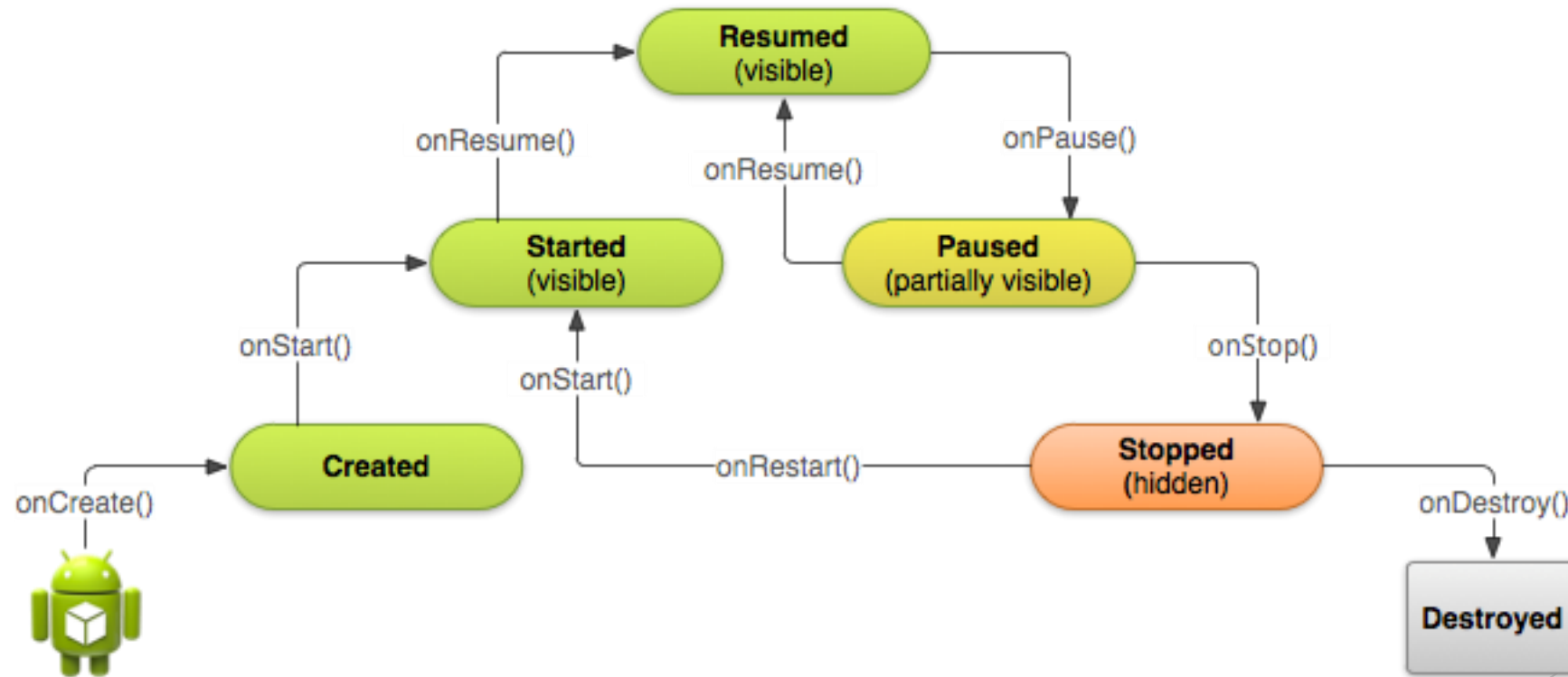


Android - III. előadás

Miről lesz ma szó?

