



A n d r o i d

Основни UI елементи

Съдържание

- XML
- Basic Views
- Lollipop views
- Стартиране на проект
 - Емулатор
 - Устройство
- Demo

xml

ОСНОВИ

- XML е дескриптивен език за предаване на информация, но в Андроид се използва за описване на потребителския интерфейс
- В Андроид, всеки XML елемент има съответстващ Java клас със същото име
- UI за всеки екран може да има само един root елемент.
- Задължително в него се декларира и редът
 - `xmlns:android="http://schemas.android.com/apk/res/android"`
- Задължително за всеки елемент е да има свойствата
 - `android:layout_width="match_parent"`

Ширина и височина на елемент

- `android:layout_width="match_parent"`
- `android:layout_height="match_parent"`
- Дефинират размера на елемента
- Могат да имат три стойности
 - `match_parent` - приема размера на родителския елемент
 - `wrap_content` - заема минималното му необходимо място
 - `48dp` - може и да се задават конкретни стойности
 - числата е препоръчително да са степени на 2ката
 - `dp` е независим пиксел, имащ предвид съотношението между размера на екрана и наситетеността му

view

View

- Базовия клас за всички елементи.
- Има свойствата:
 - `margin`, `marginLeft` (`Right`, `Top`, `Bottom`)
 - `padding`, `paddingLeft` (`Right`, `Top`, `Bottom`)
 - `background` (приема цвят или картинка)
 - `clickable` (`true/false`)
 - `id` (задава се от потребителя)
 - когато се задава ново `id`, трябва да се използва знака `+`

layouts

Layouts

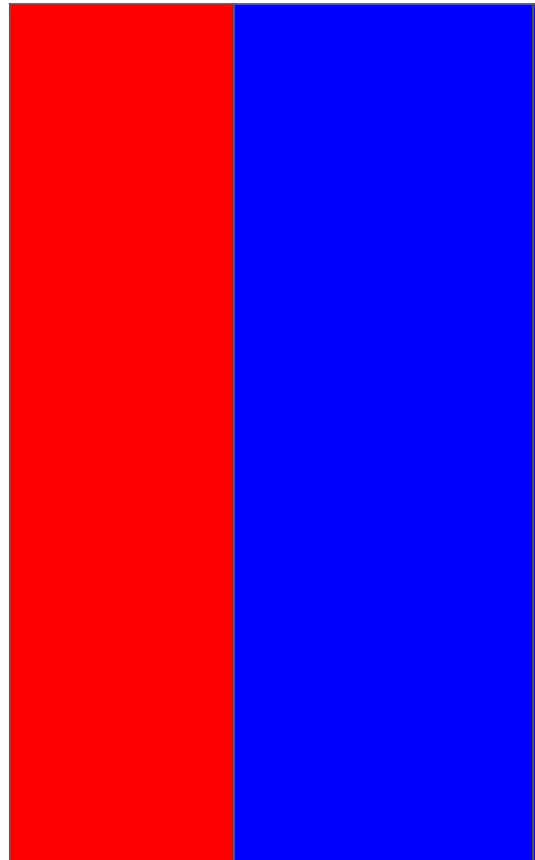
- Този тип елементи нямат визуална репрезентация на екрана
- Те служат за контейнери
- Указват на елементите в тях, как да бъдат подредени
- Не се препоръчва използването на повече от 5 вложени layout-а. Повече от това би създавало лаг, заради проблеми със сложността на пресмятане при изрисоване (на модерни телефони не се препоръчват повече от 10)
- Ако достигнете вложеност 5, помислете дали да не смените вида на layout-ите

