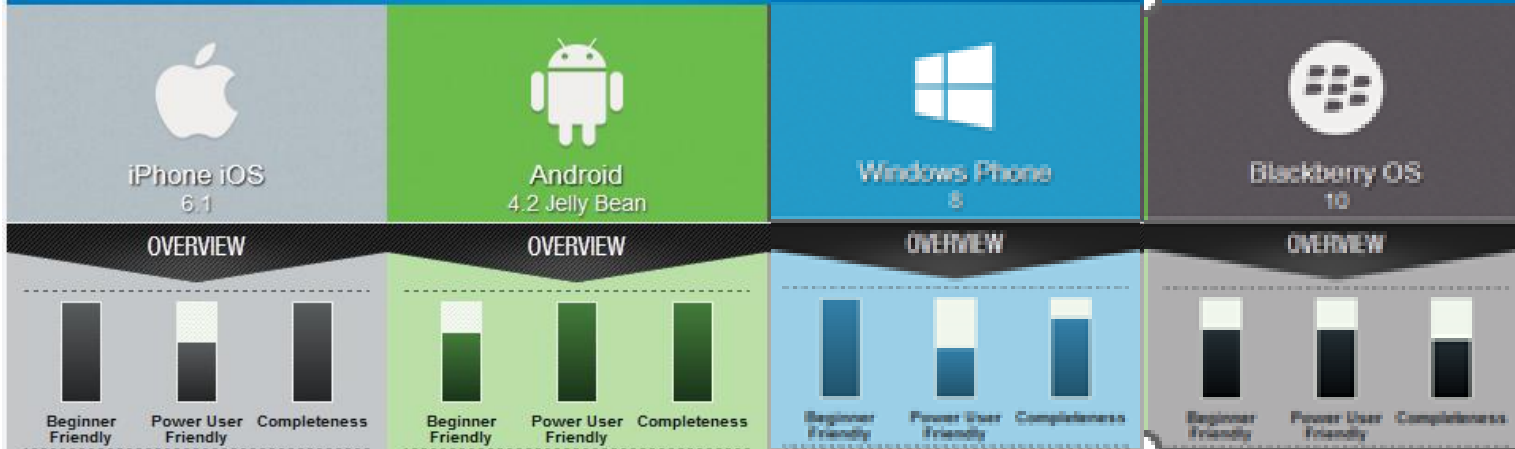


Programowanie urządzeń mobilnych

Wprowadzenie do systemu Android



<http://myphonedeads.co.uk/blog/33-the-smartphone-os-complete-comparison-chart>



Wojna platform

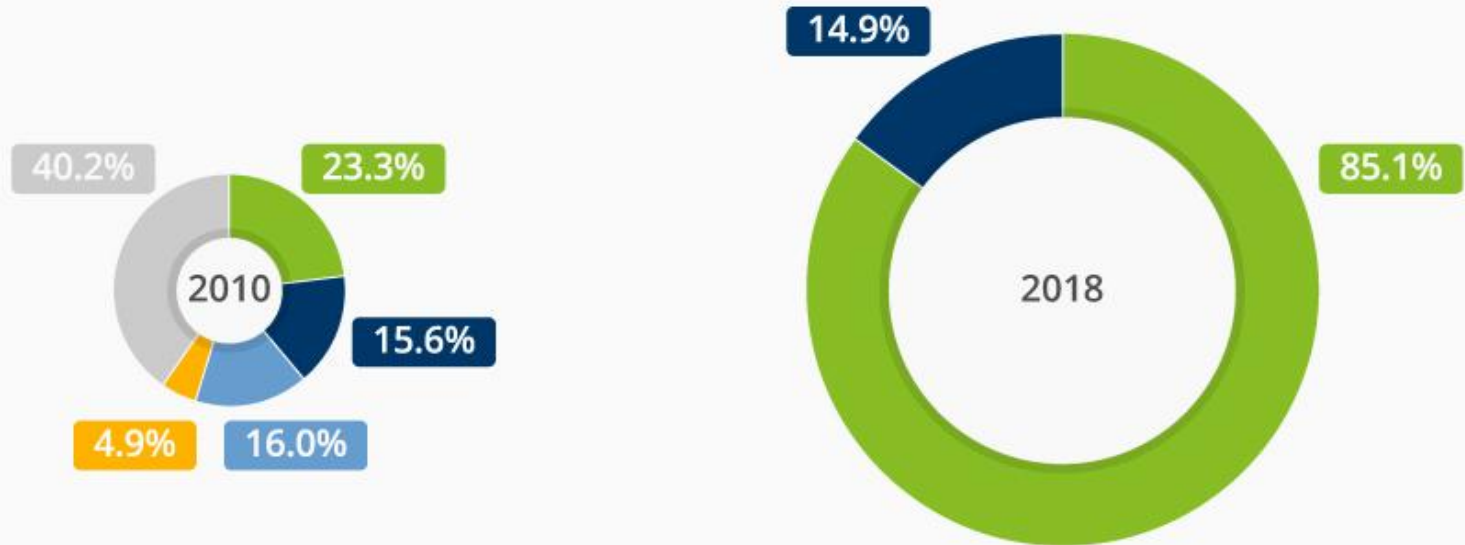
www.techradar.com/news/phone-and-communications/mobile-phones/ios7-vs-android-jelly-bean-vs-windows-phone-8-vs-bb10-1159893

Segmentacja rynku

The Smartphone Duopoly

Worldwide smartphone market share by operating system (based on unit shipments)

● Android ● iOS ● BlackBerry ● Windows Phone ● Others



Total sales

305m

1,405m



@StatistaCharts

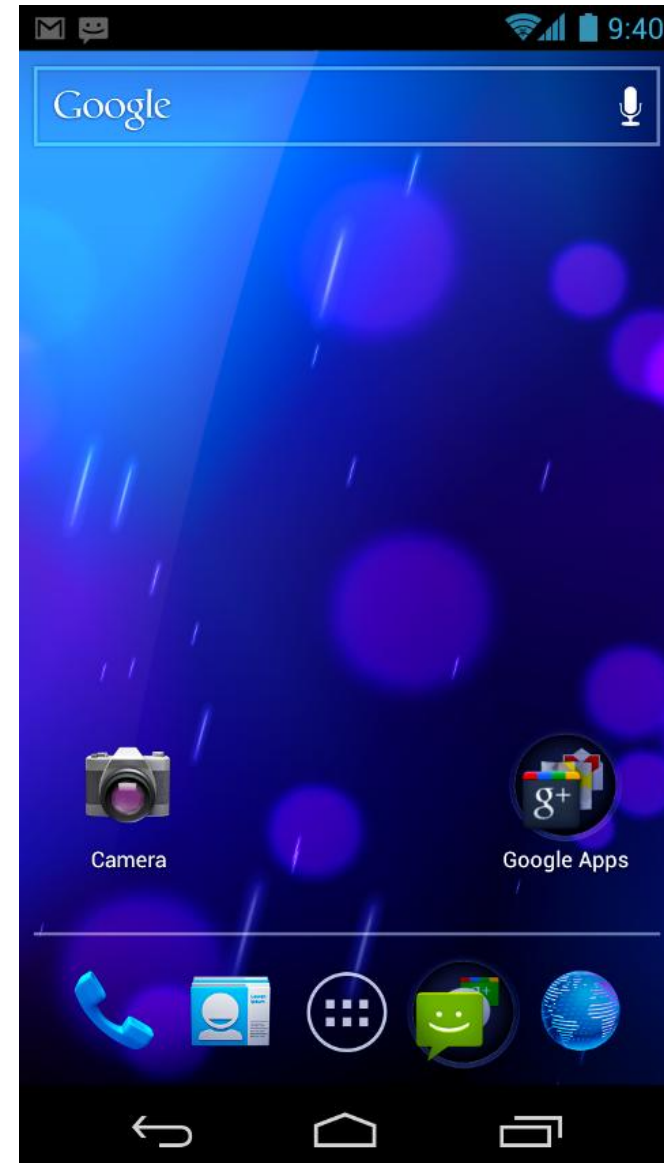
Source: IDC

statista

<https://www.statista.com/chart/3268/smartphone-os-market-share/>

Android – co to takiego?

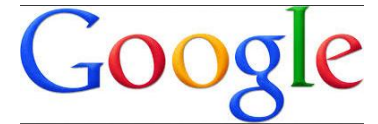
- ▶ System operacyjny bazujący na jądrze i filozofii Linux'a, przystosowany do urządzeń mobilnych.
- ▶ Tworzony przez Open Handset Alliance (w tym Google Inc.)
- ▶ Projekt typu Open Source.



Krótką historia Androida



ANDROID



2003

- Powstaje **Android Inc.**
- Andy Rubin, Rich Miner, Nick Sears, Chris White

2005

- Android Inc. przejęty przez Google
- Google chciało stworzyć „*a flexible and upgradable system*”

2007

- Świat ujrzał iPhone'a
- Powstaje Open Handset Alliance
- Pierwsza prezentacja Androida

2008

- Pierwsze urządzenie z Androidem - HTC Dream

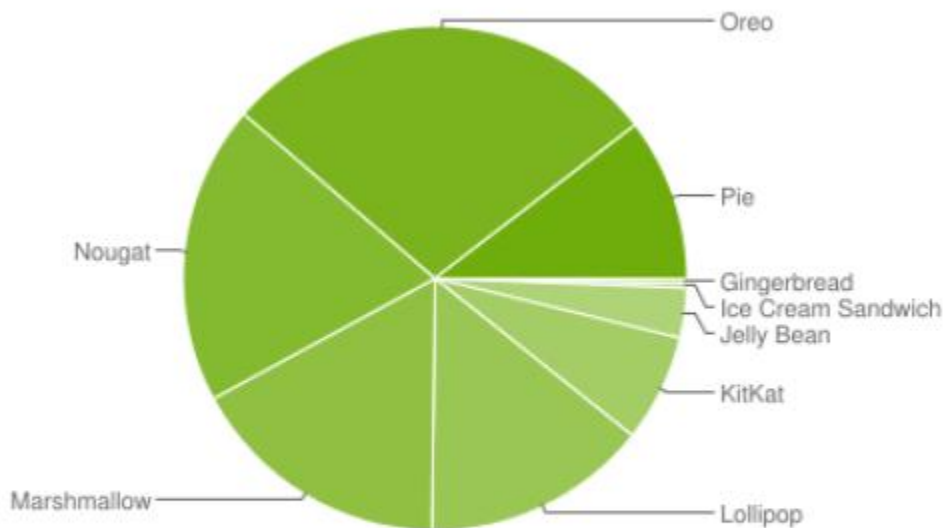


Po co był Android?