- ▶ Még egy kicsit az Activity-ről
- ▶ Különböző kijelzők és méretek támogatása
- ▶ Felhasználói felület tervezése
- ▶ Layoutok
- ▶ UI elemek és sajátosságaik

Egy Activity élete és halála



Állapotok közötti váltás

- ▶ Minden állapotváltásnál
- ▶ A megfelelő meghívódik
 - ▶ Ha nem definiáltuk felül, akkor az őosztályé (Activity)
 - ▶ Ha felüldefiniáltuk, akkor először mindig meg kell hívni az őosztályét!
 - ▶ `super.onCreate()`
- ▶ A rendszer ezeket automatikusan meghívja
- ▶ A fejlesztőnek feladata ezeket logikával feltölteni

Activity váz

```
public class MyActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        //setContentView(R.layout.activity_my);  
  
        //Létrejövünk...  
    }  
  
    @Override  
    protected void onStart() {  
        super.onStart();  
  
        //Elindulunk...  
    }  
  
    @Override  
    protected void onResume() {  
        super.onResume();  
  
        //Láthatóvá válunk...  
    }  
}
```

```
    @Override  
    protected void onPause() {  
        super.onPause();  
  
        //Háttérbe szorulunk...  
    }  
  
    @Override  
    protected void onStop() {  
        super.onStop();  
  
        //Leállunk...  
    }  
  
    @Override  
    protected void onDestroy() {  
        super.onDestroy();  
  
        //Rövid kis életünk itt véget ért.  
    }  
}
```

Erőforrások (kód)testközelben

- ▶ Strings.xml-be:

```
<string name="welcome_message">Üdvözöllek, kedves %s!</string>
```

- ▶ Használat kódban:

```
String username = "Petőfi Sándor";  
String customWelcomeMessage = getString(R.string.welcome_message, username);
```

- ▶ Ugyanígy lehet bármilyen konstanst használni
- ▶ Miért jó?
 - ▶ Layoutokból is elérhetőek (bár nem paraméterezhetőek)
 - ▶ Egy helyen tartható minden szöveg (internationalization)
 - ▶ Akár egyszerű szkriptekkel generálható is táblázatból

Logolás Androidon

- ▶ android.util.Log
 - ▶ Log.d → debug
 - ▶ Log.e → error
 - ▶ Log.i → info
 - ▶ Log.w → warning
 - ▶ Log.v → verbose

```
Log.d("TAG", "üzenet");
```

```
1 Logcat:
2 02-02 18:52:57.132: VERBOSE/ProtocolEngine(24): DownloadRate 104166 bytes per sec. Downloaded B
3 08-03 13:31:16.196: DEBUG/dalvikvm(2227): HeapWorker thread shutting down
4 08-03 13:31:16.756: INFO/dalvikvm(2234): Debugger is active
5 08-03 16:26:45.965: WARN/ActivityManager(564): Launch timeout has expired, giving up wake lock!
6 08-04 16:19:11.166: ERROR/AndroidRuntime(4687): Uncaught handler: thread main exiting due to un
7 08-04 16:24:11.166: ASSERT/Assertion(4687): Expected true but was false
```

Különböző készülékek támogatása

- ▶ Végtelen gyártó és készüléktípus
- ▶ Végtelen képernyőméret és arány
- ▶ Az Android op rendszer automatizál
- ▶ Képek optimalizálása fontos!
- ▶ Lehet specifikálni a megjelenítést

