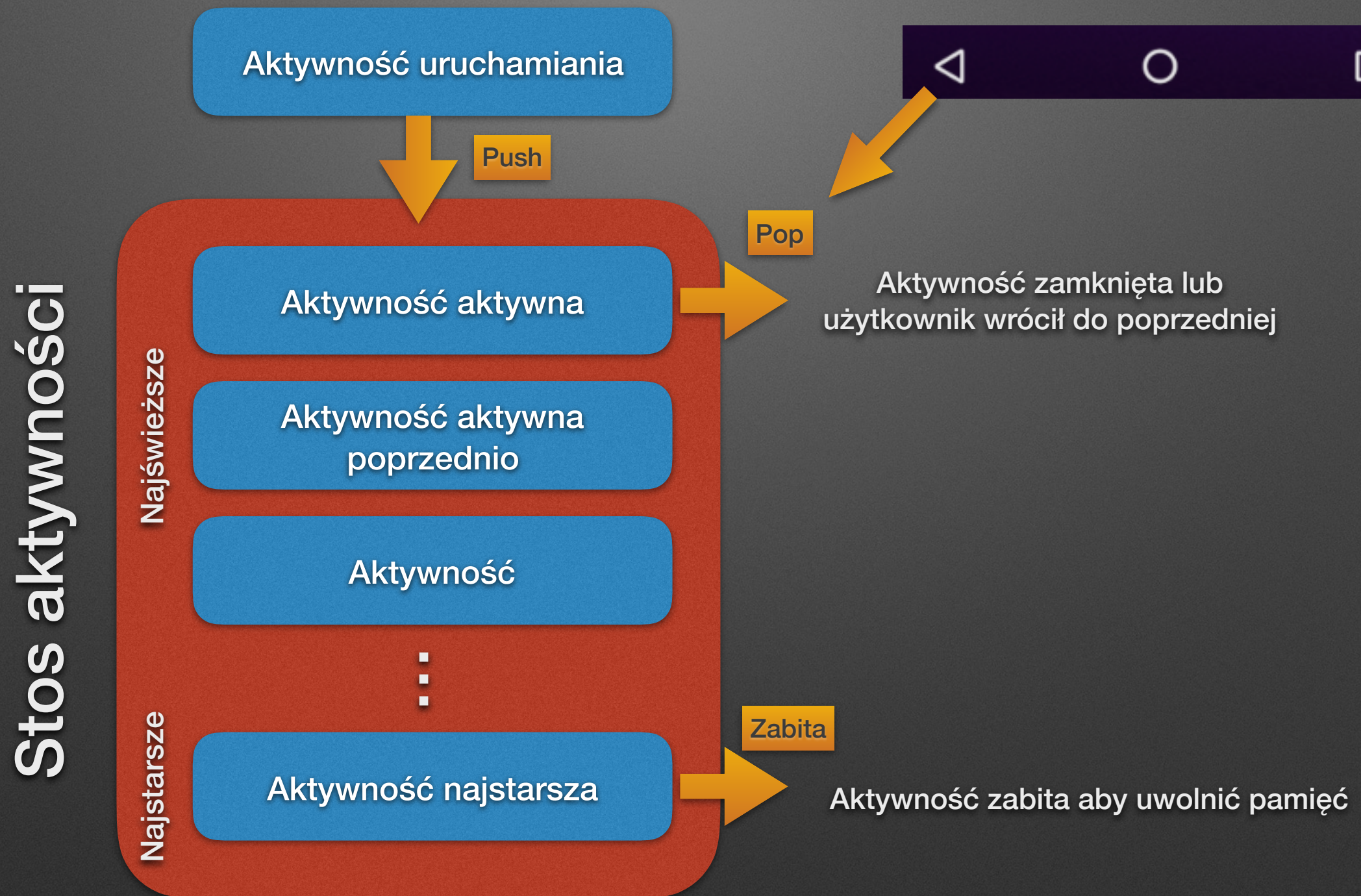
Proces pusty

Najniższy priorytet

Czas życia aplikacji

- Stan aplikacji jest wprost zależny od stanu jej aktywności.
- W trakcie działania systemu wszystkie aktywności są przechowywane na stosie (Activity Stack).
- Nowa aktywność jest zawsze umieszczana na szczycie stosu.
- Po zamknięciu aktywności (np. przyciskiem „Back”), jest ona zdejmowana ze stosu, a uaktywniana jest ta, która była poprzednio.

Czas życia aplikacji



Czas życia aplikacji

- Aktywność może znajdować się w jednym ze stanów:
 - Aktywna - na szczycie stosu, widoczna na ekranie i w interakcji z użytkownikiem,
 - Wstrzymana (Paused) - jest widoczna na ekranie, ale może być częściowo przysłonięta. W każdej chwili jest gotowa do powrotu na wierzch stosu.
 - Zatrzymana - nie jest widoczna, jest stan jest zachowany, ale w stosunku do wstrzymanej jej usunięcie jest bardziej prawdopodobne.
 - Zabita - aktywność została usunięta ze stosu, jej pamięć zwolniona, a ponowne wyświetlenie wymaga inicjalizacji.

Czas życia aplikacji

- Jeszcze jednym powodem wymuszenia przeładowania aktywności może być zmiana konfiguracji, np. zmiana orientacji ekranu, czcionki systemowej, ...
- Aktywność jest wtedy niszczone i inicjalizowana ponownie.
- Aktywność można skonfigurować aby była niewrażliwa na pewne zmiany.

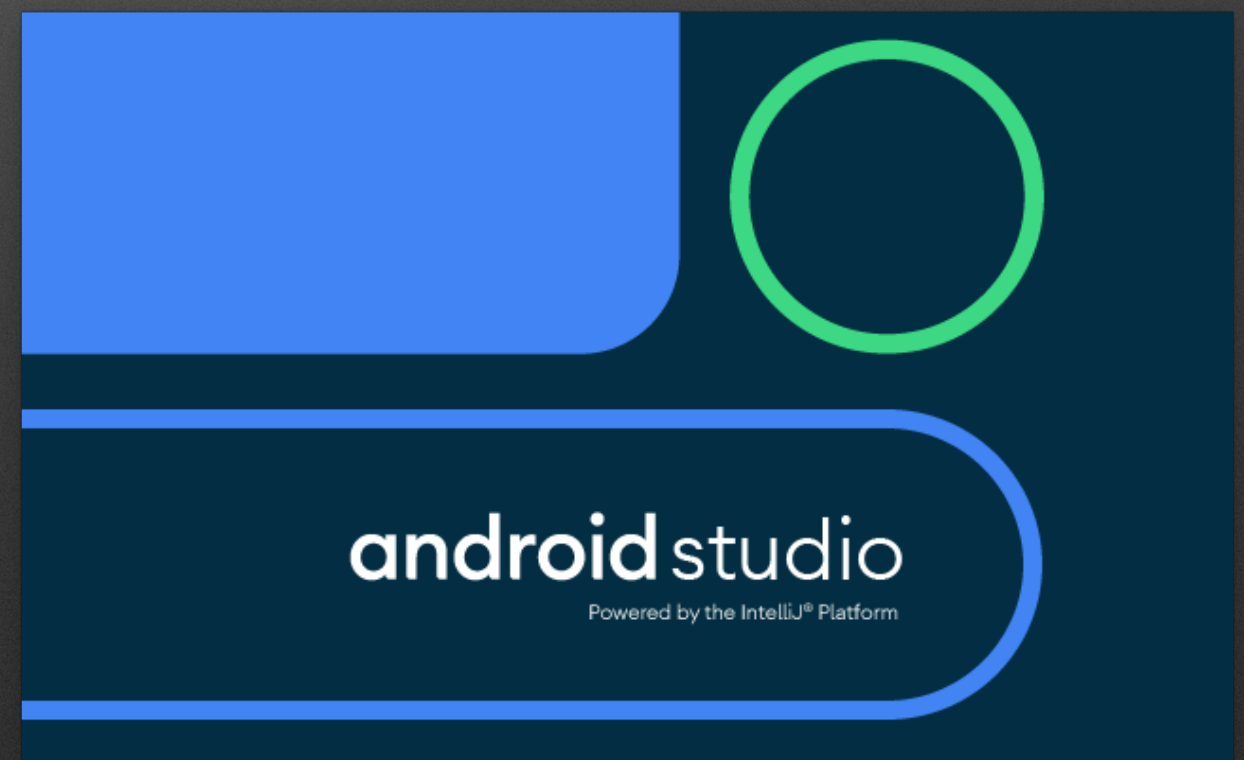
Narzędzia programistyczne

- Android Studio - oficjalne środowisko IDE od Google
- IntelliJIDEA
- Eclipse
- Android-IDE
- Cordova
- Corona
- Xamarin
- CppDroid
- ...