Wersje Androida

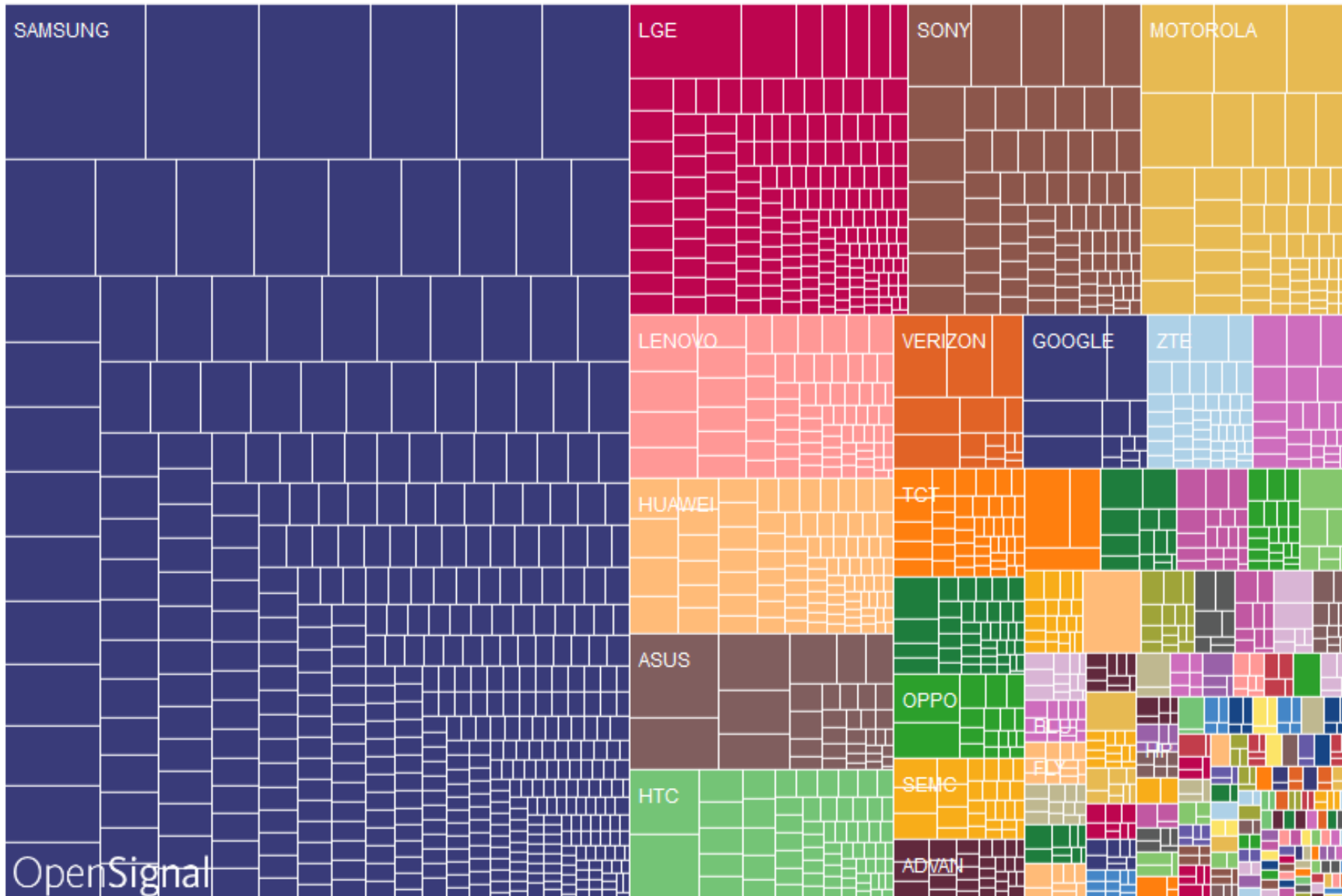
Na którą wersję systemu pisać aplikacje?

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.2%
4.2.x		17	1.5%
4.3		18	0.5%
4.4	KitKat	19	6.9%
5.0	Lollipop	21	3.0%
5.1		22	11.5%
6.0	Marshmallow	23	16.9%
7.0	Nougat	24	11.4%
7.1		25	7.8%
8.0	Oreo	26	12.9%
8.1		27	15.4%
9	Pie	28	10.4%



*Data collected during a 7-day period ending on May 7, 2019.
Any versions with less than 0.1% distribution are not shown.*

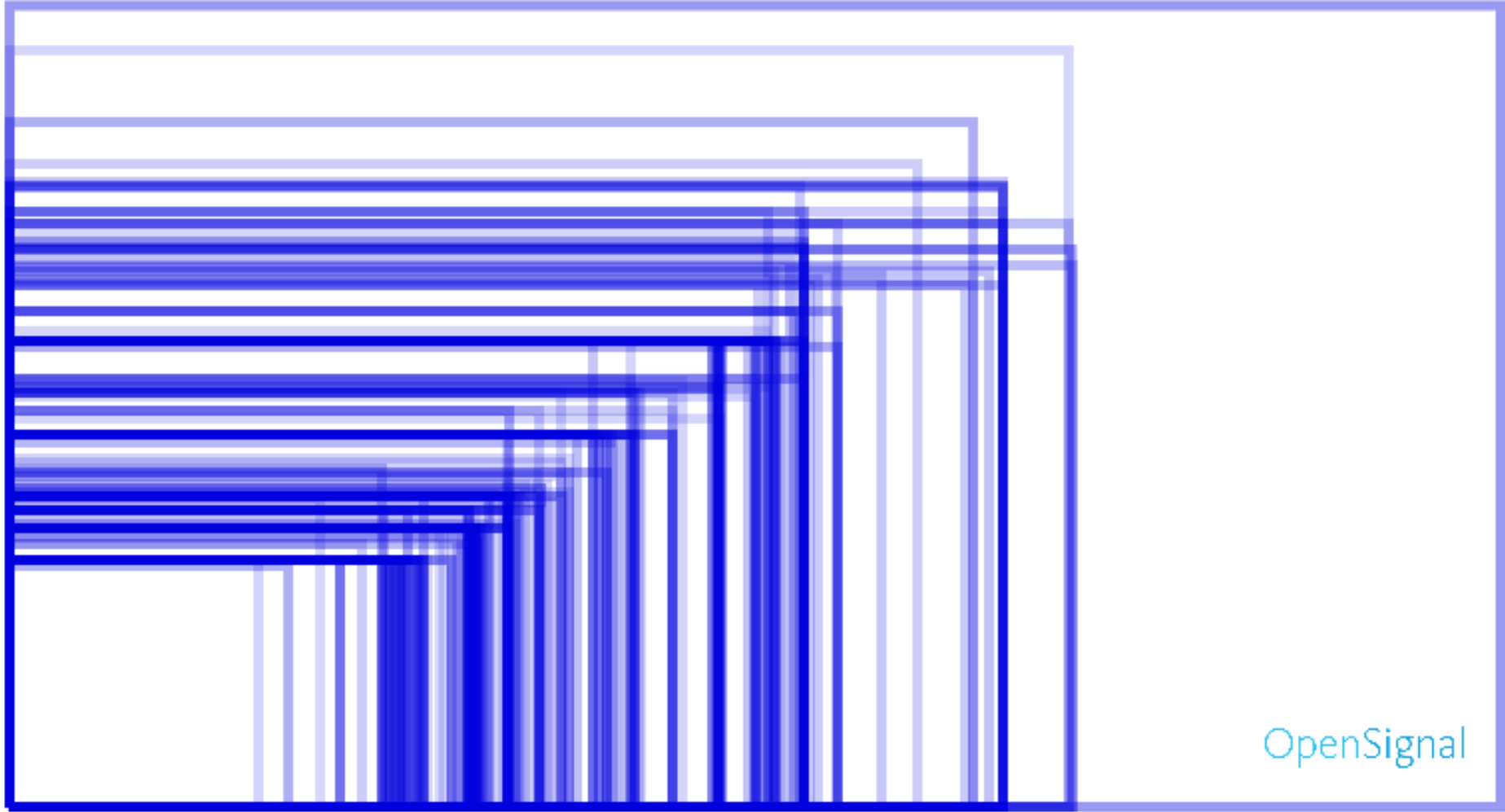
Fragmentacja – producenci



Fragmentacja – urządzenia

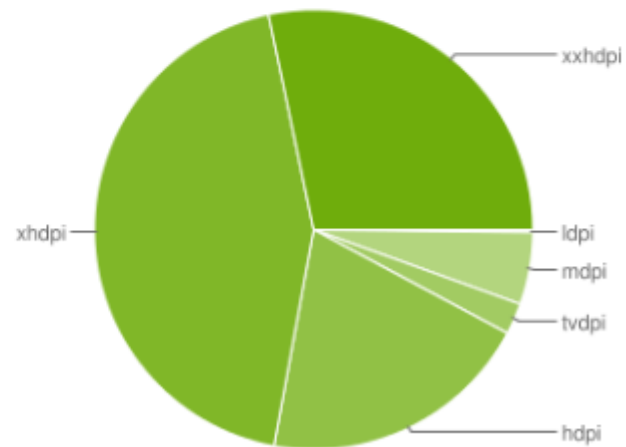
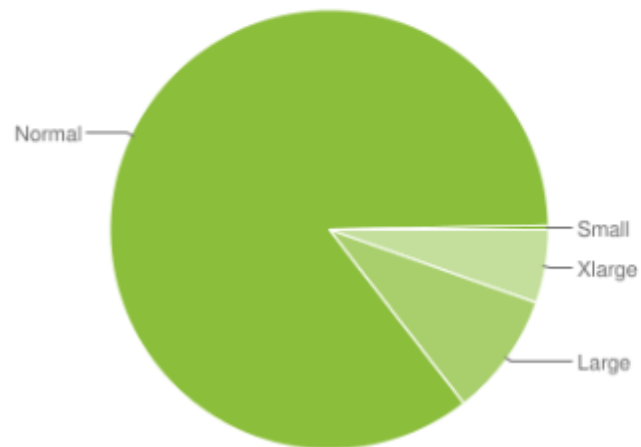


Fragmentacja – rozmiar ekranu



Fragmentacja – rozmiar ekranu c.d.

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	0.1%				0.1%		0.2%
Normal		0.4%	0.3%	17.0%	41.1%	25.9%	84.7%
Large		1.8%	2.0%	0.7%	2.6%	2.1%	9.2%
Xlarge		3.5%		1.9%	0.5%		5.9%
Total	0.1%	5.7%	2.3%	19.6%	44.3%	28.0%	



Data collected during a 7-day period ending on August 30, 2020.

Any screen configurations with less than 0.1% distribution are not shown.

Niezależność od gęstości

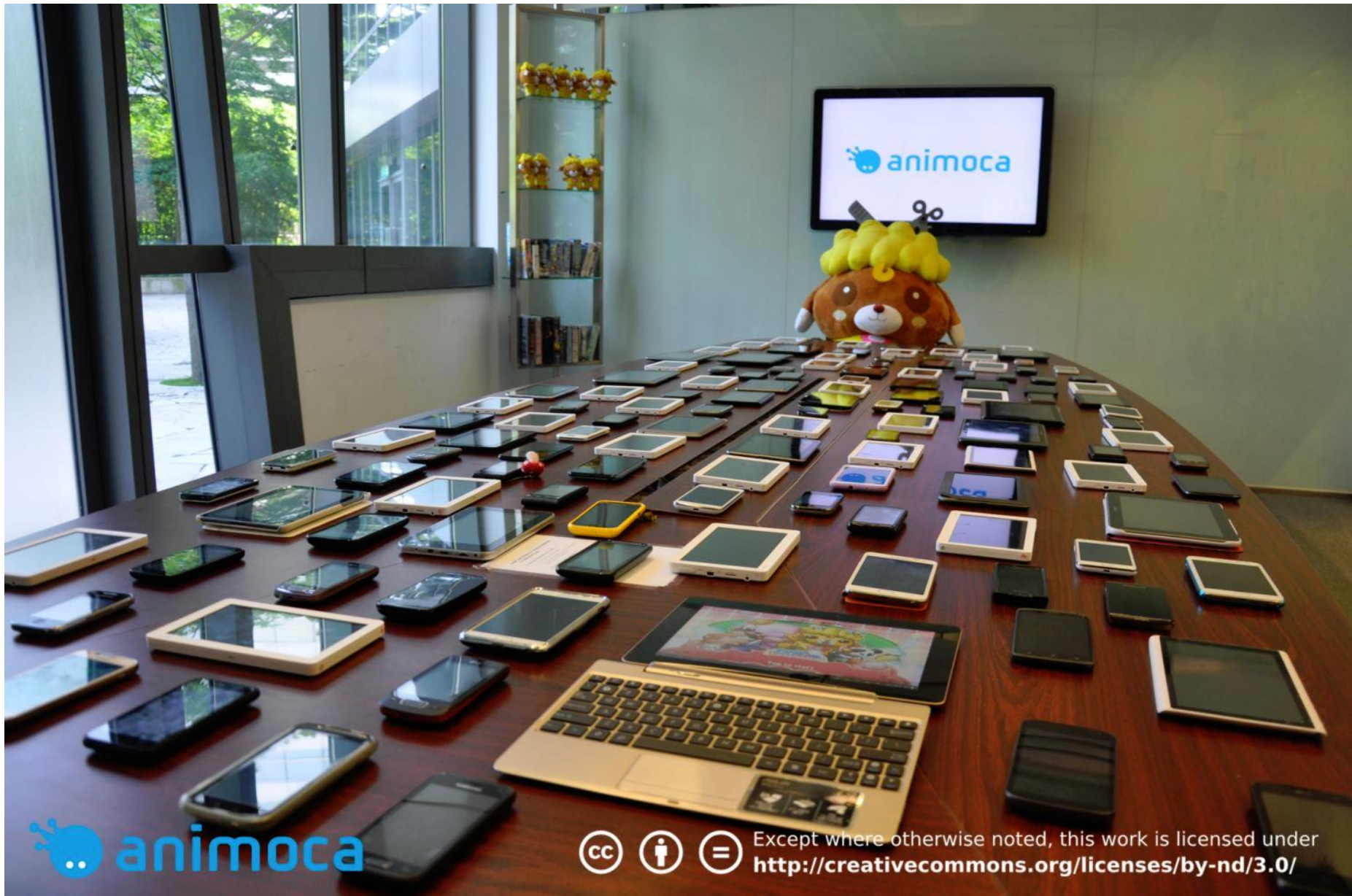


Figure 2. Example application without support for different densities, as shown on low, medium, and high density screens.