LinearLayout

- Елементите в този лейаут се подреждат един след друг, вертикално или хоризонтално
- Задължително свойство е `android:orientation="vertical/horizontal"`
- `android:gravity` - указва на децата на лейаута как да се подредят - центрирано, вляво, вдясно
- децата на `LinearLayout` могат да имат свойството `weight`. То разпределя цялото пространство на родителя пропорционално между всички елементи с `weight`, в зависимост от стойността

android:weight

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <View
        android:background="#f00"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        weight="2" />
    <View
        android:background="#0f0"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        weight="3" />
</LinearLayout>
```



- Направете екран подобен на този

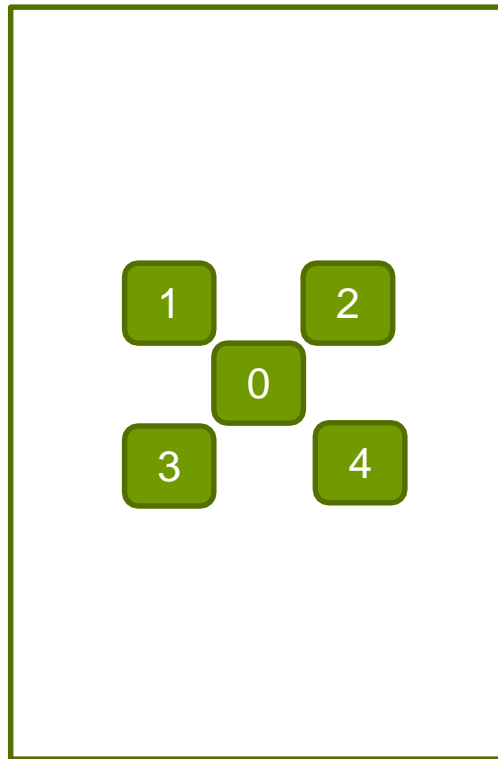
A login form with a green border. It contains three green rounded rectangular elements: a top field labeled "Име" (Name), a middle field labeled "Парола" (Password), and a bottom row with two buttons labeled "Влез" (Login) and "Край" (Logout).

Име	
Парола	
Влез	Край

RelativeLayout

- Елементите в този лейаут се подреждат един спрямо друг и всеки елемент трябва сам да укаже къде иска да бъде позициониран
- Елементите в RelativeLayout могат да имат свойствата:
 - `android:layout_alignParentBottom="true"` (Top, Left, Right)
 - `android:layout_toRightOf="@id/time"` (`toLeftOf`)
 - `android:layout_below` / `android:layout_above`
 - `android:layout_alignLeft` (Right, Top, Bottom)
 - `android:layout_centerHorizontal="true"`
 - `android:layout_centerVertical="true"`

- Бутон 0 е точно в средата,
останалите са точно в ъглите му



TableLayout -> GridLayout

- Поддържа децата си в таблица

```
<GridLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:rowCount="2"
```

```
    android:columnCount="2">
```

```
        <View android:layout_row="0" android:layout_column="0" />
```

```
        <View android:layout_row="0" android:layout_column="1" />
```

```
        <View android:layout_row="1" android:layout_column="0" />
```

```
        <View android:layout_row="1" android:layout_column="1" />
```

```
</GridLayout>
```

****Спестени са width & height на децата за краткост**

FrameLayout

- Създаден за да държи в себе си единствено дете
- Основно се използва за да бъдат добавяни и премахвани от него елементи по време на изпълнение на приложението

ScrollView

- Позволява на съдържанието си да бъде скролирано.
- Трябва да има само един дъщерен елемент. Най-често се използва `LinearLayout`.

```
<ScrollView>  
    <LinearLayout>  
        <!-- Here is the scrollable content -->  
    </LinearLayout>  
</ScrollView>
```

user input views

TextView

- Използва се за да показва текст
- Текстът не може да бъде променян от потребителя
- СВОЙСТВА

