

Mobil alkalmazásfejlesztés - Menük, Navigáció

Dr. Bilicki Vilmos

Szoftverfejlesztés Tanszék



A fóliához felhasznált anyagok:

Google: Android Developer Fundamentals (Version 2), <https://developer.android.com/courses/fundamentals-training/overview-v2>

Az előző előadás áttekintése:

- ▶ Menük és kiválasztók
 - Áttekintés
 - Alkalmazás sáv opciók menüvel
 - Kontextuális menük
 - Popup menük
 - Dialógus-ok
 - Picker-ek
- ▶ Felhasználói navigáció
 - Vissza
 - Hierarchikus



A mai alkalom áttekintése

- ▶ RecyclerView
 - Komponensek
 - Megvalósítása
- ▶ Egyedi felhasználói élmény
 - Drawables
 - Stílusok
 - Témák
- ▶ Material design
 - Az alapelv
 - Képek
 - Színek
 - Elrendezések
 - Komponensek
- ▶ Adaptív elrendezés erőforrások
 - Adaptív erőforrások
 - Alternatív erőforrások





RecyclerView

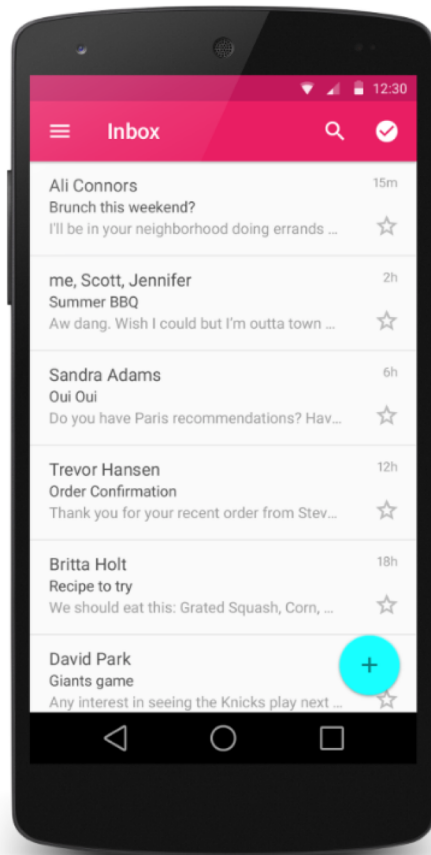


Figure 1. A list using `RecyclerView`

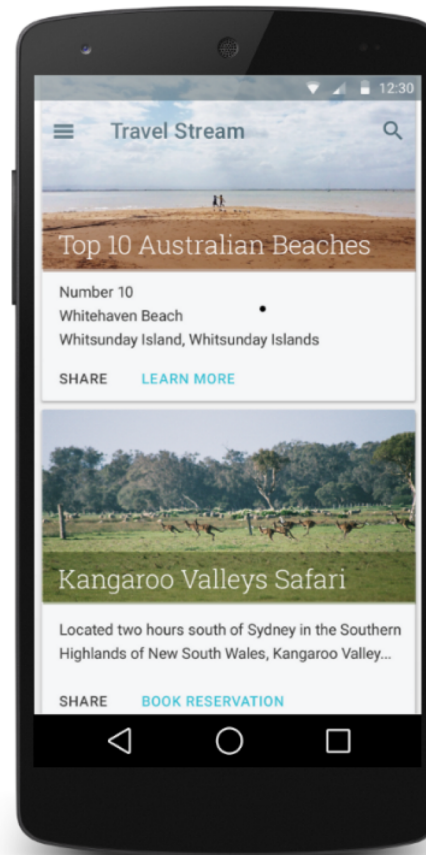
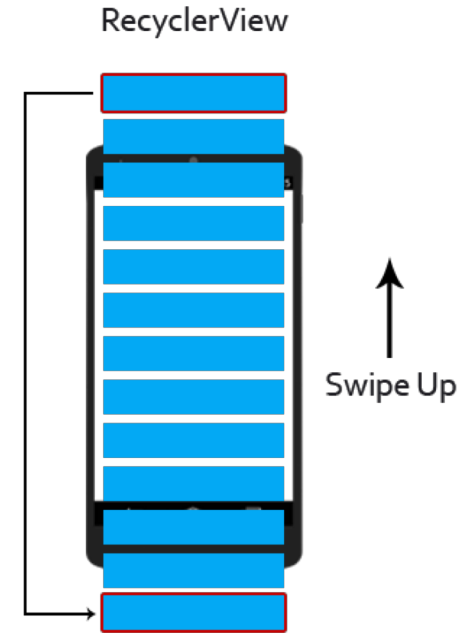
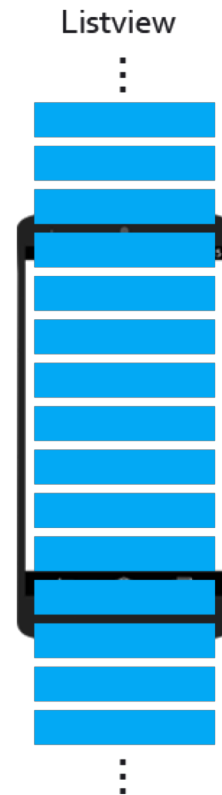
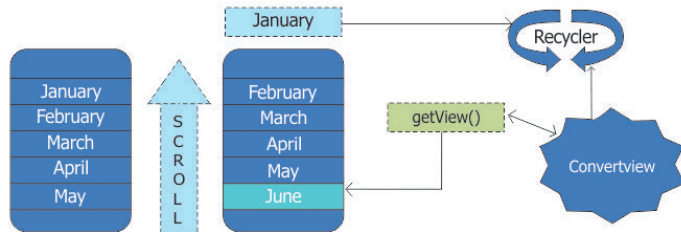