Figure 3. Example application with good support for different densities (it's density independent), as shown on low, medium, and high density screens.

Zbiór urządzeń testowych



Anatomia Androidida

APPLICATIONS

Home

Contacts

Phone

Browser

...

APPLICATION FRAMEWORK

Activity Manager

Window Manager

Content Providers

View System

Notification Manager

Package Manager

Telephony Manager

Resource Manager

Location Manager

XMPP Service

LIBRARIES

Surface Manager

Media Framework

SQLite

OpenGL|ES

FreeType

WebKit

SGL

SSL

libc

ANDROID RUNTIME

Core Libraries

Dalvik Virtual Machine

LINUX KERNEL

Display Driver

Camera Driver

Bluetooth Driver

Flash Memory Driver

Binder (IPC) Driver

USB Driver

Keypad Driver

WiFi Driver

Audio Drivers

Power Management

Najniższa warstwa - kernel

- ▶ Android kernel = Linux kernel po drobnych modyfikacjach
- ▶ Kernel zapewnia m.in.:
 - Warstwę abstrakcji sprzętowej (HAL)
 - Zarządzanie pamięcią
 - Zarządzanie procesami
 - Zarządzanie siecią
 - Zarządzaniem zasilaniem
- ▶ Linux shell komendą: `adb shell`
- ▶ Czemu właśnie jądro Linux'a?



Biblioteki natywne

- ▶ Kod napisany w C lub C++

Przykładowe elementy:

- ▶ **Surfaces Manager** – zarządzanie wyświetlaniem ekranów
- ▶ **OpenGL|ES** – zarządzanie grafiką 2D i 3D
- ▶ **Media Framework** – kodeki (MPEG 4, H.264, MP3, AAC)
- ▶ **FreeType** – rendering czcionek
- ▶ **SQLite** – przechowywanie danych
- ▶ **Webkit** – silnik przeglądarki internetowej



Dalvik

- ▶ Android Runtime – dostosowano do warunków urządzeń wbudowanych (ograniczone zasoby procesora, pamięci, baterii)
- ▶ Dalvik – własna implementacja Wirtualnej Maszyny (Javy)

Różnice względem JVM:

- ▶ Dalvik operuje na plikach .dex
- ▶ Inny zestaw bibliotek niż w JDK
- ▶ Bardziej zwarta i efektywna implementacja
- ▶ Dalvik jest „register-based VM”

Po co tworzyć własną WM?



Art (od 4.4 i L)

Włączenie:

- ▶ W Androidzie 4.4 ART (Android Runtime) należy uaktywnić w menu Ustawienia -> Opcje programistyczne.

Zalety:

- ▶ (AOT) Ahead-of-time compilation
- ▶ Ulepszony mechanizm garbage collector
- ▶ Ulepszone mechanizmy odplukiwania (debuger)

```
java.lang.NullPointerException: Attempt to write to field 'int android.accessibilityservice.AccessibilityServiceInfo.flags' on a null object reference
```

```
java.lang.NullPointerException: Attempt to invoke virtual method 'java.lang.String java.lang.Object.toString()' on a null object reference
```

Wady:

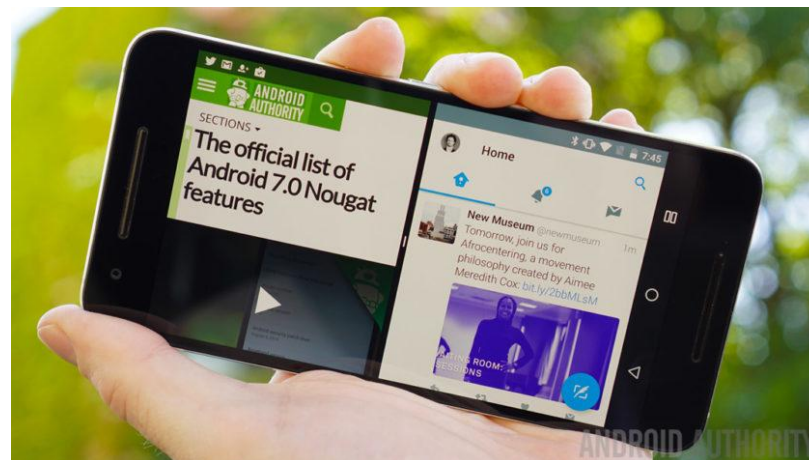
- ▶ Nie wszystkie aplikacje będą w 100% funkcjonować poprawnie

Marshmallow czyli Android 6.0

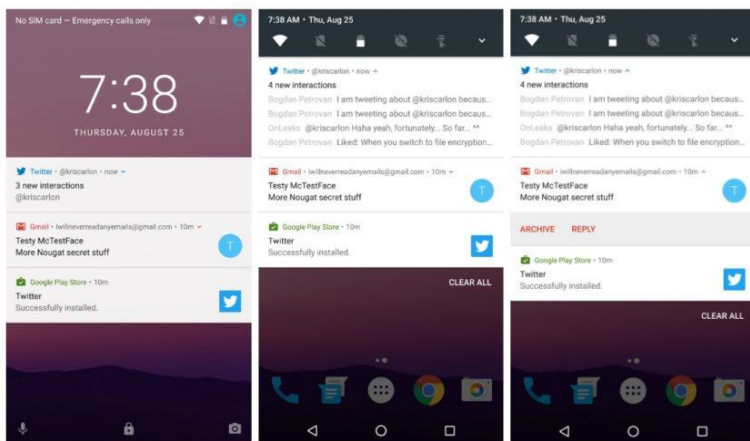
- ▶ Zmiany w uprawnieniach aplikacji – system pyta o dane uprawnienie (za pierwszym razem) dokładnie w momencie kiedy jest potrzebne.
- ▶ Wsparcie dla czytników linii papilarnych
- ▶ Wsparcie dla USB typu C
- ▶ Usprawnienia asystenta głosowego
- ▶ Wprowadzenie obsługi płatności Android Pay
- ▶ Ulepszenia w systemie oszczędzania energii.

Nugat – co nowego

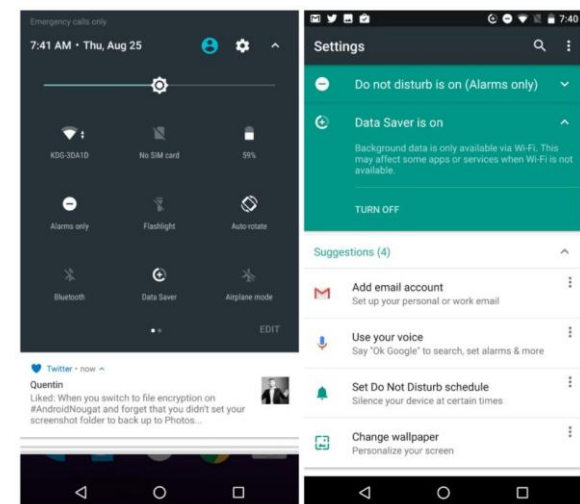
- ▶ Tryb split screen
- ▶ Priorytety w Powiadomieniach



Ulepszony tryb drzemki Zmiana DPI

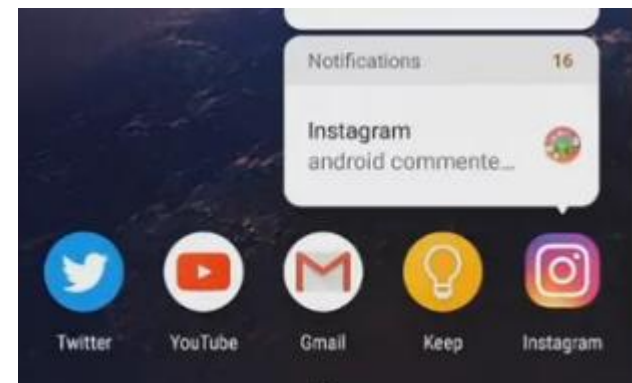


- ▶ Tryb nie przeszkadzać
- ▶ Wsparcie dla Vulkan, Java 8



Oreo – co nowego

- ▶ Większe restrykcje dla procesów w tle
- ▶ Notification Dots
- ▶ Tryb Obraz-w-obrazie (PiP)
- ▶ Ambient display



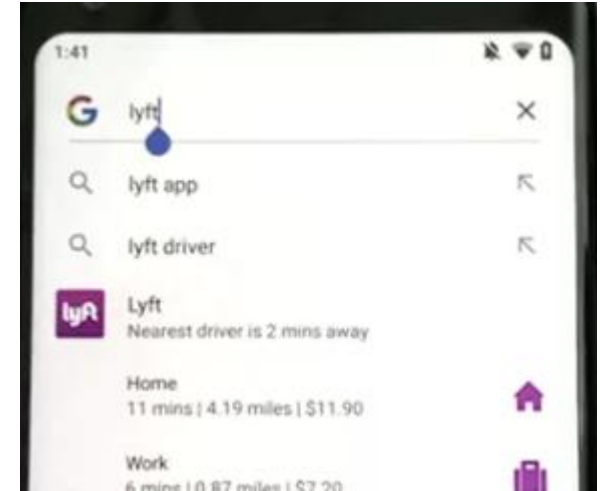
Usprawnienia w wypełnianiu formularzy



- ▶ Wsparcie dla kodeków typu LDAC
- ▶ Asystent do połączeń WIFI

Pie – co nowego

- ▶ Adaptacyjna jasność i bateria
- ▶ Obsługa za pomocą gestów
- ▶ Tryb Slices



Ulepszone zaznaczanie tekstu



- ▶ Wsparcie dla HDR i ekranów z wcięciem
- ▶ Ograniczenia w dostępie do mikrofonu, kamery i sensorów