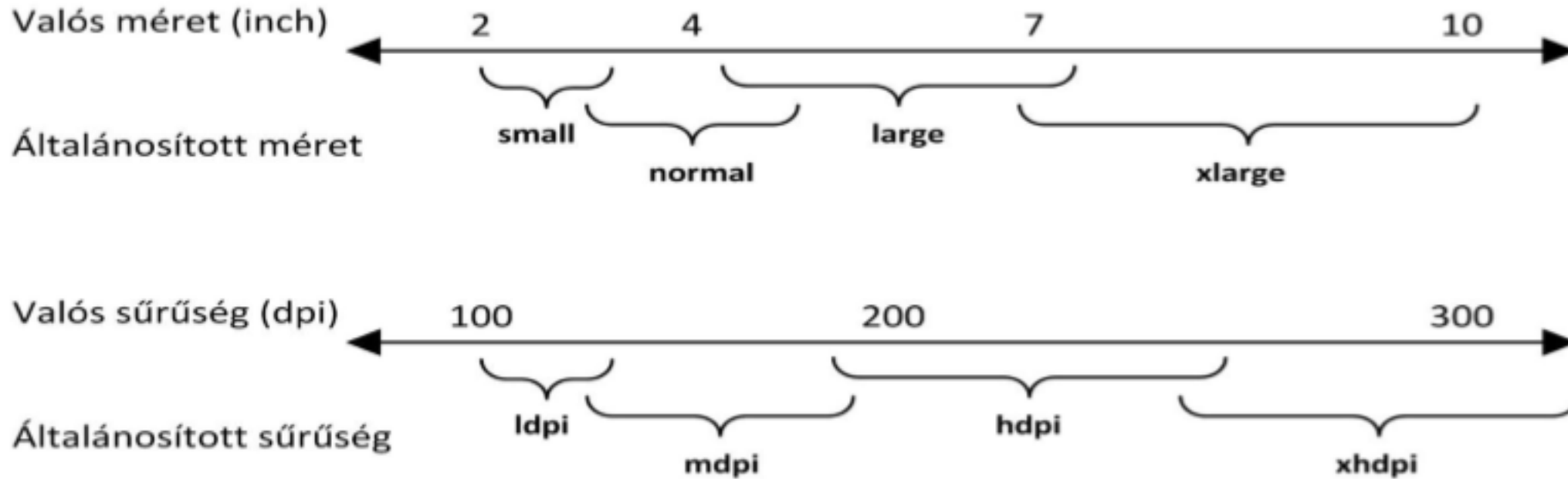
Fogalmak

- ▶ Képernyő méret (screen size)
 - ▶ Fizikai képátló
 - ▶ 4 kategória: small, normal, large és extra large (xlarge)
- ▶ Pixelsűrűség (screen density - dpi)
 - ▶ Pixelek száma egy egységen (inch)
 - ▶ Itt is 4 kategória: low, medium, high, extra high (xhigh)
 - ▶ Újabban van xxhigh és xxlarge
- ▶ Orientáció
 - ▶ Álló (portrait) és Fekvő (landscape)
 - ▶ Változtathatja a rendszer (felhasználói tevékenység miatt) egy alkalmazás (videolejátszás)
 - ▶ Rögzíthető is (játék)