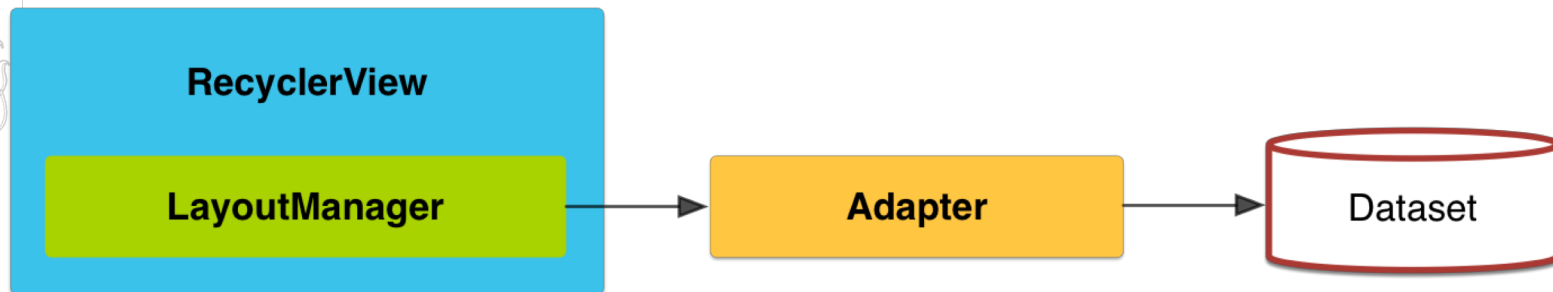
Figure 2. A list also using `CardView`

RecyclerView

- Komponentek
- Megvalósítása

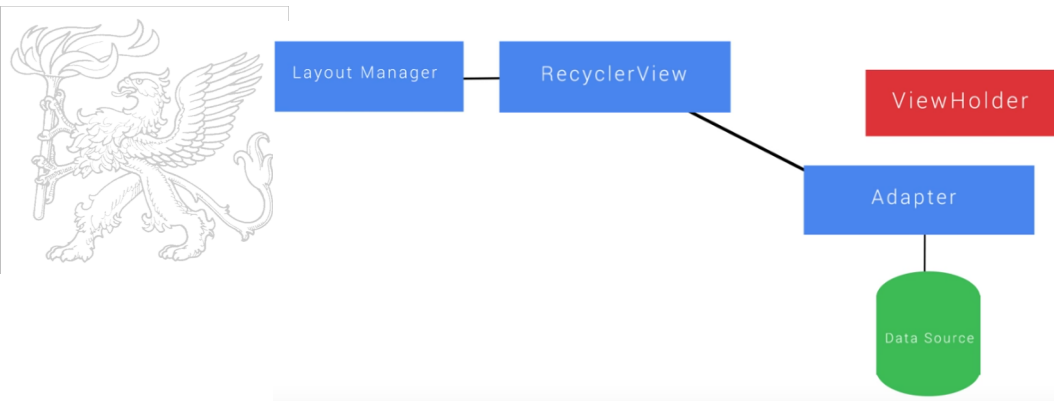


©WiselTeach

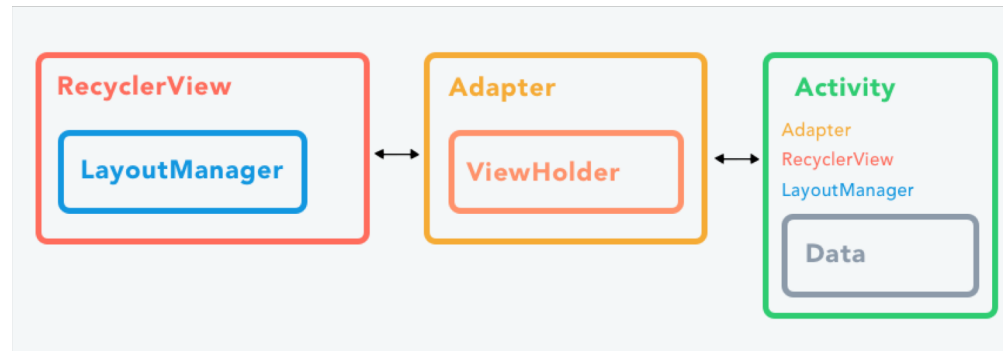


RecyclerView

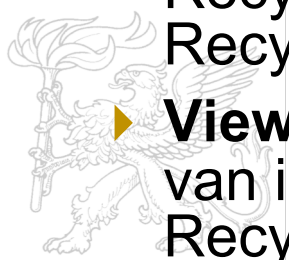
- A [RecyclerView](#) egy nagy adathalmaz kezelésére is alkalmas görgethető tároló
- Hatékony mert
 - Újrahasznosít adott számú nézet elemet hasznosít
 - Az adat váltótást gyorsan követi



Áttekintés

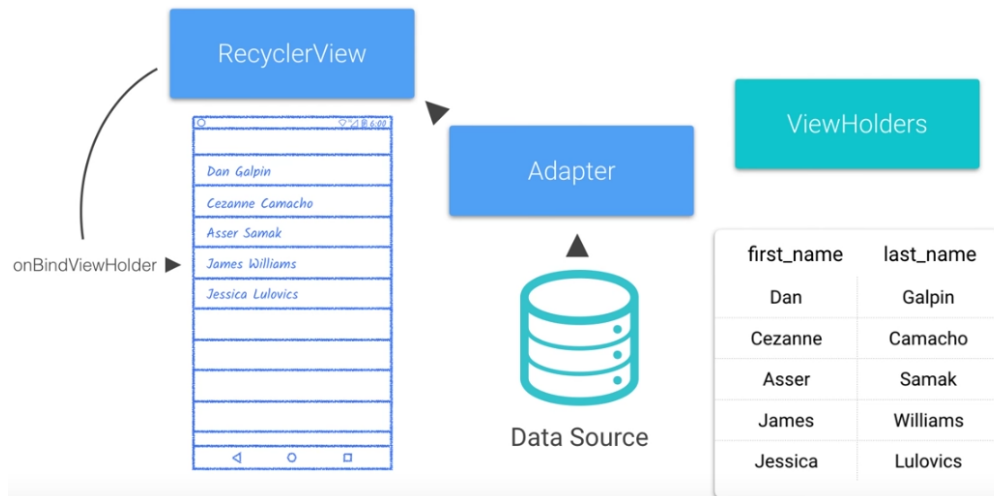
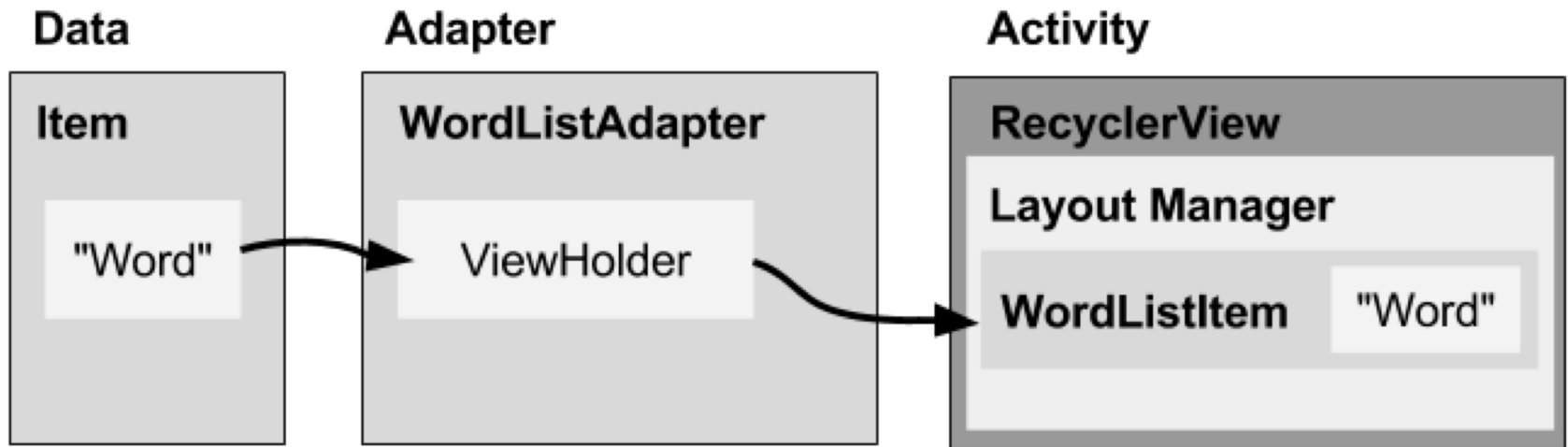


