



# Lecture 10: Fragment & Dialog Fragment

## 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 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

## LECTURE 10: FRAGMENT & DIALOG FRAGMENT TOPICS

- ▶ More menu items
- ▶ Dialog fragment
- ▶ IDataReceived
- ▶ Fragment

# MENU

- ▶ Add new menu items to **menu\_main.xml**
- ▶ **Note:** About us icon has changed from a person icon to an information icon

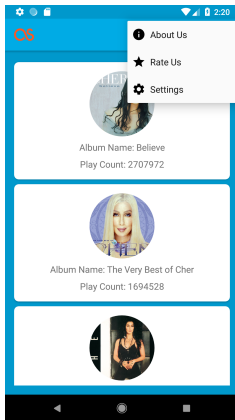
```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context=".activities.MainActivity">

    <item
        android:id="@+id/action_main"
        android:icon="@drawable/ic_search_white_24dp"
        android:title="Main"
        app:showAsAction="ifRoom" />

    <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_info_black_24dp"
                android:title="@string/action_about_us"
                app:showAsAction="ifRoom" />
            <item
                android:id="@+id/action_rate_us"
                android:icon="@drawable/ic_star_black_24dp"
                android:title="@string/action_rate_us"
                app:showAsAction="ifRoom" />
            <item
                android:id="@+id/action_settings"
                android:icon="@drawable/ic_settings_black_24dp"
                android:title="@string/action_settings"
                app:showAsAction="ifRoom" />
        </menu>
    </item>
</menu>
```

# MENU

- Output - nothing different, just new menu items



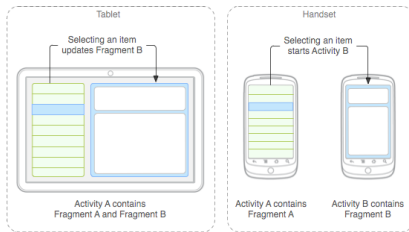
# MENU - MainActivity.kt

- ▶ Add menu item references to the when statement
- ▶ **Note:** the **showDialog()** & new activity called **SettingsActivity.kt**

```
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.custom_about_us)  
            aboutUsAlertDialog.show(  
                "About Us"  
            )  
            true  
        }  
        R.id.action_rate_us -> {  
            showDialog()  
            true  
        }  
        R.id.action_settings -> {  
            startActivity(Intent( packageContext: this@MainActivity, SettingsActivity::class.java))  
            true  
        }  
        else -> super.onOptionsItemSelected(item)  
    }  
}
```

# WHAT ARE FRAGMENTS?

- ▶ Primarily to support more dynamic and flexible UI designs on large screens such as tablets
  - ▶ Due to a tablet's screen being much larger than that of a mobile device
  - ▶ More space to combine and interchange UI components
- ▶ Example: news application
  - ▶ One fragment displays a list of articles on the left hand-side
  - ▶ Another fragment displays a single article on the right hand-side
  - ▶ Both fragments appear in one activity
  - ▶ Each fragment has its own set of life cycle callback methods & events



# CREATING A FRAGMENT

- ▶ You must create a subclass of **Fragment**
- ▶ The **Fragment** class has code that looks a lot like an **Activity**
- ▶ `onCreateView`
  - ▶ System calls this when it's time for the fragment to draw its user interface for the first time
  - ▶ To draw, you must return a **View** that is the root of your fragment's layout



# SHOW DIALOG

- ▶ Private function in **MainActivity.kt**
- ▶ Displays the dialog fragment via the fragment support manager

```
private fun showDialog() {  
    val dialogFragment = RateUsDialogFragment( listener: this)  
    dialogFragment.show(supportFragmentManager, tag: null)  
}
```

# DIALOG FRAGMENT

- ▶ Subclass of **DialogFragment**
- ▶ Takes **IDataReceived** as its argument - we will look at this soon

```
class RateUsDialogFragment(private val listener: IDataReceived) : DialogFragment() {  
    override fun onCreateView(  
        inflater: LayoutInflater,  
        container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View? {  
        isCancelable = false  
        return inflater.inflate(R.layout.dialog_fragment_rate_us, container, attachToRoot: false)  
    }  
  
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
        super.onViewCreated(view, savedInstanceState)  
        btnNoThanks.setOnClickListener { dismiss() }  
        btnRateNow.setOnClickListener { lt: View!  
            listener.onDataReceived( data: "Thank you! Your feedback is very helpful for us.")  
            dismiss()  
        }  
    }  
}
```

