



SZOFTVERFEJLESZTÉS ANDROID PLATFORMRA

DR. ISZÁLY GYÖRGY BARNA

ANIMÁCIÓK

- Háromféle animáció áll rendelkezésünkre
 - Tweened animáció
 - Frame by frame animáció
 - Property animation
- Az animációkat erőforrás állományokban érdemes megadni, melyeket a */res/anim* alkönyvtárba kell elhelyeznünk.

TWEENED ANIMATION

- Négy plusz egy animációs forma közül választhatunk:
 - *alpha* – az átlátszóság állítható vele
 - *scale* – a méretezés beállítására szolgál
 - *translate* – a mozgás megvalósítását végzi
 - *rotate* – a forgatás beállítása valósítható meg vele
 - *+1 – set* – csoportba foghatjuk az animációs elemeket, így összetett animációk készíthető
- Az animáció lefolyását befolyásolhatjuk.
 - `android:duration` – az animáció időtartama adható meg vele miliszekundumban
 - `android:fillAfter` – ha értéke igaz, akkor az animáció befejeződése után alkalmazza a módosításokat
 - `android:fillBefore` – ha értéke igaz, akkor az animáció elindítása előtt alkalmazza a módosításokat
 - `android:interpolator` – az animáció sebességét változtatja meg. Ezáltal egy animáció lehet gyorsuló, lassuló, ismétlődő stb.
 - `android:repeatCount` – az animáció ismétlődésének számát adja meg.
 - `android:startOffset` – az animáció indításának késleltetése adható meg vele miliszekundumban

PÉLDA ÖSSZETETT TWEENED ANIMÁCIÓRA

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
```

```
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:fromDegrees="0" android:toDegrees="360"
```

```
    android:toYScale="0.0" android:pivotX="40%"
```

```
    android:pivotY="30%,, android:duration="2000" />
```

```
<alpha android:fromAlpha="1.0"
```

```
    android:toAlpha="0.0"
```

```
    android:duration="4000"
```

```
/>
```

```
<scale android:fromXScale="1.0"
```

```
    android:toXScale="0.0,, android:fromYScale="1.0"
```

```
    android:toYScale="0.0,, android:pivotX="0"
```

```
    android:pivotY="0,, android:duration="2000"
```

```
/>
```

```
</set>
```

Összetett animáció

Elforgatás 360 fokban

Elhalványulás, vagy megjelenés

Skálázás

ANIMÁCIÓ HÍVÁSA PROGRAMBÓL

```
final Animation a = AnimationUtils.loadAnimation(this,  
R.anim.textanimation);
```

```
a.reset();
```

```
final TextView animText = (TextView) findViewById(R.id.Animszoveg);
```

```
animText.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View v) {
```

```
        animText.startAnimation(a);
```

```
    }
```

```
});
```

FRAME BY FRAME ANIMÁCIÓ

- Egy *animation-list* elemben fel kell sorolnunk az egymás után megjelenítendő képeket a hozzájuk tartozó időtartammal.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:oneshot="true">
```

```
  <item android:drawable="@drawable/kocka1,, android:duration="200"/>
```

```
  <item android:drawable="@drawable/kocka2,, android:duration="200"/>
```

```
  <item android:drawable="@drawable/kocka3,, android:duration="200"/>
```

```
  <item android:drawable="@drawable/kocka4,, android:duration="200"/>
```

```
  <item android:drawable="@drawable/kocka5,, android:duration="200"/>
```

```
  <item android:drawable="@drawable/kocka6,, android:duration="200"/>
```

```
</animation-list>
```

FRAME BY FRAME ANIMÁCIÓ INDÍTÁSA PROGRAMBÓL

```
ImageView kep=(ImageView)
this.findViewById(R.id.imageView_kocka);

kep.setImageResource(R.drawable.kocka_animation);

final AnimationDrawable kockaanim=(AnimationDrawable)
kep.getDrawable();

kep.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        kockaanim.start();
    }
});
```

PROPERTY ANIMÁCIÓ

- Android API 12-es verziójától
- Ezzel szinte minden elem animálható
- Ugyanazok az animációk hajthatók végre vele, mint a tweened animációkkal, de nem csak View elemeken

- Példa:

```
final TextView prop_anim_szoveg=(TextView)  
this.findViewById(R.id.textView_prop);  
prop_anim_szoveg.setAlpha(0);  
prop_anim_szoveg.animate().setDuration(4000);  
prop_anim_szoveg.animate().alpha(1);
```

Az előtűnés animáció paramétereinek beállítása és animálása

PROPERTY ANIMÁCIÓ PÉLDA

- Példa:

```
Integer colorFrom = getResources().getColor(R.color.feher);
```

```
Integer colorTo = getResources().getColor(R.color.zold);
```

```
final ValueAnimator colorAnimation =  
ValueAnimator.ofObject(new
```

```
ArgbEvaluator(), colorFrom, colorTo);
```

```
colorAnimation.setDuration(4000);
```

A ValueAnimator osztállyal egy intervallumon belüli animáció határozható meg. Jelen esetben a kiinduló érték a fehér színtől, jutunk el a zöld színig.

PROPERTY ANIMÁCIÓ PÉLDA

```
colorAnimation.addUpdateListener(new AnimatorUpdateListener(){
    @Override
        public void onAnimationUpdate(ValueAnimator arg0) {
            prop_anim_szoveg.setTextColor((Integer)arg0.getAnimatedValue());
        }
});

prop_anim_szoveg.setOnClickListener(new View.OnClickListener() {
    @Override
        public void onClick(View v) {
            // A színátmenet animáció indítása érintésre
            colorAnimation.start();
        }
});
```

Az `onAnimationUpdate()` metódust hívja meg az animáció minden egyes frame-je. A paraméterként kapott `ValueAnimator` tartalmazza az animációhoz kiszámított aktuális értéket.

MENÜK

- Legegyszerűbben erőforrások felhasználásával hozhatjuk létre a menüt
- A `/res/menu` alkönyvtárban hozzuk létre az XML erőforrás állományt
- `<menu>` elem a gyökérelem, és több `<menu>` elem is egymásba ágyazható - több szintű menü
- `<item>` menüpontok
- Példa az erőforrás állományra:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/menu1" android:title="@string/menupont1"/>
  <item android:id="@+id/menu2" android:title="@string/menupont2">
    <menu>
      <item android:id="@+id/almenu1" android:title="@string/almenu1"/>
      <item android:id="@+id/almenu2" android:title="@string/almenu2"/>
    </menu>
  </item>
  <item android:id="@+id/menu3" android:title="@string/menupont3"/>
</menu>
```

MENÜ BEÁLLÍTÁSA A PROGRAMBAN

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.menuk, menu);  
    return true;  
}
```

MENÜ KEZELÉSE PROGRAMBÓL

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch(item.getItemId()){  
        case R.id.menu1: Toast.makeText(MenukActivity.this, "Az első menüpontot választottad", Toast.LENGTH_LONG).show();  
            return true;  
        case R.id.almenu1: Toast.makeText(MenukActivity.this, "Az első almenüt választottad", Toast.LENGTH_LONG).show();  
            return true;  
        case R.id.almenu2: Toast.makeText(MenukActivity.this, "A második almenüt választottad", Toast.LENGTH_LONG).show();  
            return true;  
        case R.id.menu3: Toast.makeText(MenukActivity.this, "A harmadik menüpontot választottad", Toast.LENGTH_LONG).show();  
            return true;  
        default: return super.onOptionsItemSelected(item);  
    }  
}
```