- ▶ **Adat**
- ▶ **RecyclerView** lista elemeket tartalmazó görgető lista — RecyclerView
- ▶ **Elrendezés** az adat egy eleméhez — XML fájl
- ▶ **Elrendezés menedzser** kezeli az UI komponensek szervezését a nézetben — RecyclerView.LayoutManager
- ▶ **Adapter** köti össze az adatot a RecyclerView-val — RecyclerView.Adapter
- ▶ **ViewHolder** egy elem megjelenítéséhez van információja — RecyclerView.ViewHolder





A komponensek együttműködése

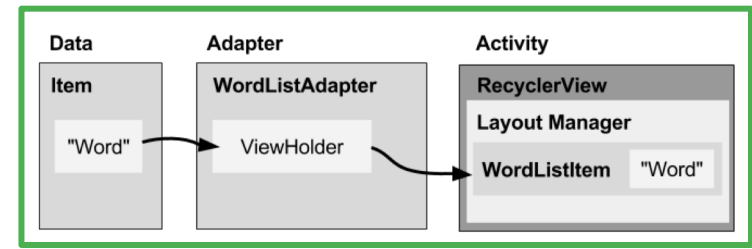


Elrendezéskezelő

- ▶ Minden ViewGroup-nak van elrendezés kezelője
- ▶ Az egyes View elemeket kezeli a RecyclerView-en belül
- ▶ Újrahasznosítja a már nem látható View elemeket
- ▶ Beépített elrendezés kezelők:
 - LinearLayoutManager
 - GridLayoutManager
 - StaggeredGridLayoutManager
- ▶ Kiegészíthetjük RecyclerView.LayoutManager



Adapter

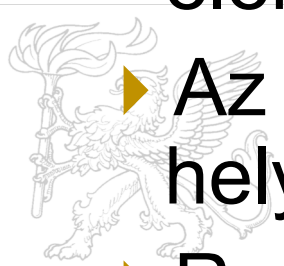
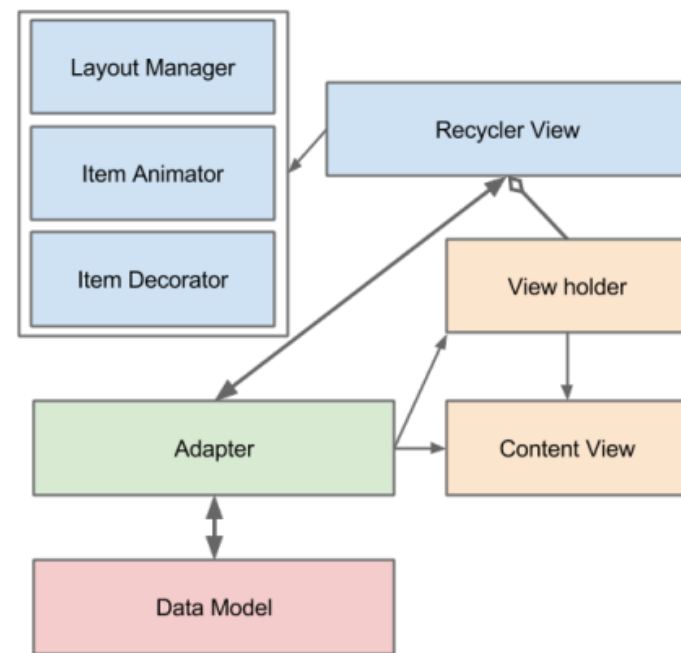
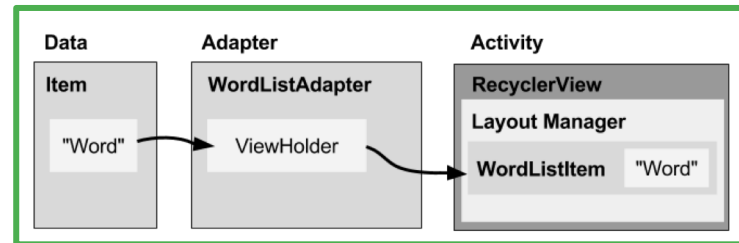


- ▶ Segít az egyébként nem kompatibilis interfészek illesztésében
 - Például: Kiolvassa az adatbázis kurzor segítségével és karakterláncként átadja a View-nak
- ▶ Közvetítő az adat és a View között
- ▶ Kezeli a View elemek létrehozását, frissítését, törlését az adat változásának függvényében.
- ▶ RecyclerView.Adapter



ViewHolder

- ▶ Az adatper használja, hogy egy View elemet kitöltsön adattal
- ▶ Az elrendezés XML erőforrás fájlban adott
- ▶ Lehetnek kattintható elemei
- ▶ Az elrendezés kezelő helyezi el
- ▶ RecyclerView.ViewHolder



RecyclerView megvalósítása

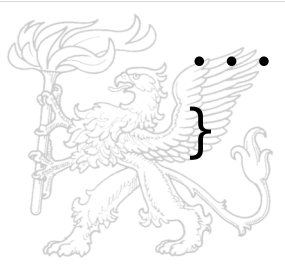
1. Adjuk hozzá a RecyclerView függőséget a build.gradle-hez (amennyiben kell)
2. Adjuk hozzá a RecyclerView-et az elrendezéshez
3. Hozzunk létre egy elemhez tartozó XML elrendezést
4. Bővítsük a RecyclerView.Adapter-t
5. Bővítsük a RecyclerView.ViewHolder-t
6. Az Activity-ben lévő onCreate()-ben, hozzuk létre a RecyclerView-et megfelelő adapterrel és elrendezés kezelővel