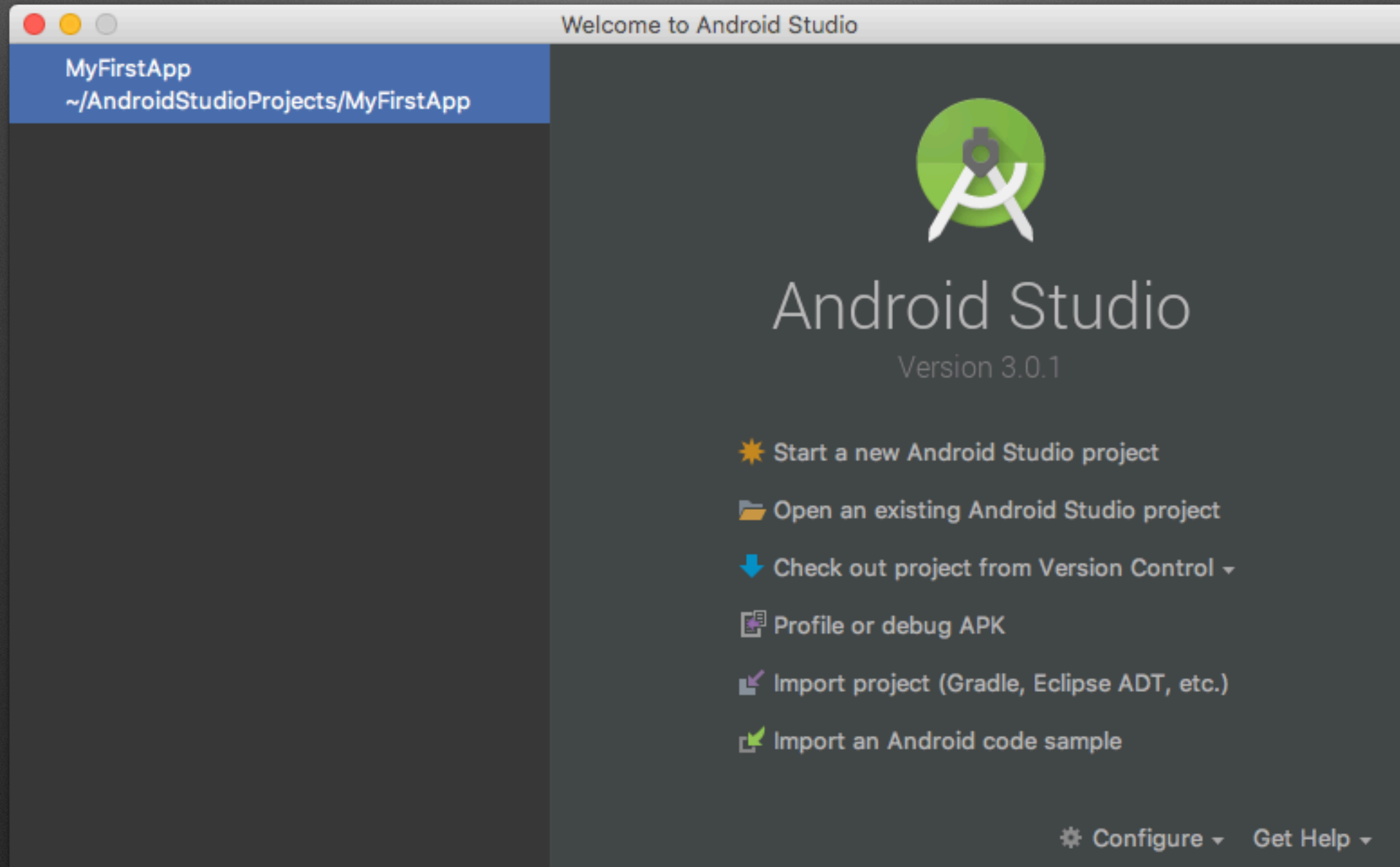
Android Studio



- Dostosowana do potrzeb Androida wersja IntelliJ IDEA
- Dostępne dla Windows, macOS i Linux'a
- Początki od maja 2013 r.
- Aktualnie wersja 3.6.1

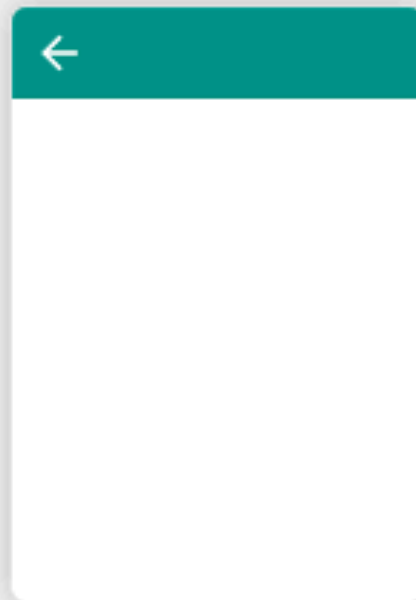


Android Studio





Configure Your Project



Empty Activity

Creates a new empty activity

Name

My Application

Package name

com.example.myapplication

Save location

/Users/bartlomiejpredki/Desktop/9.45/MyApplication

Language

Kotlin

Minimum SDK

API 23: Android 6.0 (Marshmallow)

Your app will run on approximately **62.6%** of devices.[Help me choose](#)

Use legacy android.support libraries ?

Cancel

Previous

Next

Finish

**ANDROID PLATFORM
VERSION****API LEVEL****CUMULATIVE
DISTRIBUTION**

4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99.6%
4.2 Jelly Bean	17	98.1%
4.3 Jelly Bean	18	95.9%
4.4 KitKat	19	95.3%
5.0 Lollipop	21	85.0%
5.1 Lollipop	22	80.2%
6.0 Marshmallow	23	62.6%
7.0 Nougat	24	37.1%
7.1 Nougat	25	14.2%
8.0 Oreo	26	6.0%
8.1 Oreo	27	1.1%

Marshmallow**Security**

Fingerprint Authentication
Confirm Credential

System

App Linking
Adoptable Storage Devices

Multimedia

4K Display Mode
Support for MIDI
Create digital audio capture and
playback objects
APIs to associate audio and input
devices
List of all audio devices
Updated video processing APIs
Flashlight API
Reprocessing Camera2 API
Updated ImageWriter objects and
Image Reader class

User Input

Voice Interactions
Assist API
Bluetooth Stylus Support

User Interface

Themeable ColorStateLists

Wireless & Connectivity

Hotspot 2.0
Improved Bluetooth Low Energy Scanning

Android for Work

Controls for Corporate-Owned, Single-Use
devices
Silent install and uninstall of apps by
Device Owner
Silent enterprise certificate access
Auto-acceptance of system updates
Delegated certificate installation
Data usage tracking
Runtime permission management
Work status notification