Sűrűségfüggetlen pixel

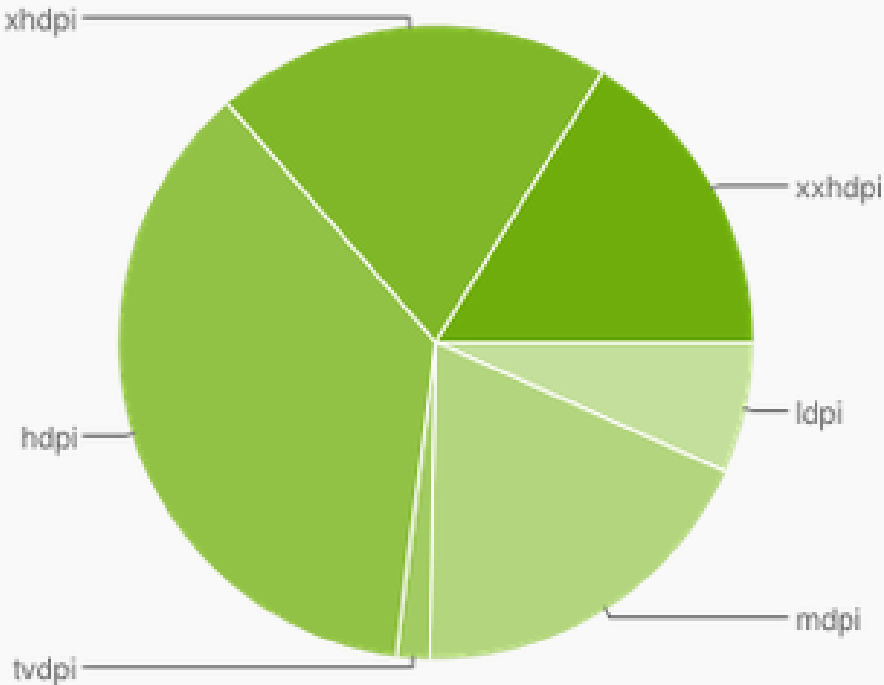
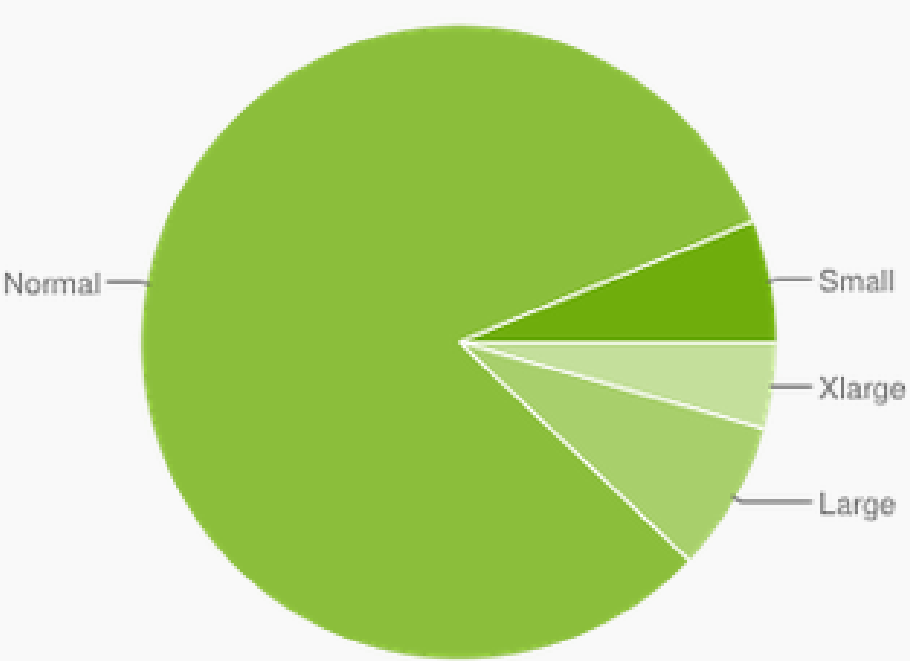
- ▶ Pixel nem jó mérőszám, ahány képernyősűrűség, annyi féleképpen reprezentálódik
- ▶ dp (dip - density independent pixel)
- ▶ Virtuális pixel egység
- ▶ Kibökött középérték alapján 160dpi-is képernyő 1 pixele az 1dp
- ▶ Rendszer számol és skáláz, ha nem ekkora
- ▶ $Px = dp * (dpi / 160)$
- ▶ Pl. 240 dpi-s kijelzőn 1dp:
- ▶ $1 * (240 / 160) = 1.5pixel$

Kategóriák ábrázolva



- ▶ Minimumok:
 - ▶ xlarge: legalább 960dp x 720dp
 - ▶ large: legalább 640dp x 480dp
 - ▶ normal: legalább 470dp x 320dp
 - ▶ small: legalább 426dp x 320dp

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	6.2%						6.2%
Normal		10.6%		35.7%	19.2%	16.2%	81.7%
Large	0.5%	4.3%	1.7%	0.6%	0.6%		7.7%
Xlarge		3.7%		0.3%	0.4%		4.4%
Total	6.7%	18.6%	1.7%	36.6%	20.2%	16.2%	



*Data collected during a 7-day period ending on September 9, 2014.
Any screen configurations with less than 0.1% distribution are not shown.*

Példák



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.

