



## Lecture 09: Alert Dialog Builder & User Interfaces 2

IN721: Design and Development of Applications for Mobile Devices  
Semester One, 2020

Kaiako: Grayson Orr

Te Kura Matatini ki Otago, Ōtepoti, Aotearoa

Friday, 13 March

## LECTURE 08: PROGRESS DIALOG & WEB VIEW TOPICS

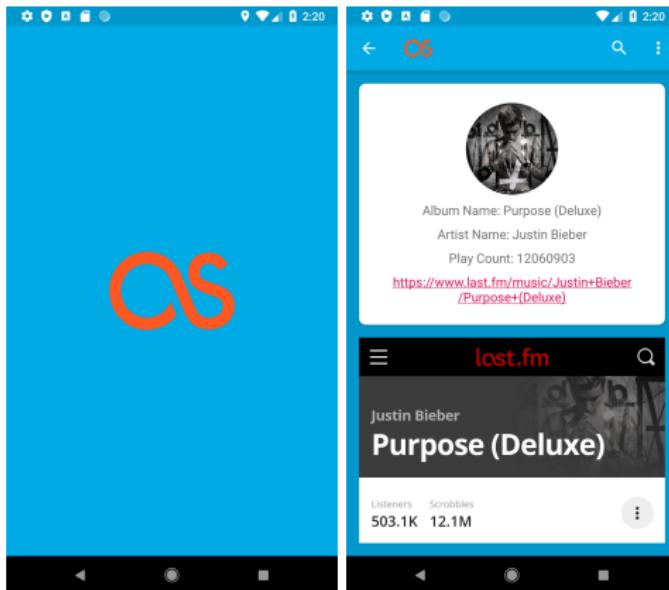
- ▶ Custom style
- ▶ Custom progress dialog/bar
- ▶ Web view

# LECTURE 09: ALERT DIALOG BUILDER & USER INTERFACES

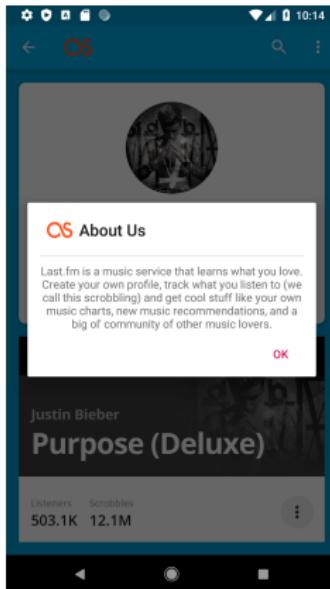
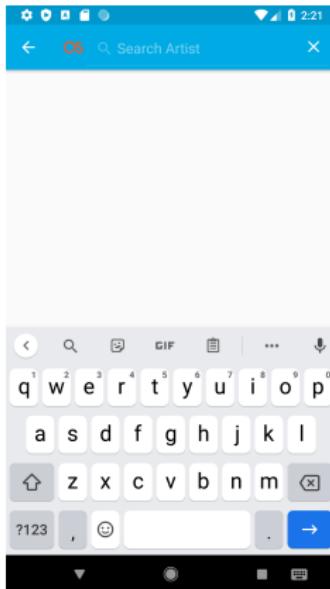
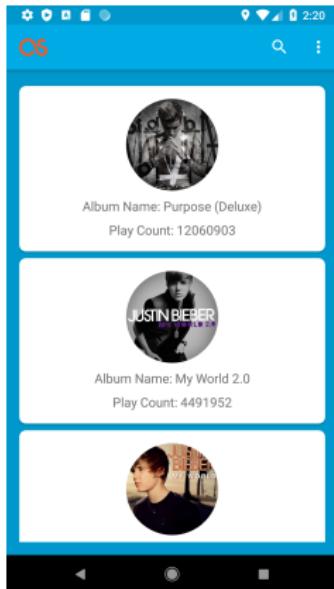
## 2 TOPICS