<https://developer.android.com/about/versions/marshmallow/android-6.0.html>

Cancel

OK



Select a Project Template

Phone and Tablet

Wear OS

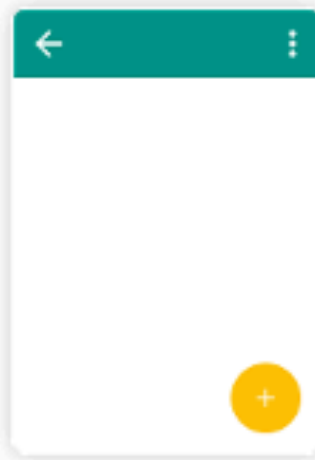
TV

Automotive

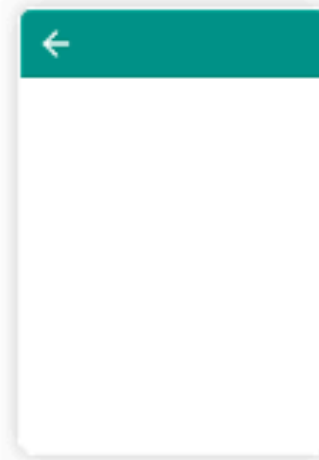
Android Things



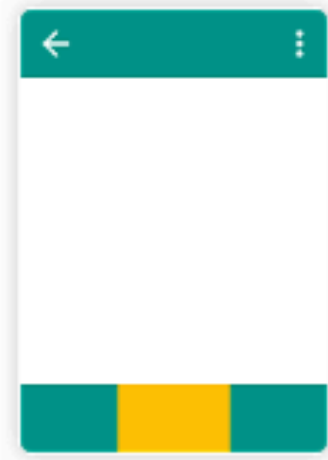
No Activity



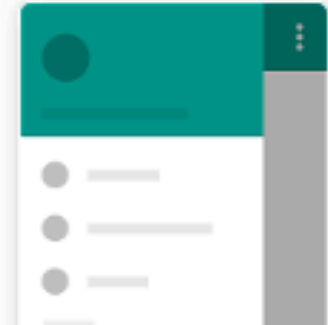
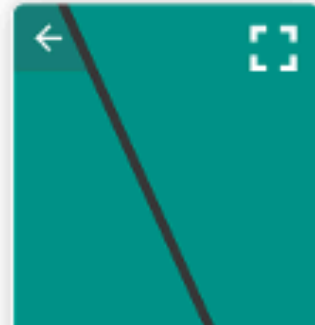
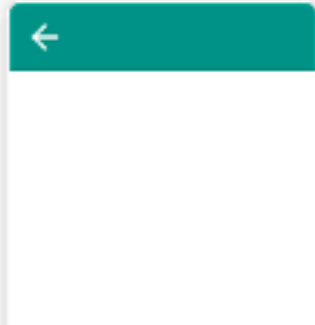
Basic Activity



Empty Activity



Bottom Navigation Activity



Empty Activity

Creates a new empty activity

Cancel

Previous

Next

Finish



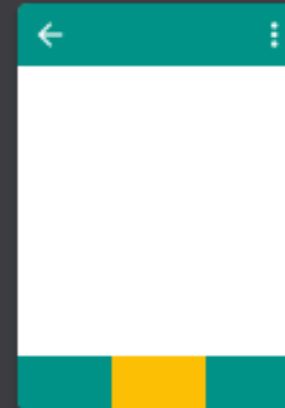
Add an Activity to Mobile



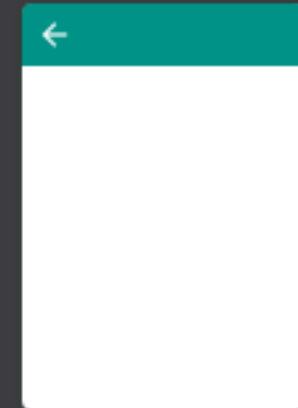
Add No Activity



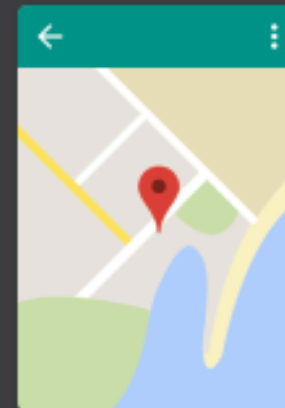
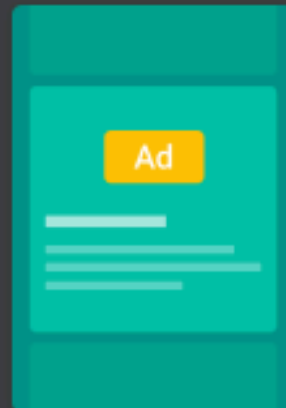
Basic Activity