```
<TextView
    android:layout_width="match_parent"
    android:text="Neshto"
    android:textSize="35sp"
    android:textStyle="bold"
    android:textAlignment="center"
    android:textAllCaps="true"
    android:padding="18dp"
    android:textColor="#fff"
    android:layout_height="wrap_content"
    android:background="#AA3456" />
```

EditText

- За въвеждане на текст от потребителя
- Наследява TextView
- Свойства
 - android:hint - показва нещо, когато полето е празно
 - android:inputType - какъв вид текст може да бъде показван тук, определя какъв тип клавиатура ще се покаже

Button

- Бутон. Наследява TextView
- Свойства:
 - [android:onClick](#) - вместо да се задава програмно, може чрез това свойство да се укаже, кой метод да бъде изпълнен при натискане на бутона

ImageView

- Елемент, служещ за визуализация на картинка
- Свойства:
 - `src` - указва, коя е картинката
 - `scaleType` - указва, как да бъде скалирана картинката

CheckBox & RadioButton

- За да бъдат радио бутоните свързани, те трябва да са в обща `RadioGroup`

```
<RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/radio_male"
        android:checked="true" />

    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/radio_female" />

</RadioGroup>
```

ToggleButton -> Switch

```
<Switch  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Play with the Switch"  
>
```



Spinner

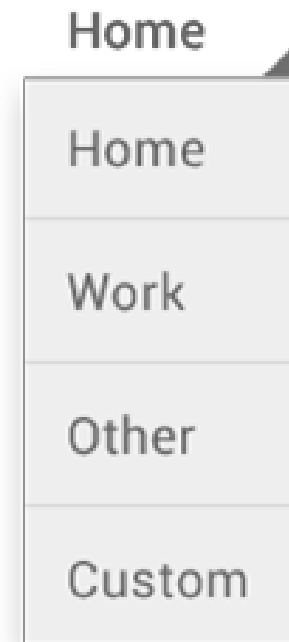
- Падащо меню
- Стойностите в падащото меню се задават програмно, не през xml-a