- ▶ User interfaces
  - ▶ Circle image view
  - ▶ Nested menu items
  - ▶ Toolbar icon
- ▶ Text view web links
- ▶ Alert dialog builder

# APPLICATION SCREENSHOTS



# APPLICATION SCREENSHOTS



# CIRCLE IMAGE VIEW

- ▶ Add the following to build.gradle
- ▶ Circular images - a recommended technique by Romain Guy
- ▶ <https://github.com/hdodenhof/CircleImageView>

```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"  
    implementation 'com.android.support:appcompat-v7:28.0.0'  
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'  
    implementation 'com.android.support:design:28.0.0'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'com.android.support.test:runner:1.0.2'  
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  
    implementation 'com.android.support:recyclerview-v7:28.0.0'  
    implementation 'com.squareup.picasso:picasso:2.5.2'  
    implementation 'de.hdodenhof:circleimageview:3.1.0'  
}
```

# CIRCLE IMAGE VIEW

- ▶ Change **ImageView** tag to  
**de.hdodenhof.circleimageview.CircleImageView** tag
- ▶ Make sure you remove **android:scaleType="fitCenter"**
- ▶ Check the documentation for additional attributes

```
<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/imvAlbumImage"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="16dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/ic_image_black_48dp" />
```

# CIRCLE IMAGE VIEW

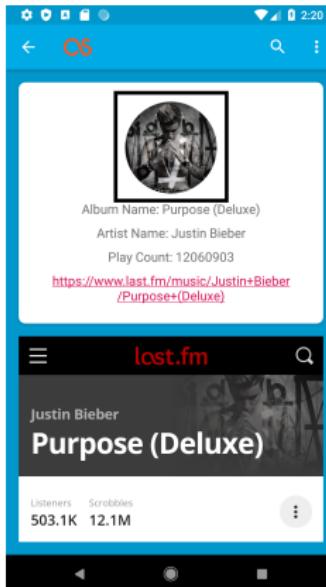
- ▶ In LastFmViewHolder, import the required library
- ▶ Change the view type to **CircleImageView**

```
import android.support.v7.widget.RecyclerView
import android.view.View
import android.widget.TextView
import de.hdodenhof.circleimageview.CircleImageView
import graysono.com.cp09progressdialogwebview.R

class LastFmViewHolder(view: View) : RecyclerView.ViewHolder(view) {
    var imgAlbum: CircleImageView = view.findViewById(R.id.imgAlbum)
    var txvName: TextView = view.findViewById(R.id.txvName)
    var txvPlayCount: TextView = view.findViewById(R.id.txvPlayCount)
}
```

# CIRCLE IMAGE VIEW

- ▶ DetailsActivity.kt
- ▶ As part of your practical, you will be required to change the ImageView tag in the **RecyclerView**



## NESTED MENU ITEMS

- ▶ In **menu\_main.xml**, add nested menu items
- ▶ Create a menu within an item
- ▶ Drawable icon next to title

```
<item
    android:icon="@drawable/ic_more_vert_white_24dp"
    android:title=""
    app:showAsAction="ifRoom">
    <menu>
        <item
            android:id="@+id/action_about"
            android:icon="@drawable/ic_person_black_24dp"
            android:title="About Us"
            app:showAsAction="ifRoom"/>
        <item
            android:id="@+id/action_settings"
            android:icon="@drawable/ic_settings_black_24dp"
            android:title="Settings"
            app:showAsAction="ifRoom"/>
    </menu>
</item>
```

