



# Lecture 02: Activities & Intents

## IN721: Mobile Application Development

### Semester One, 2020

Kaiako: Grayson Orr

Te Kura Matatini ki Otago, Ōtepoti, Aotearoa

Friday, 21 February

# LECTURE 01: TOUCH & INPUT TOPICS

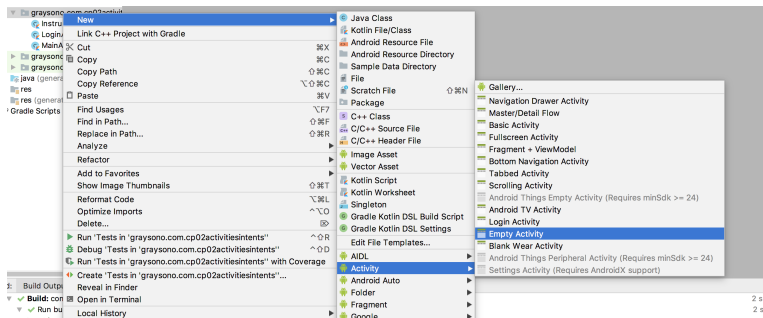
- ▶ Brief History
- ▶ Linux Kernel
- ▶ Software Stack
- ▶ Open-Source Community
- ▶ QEMU
- ▶ Kotlin
- ▶ Android Studio
- ▶ ArrayAdapter
- ▶ Inner class

# LECTURE 02: ACTIVITIES & INTENTS TOPICS

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

# ACTIVITIES: CREATING A NEW ACTIVITY

## ► New > Activity > Empty Activity



# ACTIVITIES: CREATING A NEW ACTIVITY

- ▶ Activities are added to the AndroidManifest.xml



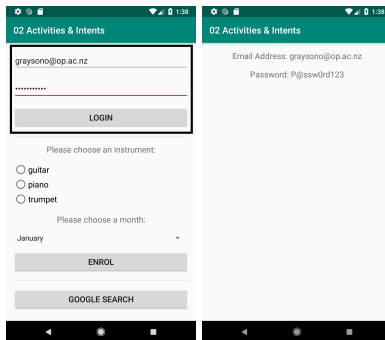
```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3          package="graysono.com.cp02activitiesintents">
4
5      <application
6          android:allowBackup="true"
7          android:icon="@mipmap/ic_launcher"
8          android:label="@string/app_name"
9          android:roundIcon="@mipmap/ic_launcher_round"
10         android:supportRtl="true"
11         android:theme="@style/AppTheme">
12         <activity android:name=".MainActivity">
13             <intent-filter>
14                 <action android:name="android.intent.action.MAIN">
15
16                 <category android:name="android.intent.category.LAUNCHER">
17
18             </intent-filter>
19         </activity>
20         <activity android:name=".LoginActivity">
21
22         <activity android:name=".InstrumentActivity">
23
24     </application>
25 </manifest>
```

# INTENTS: COMMUNICATION BETWEEN ACTIVITIES

- ▶ Data structure holding an abstract description of an action to be performed
- ▶ Two use cases:
  - ▶ XActivity passes data directly to YActivity when it transfers control
  - ▶ XActivity requests from YActivity - YActivity will return the result of XActivity

# INTENTS: PASSING DATA WHEN LAUNCHING ACTIVITY

- Data is passed into an activity when launched via an intent using `Intent.putExtra(String name, Bundle value)`



# INTENTS: PASSING DATA WHEN LAUNCHING ACTIVITY

- ▶ Intent - packageContext & class
- ▶ Key/value
  - ▶ Name = email\_address
  - ▶ Value = email edit text value
- ▶ Inner class - View.OnClickListener
- ▶ Pass multiple key/values

```
val intent = Intent( packageContext: this@MainActivity, LoginActivity::class.java)
intent.putExtra( name: "email_address",
    getString(R.string.email_address_output, edtEmail.text))
intent.putExtra( name: "password",
    getString(R.string.password_output, edtPassword.text))
startActivity(intent)
```



# INTENTS: RECEIVING DATA WHEN LAUNCHING ACTIVITY

- ▶ How do we get the data from XActivity to YActivity?
  - ▶ Get a reference to the intent launched
  - ▶ getStringExtra(String name)

```
7 class LoginActivity : AppCompatActivity() {  
8     override fun onCreate(savedInstanceState: Bundle?) {  
9         super.onCreate(savedInstanceState)  
10        setContentView(R.layout.activity_login)  
11        txvEmailAddress.text = intent.getStringExtra( name: "email_address")  
12        txvPassword.text = intent.getStringExtra( name: "password")  
13    }  
14 }
```

# INTENTS: RECEIVING DATA WHEN LAUNCHING ACTIVITY

- ▶ Syntactic alternative
  - ▶ `intent.extras`

```
8 class LoginActivity : AppCompatActivity() {  
9     override fun onCreate(savedInstanceState: Bundle?) {  
10         super.onCreate(savedInstanceState)  
11         setContentView(R.layout.activity_login)  
12         val bundledData: Bundle? = intent.extras  
13         txvEmailAddress.text = bundledData?.getString( key: "email_address")  
14         txvPassword.text = bundledData?.getString( key: "password")  
15     }  
16 }
```

# INTENTS: EXPLICIT

- ▶ Specify which application will satisfy the intent
  - ▶ Target application's package name or a component class name is supplied
  - ▶ Used to start a component in your own application - you know the class name of the activity
  - ▶ For example, starting a new activity within your application in response to a user action - button click

```
startActivity(Intent( packageContext: this@MainActivity, LoginActivity::class.java))
```

# INTENTS: IMPLICIT

- ▶ Do not name a specific component
- ▶ Allows a component from another application to handle it
- ▶ For example, showing the user a location on a map, a request can be made to another application...Google Maps

```
startActivity(Intent(Intent.ACTION_VIEW, Uri.parse(uriString: "https://www.google.com")))
```

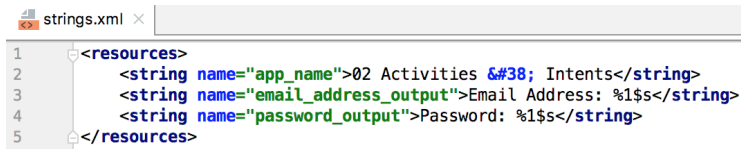
# ONBACKPRESSED

- ▶ override fun onBackPressed()

```
override fun onBackPressed() {  
    super.onBackPressed()  
    val intent = Intent( packageContext: this@LoginActivity, MainActivity::class.java)  
    startActivity(intent)  
}
```

# STRING PLACEHOLDER

## ► strings.xml



```
strings.xml x
1 <resources>
2   <string name="app_name">02 Activities &#38; Intents</string>
3   <string name="email_address_output">Email Address: %1$s</string>
4   <string name="password_output">Password: %1$s</string>
5 </resources>
```

# 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