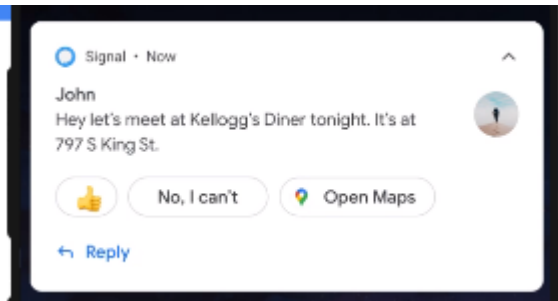


Android 10 – co nowego

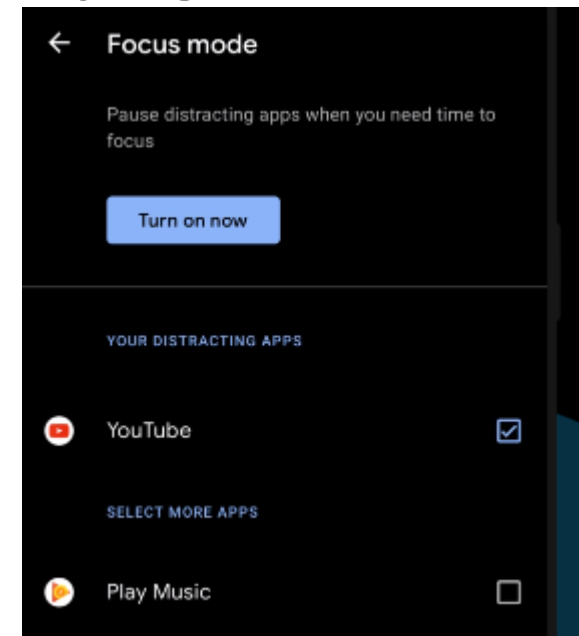
- ▶ Live caption
- ▶ Wzmacniacz dźwięku
- ▶ Smart replay



Ulepszona nawigacja gestami

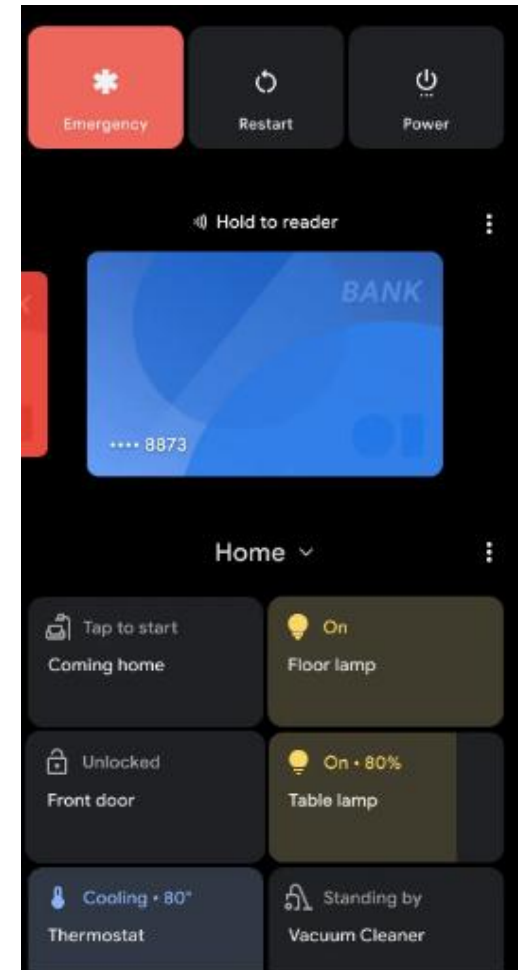
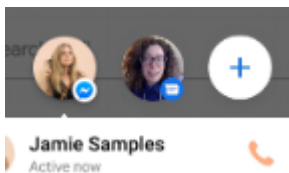
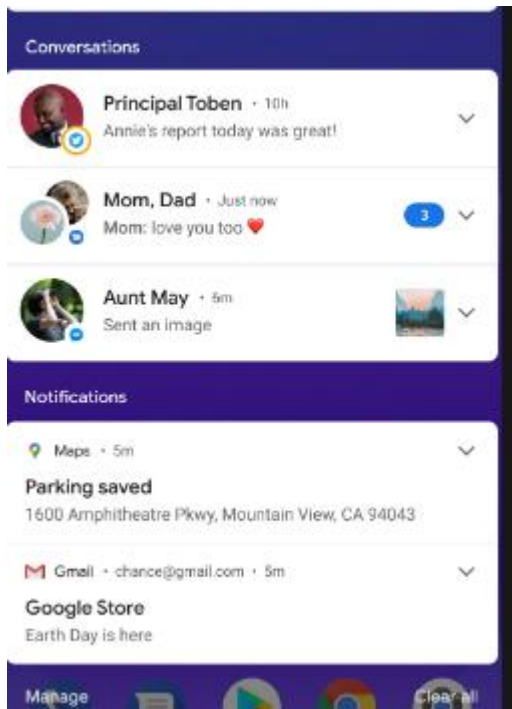


- ▶ Ciemny motyw
- ▶ Ulepszenia w zarządzaniu prywatnością
- ▶ Tryb focus



Android 11 – co nowego

- ▶ Konwersacje oddzielone od innych powiadomień.
- ▶ Chmurki na wzór Messenger'a.
- ▶ Nowe menu zasilania (SmartHome)
- ▶ Łatwy wybór urządzenia audio.
- ▶ One-time permissions.
- ▶ Wbudowane nagrywanie ekranu.



Android 12 – co nowego

- ▶ Material You.
- ▶ Widżet konwersacji.
- ▶ Lepsze wsparcie dla osób niepełnosprawnych
- ▶ Powiadomienia o użyciu kamery i mikrofonu.
- ▶ Lokalizacja dokładna lub przybliżona.

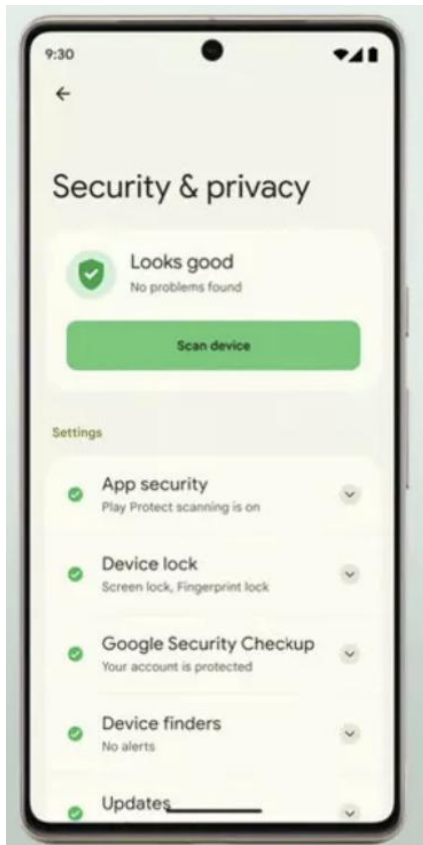


Area magnification

A new window magnifier lets you zoom in on a part of your screen without having to lose context on the rest of the screen content.



Android 13 – co nowego



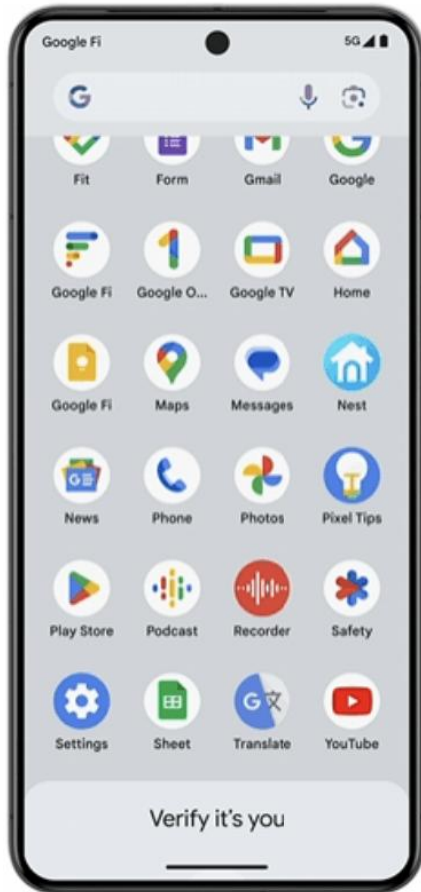
- ▶ Nowy wygląd ikon (dostosują się do barw tapety ekranu głównego).
- ▶ Lepsza kontrola nad multimediami.
- ▶ Ustawianie języka dla danej aplikacji.
- ▶ Nowe ustawienia prywatności użytkowników i powiadomień.
- ▶ Lepsza współpraca pomiędzy urządzeniami.
- ▶ Pasek zadań na tabletach.
- ▶ Poprawiony tryb wielu okien i pełnoekranowy.
- ▶ Tablety rozpoznają dłoń i stylus jako osobne dotknięcia.

Android 14 – co nowego



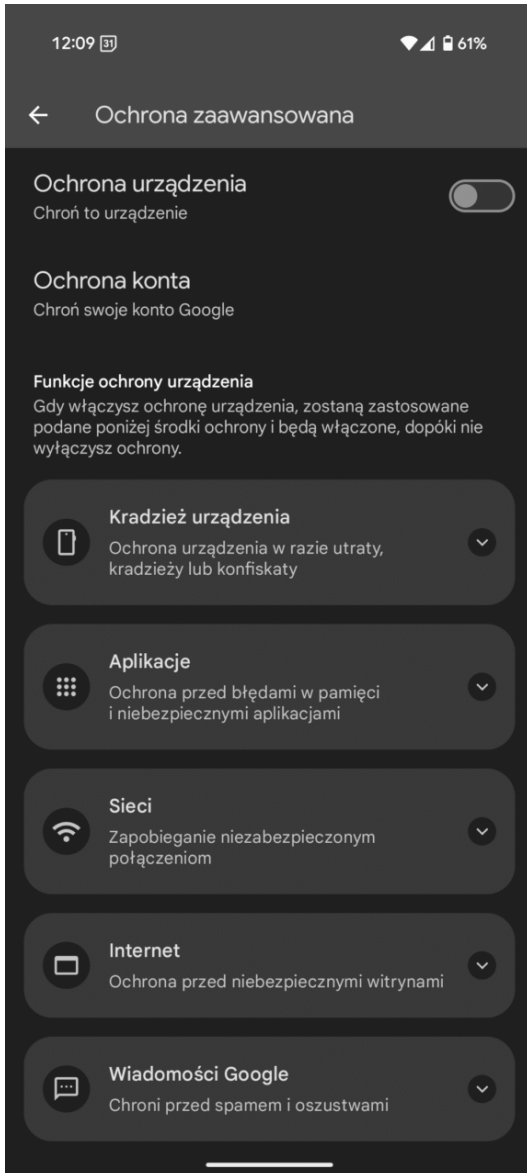
- ▶ Natywna obsługa formatu Ultra HDR.
- ▶ Zmiany w konfiguracji ekranu blokady.
- ▶ Generator tapet bazujący na AI.
- ▶ Telefon może służyć jako kamera internetowa.
- ▶ Usługa Health Connect agregująca dane z różnych źródeł.
- ▶ Lepsza obsługa dla osób z dostosowaniami: aparatów słuchowych, kontrola rozmiaru czcionki, powiększanie treści lupą itp.
- ▶ Monochromatyczny motyw graficzny.

