



# Lecture 03: User Interfaces & Material Design

## IN721: Design and Development of Applications for Mobile Devices

### Semester One, 2020

Kaiako: Grayson Orr

Te Kura Matatini ki Otago, Ōtepoti, Aotearoa

Wednesday, 26 February

## LECTURE 02: ACTIVITIES & INTENTS TOPICS

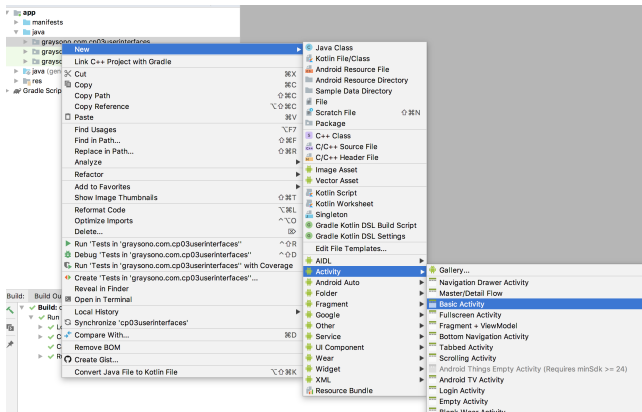
- ▶ Activities
- ▶ Intents
  - ▶ Data passing
  - ▶ Implicit
  - ▶ Explicit
- ▶ OnBackPressed
- ▶ String placeholder

# LECTURE 03: USER INTERFACES & MATERIAL DESIGN TOPICS

- ▶ Basic activity
- ▶ BottomNavigationView
- ▶ Menus
- ▶ Splash screen
- ▶ Material design

# BASIC ACTIVITY

- ▶ Create a new basic activity
- ▶ A menu resource directory & menu XML file will be generated
- ▶ A activity & content XML file will be generated
  - ▶ Activity XML file references the content XML file
  - ▶ Add widgets to the content XML file



# MainActivity.KT

- ▶ onCreateOptionsMenu
- ▶ onOptionsItemSelected

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        setSupportActionBar(toolbar)  
        val fab = findViewById<FloatingActionButton>(R.id.fab)  
        fab.setOnClickListener(YouTubeButtonOnClickListener())  
    }  
  
    override fun onCreateOptionsMenu(menu: Menu): Boolean {  
        menuInflater.inflate(R.menu.menu_main, menu)  
        return true  
    }  
  
    override fun onOptionsItemSelected(item: MenuItem): Boolean {  
        return when (item.itemId) {  
            R.id.action_settings -> true  
            else -> super.onOptionsItemSelected(item)  
        }  
    }  
  
    inner class YouTubeButtonOnClickListener : View.OnClickListener {  
        override fun onClick(view: View) {  
            startActivity(Intent(Intent.ACTION_VIEW,  
                Uri.parse("https://www.youtube.com")))  
        }  
    }  
}
```

# MENUS: TOOLBAR

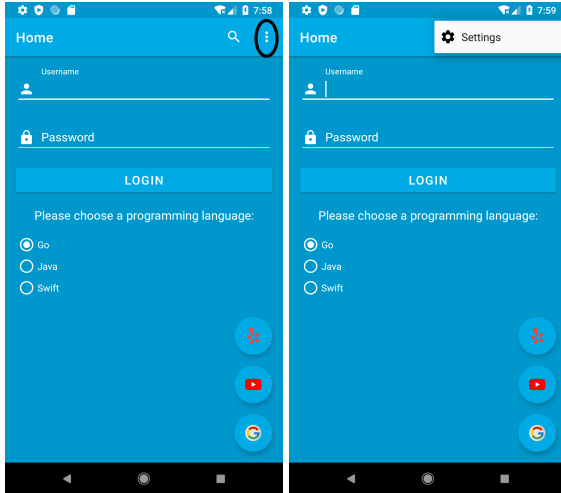
- ▶ menu\_main.xml
- ▶ <menu> tag
- ▶ <item> tag



```
1 <menu xmlns:android="http://schemas.android.com/apk/res/android"
2       xmlns:app="http://schemas.android.com/apk/res-auto"
3       xmlns:tools="http://schemas.android.com/tools"
4       tools:context="com.example.graysono.test.MainActivity">
5   <item
6       android:id="@+id/action_settings"
7       android:orderInCategory="100"
8       android:title="Settings"
9       app:showAsAction="never" />
10 </menu>
```

# MENUS: TOOLBAR

- ▶ Vertical ellipsis
- ▶ Click to show menu items



# BOTTOMNAVIGATIONACTIVITY.KT

- ▶ Create a new empty activity
- ▶ You can create a bottom navigation activity, but generates a lot of boilerplate. We won't do this