Adjuk hozzá a függőségeket az app/build.gradle-hoz

Adjuk hozzá a RecyclerView függőséget a build.gradle-hoz amennyiben szükséges:

```
dependencies {  
    ...  
    compile 'com.android.support:recyclerview-  
v7:26.1.0'
```



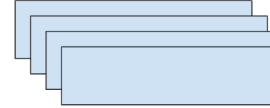
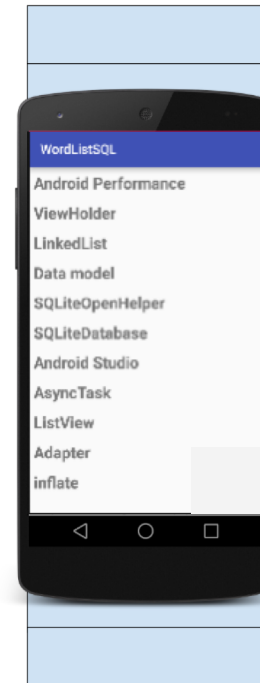
Adjuk hozzá a RecyclerView-et az elrendezés XML-hez

```
<android.support.v7.widget.RecyclerView  
    android:id="@+id/recyclerview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
</android.support.v7.widget.RecyclerView>
```



Hozzuk létre egy elemhez tartozó elrendezést

```
<LinearLayout ...>
    <TextView
        android:id="@+id/word"
        style="@style/word_title" />
</LinearLayout>
```



Valósítsuk meg az adaptert

```
public class WordListAdapter
    extends RecyclerView.Adapter<WordListAdapter.WordViewHolder> {

    public WordListAdapter(Context context,
                            LinkedList<String> wordList) {
        mInflater = LayoutInflater.from(context);
        this.mWordList = wordList;
    }
}
```



3 kötelező metódusa van

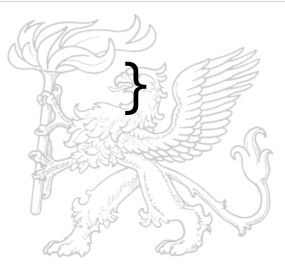
- `onCreateViewHolder()`
- `inBindViewHolder()`
- `getItemCount()`



onCreateViewHolder()

@Override

```
public ViewHolder onCreateViewHolder(  
    ViewGroup parent, int viewType) {  
    // Create view from layout  
    View itemView = inflater.inflate(  
        R.layout.wordlist_item, parent, false);  
    return new ViewHolder(itemView, this);  
}
```



onBindViewHolder()

```
@Override  
public void onBindViewHolder(  
    WordViewHolder holder, int position) {  
    // Retrieve the data for that position  
    String mCurrent = mWordList.get(position);  
    // Add the data to the view  
    holder.wordItemView.setText(mCurrent);  
}
```



getItemCount()

```
@Override  
public int getItemCount() {  
    // Return the number of data items to display  
    return mWordList.size();  
}
```



Hozzuk létre a nézet befogadót

```
class WordViewHolder extends RecyclerView.ViewHolder { //.. }
```

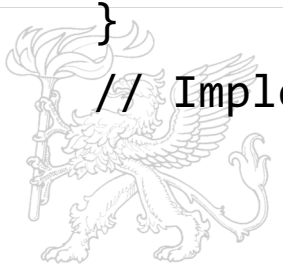
Amennyiben kattintást szeretnénk kezelni

```
class WordViewHolder extends RecyclerView.ViewHolder  
    implements View.OnClickListener { //.. }
```



Nézet befogadó konstruktor

```
public WordViewHolder(View itemView, WordListAdapter adapter) {  
    super(itemView);  
    // Get the layout  
    wordItemView = itemView.findViewById(R.id.word);  
    // Associate with this adapter  
    this.mAdapter = adapter;  
    // Add click listener, if desired  
    itemView.setOnClickListener(this);  
}  
// Implement onClick() if desired
```



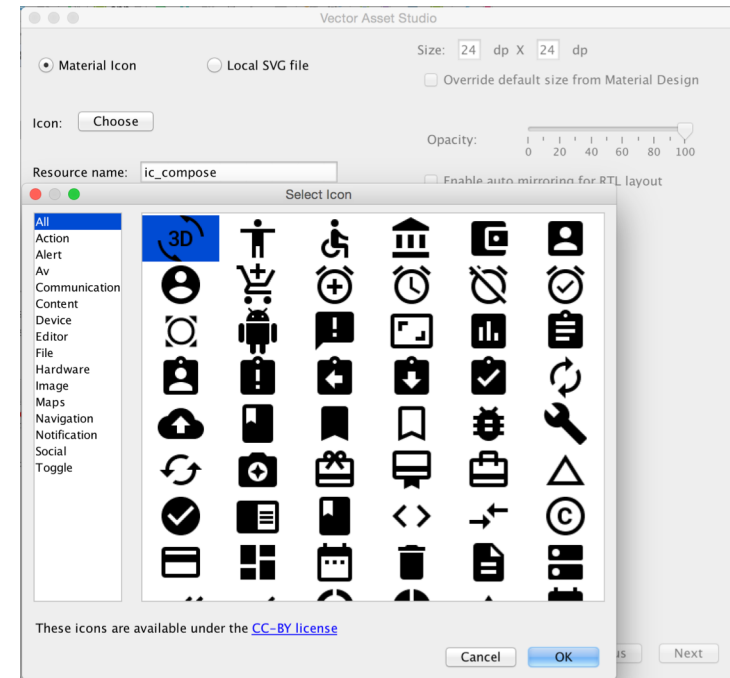
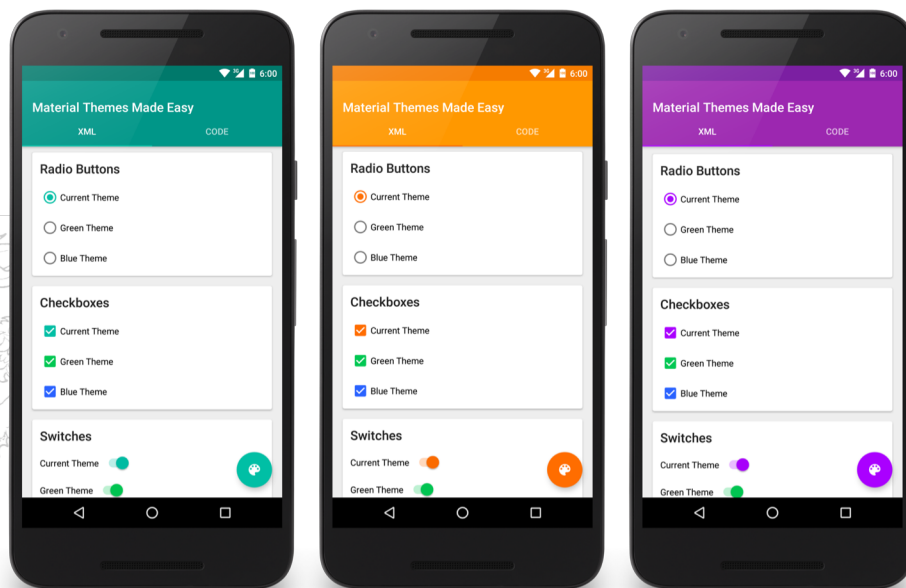
Hozzuk létre a RecyclerView-et az Activity onCreate()-ben

```
mRecyclerView = findViewById(R.id.recyclerview);  
mAdapter = new WordListAdapter(this, mWordList);  
mRecyclerView.setAdapter(mAdapter);  
mRecyclerView.setLayoutManager(new  
    LinearLayoutManager(this));
```