Android 15 – co nowego



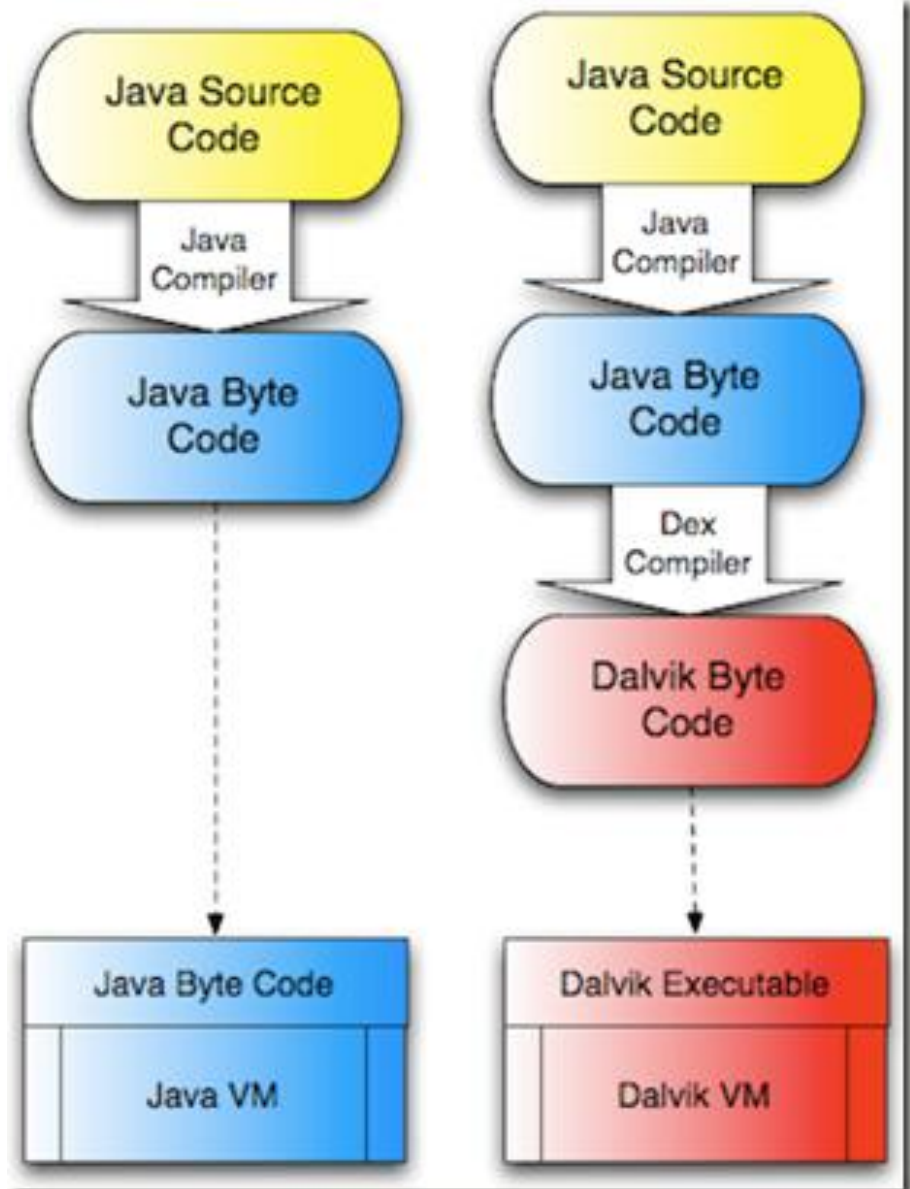
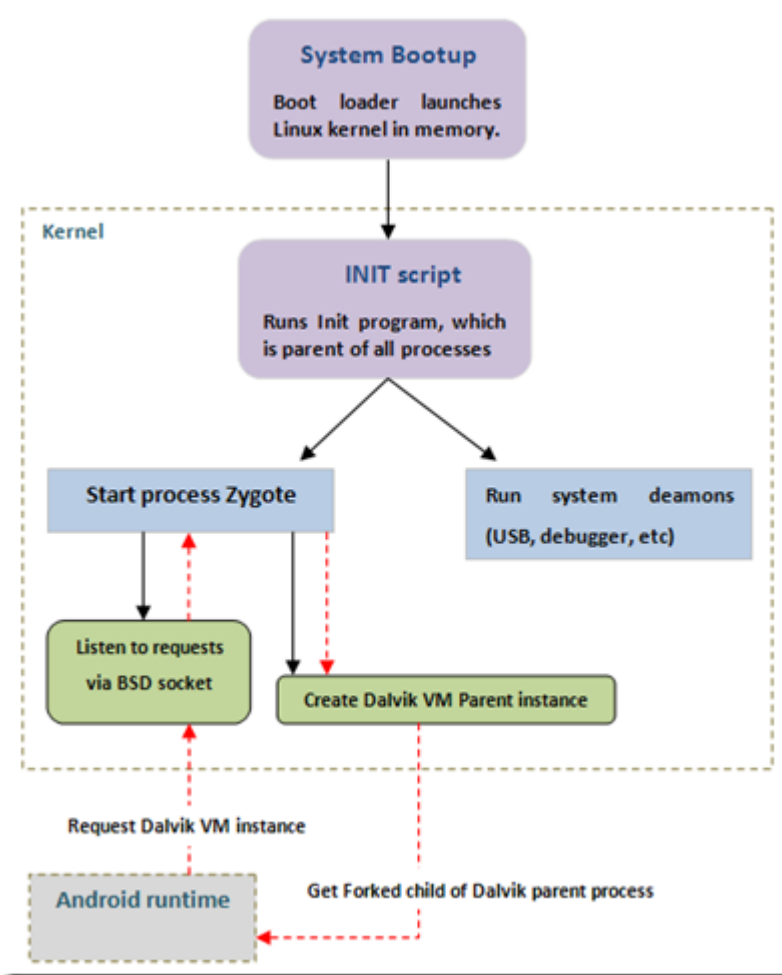
- ▶ Ukryta przestrzeń (sejf) na aplikacje.
- ▶ Wsparcie dla łączności satelitarnej.
- ▶ Dzielenie się muzyką (Bluetooth Audio Sharing).
- ▶ Ulepszenia w zarządzaniu pamięcią i akumulatorem.
- ▶ Rozszerzone zarządzanie lokalizacją i prywatnością.
- ▶ Widżety na ekranie blokady.
- ▶ Rozszerzenie funkcji portfela Google.

Android 16 – co nowego



- ▶ Mini odtwarzacz muzyki na pasku powiadomień z ciemniejszą okładką albumu i bardziej zaawansowanym sterowaniem.
- ▶ Ekran ostatnich aplikacji z nowymi przyciskami: „zrzut ekranu” i „wybierz”.
- ▶ Dodatkowe skróty dostępne po przytrzymaniu ikony aplikacji.
- ▶ W ustawieniach baterii można teraz sprawdzić kondycję ogniwa – wyrażoną w procentach pojemności (funkcja znana z iOS).
- ▶ Nowa zakładka „Ochrona zaawansowana” umożliwia zabezpieczenie danych w przypadku kradzieży, ochronę przed niezabezpieczonymi połączeniami Wi-Fi itp.

Działanie Dalvik VM



Czemu kod Javy nie jest bezpośrednio kompilowany do Dalvik Byte Code?

Application Framework

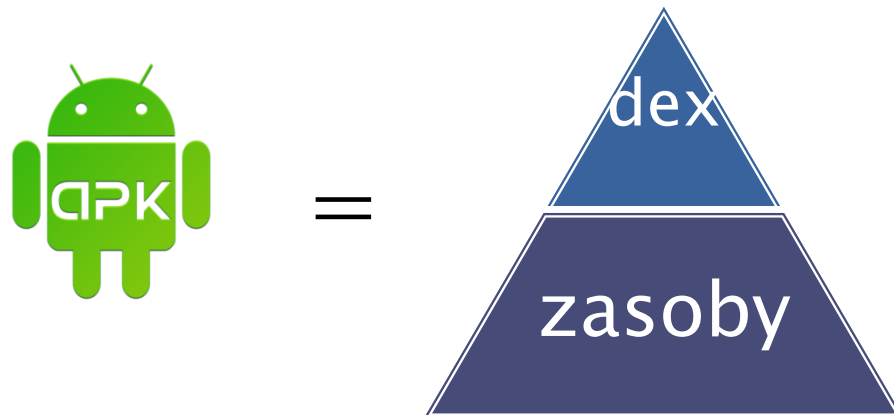
Główne komponenty

- ▶ **Activity Manager** – zarządza cyklem życia (i nawigacją) aplikacji
- ▶ **Package Manager** – zarządzanie zainstalowanymi aplikacjami
- ▶ **Content Providers** – zarządzanie udostępnianiem danych między aplikacjami
- ▶ **View System** – zarządzanie warstwą GUI oraz generowanymi zdarzeniami



Aplikacje

- ▶ Plik wykonywalny Dalvik + Zasoby = APK
- ▶ Aplikacje muszą być podpisane
 - Istnieje też debug key
- ▶ Dostępne są różne sklepy: Google Play, Amazon AppStore, GetJar, AppBrain, F-Droid itp.



APPLICATIONS

Home

Contacts

Phone

Browser

...

Nowy projekt

New Android Application

New Android Application

Creates a new Android Application



Application Name: Hello World

Project Name: HelloWorld

Package Name: pl.tomaszx.helloworld

Minimum Required SDK: API 8: Android 2.2 (Froyo)

Target SDK: API 19: Android 4.4 (KitKat)

Compile With: API 19: Android 4.4 (KitKat)

Theme: Holo Light with Dark Action Bar

- HelloWorld
 - src
 - pl.tomaszx.helloworld
 - MainActivity.java
 - gen [Generated Java Files]
 - pl.tomaszx.helloworld
 - BuildConfig.java
 - R.java
 - Android 4.4
 - Android Private Libraries
 - assets
 - bin
 - libs
 - res
 - drawable-hdpi
 - drawable-ldpi
 - drawable-mdpi
 - drawable-xhdpi
 - drawable-xxhdpi
 - layout
 - activity_main.xml
 - menu
 - main.xml
 - values
 - values-sw600dp
 - values-sw720dp-land
 - values-v11
 - values-v14
 - AndroidManifest.xml
 - ic_launcher-web.png
 - proguard-project.txt
 - project.properties

```
package pl.tomaszx.helloworld;
```

```
import android.os.Bundle;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
```

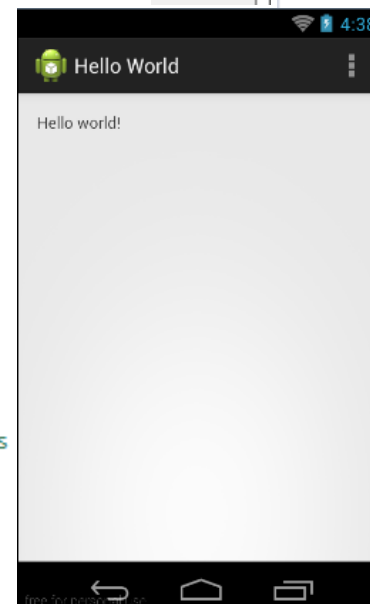
```
        // Inflate the menu; this adds items to the action bar if it is
```

```
        getMenuInflater().inflate(R.menu.main, menu);
```

```
        return true;
```

```
    }
```

```
}
```



Emulated Device Key	Keyboard Key
Home	HOME
Menu (left softkey)	F2 or Page-up button
Star (right softkey)	Shift-F2 or Page Down
Back	ESC
Call/dial button	F3
Hangup/end call button	F4
Search	F5
Power button	F7
Audio volume up button	KEYPAD_PLUS, Ctrl-F5
Audio volume down button	KEYPAD_MINUS, Ctrl-F6
Camera button	Ctrl-KEYPAD_5, Ctrl-F3
Switch to previous layout orientation (for example, portrait, landscape)	KEYPAD_7, Ctrl-F11
Switch to next layout orientation (for example, portrait, landscape)	KEYPAD_9, Ctrl-F12
Toggle cell networking on/off	F8
Toggle code profiling	F9 (only with <code>-trace</code> startup option)
Toggle fullscreen mode	Alt-Enter
Toggle trackball mode	F6
Enter trackball mode temporarily (while key is pressed)	Delete
DPad left/up/right/down	KEYPAD_4/8/6/2
DPad center click	KEYPAD_5
Onion alpha increase/decrease	KEYPAD_MULTIPLY(*) / KEYPAD_DIVIDE(/)

Transfer plików i debug

Device Explorer

Pixel 2 API 28 Android 9.0 ("Pie")

