Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Elektronsko in mobilno poslovanje

VAJE 4

Jan Meznarič
Simon Vrhovec

Ljubljana
16. 11. 2016

Prilagodljiv GUI

- Za naprave z različnimi velikostmi in gostotami zaslonov lahko pripravimo različne vmesnike.
- To je smiselno zlasti, kadar so razlike zelo velike (npr. tablica napram telefonu).
- Kadar so razlike v velikostih manjše, pa poskrbimo za prilagajanje velikosti elementov.

Prilagodljiv GUI

- Gradnik, na katerega tipično postavljamo ostale elemente, je *layout*. Gre za *t.i.* razporeditveni gradnik.

- Na razpolago imamo več tipov *layout-ov*
  - LinearLayout
  - RelativeLayout
  - TableLayout
  - GridLayout
  - (AbsoluteLayout)

LinearLayout

- Gradnik **LinearLayout** je eden izmed razredov razporeditvenih gradnikov, znotraj katerih opredelimo enega ali več podelementov (angl. child).
- Podelementi so razporejeni drug za drugim v en stolpec ali eno vrstico :
  - Navpičen seznam, en stolpec, en podelement na vrstico.
  - Vodoraven seznam, ena vrstica, višine najvišjega podelementa.
- Pogosteje uporabljeni atributi:
  - **android:orientation**: lahko zavzame vrednosti »**vertical**«, če želimo podelement porazdeliti v stolpec, ali »**horizontal**«, če jih želimo porazdeliti v vrstico.
- http://developer.android.com/guide/topics/ui/layout/linear.html

TableLayout

- **TableLayout** je razporeditveni gradnik, ki razporedi posamezne podelemente v tabelo.

- Vsak objekt razreda TableLayout je sestavljen iz enega ali več objektov razreda **TableRow**, pri čemer vsak izmed njih predstavlja vrstico tabele.

- Razred TableRow razširja razred LinearLayout.

- http://developer.android.com/reference/android/widget/TableLayout.html

RelativeLayout

- Bolj kompleksen, ampak pogosto uporabljen.
- Razporeditev določimo bodisi glede na posamezne podelemente bodisi glede na sam vsebnik.
- Včasih bolj učinkovito in pregledno kot večkratno gnezdenje LinearLayout .

RelativeLayout: Atributi pozicije

- **android:layout_alignParentTop**: če je »true«, se zgornji rob podelementa X ujema z zgornjim robom razporeditvenega gradnika R.
- **android:layout_centerVertical**: če je »true«, je vertikalna sredina podelementa X enaka vertikalni sredini razporeditvenega gradnika R.
- **android:layout_centerHorizontal**: če je »true«, je horizontalna sredina podelementa X enaka horizontalni sredini razporeditvenega gradnika R.
- **android:layout_below**: Zgornji rob podelementa X je pod podelementom Y.
- **android:layout_toRightOf**: Levi rob podelementa X je desno od podelementa Y.
- http://developer.android.com/guide/topics/ui/layout/relative.html

Vaja 6

- Kalkulator dopolnite tako, da boste imeli na zaslonu tipkovnico v obliki kvadrata (tipke od 1 do 9).
- S pritiski na tipke naj se polnijo polja EditText.
- Uporabite LinearLayout.
- Namig: Layoute lahko gnezdite.

Merske enote

- **px** – slikovne točke (angl. pixel): pike na ekranu.
- **mm** – milimetri: velikost milimetra.
- **in** – palci (angl. inch): velikost palca.
- **dp** (ali dip) – točke, neodvisne od gostote (angl. density-independent pixels):
  - Abstraktna enota, ki temelji na gostoti ekrana.
  - Na ekranu s 160 pikami na palec velja: 1dp = 1px.
  - To pomeni, da je pretvorba iz px v dp preprosta:

    $$px = dp * (dpi / 160)$$
- **sp** – točke, neodvisne od merila (angl. scale-independent pixels):
  - Podobno kot dp, vendar skalirano glede na uporabnikovo izbiro velikosti pisave.
  - Uporabljamo ga pri opredeljevanju velikosti pisave, ki je tako prilagojena gostoti ekrana in uporabniški izbiri.

Vaja 7

- Poskrbite, da bodo tipke vedno prikazane čez cel spodnji del zaslona ne glede na ločljivost naprave.
- Namig:
  - Za razporeditev tipk uporabite LinearLayout - uporabite skupen vertikalen Layout v katerem so gnezdeni trije horizontalni Layouti
  - Uporabite lastnost layout_weight

Prilagojen izgled

- Izgled vseh elementov na zaslonu je mogoče prilagajati.
- Rišemo lahko lastne vektorske oblike (angl. shape) ali uporabljamo v ta namen pripravljene rastrske slike.
- Posebna oblika rastrskih slik 9-patch nam omogoča prilagajanje njihovega prikaza (širjenje, oženje).
- Izdelamo lahko tudi enotno temo za celotno aplikacijo.

Enostaven primer prilagoditve izgleda tipke

- V posamezni tipki lahko definiramo lasten stil ozadja:
  - android:background="*@drawable/style_button_simple*"
- Stil:
  - v drawable dodamo datoteko *style_button_simple.xml*:

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid
        android:color="#AA222222" />
    <stroke
        android:width="0dp"
        android:color="#AA222222"/>
    <corners
        android:radius="0dp"/>
</shape>
```

- https://developer.android.com/guide/topics/resources/drawable-resource.html#Shape

Prilagoditev izgleda tipke z upoštevanjem stanj

- Izdelamo selector *(.xml)*, ki ga dodamo v imenik *drawable*.
- Za vsako stanje izdelamo svoj stil.
- Primer selectorja:

```
<?xml version="1.0" encoding="utf-8"?>
<selector
    xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:state_selected="true"
        android:drawable="@drawable/style_button_focus" >
    </item>
    <item
        android:state_pressed="true"
        android:drawable="@drawable/style_button_pressed" >
    </item>
    <item
        android:drawable="@drawable/style_button_normal" >
    </item>
</selector>
```

Vaja 8

- Trem izbranim tipkam na kalkulatorju prilagodite izgled.

- Namig:
  - Uporabite *selector*.
  - Selectorjeva stanja naj bodo določena v posameznih ločenih stilih (normalno stanje, fokus, pritisnjeno stanje).
  - Selector uporabite kot *background* za izbrane tipke.