Felhasználói élmény

- ▶ Rajzolható elemek
- ▶ Stílusok
- ▶ Témák



Drawables- Rajzolható elemek

- ▶ Drawable—általános Android osztály grafika megvalósítására
- ▶ Minden drawable a res/drawable projekt mappában van tárolva

[Bitmap File](#)

[Nine-Patch File](#)

[Layer List Drawable](#)

[Shape Drawable](#)

[State List Drawable](#)

[Level List Drawable](#)

[Transition Drawable](#)

[Vector Drawable](#)

... és sok más

Egyedi Drawable



Bittérképek

- PNG (.png), JPG (.jpg), vagy GIF (.gif) formátumok
- Tömörítetlen BMP (.bmp)
- WebP (4.0 vagy magasabb)
- Egy BitmapDrawable adattípust hoz létre
- Közvetlenül a res/drawables-be kerül



Hivatkozás Drawable-ra

- XML: @[package:]drawable/fájlnév

```

<ImageView
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:src="@drawable/myimage" />
    
```

- Java kód: R.drawable.fájlnév

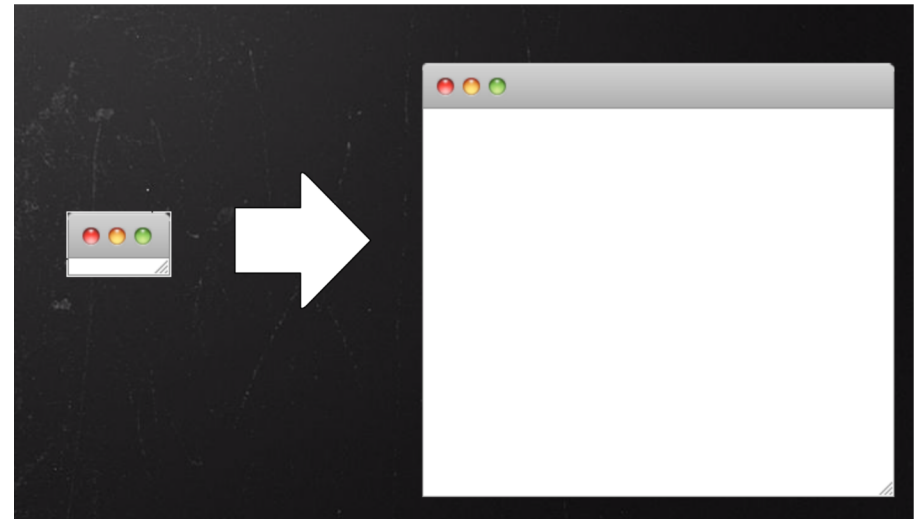
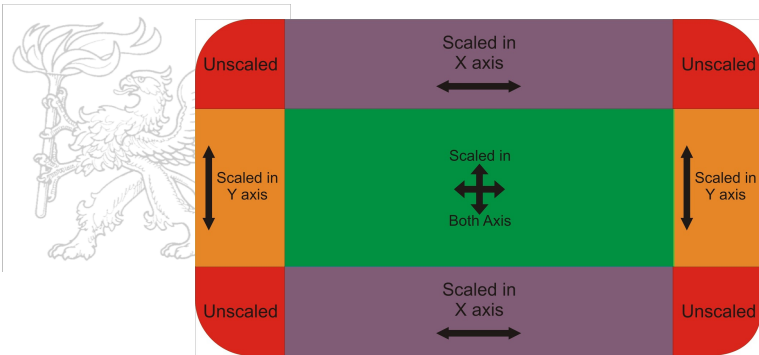
```

Resources res = \_getResources\(\);
Drawable drawable = res.getDrawable(R.drawable.myimage);
    
```



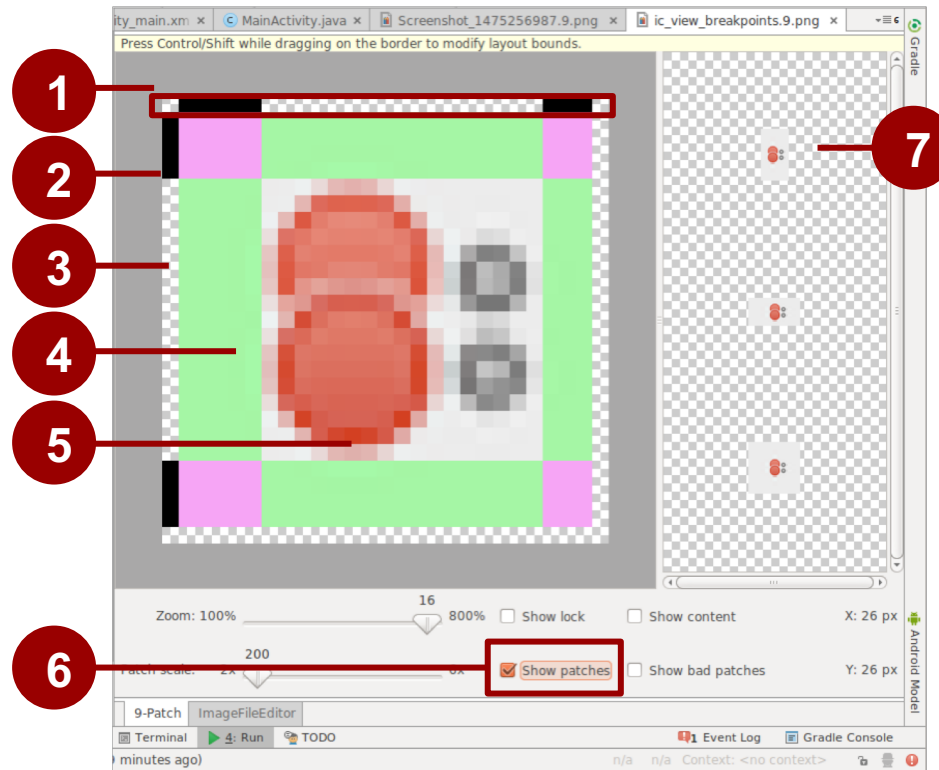
9 patch image

- ▶ Nine-patch image (UIImage)
- ▶ PNG széthúzható régiókkal
- ▶ Gyakran az egyes UI elemek háttere
- ▶ Pl.: Gomb háttere mely mérete a címkével változik