Files Processes


Name	Permissions	Date	Size
> acct	dr-xr-xr-x	2023-10-07 08:38	0 B
> bin	lrw-r--r--	2009-01-01 03:00	11 B
> cache	drwxrwx---	2009-01-01 03:00	4 KB
> config	drwxr-xr-x	2023-10-07 08:38	0 B
> d	lrw-r--r--	2009-01-01 03:00	17 B
> data	drwxrwx--x	2023-06-26 03:50	4 KB
> dev	drwxr-xr-x	2023-10-07 08:38	2,6 KB
> etc	lrw-r--r--	2009-01-01 03:00	11 B
> lost+found	drwx-----	2009-01-01 03:00	16 KB
> mnt	drwxr-xr-x	2023-10-07 08:38	240 B
> odm	drwxr-xr-x	2009-01-01 03:00	4 KB
> oem	drwxr-xr-x	2009-01-01 03:00	4 KB
> proc	dr-xr-xr-x	2023-10-07 08:38	0 B
> product	lrw-r--r--	2009-01-01 03:00	15 B
> sbin	drwxr-x---	2009-01-01 03:00	4 KB
> sdcard	lrw-r--r--	2009-01-01 03:00	21 B
> storage	drwxr-xr-x	2023-10-07 08:38	100 B
> sys	dr-xr-xr-x	2023-10-07 08:38	0 B
> system	drwxr-xr-x	2009-01-01 03:00	4 KB
> vendor	drwxr-xr-x	2009-01-01 03:00	4 KB
bugreports	lrw-r--r--	2009-01-01 03:00	50 B
charger	lrw-r--r--	2009-01-01 03:00	13 B

Device Manager

Notifications

Gradle

Device Explorer



motion for personal use - Galaxy S2 - 4.1.1 - API 16 - 480x800 (480x8...

or personal use

ivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER

Dostęp do powłoki systemowej

adb shell

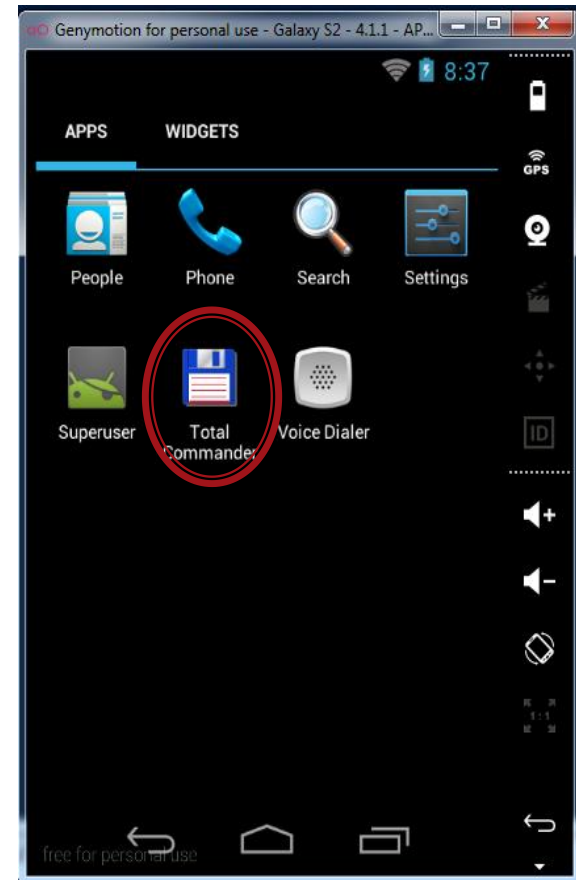
```
PS D:\adt-bundle-windows-x86_64-20131030\sdk\platform-tools> .\adb.exe shell
root@android:/ # ls -al
ls -al
drwxr-xr-x root root 2014-02-25 20:19 acct
drwxrwx--- system cache 2014-02-23 17:47 cache
dr-x----- root root 2014-02-25 20:19 config
lrwxrwxrwx root root 2014-02-25 20:19 d -> /sys/kernel/debug
drwxrwx--x system system 2014-02-23 17:53 data
-rw-r--r-- root root 116 1970-01-01 00:00 default.prop
drwxr-xr-x root root 2014-02-25 20:19 dev
lrwxrwxrwx root root 2014-02-25 20:19 etc -> /system/etc
-rw-r----- root root 625 1970-01-01 00:00 fstab.vbox86
-rwxr-x--- root root 206451 1970-01-01 00:00 init
-rwxr-x--- root root 2344 1970-01-01 00:00 init.goldfish.rc
-rwxr-x--- root root 15537 1970-01-01 00:00 init.rc
-rwxr-x--- root root 1637 1970-01-01 00:00 init.trace.rc
-rwxr-x--- root root 3915 1970-01-01 00:00 init.usb.rc
-rwxr-x--- root root 872 1970-01-01 00:00 init.vbox86.rc
drwxrwxr-x root system 2014-02-25 20:19 mnt
dr-xr-xr-x root root 2014-02-25 20:19 proc
drwx----- root root 2012-10-01 14:14 root
drwxr-x--- root root 1970-01-01 00:00 sbin
lrwxrwxrwx root root 2014-02-25 20:19 sdcard -> /mnt/sdcard
dr-xr-xr-x root root 2014-02-25 20:19 sys
drwxr-xr-x root root 1970-01-01 00:00 system
-rw-r--r-- root root 272 1970-01-01 00:00 ueventd.goldfish.rc
-rw-r--r-- root root 3879 1970-01-01 00:00 ueventd.rc
-rw-r--r-- root root 30 1970-01-01 00:00 ueventd.vbox86.rc
lrwxrwxrwx root root 2014-02-25 20:19 vendor -> /system/vendor
root@android:/ #
```

Możliwości ADB

```
PS D:\adt-bundle-windows-x86_64-20131030\sdk\platform-tools> .\adb.exe -h
Android Debug Bridge version 1.0.31

-a          - directs adb to listen on all interfaces for a connection
-d          - directs command to the only connected USB device
            - returns an error if more than one USB device is present.
-e          - directs command to the only running emulator.
            - returns an error if more than one emulator is running.
-s <specific device> - directs command to the device or emulator with the given
            - serial number or qualifier. Overrides ANDROID_SERIAL
            - environment variable.
-p <product name or path> - simple product name like 'sooner', or
            - a relative/absolute path to a product
            - out directory like 'out/target/product/sooner'.
            - If -p is not specified, the ANDROID_PRODUCT_OUT
            - environment variable is used, which must
            - be an absolute path.
-H          - Name of adb server host (default: localhost)
-P          - Port of adb server (default: 5037)
devices [-l] - list all connected devices
            - ('-l' will also list device qualifiers)
connect <host>[:<port>] - connect to a device via TCP/IP
            - Port 5555 is used by default if no port number is specified.
disconnect [<host>[:<port>]] - disconnect from a TCP/IP device.
            - Port 5555 is used by default if no port number is specified.
            - Using this command with no additional arguments
            - will disconnect from all connected TCP/IP devices.

device commands:
adb push <local> <remote> - copy file/dir to device
adb pull <remote> [<local>] - copy file/dir from device
adb sync [-l <directory>] - copy host->device only if changed
                        - (-l means list but don't copy)
                        - (see 'adb help all')
adb shell - run remote shell interactively
adb shell <command> - run remote shell command
adb emu <command> - run emulator console command
adb logcat [-f <filter-spec>] - View device log
```



```
PS D:\adt-bundle-windows-x86_64-20131030\sdk\platform-tools> .\adb.exe devices
List of devices attached
192.168.56.101:5555    device

PS D:\adt-bundle-windows-x86_64-20131030\sdk\platform-tools> .\adb.exe pull data/app/ApiDemos.apk c:/ApiDemos.apk
5056 KB/s (3050203 bytes in 0.589s)

PS D:\adt-bundle-windows-x86_64-20131030\sdk\platform-tools> .\adb.exe uninstall data/app/ApiDemos.apk
Failure

PS D:\adt-bundle-windows-x86_64-20131030\sdk\platform-tools> .\adb.exe install C:\tcandroid204.apk
5988 KB/s (1202118 bytes in 0.196s)
pkg: /data/local/tmp/tcandroid204.apk
Success
```

Komendy powłoki systemowej

- ▶ Przydatne komendy powłoki systemowej:
 - **ls** listing plików i katalogów
 - **mkdir** stwórz nowy katalog
 - **rmdir** usuń katalog
 - **rm -r** usuń katalog(i) z plikami
 - **rm** usuń pliki
 - **mv** przeniesienie/zmiany nazwy plików
 - **cat** podgląd zawartości plików
 - **cd** zmiana aktualnej ścieżki
 - **pwd** aktualna ścieżka
 - **df** wolne miejsce na dysku
 - **chmod** zmiana uprawnień (do pliku/katalogu)
 - **date** wyświetlenie daty
 - **exit** zakończenie sesji
- ▶ Kiedyś nie było komendy kopiowania (cp), ale można ją zastąpić poleceniem cat:

```
#cat data/app/ApiDemos.apk > cache/apiDemos.apk
```

Wysyłanie wiadomości SMS / wykonywanie połączeń - Emulator

- ▶ telnet localhost **5554**
- ▶ sms send <numer> <treść>
- ▶ gsm call <numer dzwoniącego>

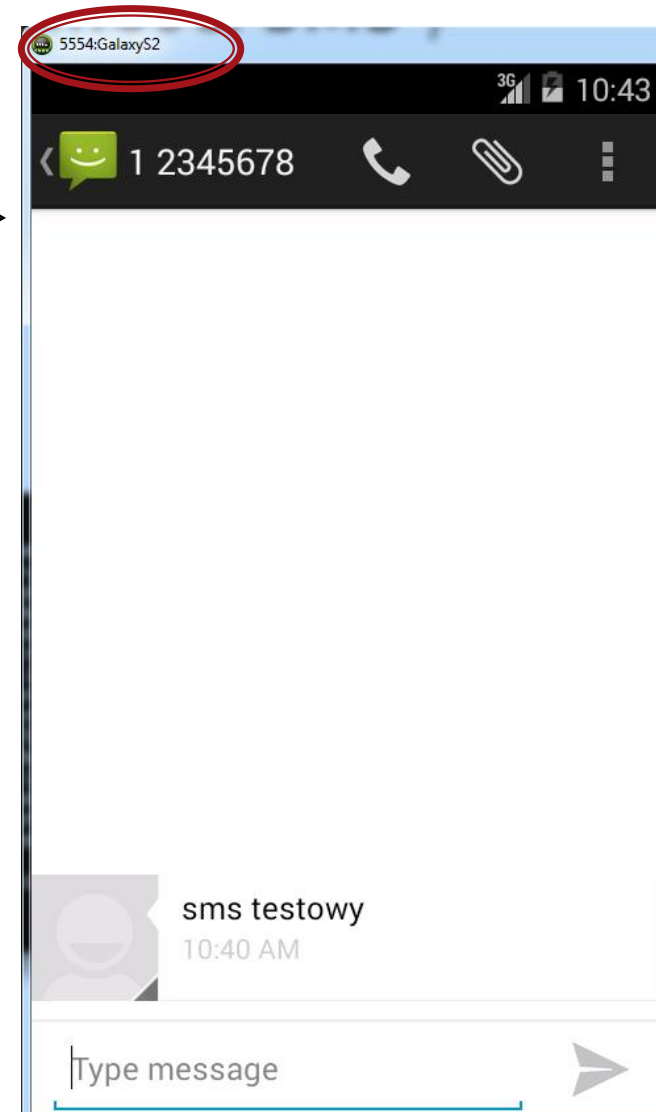
Android console command help:

help	print a list of commands
event	simulate hardware events
geo	Geo-location commands
gsm	GSM related commands
cdma	CDMA related commands
kill	kill the emulator instance
network	manage network settings
power	power related commands
quit	quit control session
redir	manage port redirections
sms	SMS related commands
avd	control virtual device execution
window	manage emulator window
qemu	QEMU-specific commands
sensor	manage emulator sensors