Bottom Navigation Activity



Empty Activity



Cancel

Previous

Next

Finish



Configure Activity



Creates a new empty activity



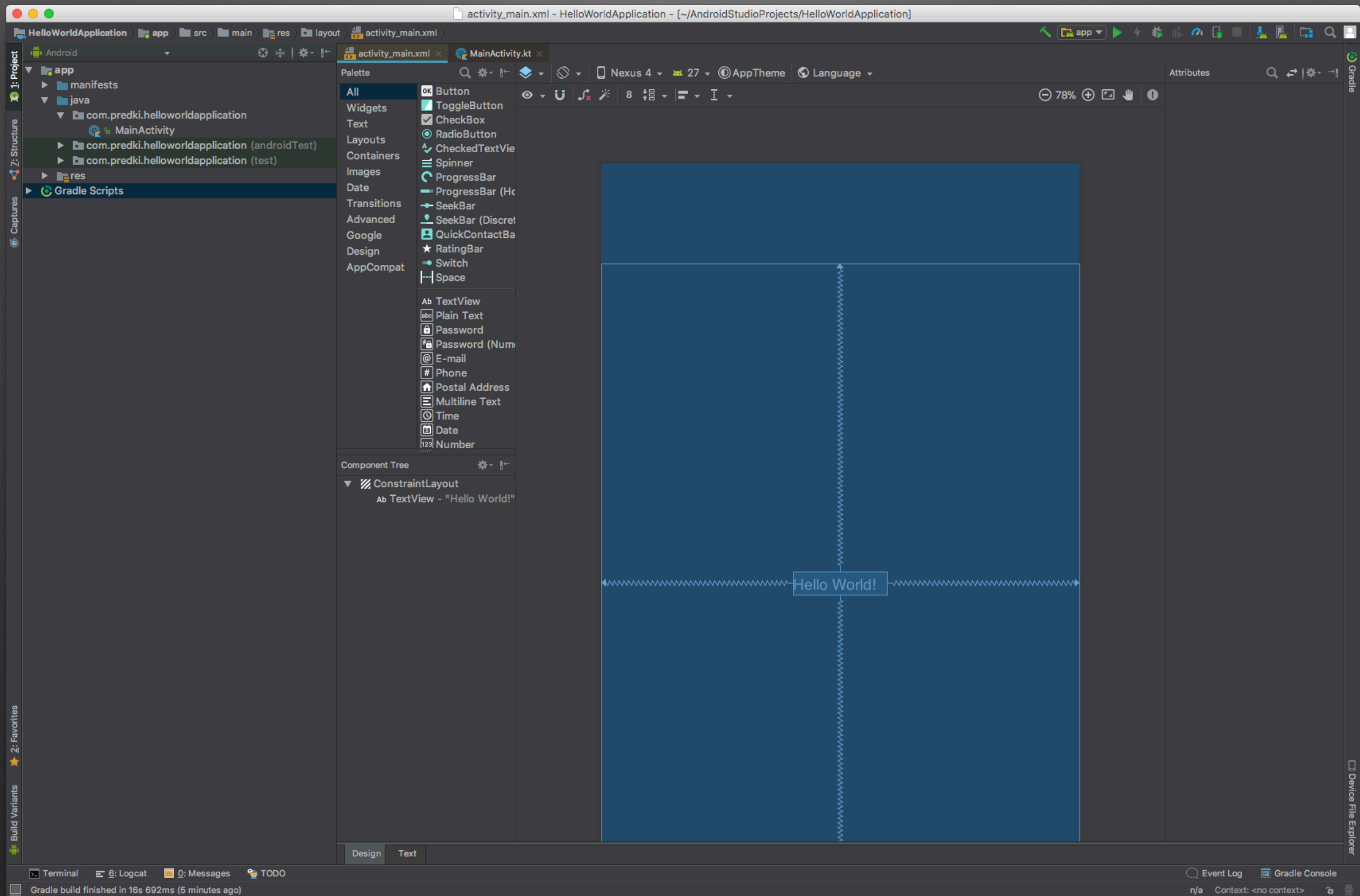
Activity Name

☒ Generate Layout File

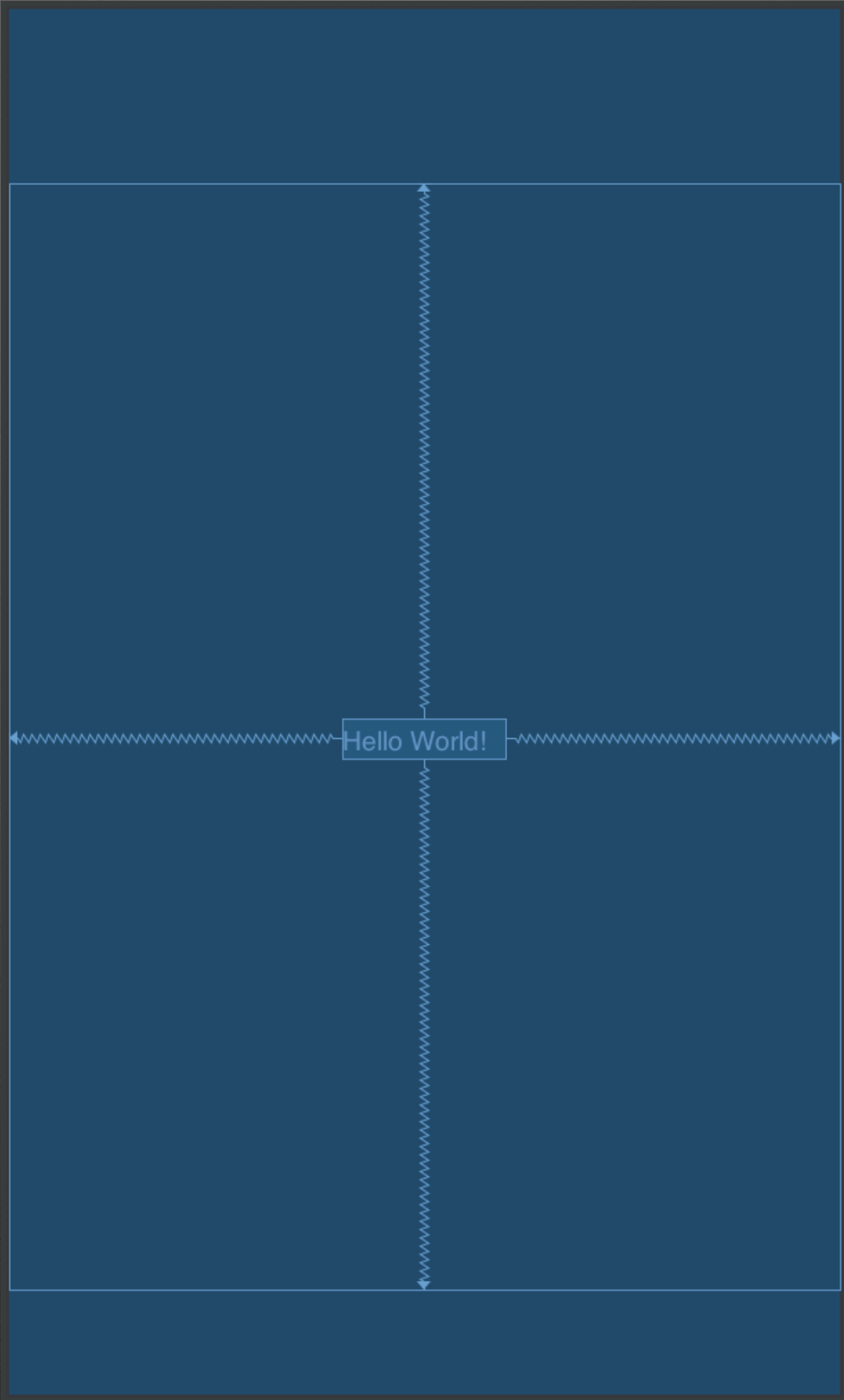
Layout Name

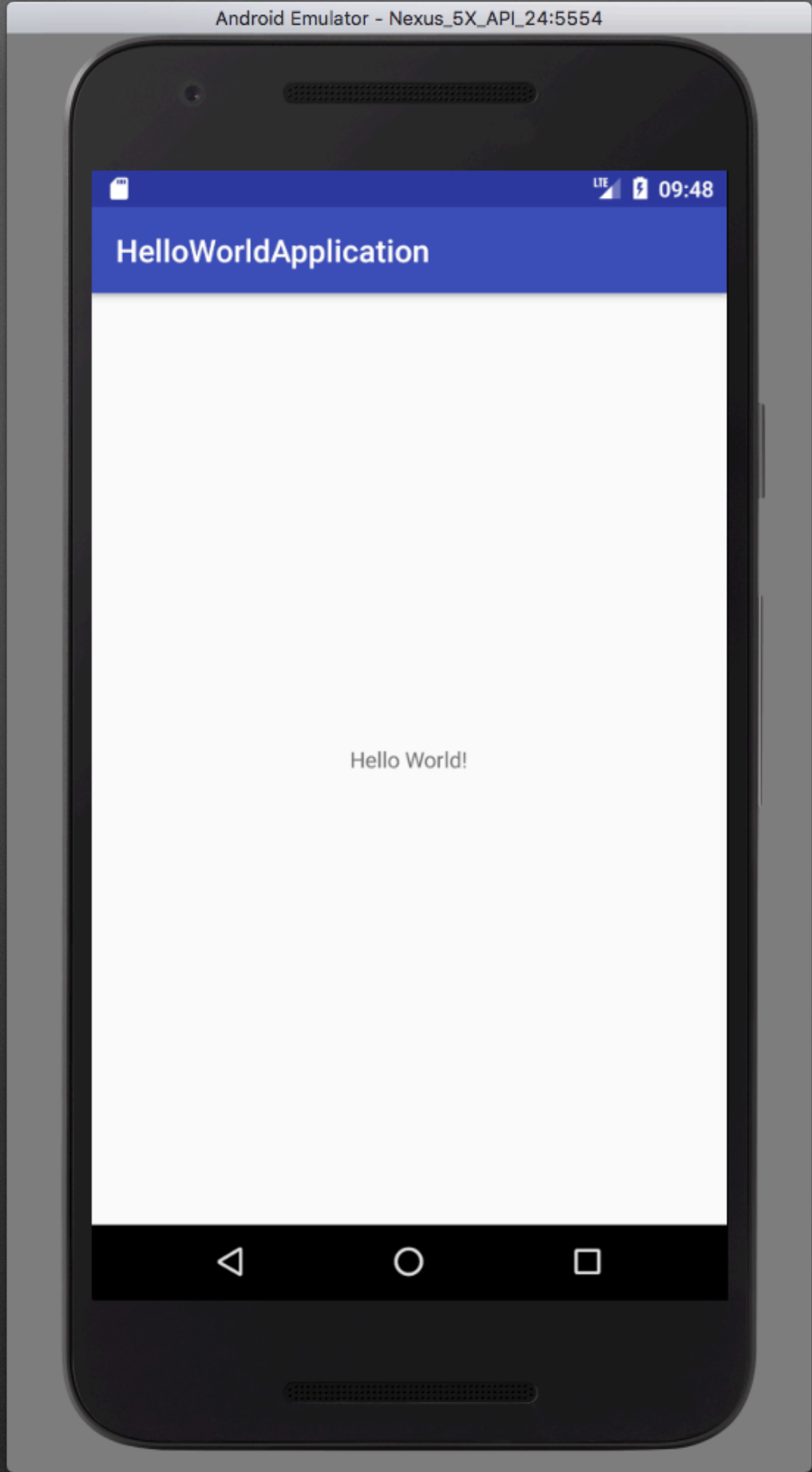
If false, this activity base class will be Activity instead of AppCompatActivity