# DIALOG FRAGMENT

## ► onCreateView

- You will need to convert your project to AndroidX, in order to define this
- Take **LayoutInflater**, **ViewGroup**, & **Bundle** as its argument
- Set the dialog to be **not** cancelable
- Return the inflated layout - **dialog\_fragment\_rate\_us**

```
class RateUsDialogFragment(private val listener: IDataReceived) : DialogFragment() {  
    override fun onCreateView(  
        inflater: LayoutInflater,  
        container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View? {  
        isCancelable = false  
        return inflater.inflate(R.layout.dialog_fragment_rate_us, container, attachToRoot: false)  
    }  
  
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
        super.onViewCreated(view, savedInstanceState)  
        btnNoThanks.setOnClickListener { dismiss() }  
        btnRateNow.setOnClickListener { it: View!  
            listener.onDataReceived( data: "Thank you! Your feedback is very helpful for us.")  
            dismiss()  
        }  
    }  
}
```

# DIALOG FRAGMENT

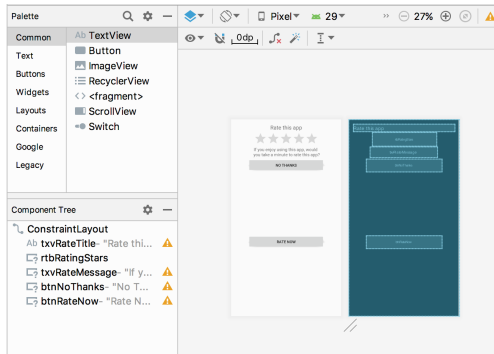
## ► onViewCreated

- Two button click listeners
- One dismisses the dialog fragment
- The allows the **context/other fragments** to use this data, then dismisses the dialog fragment

```
class RateUsDialogFragment(private val listener: IDataReceived) : DialogFragment() {  
    override fun onCreateView(  
        inflater: LayoutInflater,  
        container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View? {  
        isCancelable = false  
        return inflater.inflate(R.layout.dialog_fragment_rate_us, container, attachToRoot: false)  
    }  
  
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
        super.onViewCreated(view, savedInstanceState)  
        btnNoThanks.setOnClickListener { dismiss() }  
        btnRateNow.setOnClickListener { it: View!  
            listener.onDataReceived( data: "Thank you! Your feedback is very helpful for us.")  
            dismiss()  
        }  
    }  
}
```

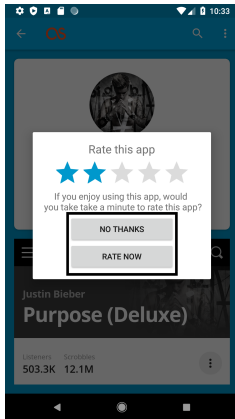
# DIALOG FRAGMENT XML

- ▶ **dialog\_fragment\_rate\_us.xml**
- ▶ In order of widgets:
  - ▶ Text view, rating bar, text view, button & button
- ▶ The position of the last button may look strange. It is constrain to the bottom so that the margin between the top & bottom is even at 16dp





# DIALOG FRAGMENT OUTPUT

- ▶ When the user clicks on the either button, it should perform the actions aforementioned



# IDataReceived

- ▶ Create a new interface called IDataReceived
- ▶ Remember, this goes in the interfaces directory
- ▶ Performs an action when data is received from a fragment

```
 interface IDataReceived {  
    fun onDataReceived(data: String)  
 }
```

# IDataRECEIVED

- Implement the interface in the **MainActivity.kt**

```
class MainActivity : BaseActivity(), IDataDownloadAvailable,  
    IDataDownloadComplete, IRecyclerViewItem, IDataReceived {
```



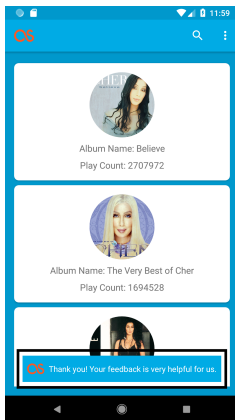
# IDATARECEIVED

## ► Custom toast - practical 03

```
override fun onDataReceived(data: String) {  
    val inflater = LayoutInflater  
    val layout: View = inflater.inflate(  
        R.layout.custom_toast,  
        findViewById(R.id.layoutRateUs)  
    )  
    val toast = Toast( context: this@MainActivity)  
    val toastTextView = layout.findViewById(R.id.txvRateUsToast) as TextView  
    toastTextView.text = data  
    toast.duration = Toast.LENGTH_LONG  
    toast.view = layout  
    toast.show()  
}
```

# DATA RECEIVED

- ▶ Output - custom toast at the bottom
- ▶ **Note:** toasts can be positioned anywhere on the screen