```
<Spinner
```

```
    android:id="@+id/planets_spinner"
```

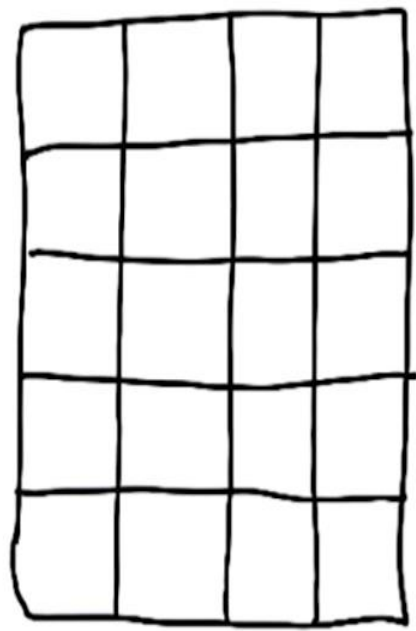
```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content" />
```

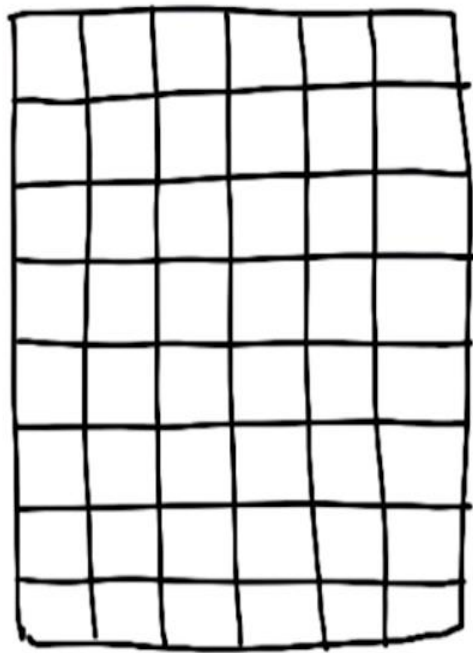


Size: dp

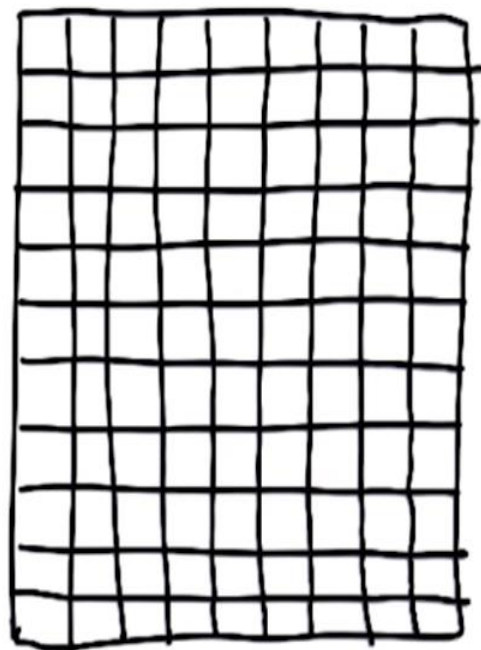
DENSITY - INDEPENDENT PIXELS



Medium Resolution Device



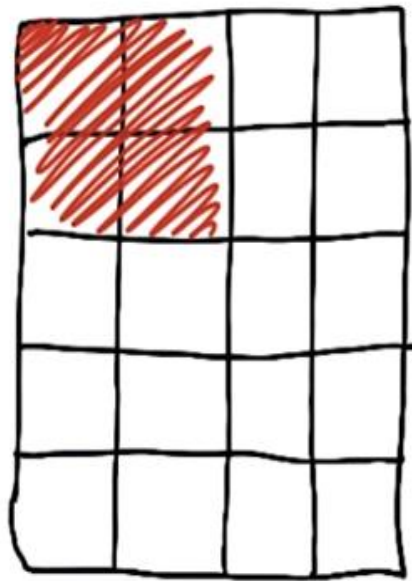
High Resolution Device



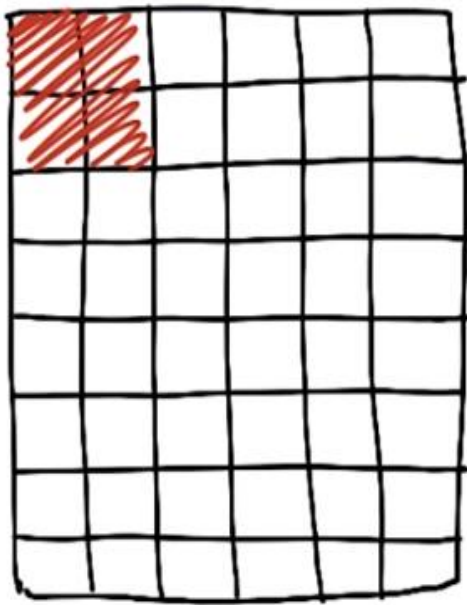
Extra-High Resolution Device

DENSITY - INDEPENDENT PIXELS

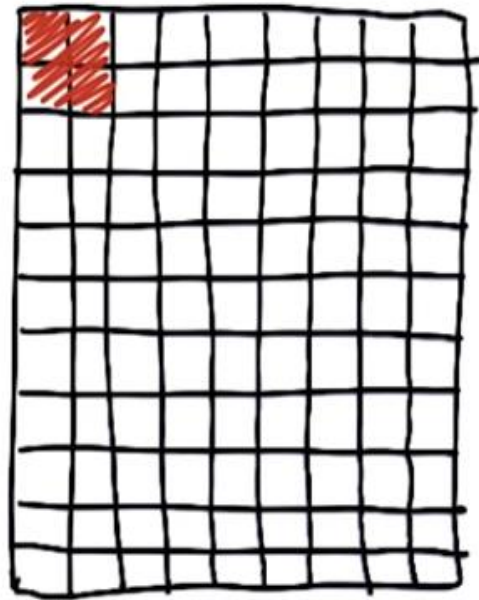
2 pixels by 2 pixels



Medium Resolution Device



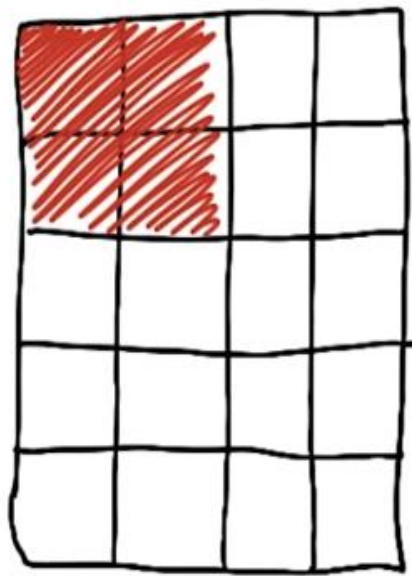
High Resolution Device



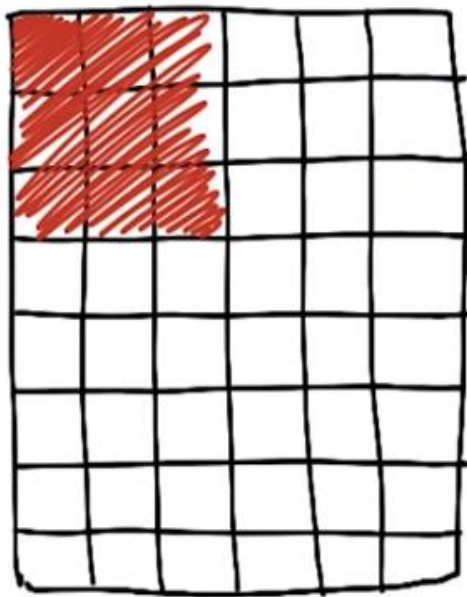
Extra-High Resolution Device

DENSITY - INDEPENDENT PIXELS

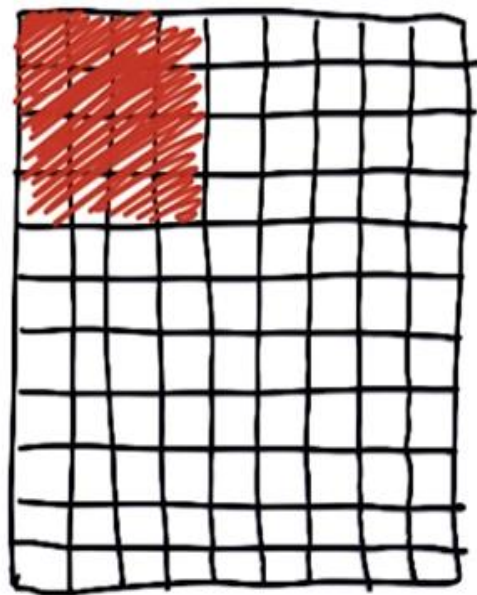
2 dp by 2 dp



Medium Resolution Device



High Resolution Device



Extra-High Resolution Device

lollipop views

ToolBar

- До Андроид KitKat ActionBar-а се задава програмно и е труден за кustomизиране
- Андроид Лолипоп въвежда елемента ToolBar който става част от xml йерархията
- Свойства:
 - [android:navigationIcon](#)

```
<android.support.v7.widget.Toolbar  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:background="@color/ColorPrimary"  
    android:elevation="4dp"    />
```

CardView

Обикновен контейнер, който се стилизира като карта. Поддържа свойството elevation, което показва сянка.

```
<android.support.v7.widget.CardView
    android:layout_gravity="center"
    android:layout_width="200dp"
    android:layout_height="200dp"
    card_view:cardCornerRadius="4dp">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</android.support.v7.widget.CardView>
```


как се стартира проект

С емулятор

- Стартирайте AVD Manager /иконка с телефонче/
- Създайте ново виртуално устройство
- Изберете му име, операционна система, Image /изберете x86 при възможност/, RAM
- Стартирайте проекта от зелената стрелка
- Изберете устройството си в падащия списък на диалоговия прозорец

С устройство

- Инсталирайте подходящите драйвери за устройството ви
- Закачете го
- В настройките на устройството включете дебъг режима
 - Идете в настройки -> За телефона /About/
 - Кликнете 14 пъти върху модела на телефона
 - Идете в новопоявилата се опция Дебъг, включете я
- Стартирайте приложението
- На устройството потвърдете, че се доверявате на този компютър