☒ Backwards Compatibility (AppCompat)CancelPreviousNextFinish




```
1      package com.predki.helloworldapplication
2
3      import android.support.v7.app.AppCompatActivity
4      import android.os.Bundle
5
6      class MainActivity : AppCompatActivity() {
7
8          override fun onCreate(savedInstanceState: Bundle?) {
9              super.onCreate(savedInstanceState)
10             setContentView(R.layout.activity_main)
11         }
12     }
```



Czas życia aktywności

- Obsługiwany przez metody:
 - onCreate - wywoływana w momencie tworzenia aktywności - ma argument, w którym może otrzymać informacje o poprzednim stanie aktywności
 - onRestart - gdy aktywność ma być wznowiona po wcześniejszym zatrzymaniu przez system
 - onStart - uruchamiana zaraz po onCreate lub onRestart, gdy aktywność ma stać się widoczna

Czas życia aktywności

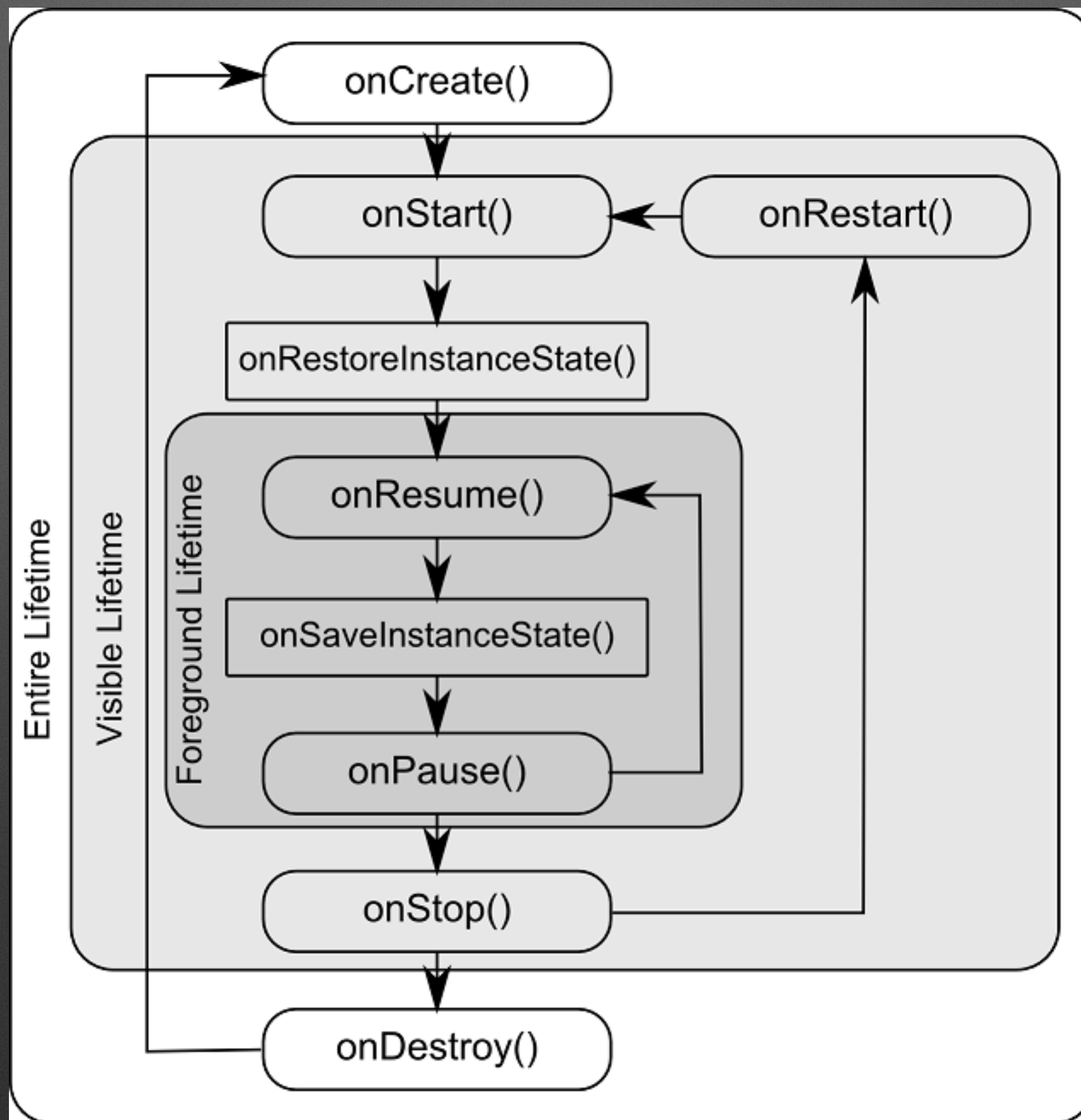
- onResume - aktywność jest aktualnie widoczna, a użytkownik może wchodzić z nią w interakcję.
- onPause - gdy aktywność ma być wstrzymana.
- onStop - aktywność nie jest już widoczna dla użytkownika.
- onDestroy - aktywność zostanie zabita, z własnej inicjatywy lub przez system

Czas życia aktywności

- `onConfigurationChanged` - gdy zajdzie zmiana konfiguracji, która nie spowoduje restartu aktywności.
- `onRestoreInstanceState` - wywoływana po metodzie `onStart` gdy aktywność ma odtworzyć swój zachowany stan
- `onSaveInstanceState` - wywoływana przed zabiciem aktywności, gdy aktywność ma zachować swój stan dynamiczny

Czas życia aktywności

- Przeciążając te metody należy pamiętać, że oprócz dwóch ostatnich należy zawsze wywołać metodę odziedziczoną `super.on....`
- Standardowe kontrolki są w stanie same zachować swój stan pomiędzy zmianami stanu aktywności (można to wyłączyć - `android:saveEnabled=„false”`)
- Do przechowania stanu innych elementów można wykorzystać klasę `Bundle`