# FRAGMENT

## ► onCreateView

- The same as the dialog fragment - inflate **fragment\_profile**

```
class ProfileFragment : Fragment() {  
  
    override fun onCreateView(  
        inflater: LayoutInflater,  
        container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View {  
        return inflater.inflate(R.layout.fragment_profile, container, attachToRoot: false)  
    }  
  
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
        super.onViewCreated(view, savedInstanceState)  
        val fragmentManager? = activity?.supportFragmentManager  
        view.findViewById<Button>(R.id.btnPersonalInfoSave).setOnClickListener { it: View!  
            fragmentManager?.beginTransaction()?.remove( fragment: this)?.commit()  
        }  
    }  
}
```

# FRAGMENT

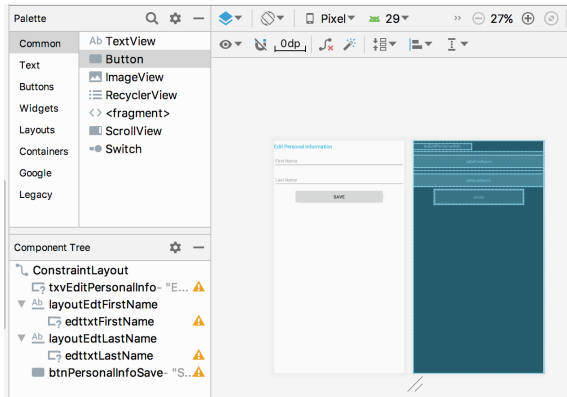
## ► onViewCreated

- Find view by id in a fragment
- You can't directly reference the id. It will stop your application

```
class ProfileFragment : Fragment() {  
    override fun onCreateView(  
        inflater: LayoutInflater,  
        container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View {  
        return inflater.inflate(R.layout.fragment_profile, container, attachToRoot: false)  
    }  
  
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
        super.onViewCreated(view, savedInstanceState)  
        val fragmentManager: FragmentManager? = activity?.supportFragmentManager  
        view.findViewById<Button>(R.id.btnPersonalInfoSave).setOnClickListener { it: View!  
            fragmentManager?.beginTransaction()?.remove( fragment: this)?.commit()  
        }  
    }  
}
```

# FRAGMENT XML

- ▶ **fragment\_profile.xml**
- ▶ In order of widgets:
  - ▶ Text view, edit text, edit text & button
  - ▶ Ignore the warning signs



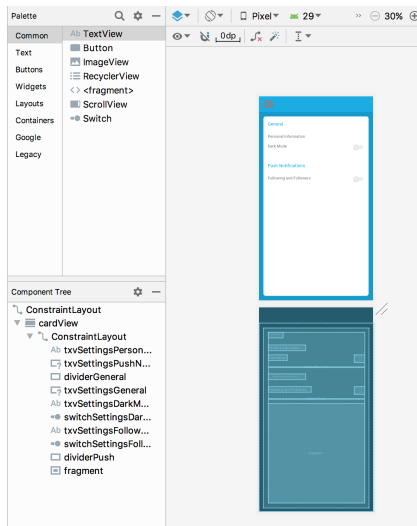
# FRAGMENT - SETTINGSACTIVITY.KT

- ▶ Create a new class called **SettingsActivity.kt** via the basic activity wizard
- ▶ **ProfileFragment** object
- ▶ Inner class - begin transaction – > replace the view with the fragment – > commit

```
class SettingsActivity : BaseActivity() {  
    private val fragManager: FragmentManager = supportFragmentManager  
    private val profileFrag = ProfileFragment()  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_settings)  
        displayToolbar(isHomeEnabled: true)  
        txvSettingsPersonalInformation.setOnClickListener(PersonalInfoButtonOnClickListener())  
    }  
  
    inner class PersonalInfoButtonOnClickListener : View.OnClickListener {  
        override fun onClick(view: View) {  
            fragManager.beginTransaction()  
                .replace(R.id.fragment, profileFrag)  
                .commit()  
        }  
    }  
}
```

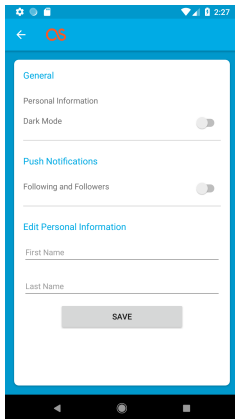
# FRAGMENT - SETTINGSACTIVITY.KT

## ► activity\_settings.xml



# FRAGMENT - OUTPUT

- Click on the text view to fire event





# EXAM 02

- ▶ Series of tasks covering lectures 05-08
- ▶
- ▶ Worth 6% of your final mark for the Design and Development of Applications for Mobile Devices course
- ▶ Release: Today at 5pm
- ▶ Deadline: Friday, 27 March at 5pm