Erőforrások elhelyezése, a választás algoritmus

- ▶ res/ mappán belül
- ▶ <eroforras_neve>-<tipus>
 - ▶ Ahol az erőforrás neve lehet pl. drawable, layout, values...
 - ▶ A típus pedig a minősítő, lehet belőle több is egymás után kötőjellel, pl. hdpi, xlarge, port, en
- ▶ Típus minősítők:
 - ▶ Méret: small, normal, large, xlarge
 - ▶ Sűrűség: ldpi, mdpi, hdpi, xhdpi, xxhdpi, xxxhdpi 😊
 - ▶ Irány: land, port
 - ▶ Egyedi: sw720dp, h600dp
- ▶ Hogyan találja meg a rendszer, hogy melyik lesz felhasználva?

View / ViewGroup

▶ View

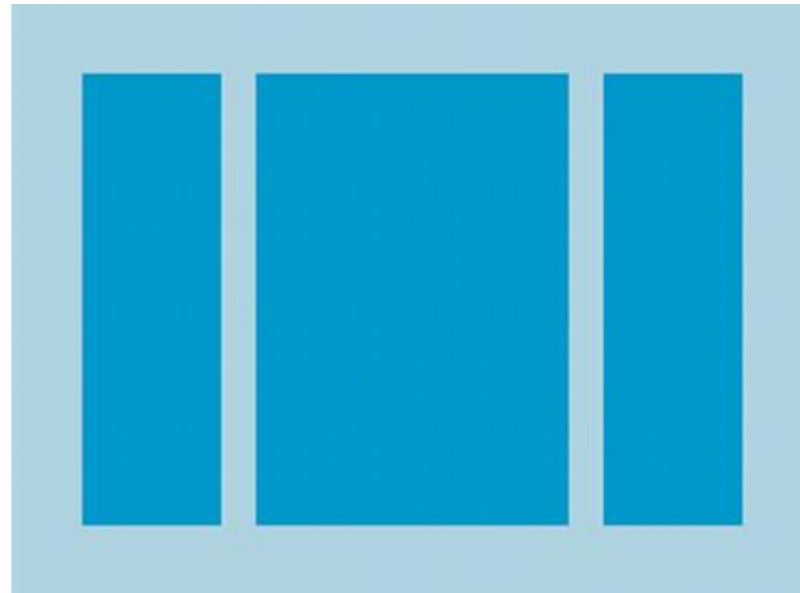
- ▶ Minden megjelenítendő elem ősatya
- ▶ Kötelező attribútumok innen (width, height, id)
- ▶ Neki van onclick figyelő berendezése (ezért kapunk ott mindig paraméterként egy View-t)

▶ ViewGroup

- ▶ Minden olyan elem ősatya, amely tartalmazhat View-kat
- ▶ Ő is View egyébként

LinearLayout

- ▶ Kötelező attributum: orientation
 - ▶ Land
 - ▶ Port
- ▶ Benne az elemeket lehet állítgatni
 - ▶ gravity
 - ▶ layout_gravity
 - ▶ layout_weight



RelativeLayout

- ▶ Benne lévő elemek pozíciója
 - ▶ Szülő (RelativeLayout)
 - ▶ Testvér elemek
- ▶ Attributum példák:
 - ▶ `layout_alignParentTop = true`
 - ▶ `layout_centerVertical = true`
 - ▶ `layout_below = @+id/masikElem`
 - ▶ `layout_toRightOf = @+id/masikElem`
 - ▶ `layout_alignBottom = @+id/masikElem`



Rajzolás példa

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

```
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="1" />
```

```
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
```

```
        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="0.5"
            android:text="2" />
```

```
        <Button
            android:id="@+id/button3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:layout_weight="0.5"
            android:text="3" />
```

```
    </LinearLayout>
```

```
    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="4" />
```

```
</LinearLayout>
```

Ennyit mára.

<http://erdekes.herokuapp.com>
gyulavari.adam+petrik@gmail.com