5COSC023W - MOBILE APPLICATION DEVELOPMENT

Lecture 4: Intents and Displaying Images

Dr Dimitris C. Dracopoulos

Intents

- A description of an operation to be performed
- Can be used to start other activities
- Can be *explicit* (starting a specific activity) or *implicit* (letting the system or the user to choose which activity to start)

Example:

```
// get the context of the application
val context = LocalContext.current

// create the intent with the destination activity
val in = Intent(context, NewActivity::class.java)

// start the new activity (i.e. from the NewActivity class)
context.startActivity(in)
```

Displaying Images

1. Copy the file with the image in the `drawable` subdirectory of resources (`res`) (filename cannot contain capitals).
2. Use a `Image` composable.

The Lost Dog Application

- Create an application which can notify owners of lost dogs if someone finds them.

The Main activity

File `MainActivity.kt` contains:

```
package com.example.thelostdogcomposableapp
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.Column
import androidx.compose.material3.Button
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.tooling.preview.Preview

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            displayGUI()
        }
    }
}
```

The Main activity (cont'd)

```
@Composable
fun displayGUI() {
    Column {
        val context = LocalContext.current
        Text("Lost Dog - Have you found her?")
        Image(
            painterResource(id = R.drawable.brittany_02625),
            contentDescription = "Image of the dog"
        )
        Button(onClick = {
            val i = Intent(context, ContactDetails::class.java)
            context.startActivity(i)
        })
        {
            Text("Contact the Owner")
        }
    }
}
```

The Second activity

File `ContactDetails.kt` contains:

```
package com.example.thelostdogcomposableapp
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.material3.Button
import androidx.compose.material3.Text
import androidx.compose.ui.platform.LocalContext

class ContactDetails : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Column {
                val context = LocalContext.current
                Text("Contact Details of the Owner")

                Button(onClick = {
                     var phoneUri = Uri.parse("tel:02079115000")
                     var i = Intent(Intent.ACTION_DIAL, phoneUri)
                     context.startActivity(i)
                }) {
                     Text("Call the owner")
                }
```

The Second activity (cont'd)

```
                Button(onClick = {
                    var i = Intent(Intent.ACTION_SEND)
                    i.setType("text/plain")
                    i.putExtra(Intent.EXTRA_EMAIL, arrayOf("brittanyOwner@gmail.com"))
                    i.putExtra(Intent.EXTRA_TEXT, "Hello, I think that I found your lost dog!")
                    i.putExtra(Intent.EXTRA_SUBJECT, "Your lost dog")
                    startActivity(i)
                }) {
                    Text("Email the Owner")
                }
            }
        }
    }
}
```

The Manifest file

Do not forget to modify the `AndroidManifest.xml` file for your project by adding to it any new activity. If you create the activity using the Android Studio menu (create new activity→Gallery→Empty Activity) it will be automatically added to the manifest.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <application
        ...tools:targetApi="33">
        <activity
            android:name=".ContactDetails"
            android:exported="false"
            android:label="@string/title_activity_contact_details"
            android:theme="@style/Theme.TheLostDogComposableApp" />
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.TheLostDogComposableApp">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Modifiers

Use to decorate or change the appearance or behaviour of Composables (UI elements).

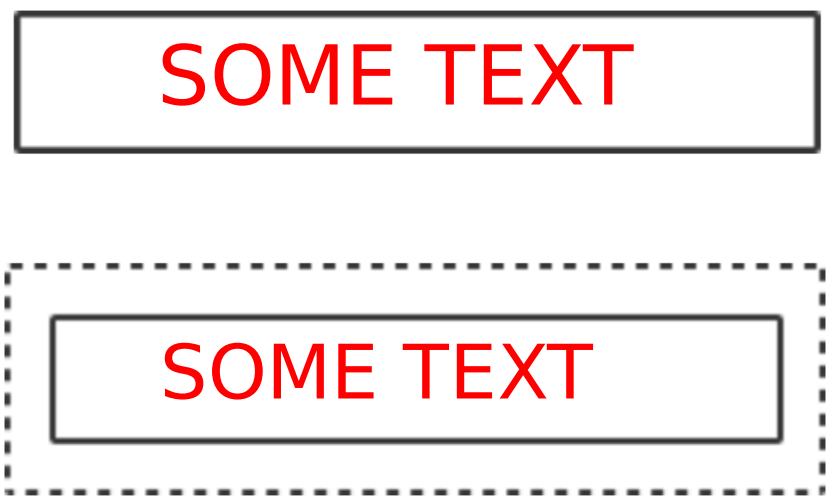
Can make UI elements:

- Clickable
- Scrollable
- Animated

Padding in Jetpack Compose

Unlike `Views` where padding and margins are different (margins is the extra space outside the border of the view, while padding is the extra space within the view, e.g. the space between the text displayed as part of the view and the border of the view).

- In Jetpack Compose padding defines extra space which is added to the composable container.
- The padding makes the Composable increasing its size.
- The content of the Composable (e.g. text or image) cannot use the extra space.



Sending Data from one Activity to Another

In the Sending Activity:

1. Create the Intent.
2. Set data or put extra data in the Intent.
3. Start the receiving (new activity) with `startActivity(intent)`.

In the Receiving Activity:

1. Get the Intent that created the Activity.
2. Retrieve the data or extras from the Intent.

Examples

```
// Setting data and extras
intent.setData(Uri.parse("http://www.google.co.uk"));
intent.setData(Uri.parse("tel:02079115000"));
intent.putExtra("Score", 56345);

// Retrieving data and extras
Uri url = intent.getData();
int score = intent.getIntExtra("score", 0);
```