Conversion App and Fundamentals of Android and Android Studio

Summarization and Lecture Information of Ch. 1-8
Android Studio 4.0 Development Essentials (Kotlin Edition)

John P. Baugh, Ph.D.
Winter 2022

Create the Project

- com.cis2818.conversion
- Oreo 8 for Android level
- Kotlin for language

Configure the Layout and App Features

1. Click the magnet to enable Autoconnections
    - If this ever fails, use the magic wand to **infer constraints**
2. Drag a **Button** onto screen underneath the text field
    - Click Infer Constraints to ensure it is constrained correctly
3. Click button and change attribute
    - **text** under Common Attributes
    - Change the text to **Convert**
4. Notice the warning in the upper right corner
    - It's because we've hardcoded the button text
    - It's better to create string resources and use those instead of hardcoding text
    - This promotes **separation of concerns**, a software engineering principle
    - It makes it easier for **internationalization** (**i18n**) – for translation of the string resources, without having to modify code
5. Fixing the string / warning issue
    - You could click on the warning and then expand it
    - There is a **Suggested Fix** at the bottom (Extract string resource)
        - Call it **convert_string**
    - **Mention that:**
        - To do this manually, you can click the button next to the Text and open the Resources, and add a new Resource

        - Then you refer to it in the Common Attributes in the Attributes panel
6. Drag a **NumberDecimal component** EditText onto the screen
    - Set the **hint** property to "dollars"
    - Click the warning and extract the string to a resource named **dollars_hint**
7. Change the **id** of the edit text to **dollarText**
8. Add missing layout constraints (**infer constraints button – magic wand icon)**
9. Examine the **activity_main.xml**
    - Sometimes it's easier to just type out or copy/paste XML data
    - XML for the design also promotes separation of concerns
    - Talk about the ConstraintLayout as the layout and parent to the TextView, Button, and EditText objects
    - Discuss attributes as well
10. To see how XML changes cause layout/design changes in real time, let's change the background color of the ConstraintLayout to a shade of red
    - In the constraint layout opening tag **add: android:background="#ff2438"**
11. Delete the background property to return the color to normal
12. Open the **res/values/strings.xml** and look at the string resources
13. Add the following:
    - **<string name="no_value_string">No Value</string>**

Adding Interaction

*3.8 in the Android Development Essentials Book 4.0

1. Add event handling to the Button
2. Under the **Button's** Common Attributes, set the **onClick** attribute to **convertCurrency**
    - The warning/red border indicates the method doesn't exist yet
3. Double-click **MainActivity.kt** to load it into the code editor
4. Add code for convertCurrency

```
fun convertCurrency(view : View) {
    if(dollarText.text.isNotEmpty()) {
        val dollarValue = dollarText.text.toString().toFloat()
        val euroValue = dollarValue * 0.85f
        textView.text = euroValue.toString()
    }
    else {
        textView.text = getString(R.string.no_value_string)
    }//end if-else
}//end convertCurrency
```

5. Run app to test it

6. Note other chapters to read through:
    - Note to read chapters 4-5 for Emulator/AVD information
        i. Includes different symbols and usage
        ii. Includes how to simulate Location information for GPS information, etc.
        iii. Also, includes how to use **Dark Theme** in the emulator as the default theme
    - Note to read Ch. 6 for Android Studio Windows and Widgets
        i. Also includes keyboard shortcuts
    - Note to read Ch. 7 for how to test on a Physical Android device
    - Note to read Ch. 8 for how to use the Code Editor within Android Studio optimally