Létrehozása

1. A széthúzható terület határai:szélesség
2. A széthúzható terület határai:hosszúság
3. -
4. Széthúzható terület
5. Nem széthúzható terület
6. Megtekintése



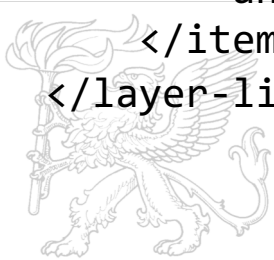
Rétegelt lista

- ▶ Különböző kép szerkesztő eszközzel létrehozhatunk rétegelt képet
- ▶ Ezeket egy-egy drawable formájában kezelhetjük Android-ban
- ▶ XML-ben tudjuk konfigurálni
- ▶ LayerDrawable



Létrehozása

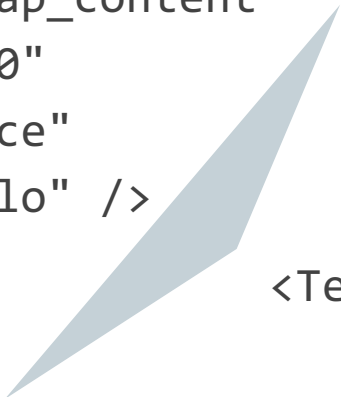
```
<layer-list>
  <item>
    <bitmap android:src="@drawable/android_red"
      android:gravity="center" />
  </item>
  <item android:top="10dp" android:left="10dp">
    <bitmap android:src="@drawable/android_green"
      android:gravity="center" />
  </item>
  <item android:top="20dp" android:left="20dp">
    <bitmap android:src="@drawable/android_blue"
      android:gravity="center" />
  </item>
</layer-list>
```



Stílusok

- ▶ Adott nézetet meghatározó attribútumok gyűjteménye
- ▶ A duplikációt csökkenti
- ▶ Kompaktabbá teszi a kódot
- ▶ Több komponens megjelenítését is kezelheti

```
TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textColor="#00FF00"  
    android:typeface="monospace"  
    android:text="@string/hello" />
```



```
<TextView  
    style="@style/CodeFont"  
    android:text="@string/hello"  
/>
```

■ styles.xml

styles.xml a res/values-ban van tárolva

```
<resources>
    <style name="CodeFont">
        <item name="android:textColor">#00FF00</item>
        <item name="android:typeface">monospace</item>
    </style>
</resources>
```

Hierarchia kezelése

```
<resources>
    <style name="CodeFont">
        <item name="android:layout_width">match_parent</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:textColor">#00FF00</item>
        <item name="android:typeface">monospace</item>
    </style>
</resources>

    <resources>
        <style name="RedCode" parent="@style/Codefont">
            <item name="android:textColor">#FF0000</item>
        </style>
    </resources>
```



Témák

- ▶ A téma egy egész aktivitásra vagy egész alkalmazásra alkalmazott stílus
- ▶ AndroidManifest.xml
- ▶ `<application android:theme="@style/AppTheme">`

```
<!-- Base application theme. -->
```

```
<style name="AppTheme"  
    parent="Theme.AppCompat.Light.DarkActionBar">
```

```
<!-- Try: Theme.AppCompat.Light.NoActionBar -->
```

```
<!-- Customize your theme here. -->
```

```
<item name="colorPrimary">@color/colorPrimary</item>
```

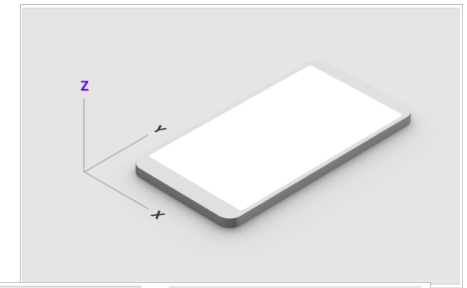
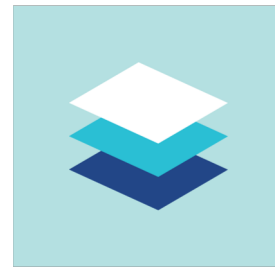
```
<item name="colorPrimaryDark">@color/colorPrimaryDark</item>
```

```
<item name="colorAccent">@color/colorAccent</item>
```

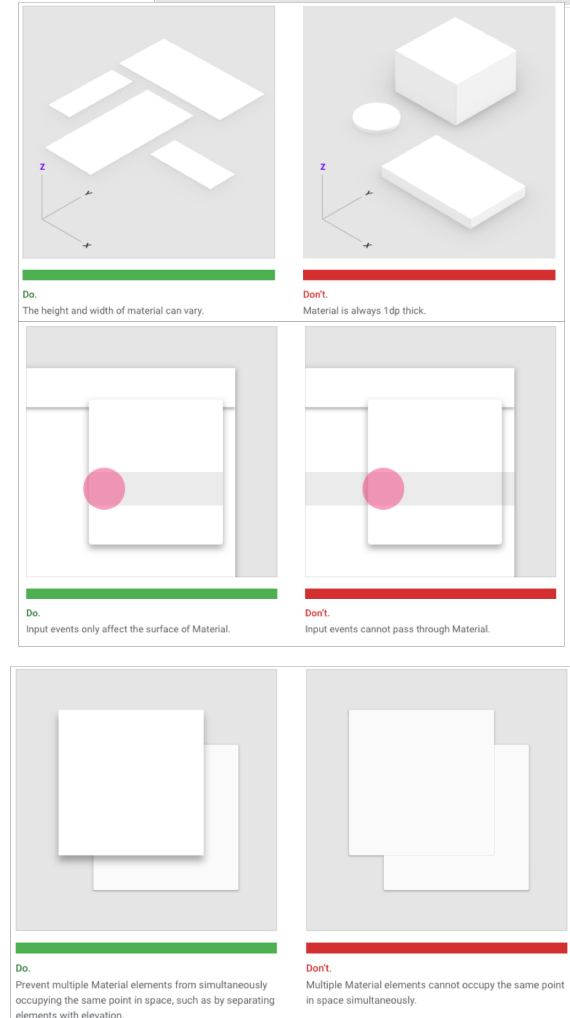
```
</style>
```



Material design



- ▶ Vizális ajánlások
- ▶ Vizuális nyelv
- ▶ A klasszikus jó tervezés és a technológia ötvözése
- ▶ Háromdimenziós környezet, könnyű, anyagi és árnyékos
- ▶ A felületek és a határok a valósághoz köthetőek
 - Árnyékuk van
 - Helyet foglalnak
 - Egymással interakcióban vannak