```
class BottomNavigationActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_bottom_navigation)  
  
        val bnv = findViewById<BottomNavigationView>(R.id.bnv)  
        bnv.setOnNavigationItemSelectedListener(OnNavigationItemSelectedListener())  
        bnv.menu.getItem(index: 1).isChecked = true  
    }  
  
    inner class OnNavigationItemSelectedListener : BottomNavigationView.OnNavigationItemSelectedListener {  
        override fun onNavigationItemSelected(item: MenuItem): Boolean {  
            return when (item.itemId) {  
                R.id.navigation_home -> {  
                    startActivity(Intent(packageContext: this@BottomNavigationActivity,  
                        MainActivity::class.java))  
                    true  
                }  
                R.id.navigation_dashboard -> true  
                else -> onNavigationItemSelected(item)  
            }  
        }  
    }  
}
```



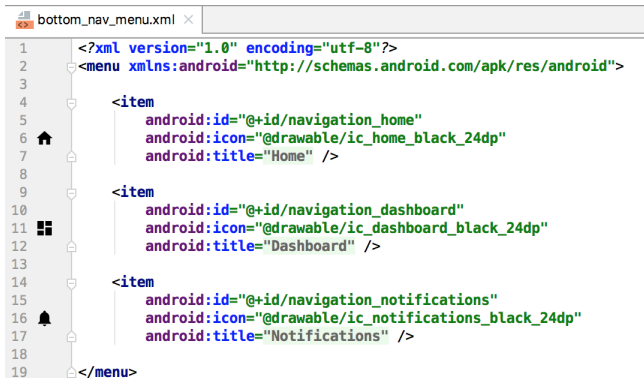
# ACTIVITY BOTTOM NAVIGATION XML

## ► BottomNavigationView widget

```
activity_bottom_navigation.xml ×
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     android:id="@+id/container"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:paddingTop="?attr/actionBarSize">
9
10     <com.google.android.material.bottomnavigation.BottomNavigationView
11         android:id="@+id/bnv"
12         android:layout_width="0dp"
13         android:layout_height="wrap_content"
14         android:layout_marginStart="0dp"
15         android:layout_marginEnd="0dp"
16         android:background="?attr/windowBackground"
17         app:layout_constraintBottom_toBottomOf="parent"
18         app:layout_constraintLeft_toLeftOf="parent"
19         app:layout_constraintRight_toRightOf="parent"
20         app:menu="@menu/bottom_nav_menu" />
21
22 </androidx.constraintlayout.widget.ConstraintLayout>
```

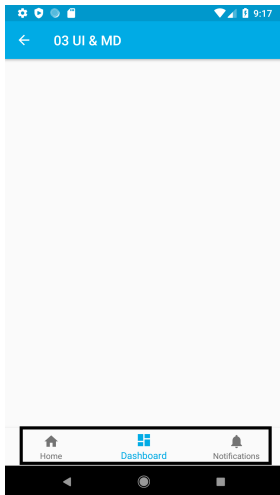
# MENUS: BOTTOM NAVIGATION

- ▶ bottom\_nav\_menu.xml
- ▶ <menu> tag
- ▶ <item> tag



# MENUS: BOTTOM NAVIGATION

- ▶ Three menu items as specified in the bottom\_nav\_menu.xml file



# SPLASH SCREEN

- ▶ Currently, the launcher activity is MainActivity.kt
- ▶ Create a new activity called SplashScreenActivity.kt
- ▶ Change the launcher activity to SplashScreenActivity.kt in the AndroidManifest.xml

```
<activity
    android:name=".SplashActivity"
    android:theme="@style/AppTheme.NoActionBar">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

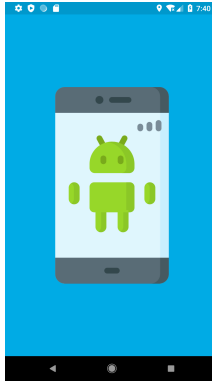
# SPLASH SCREEN

- ▶ `postDelayed`
  - ▶ Causes the runnable to be added to the message queue
  - ▶ Run after a specified amount of time elapses

```
class SplashActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_splash)  
        Handler().postDelayed({  
            startActivity(Intent( packageContext: this@SplashActivity, MainActivity::class.java))  
            finish()  
        }, resources.getInteger(R.integer.sleep).toLong())  
    }  
}
```

# SPLASH SCREEN

- ▶ Splash screen image - image view widget
- ▶ Change from SplashScreenActivity.kt to MainActivity.kt after 3000 milliseconds



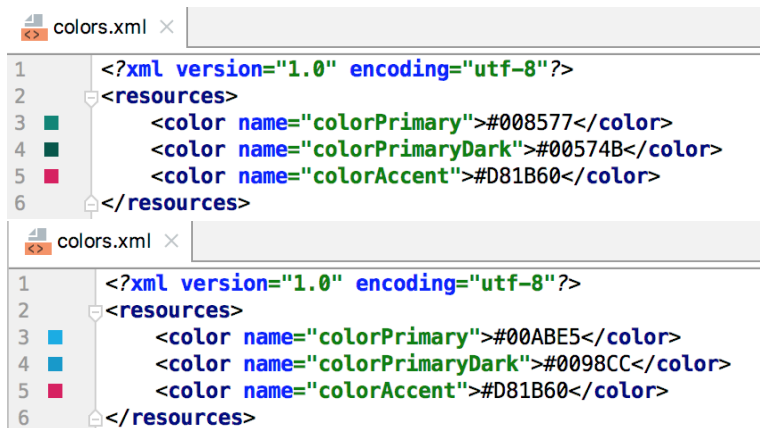
# INTEGERS RESOURCE XML FILE

- ▶ Integers resource file
  - ▶ `resources.getInteger(R.integer.sleep).toLong()`