Projektowanie UI

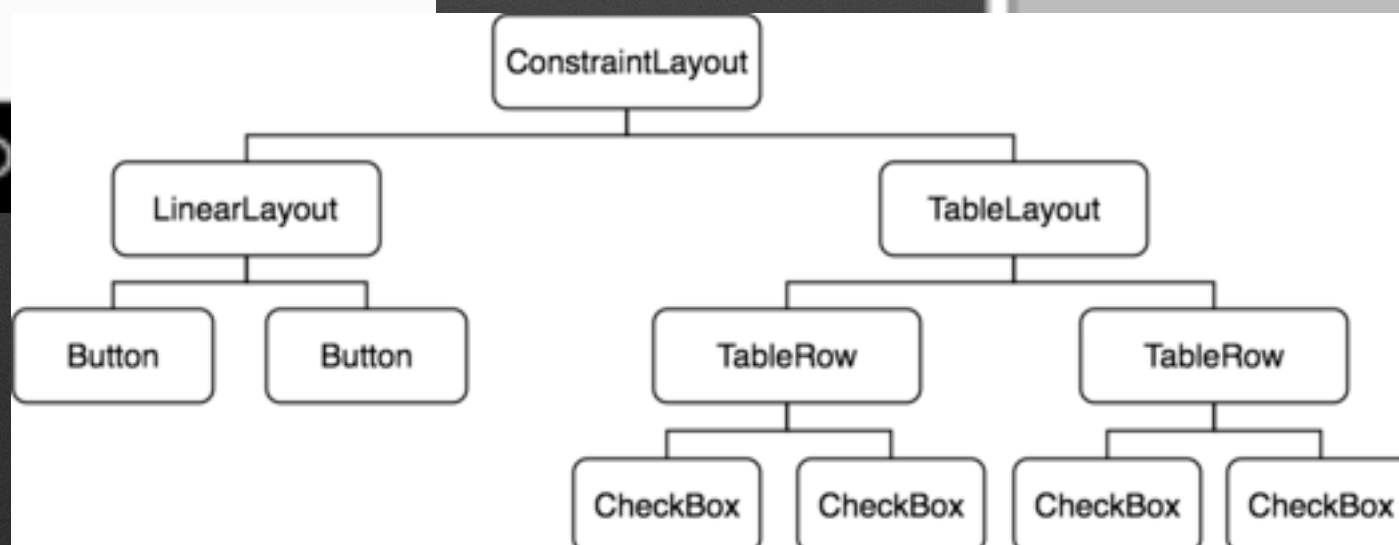
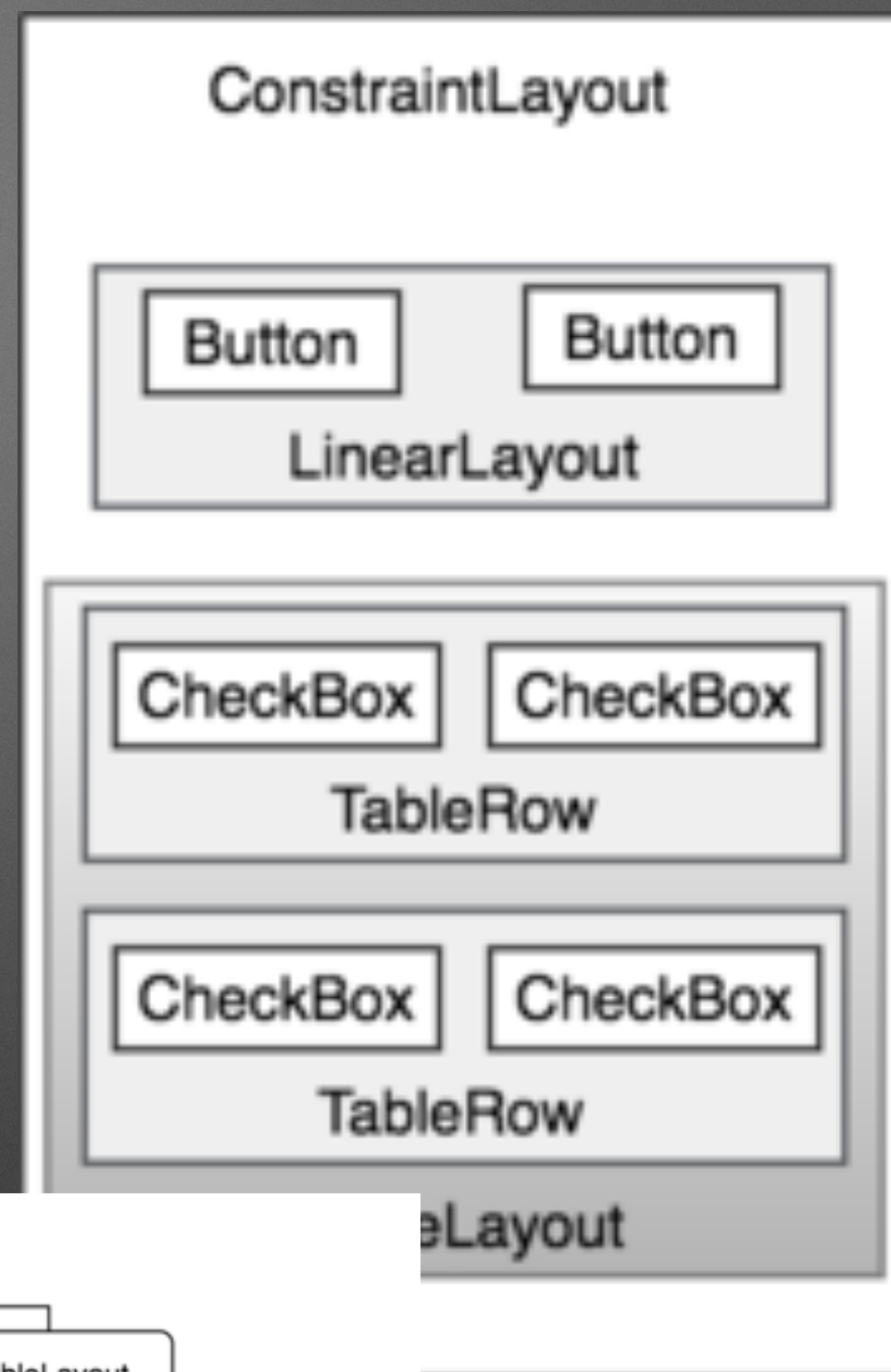
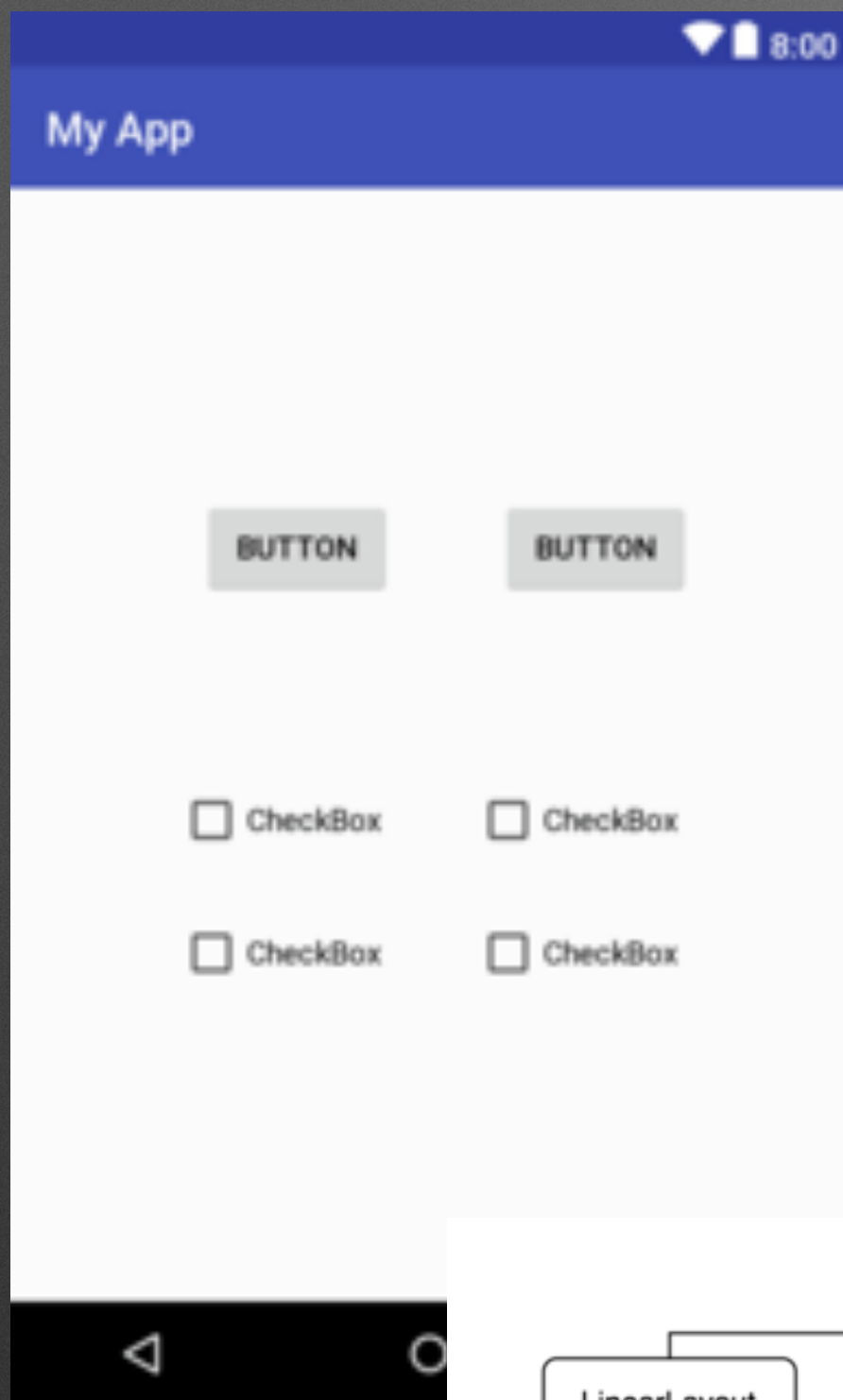
- Urządzenia z Androidem mogą pracować w wielu różnych rozdzielczościach.
- Urządzenia mogą zmieniać orientację ekranu (pion/poziom).
- Wszystkie widoczne elementy dziedziczą po klasie View.
- Czasami elementy są grupowane w obiekty dziedziczące po klasie ViewGroup.