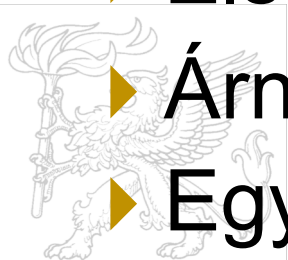


Képek használata

- ▶ Releváns
- ▶ Informatív
- ▶ Kellemes
- ▶ Szöveggel együtt
- ▶ Eredeti képek
- ▶ Fókuszra adjunk

Színek használata

- ▶ Elsődleges szín
- ▶ Árnyék
- ▶ Egy hangsúlyos szín
- ▶ styles.xml



Good choice

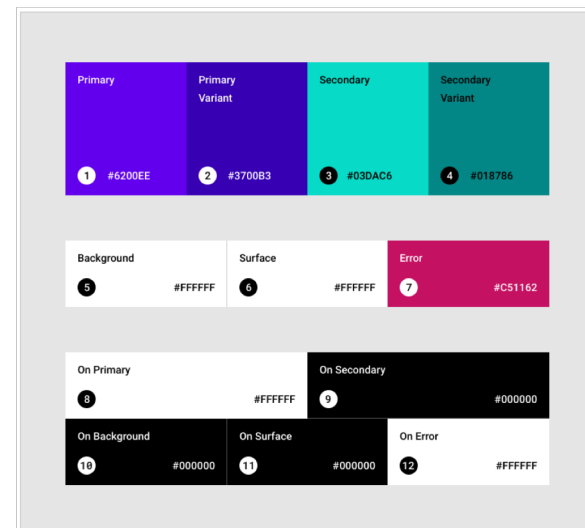
Good choice

Bad choice

Bad choice

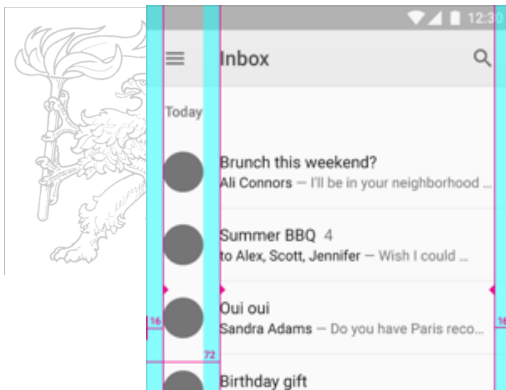
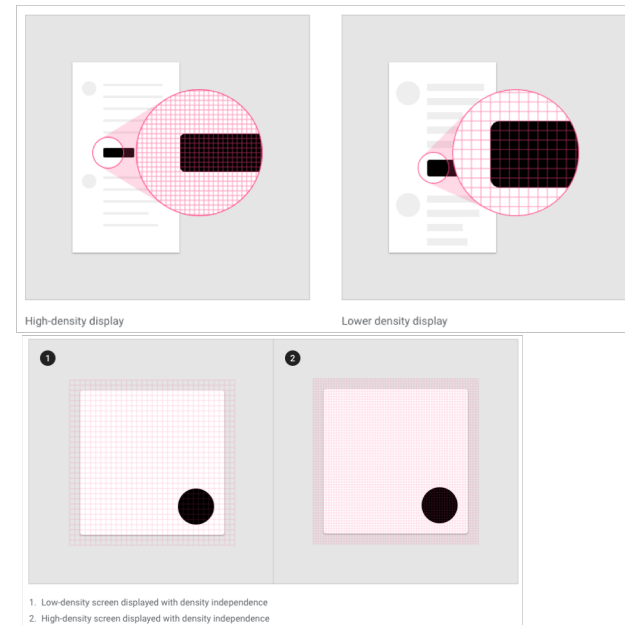
Bad choice

Good choice

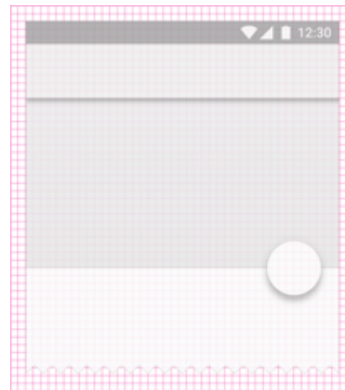


Elrendezések

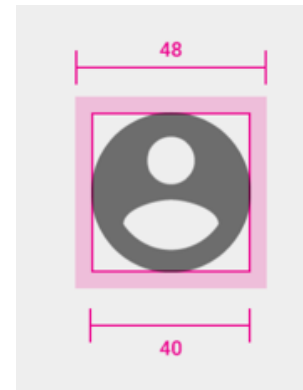
- ▶ Sűrűség független pixelek a nézetkhez – dp
- ▶ Skálázható pixelek a szövegekhez – sp
- ▶ Használjunk sablonokat, mintákat



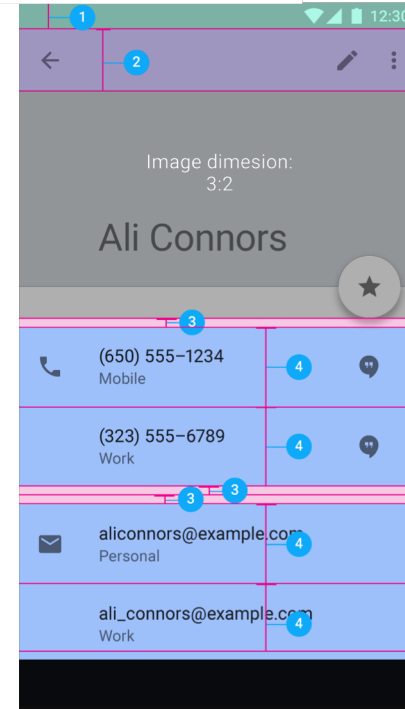
Térköz



Rács

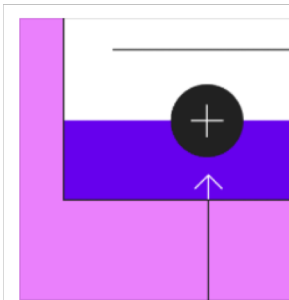
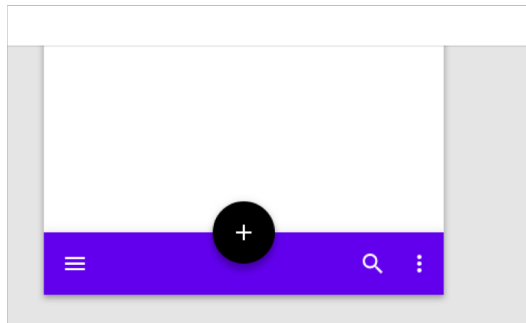