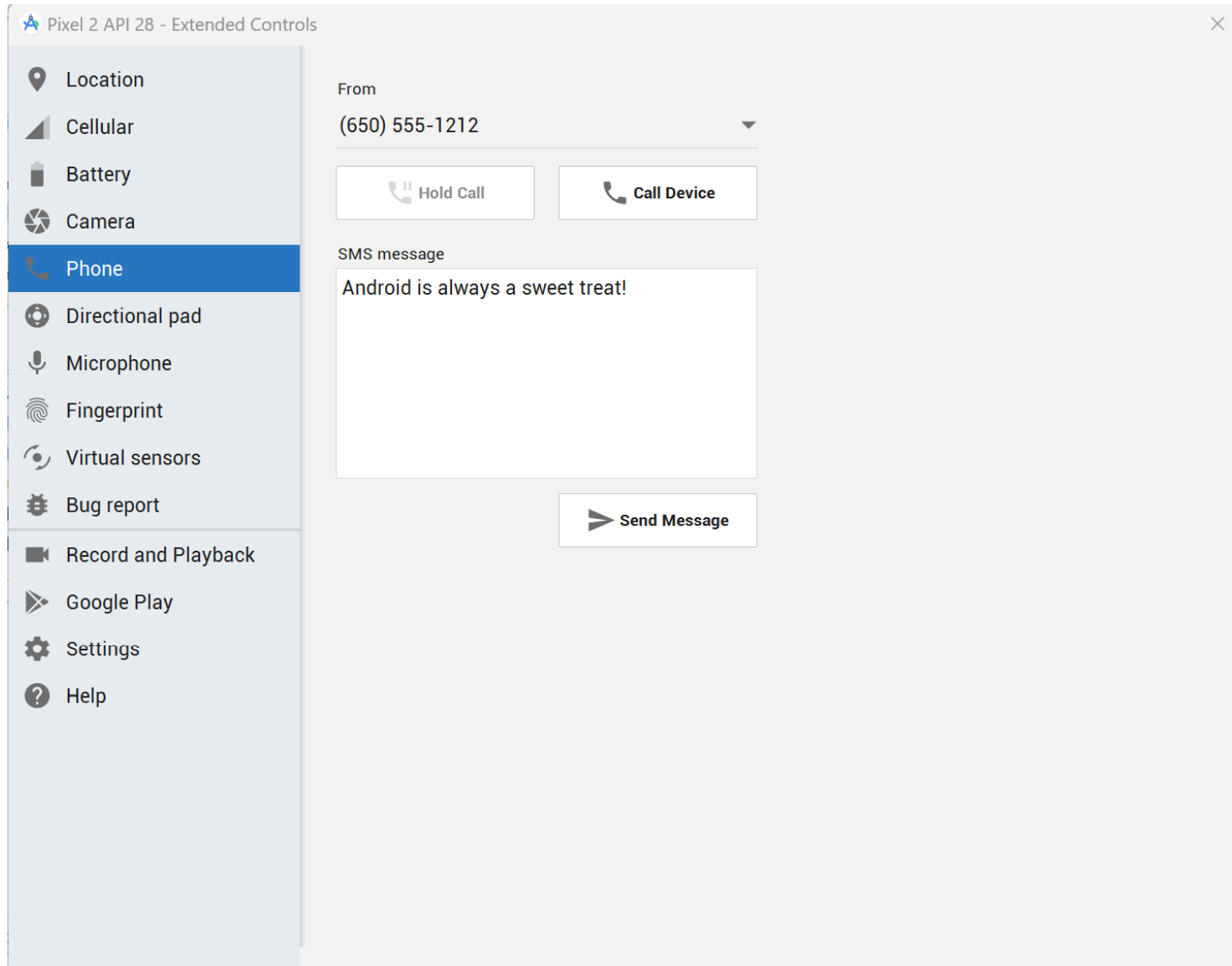
try 'help <command>' for command-specific help

OK
sms send 12345678 sms testowy

OK



SMS/Połączenia z GUI



Cykl życia aplikacji

- ▶ Każda aplikacja uruchamiana jest jako należąca do innego użytkownika.
- ▶ Tylko użytkownik o określonym ID (właściciel) ma bezpośredni dostęp do wszystkich plików danej aplikacji.
- ▶ Każda aplikacja jest izolowana od pozostałych – uruchamiana w osobnej instancji wirtualnej maszyny.
- ▶ Każda aplikacja to osobny proces systemowy. Proces jest uruchamiany, gdy jakikolwiek komponent aplikacji musi zostać uruchomiony i zakończony gdy nie jest potrzebny bądź należy zwolnić zasoby (w wyniku ich zapełnienia)

Elementy składowe aplikacji

- ▶ Aplikacja na Androida składa się z jednego lub więcej elementu składowego.
- ▶ Takim elementem może być:
 1. **Activity**
 2. **Service**
 3. **Broadcast receiver**
 4. **Content provider**



Activity (aktywność)

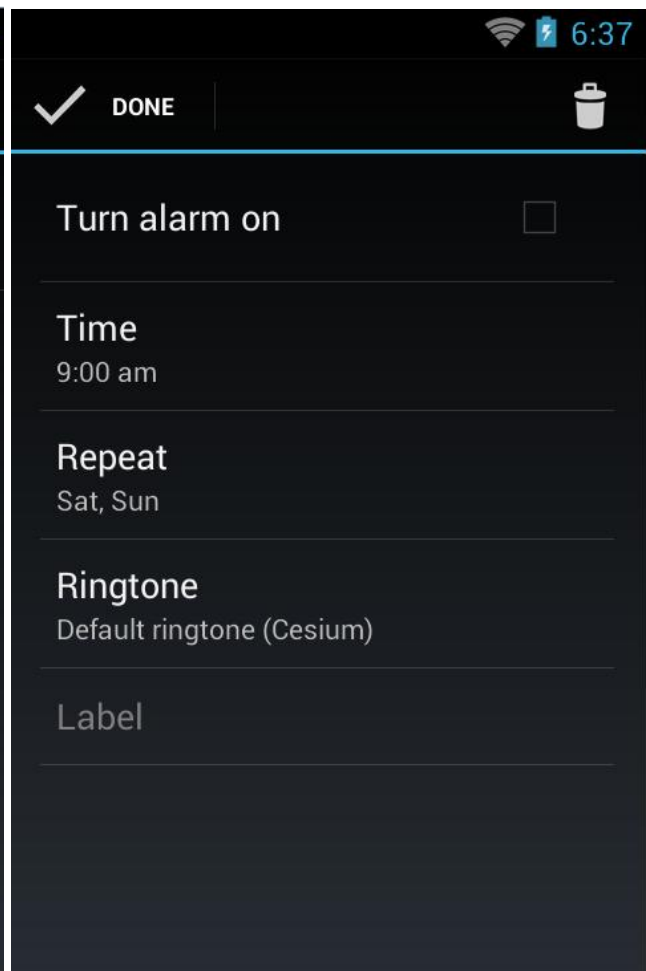
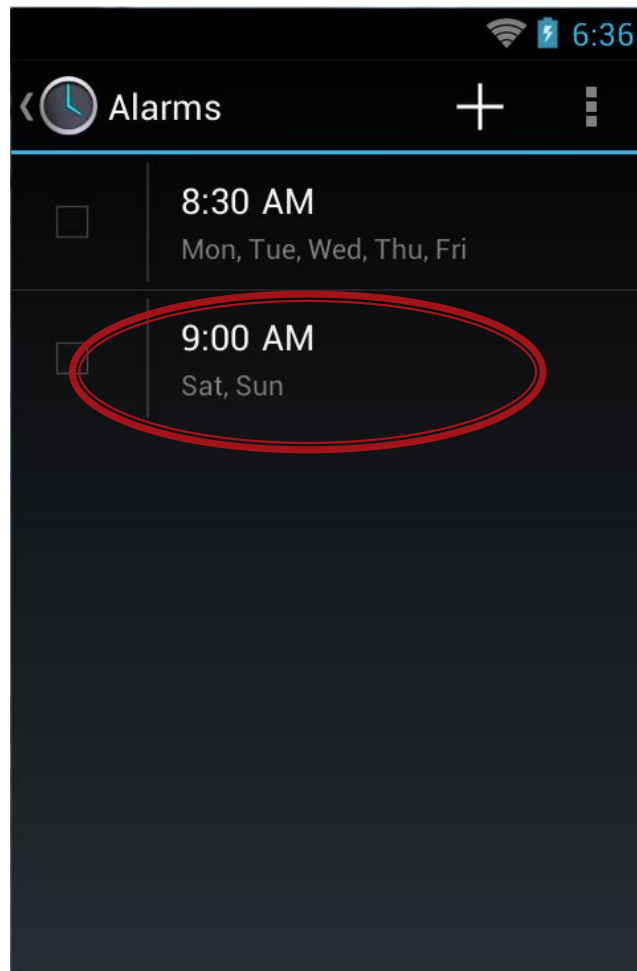
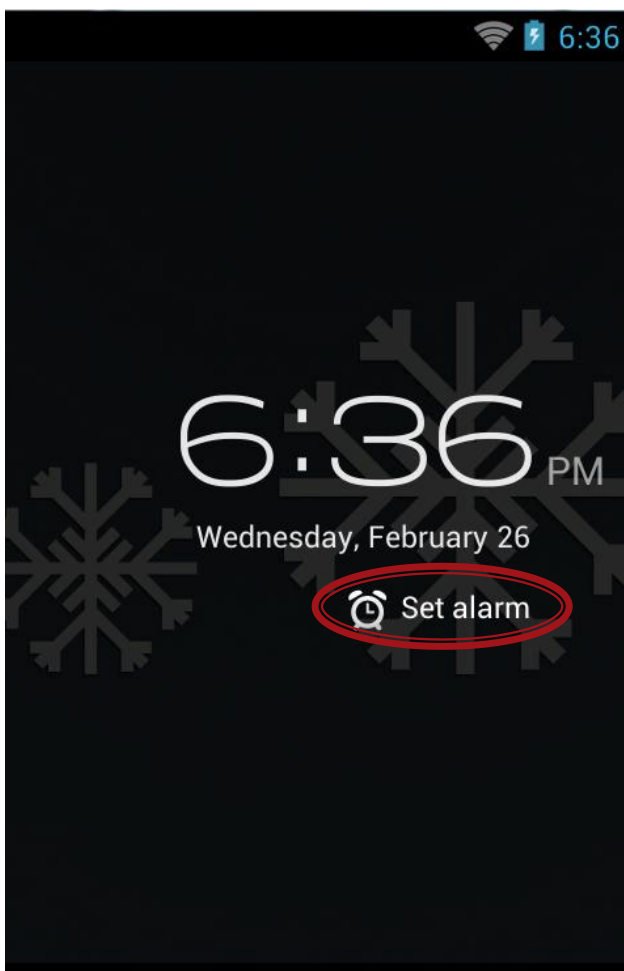
- ▶ Zwykle aplikacja składa się z jednej lub więcej aktywności.
- ▶ Tylko jedna aktywność (zwana główną) jest wybrana do wyświetlania przy pierwszym uruchomieniu aplikacji.
- ▶ Aktywność może przekazać sterowanie (i dane) do innej aktywności wykorzystując protokół komunikacyjny zwany intencją (intent).
- ▶ Aktywność zazwyczaj utożsamiana jest z pojedynczym ekranem GUI.