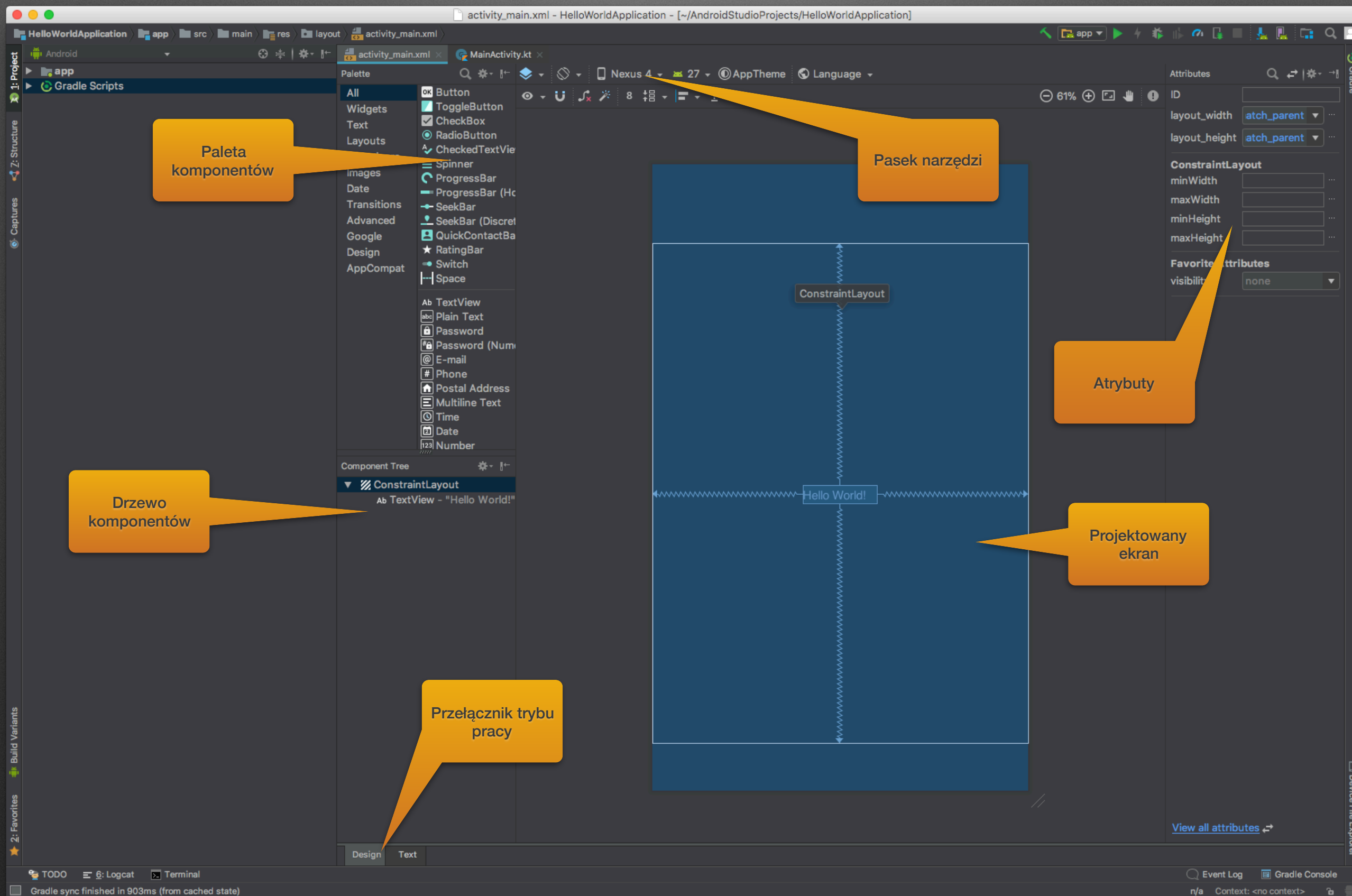
Projektowanie UI

- W rozmieszczaniu widoków na ekranie pomagają obiekty typu LayoutManagers, będące specyficznym rodzajem ViewGroup:
- ConstraintLayout - od Androida 7 podstawowy menedżer układu oparty na ograniczeniach
- LinearLayout - umieszcza elementy w wierszu lub kolumnie
- TableLayout - umieszcza elementy w macierzy wierszy/kolumn

- **FrameLayout** - blokuje fragment ekranu na najczęściej jeden widok
- **RelativeLayout** - elementy rozmieszczane są względem siebie
- **AbsoluteLayout** - elementy mają podane bezwzględne współrzędne i rozmiary
- **GridLayout** - rozmieszcza elementy na kanwie o cechach macierzy.
- **CoordinatorLayout** - od Androida 5 do zarządzania paską na górze aplikacji

- Domyślnie źródłowym menedżerem jest `ConstraintLayout`.
- Menedżery układu mogą się dowolnie zagnieżdżać, np. `GridLayout` wewnątrz `TableLayout`, wewnątrz `ConstraintLayout`.
- Wszystkie elementy tworzą strukturę drzewa widoków.