Méret



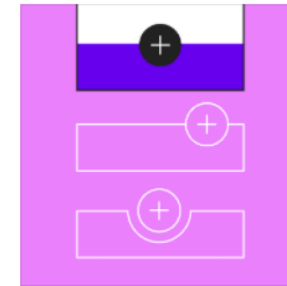
Komponensek

- ▶ Material.io
- ▶ Pl.: Alkalmazás sávok



Actionable

Bottom app bars highlight important screen actions and raise awareness of the floating



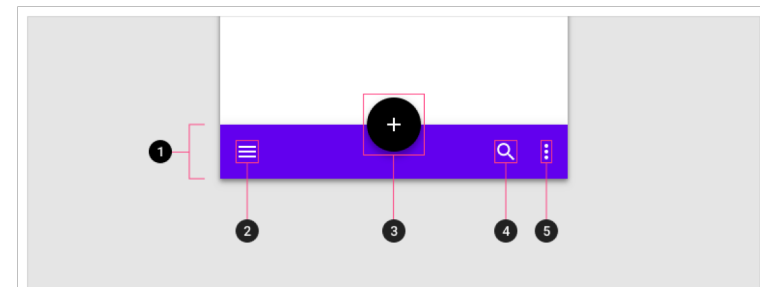
Flexible

A bottom app bar's layout and actions change based on the needs of the screen.

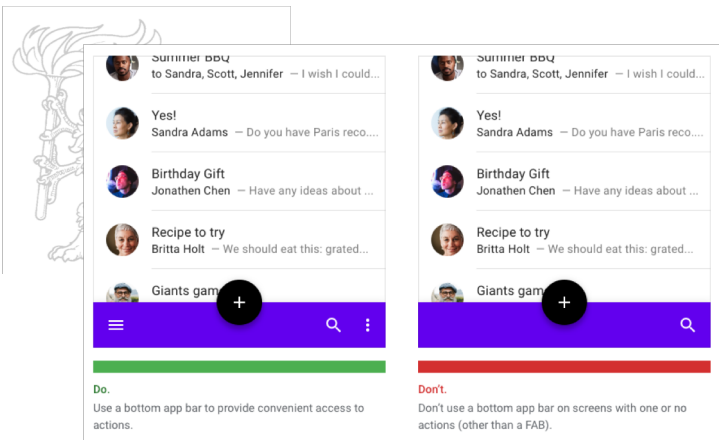


Ergonomic

The bottom app bar is easy to reach from a handheld position on a mobile device.

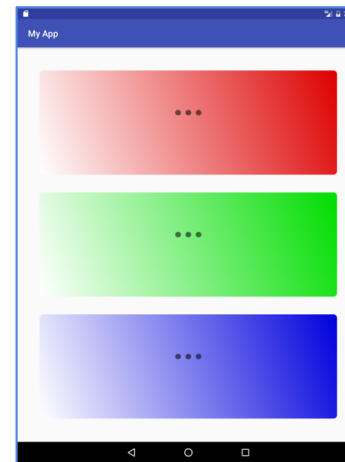
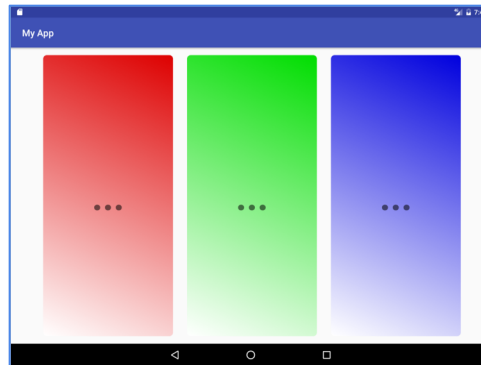


1. Container
2. Navigation drawer control
3. Floating action button (FAB)
4. Action icon
5. Overflow menu control



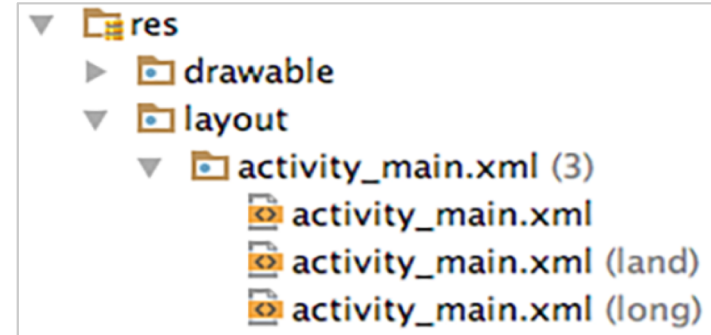
Adaptív elrendezés

- Az elrendezés a konfigurációhoz alkalmazkodik
 - Képernyő méret
 - Eszköz orientáció
 - Nyelv
 - Android verzió
- Alternatív erőforrások
 - Lokalizált karakterláncok
- Flexibilis elrendezés
 - GridLayout



Mappák

```
MyProject/
  src/
  res/
    drawable/
      graphic.png
    layout/
      activity_main.xml
      list_iteminfo.xml
    mipmap/
      ic_launcher_icon.png
    values/
      strings.xml
```



Erőforrás név konfiguráció név minősítő

drawable-hdpi	drawables a nagysűrűségű képernyőkhöz
layout-land	Fekvő orientáció elrendezés
layout-v7	Elrendezés 7-es verzióhoz
values-fr	Francia nyelvhez

A mai alkalom áttekintése

- ▶ RecyclerView
 - Komponensek
 - Megvalósítása
- ▶ Egyedi felhasználói élmény
 - Drawables
 - Stílusok
 - Témák
- ▶ Material design
 - Az alapelv
 - Képek
 - Színek
 - Elrendezések
 - Komponensek
- ▶ Adaptív elrendezés erőforrások
 - Adaptív erőforrások
 - Alternatív erőforrások



A következő előadás

- ▶ AsyncTask és AsyncTaskLoader
- ▶ Internet kapcsolat
- ▶ Üzenetszórás vevők