Przykład trzech aktywności

Aktywność 1

Aktywność 2

Aktywność 3



Usługa (Service)

- ▶ Usługa to specjalny typ komponentu, która nie posiada wizualnej reprezentacji
- ▶ Usługi zwykle uruchamiane są w tle na nieograniczony przedział czasu
- ▶ Aplikacje mogą uruchamiać własne usługi lub korzystać z już aktywnych
- ▶ Przykład:
Usługa GPS działa w tle i co jakiś czas wysyła dane lokalizacyjne do aplikacji nimi zainteresowanymi



Broadcast receiver

- ▶ Broadcast receiver to „słuchacz” oczekujący na globalnie (w obrębie całego systemu) przesyłane komunikaty.
- ▶ Nie posiada graficznego interfejsu.
- ▶ Zwykle broadcast receiver jest rejestrowany w systemie za pomocą filtra. Gdy komunikat zostanie dopasowany do danego filtra, BR jest aktywowany.
- ▶ Broadcast receiver może odpowiedzieć na komunikat wywołując określoną aktywność lub wyświetlając powiadomienie (notification).

Dostawca treści (content provider)

- ▶ Jego zadaniem jest udostępnianie zbiorów danych aplikacjom.
- ▶ Typowe zbiory danych (globalne): kontakty, zdjęcia, wiadomości, pliki audio itp.
- ▶ Globalne zbiory danych często przechowywane są w bazie SQLite.
- ▶ Dostawca treści jest warstwą abstrakcyjną – dostarcza spójne metody do pobierania, dodawania, modyfikacji i usuwania danych, bez względu na specyfikę zbioru danych.

Activity Stack

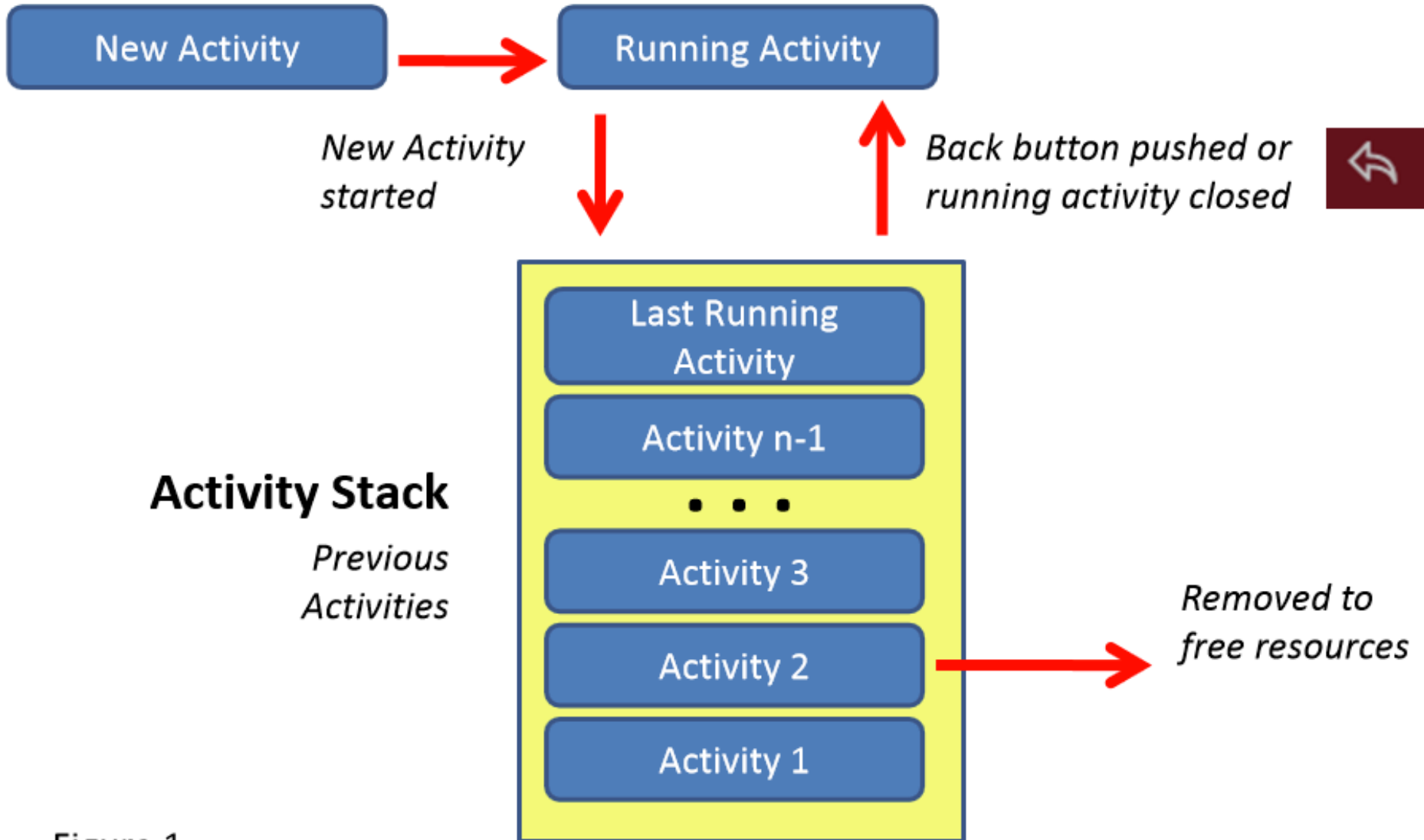


Figure 1

Implementacja callbacks

```
public class PrzykladAktywnosci extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Aktywność jest tworzona
    }
    @Override
    protected void onStart() {
        super.onStart();
        // Aktywność będzie widoczna
    }
    @Override
    protected void onResume() {
        super.onResume();
        // Aktywność jest widoczna (na pierwszym planie)
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Inna aktywność posiada focus
    }
    @Override
    protected void onStop() {
        super.onStop();
        // Aktywność nie jest już widoczna
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // Aktywność zostanie usunięta
    }
}
```

Zakończenie cyklu życia

- ▶ Metody **onPause()**, **onStop()**, **onDestroy()** mają status *killable*, czyli po zakończeniu dowolnej z nich, pozostałe nie muszą zostać wywołane (jeśli system wymusi zakończenie aplikacji).
- ▶ **onPause()** to jedyna metoda, która na pewno będzie wywołana przed zakończeniem aplikacji.
- ▶ Metoda **onPause()** powinna być wykorzystywana do zapisania stanu aplikacji.

Badanie cyklu życia aktywności - przykład

```
package pl.tomaszx.helloworld;

import android.os.Bundle;

public class MainActivity extends Activity {
    private Context context;

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

        Button btnExit = (Button) findViewById(R.id.button1);

        btnExit.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

        context = getApplicationContext();
        Toast.makeText(context, "onCreate", Toast.LENGTH_SHORT).show();
    }
}
```

Badanie cyklu życia aktywności – przykład c.d.

```
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    Toast.makeText(context, "onDestroy", Toast.LENGTH_SHORT).show();  
}
```

```
@Override  
protected void onPause() {  
    super.onPause();  
    Toast.makeText(context, "onPause", Toast.LENGTH_SHORT).show();  
}
```

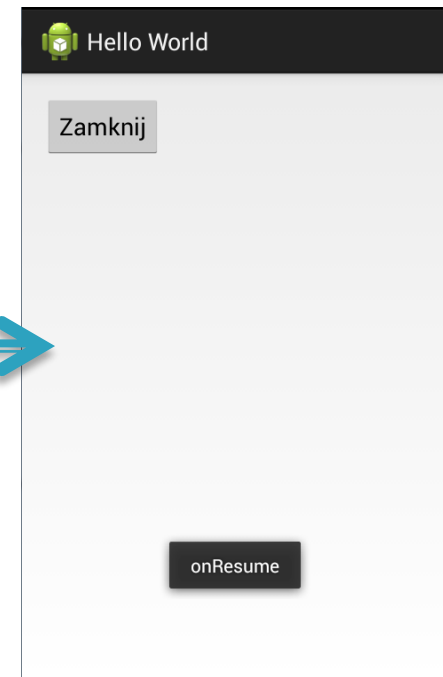
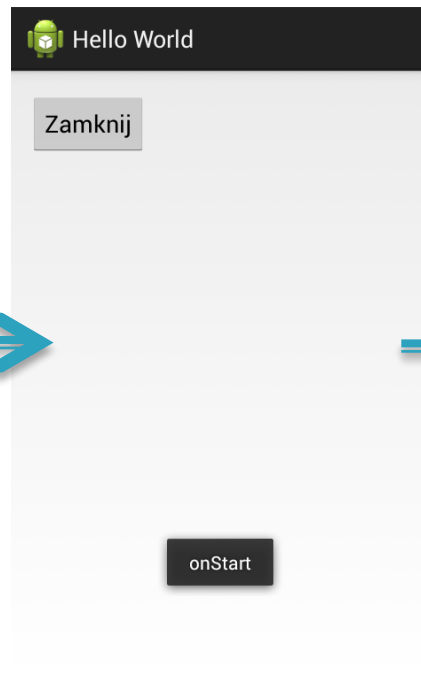
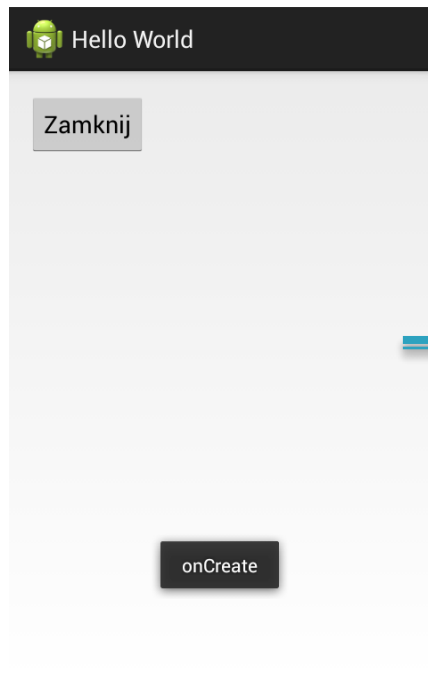
```
@Override  
protected void onRestart() {  
    super.onRestart();  
    Toast.makeText(context, "onRestart", Toast.LENGTH_SHORT).show();  
}
```

```
@Override  
protected void onResume() {  
    super.onResume();  
    Toast.makeText(context, "onResume", Toast.LENGTH_SHORT).show();  
}
```

Badanie cyklu życia aktywności – przykład c.d.

```
@Override  
protected void onStart() {  
    super.onStart();  
    Toast.makeText(context, "onStart", Toast.LENGTH_SHORT).show();  
}
```

```
@Override  
protected void onStop() {  
    super.onStop();  
    Toast.makeText(context, "onStop", Toast.LENGTH_SHORT).show();  
}
```



Android Studio Demo

The screenshot displays the Android Studio interface. The top toolbar includes icons for file operations, navigation, and execution. The menu bar contains options like File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the path: SQLite1 > app > src > main > AndroidManifest.xml. The left sidebar features the Project view (showing the app structure with folders like manifests, java, and res), the Structure view, Captures, Build Variants, and Favorites. The main editor window displays the content of the AndroidManifest.xml file, which is currently selected in the tabs. The code defines the application's package, theme, and several activities. The bottom status bar shows 'Text' and 'Merged Manifest' tabs, along with system information like '1:1 CRLF', 'UTF-8', and 'Context: <no context>'. The bottom-most status bar indicates 'Gradle build finished in 6s 281ms (8 minutes ago)'.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.pietrzyk.sqlite1">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="SQLite1"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="SQLite1"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".RefuelNewActivity"
            android:label=""
            //brak tytułu na toolbarze
        </activity>
        <activity android:name=".RefuelListActivity" />
        <activity
            android:name=".RefuelUpdateActivity"
            android:label="UpdateRefuelActivity"
            android:theme="@style/AppTheme" />
    </application>
</manifest>
```

Pytania? Problemy?

Dziękuję za uwagę!