# NESTED MENU ITEMS

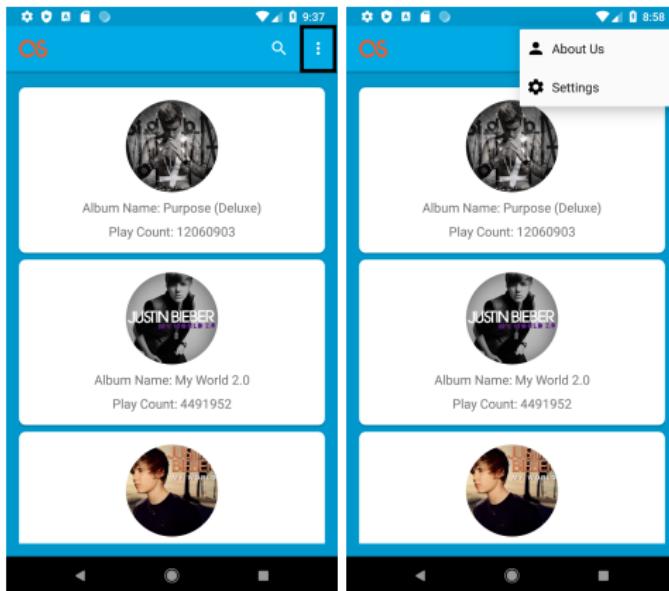
- ▶ Changes to **onOptionsItemSelected** in the **MainActivity.kt**

```
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_main -> {
            startActivity(Intent( packageContext: this@MainActivity, SearchActivity::class.java))
            true
        }
        R.id.action_about -> {
            true
        }
        else -> super.onOptionsItemSelected(item)
    }
}
```

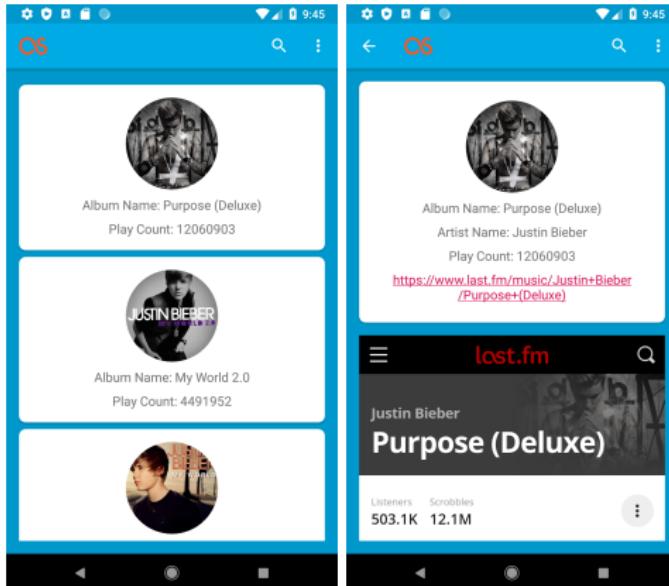
# NESTED MENU ITEMS

- ▶ Click on the vertical ellipsis
- ▶ Currently no functionality



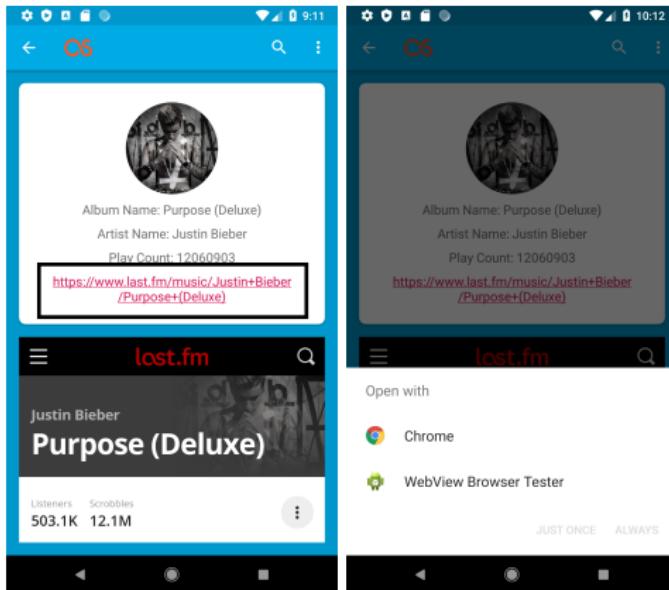
# TOOLBAR ICON

- ▶ Make a small tweak in the `activity_main.xml` & `activity_details.xml`
- ▶ Add a `TextView` widget to the **Toolbar** widget - this is nested