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <integer name="sleep">3000</integer>  
</resources>
```

# CHANGING COLORS

- ▶ Default color hex values
- ▶ Changing the color hex values



The image displays two screenshots of an XML editor, likely Android Studio, showing the contents of a file named `colors.xml`. The editor interface includes a tab at the top with a file icon, the filename `colors.xml`, and a close button. The code is displayed in a light gray background with line numbers on the left and color-coded syntax.

**Top Screenshot (Default Values):**

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <color name="colorPrimary">#008577</color>
4      <color name="colorPrimaryDark">#00574B</color>
5      <color name="colorAccent">#D81B60</color>
6  </resources>
```

**Bottom Screenshot (Modified Values):**

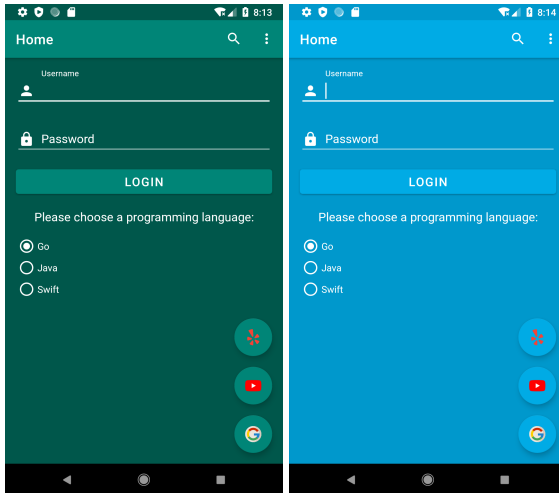
```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <color name="colorPrimary">#00ABE5</color>
4      <color name="colorPrimaryDark">#0098CC</color>
5      <color name="colorAccent">#D81B60</color>
6  </resources>
```

The changes in the bottom screenshot are highlighted with colored squares on the left margin: a light blue square for line 3, a medium blue square for line 4, and a red square for line 5.



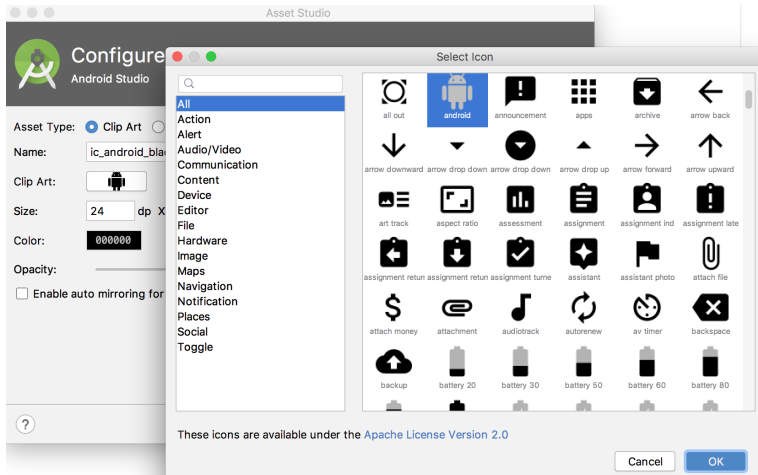
# CHANGING COLORS

- ▶ Before & after
- ▶ Adhere to Material Design best practices



# VECTOR DRAWABLES

► res > drawable > New > Vector Asset

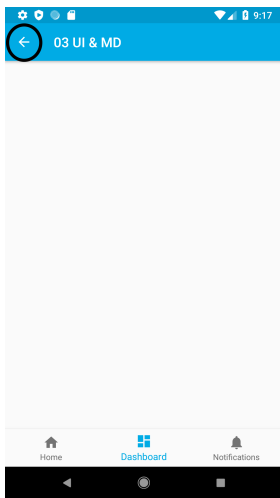


# TOOLBAR BACK BUTTON: ANDROIDMANIFEST.XML

- ▶ parentActivityName attribute

```
<activity
    android:name=".BottomNavigationActivity"
    android:label="@string/app_name"
    android:parentActivityName=".MainActivity">
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value=".MainActivity" />
</activity>
```

# TOOLBAR BACK BUTTON: ANDROIDMANIFEST.XML



# MATERIAL DESIGN

- ▶ Design system - back by open-source
- ▶ Build high-quality digital experiences for Android, iOS and the web
- ▶ <https://material.io/>
- ▶ Material Design Components (MDC) tutorials
- ▶ build.gradle
  - ▶ implementation 'com.google.android.material:material:1.0.0'

# PRACTICAL

- ▶ Series of tasks covering today's lecture
- ▶ Worth 1% of your final mark for the Design and Development of Applications for Mobile Devices course
- ▶ Deadline: Friday, 12 June at 5pm