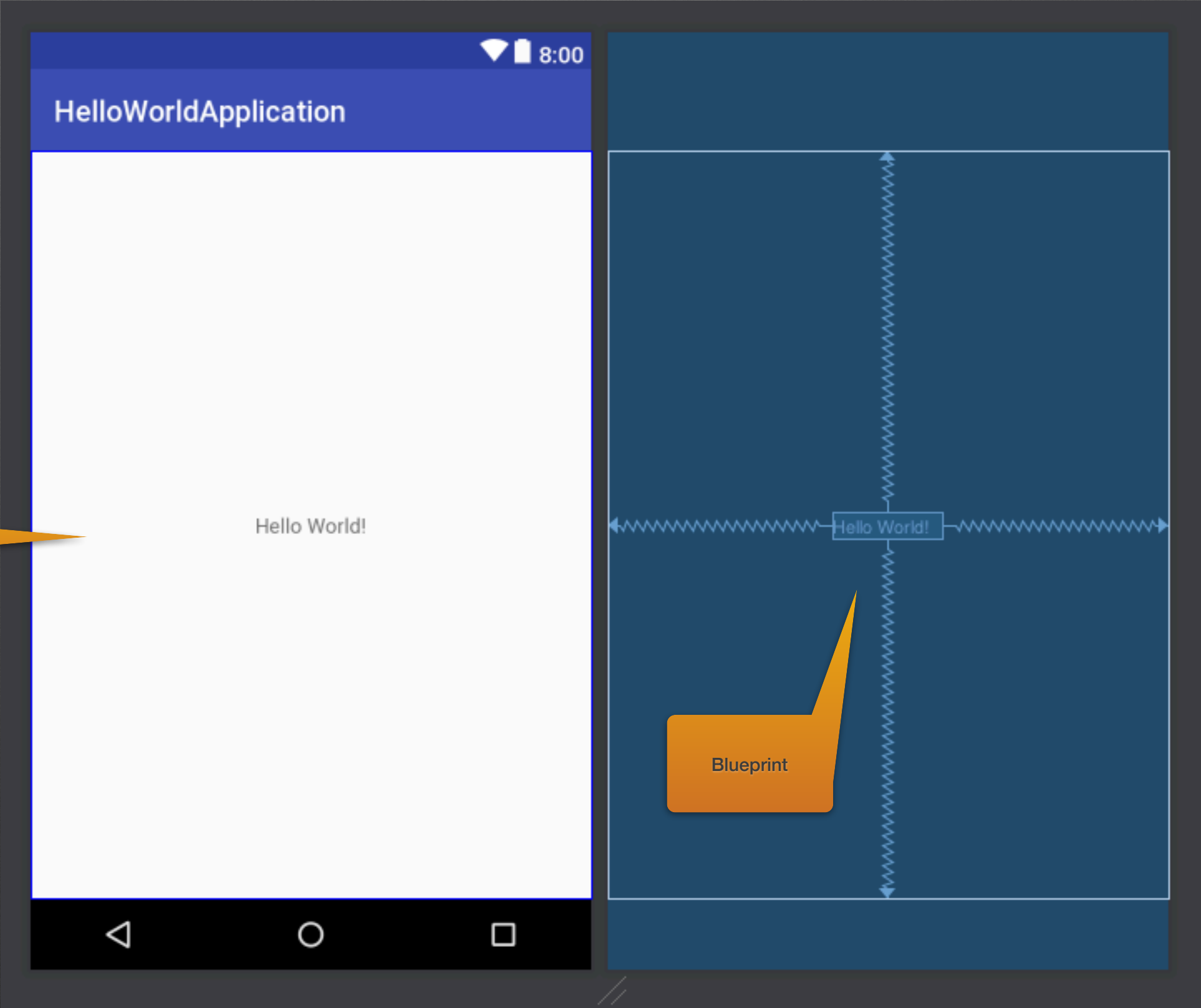
Nexus 4

Design

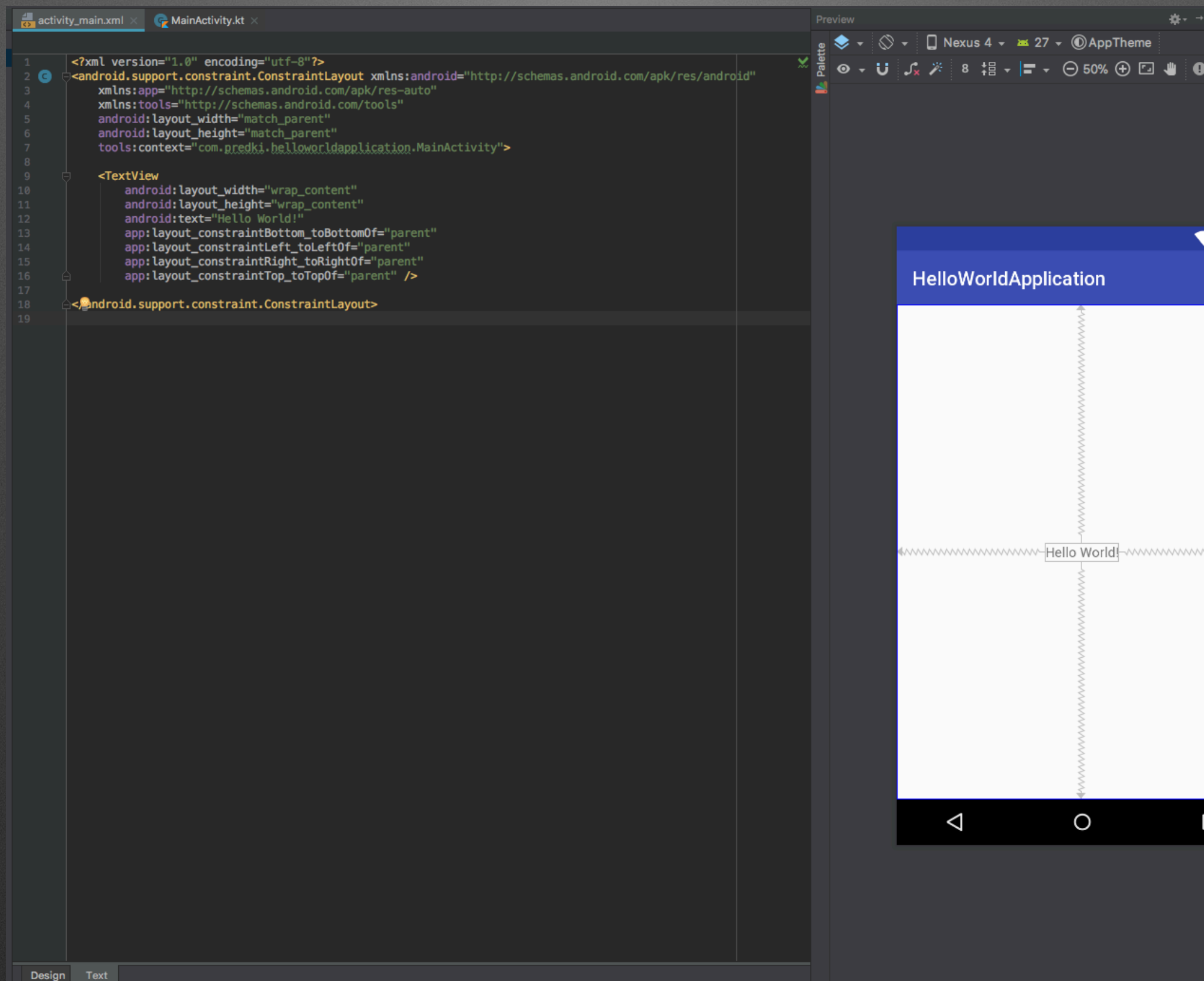
✓ Blueprint

Design + Blueprint

Force Refresh Layout



Widok tekstowy



Nexus 4

27

AppTheme

Language

3.7, 480 × 800, hdpi (Nexus One)

4.0, 480 × 800, hdpi (Nexus S)

4.7, 720 × 1280, xhdpi (Galaxy Nexus)

✓

4.7, 768 × 1280, xhdpi (Nexus 4)

5.0, 1080 × 1920, xxhdpi (Nexus 5)

5.0, 1080 × 1920, 420dpi (Pixel 2)

5.0, 1080 × 1920, xxhdpi (Pixel)

5.2, 1080 × 1920, 420dpi (Nexus 5X)

5.5, 1440 × 2560, 560dpi (Pixel XL)

5.7, 1440 × 2560, 560dpi (Nexus 6P)

6.0, 1440 × 2560, 560dpi (Nexus 6)

6.0, 1440 × 2880, 560dpi (Pixel 2 XL)

7.0, 800 × 1280, tvdpi (Nexus 7 2012)

7.0, 1200 × 1920, xhdpi (Nexus 7)

8.9, 2048 × 1536, xhdpi (Nexus 9)

9.9, 2560 × 1800, xhdpi (Pixel C)

10.1, 2560 × 1600, xhdpi (Nexus 10)

320 × 290, tvdpi (Round Chin)

280 × 280, hdpi (Square)

320 × 320, hdpi (Round)

1080p, 1920 × 1080, xhdpi (TV)

720p, 1280 × 720, tvdpi (TV)

AVD: Nexus5X_API_24

Generic Phones and Tablets

Add Device Definition...

Zdarzenia w Androidzie

- Użytkownik oraz system generują zdarzenia.
- Zdarzenia są kolejgowane na zasadzie FIFO.
- Ze zdarzeniem związane są informacje, które je opisują.
- Aby obsłużyć zdarzenie, obiekt (najczęściej widok) musi posiadać aktywne nasłuchiwanie (Listener) i odpowiednią do tego metodę.

Zdarzenia

LISTENER	METODA
onClickListener	onClick()
onLongClickListener	onLongClick()
onTouchListener	onTouch()
onCreateContextMenuListener	onCreateContextMenu()
onFocusChangeListener	onFocusChange()
onKeyListener	onKey()

Wielodotyky

- Android potrafi obsłużyć na raz wiele punktów dotyku
- Dotyk może składać się w gest
- Aby przechwycić informacje o dotyku obsługujemy zdarzenie onTouch()
- Z dotykiem związany jest obiekt MotionEvent, który zawiera informacje o punktach dotyku aktywnych w danym momencie