# TEXT VIEW WEB LINKS

- We can easily do this by adding two attributes to the **TextView** widget
  - android:autoLink="web"
  - android:linksClickable="true"



# ALERT DIALOG BUILDER

- ▶ **CustomAlertDialog.kt**
- ▶ Must the same as the **CustomProgressBar.kt**

```
class CustomAlertDialog(private val context: Context, layout: Int) {  
    private lateinit var alertDialog: AlertDialog  
    private val inflater: LayoutInflator =  
        context.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as LayoutInflator  
    private val view: View = inflater.inflate(layout, root: null)  
    private val builder = AlertDialog.Builder(context)  
  
    fun show(title: Int) {  
        builder.setTitle(title)  
        alertDialog = builder.setView(view).create()  
        builder.show()  
    }  
}
```

# ALERT DIALOG BUILDER

- ▶ **about\_us.xml** in layout resource directory
  - ▶ Horizontal divider
  - ▶ Text view

```
<View  
    android:id="@+id/divider"  
    android:layout_width="0dp"  
    android:layout_height="1dp"  
    android:layout_marginStart="16dp"  
    android:layout_marginTop="16dp"  
    android:layout_marginEnd="16dp"  
    android:background="?android:attr/listDivider"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />  
  
<TextView  
    android:id="@+id/tvAboutUs"  
    android:layout_width="0dp"  
    android:layout_height="wrap_content"  
    android:layout_marginStart="16dp"  
    android:layout_marginTop="8dp"  
    android:layout_marginEnd="16dp"  
    android:gravity="center"  
    android:text="@string/builder_about_us_message"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/divider" />
```

# ALERT DIALOG BUILDER

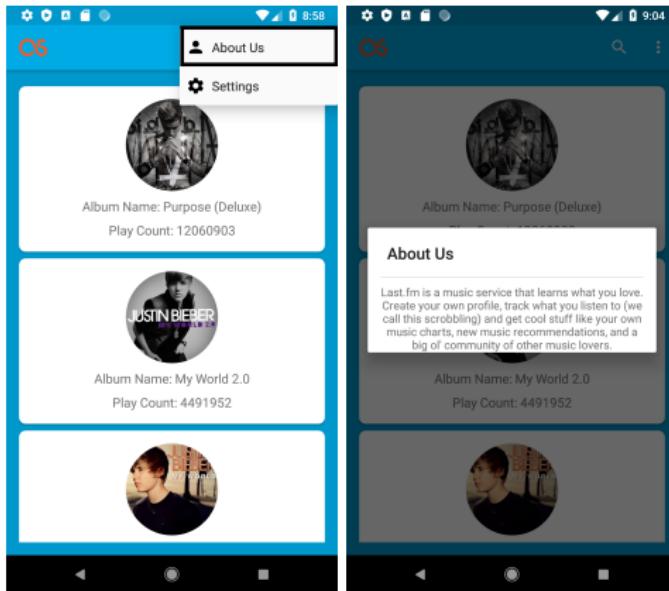
- ▶ Create **CustomAlertDialog** object in the **onOptionsItemSelected** in **MainActivity.kt**

```
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_main -> {
            startActivity(Intent( packageContext: this@MainActivity, SearchActivity::class.java))
            true
        }
        R.id.action_about -> {
            val aboutUsAlertDialog = CustomAlertDialog( context: this@MainActivity, R.layout.about_us)
            aboutUsAlertDialog.show(
                "About Us"
            )
            true
        }
        else -> super.onOptionsItemSelected(item)
    }
}
```

# ALERT DIALOG BUILDER

- ▶ Click on the **About Us** menu item
- ▶ In the practical, you will research & implement:
  - ▶ Dismiss the alert dialog via outside touch & ok button
  - ▶ Set an image next to the title



# 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

## EXAM 01 RESULTS

- ▶ Feedback & results available via Microsoft Teams
- ▶ Average: 44.63/50 or 89.26%