2. Building a Simple User Interface

Revised by Minhaeng Lee
Hierarchy of view and viewgroup

• View
  • UI object
  • E.g. Button or textfield

• ViewGroup
  • Invisible
  • Defines how the child view are laid out
  • E.g. LinearLayout, RelativeLayout

• XML is used for their hierarchy
Is this possible to make View without ViewGroup?  **No**
Create a Linear Layout

• 1. Open res/layout/activity_main.xml (might be different name)
Tip for Emulator rotation

Windows: left Ctrl + F12

Mac: Fn + Ctrl + F12
Create a Linear Layout

• 2. Delete `<TextView>` element
• 3. Change the `<RelativeLayout>` element to `<LinearLayout>`
• 4. Add the `android:orientation` attributes and set it to “horizontal”
• 5. Remove the `android:padding` attributes and the `tool:context`
Create a Linear Layout

- **LinearLayout**
  - A view group lays out in either a vertical or horizontal
  - Used with orientation attribute
  - XML order = Screen View Order
    - Top element is rendered in top view
Add a Text Field

• 1. In the content_my.xml file, within the <LinearLayout> element, define an <EditText> element with the id attribute set to @+id/edit_message.
• 2. Define the layout_width and layout_height attributes as wrap_content.
• 3. Define a hint attribute as a string object named edit_message.
Attributes Explanation

• **android:id**
  • @ is required when we are referring to any resource object from XML
  • Followed by resource type (id in here), /, then resource name
  • + is needed when you’re defining a resource ID for the first time.
  • Once it is compiled, the ID will be added in `gen/R.java` file

• **android:layout_width, android:layout_height**
  • wrap_content : as big as needed to fit the contents
  • Match_parent : as big as its parent

• **android:hint**
  • Default string to display

Files under the build folder are generated and should not be edited.

```java
public static final int decor_content_parent=0x7f00c03f;
public static final int default_activity_button=0x7f00c027;
public static final int disableHome=0x7f00000c;
public static final int edit_message=0x7f00c0050;
public static final int edit_query=0x7f00c0043;
public static final int end=0x7f00c0019;
```
Add String Resources

• Need to add string to `res/value/strings.xml`

• 1. In Android Studio, from the `res/values` directory, open strings.xml.
• 2. Add a line for a string named "edit_message" with the value, "Enter a message".
• 3. Add a line for a string named "button_send" with the value, "Send".

```xml
<resources>
    <string name="app_name">My Application</string>
    <string name="edit_message">Please edit me!</string>
    <string name="button_send">Send</string>
</resources>
```
Add a button

1. In Android Studio, from the res/layout directory, edit the activity_main.xml file.

2. Within the <LinearLayout> element, define a <Button> element immediately following the <EditText> element.

3. Set the button's width and height attributes to "wrap_content" so the button is only as big as necessary to fit the button's text label.

4. Define the button's text label with the android:text attribute; set its value to the button_send string resource you defined in the previous section.
Make the Input Box Fill in the Screen Width

1. In the content_my.xml file, assign the `<EditText>` element's `layout_weight` attribute a value of 1.

2. Also, assign `<EditText>` element's `layout_width` attribute a value of 0dp.
`<EditText android:id="@+id/edit_message"
    android:layout_weight="0.5"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="@string/edit_message" />
<Button
    android:layout_weight="0.5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_send" />
</LinearLayout>

<Ellipse android:id="@+id/edit_message"
    android:layout_weight="0.7"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="@string/edit_message" />
<Button
    android:layout_weight="0.3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_send" />
</LinearLayout>`
References

• Android Official Developer Site