# Application Development for Mobile Computer

## <Week 7 (Lab)>

Youn Kyu Lee

# Warm-up

```
package com.example.camera

import android.Manifest
import android.content.Intent
import android.os.Bundle
import android.provider.MediaStore
import android.widget.Toast
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.activity.result.contract.ActivityResultContracts
import androidx.compose.foundation.layout.*
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.camera.ui.theme.CAMERATheme

class MainActivity : ComponentActivity() {

    private val requestPermissionLauncher =
        registerForActivityResult(ActivityResultContracts.RequestPermission()) { granted
->
            if (granted) {
                openCamera()
            } else {
                Toast.makeText(this, "Camera permission denied",
Toast.LENGTH_SHORT).show()
            }
        }
```

1

```
override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            CAMERATheme {
                Scaffold(
                    modifier = Modifier.fillMaxSize(),
                ) { innerPadding ->
                    CameraScreen(
                        modifier = Modifier.padding(innerPadding),
                        onOpenCamera = {

requestPermissionLauncher.launch(Manifest.permission.CAMERA)
                        }
                    )
                }
            }
        }
    }
```

2

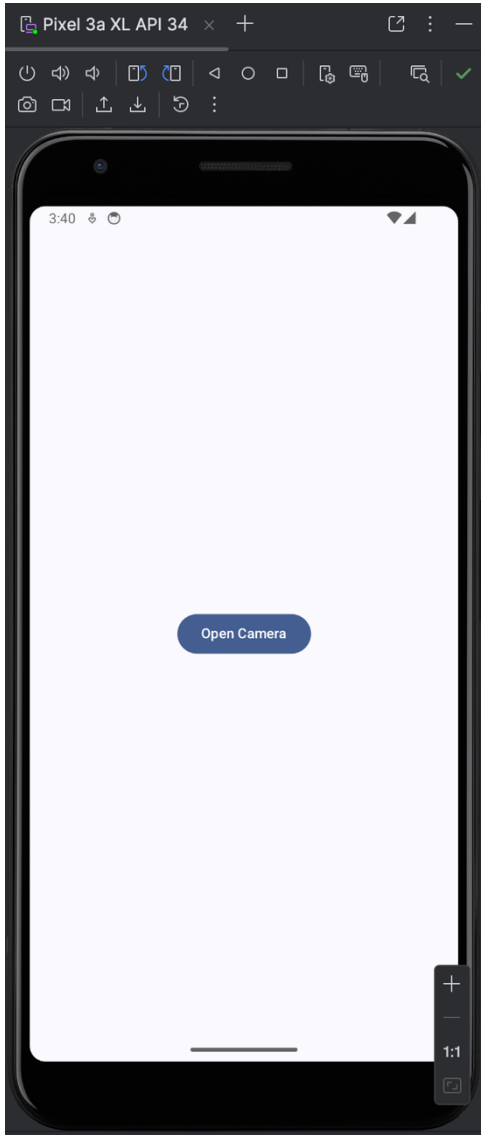
```
    private fun openCamera() {
        val intent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
        try {
            startActivity(intent)
        } catch (e: Exception) {
            Toast.makeText(this, "No camera app found",
Toast.LENGTH_SHORT).show()
        }
    }
}

@Composable
fun CameraScreen(modifier: Modifier = Modifier, onOpenCamera: () -> Unit) {
    Box(
        modifier = modifier.fillMaxSize(),
        contentAlignment = Alignment.Center
    ) {
        Button(onClick = onOpenCamera) {
            Text("Open Camera")
        }
    }
}
```

3

```
@Preview(showBackground = true)
@Composable
fun PreviewCameraScreen() {
    CAMERATheme {
        CameraScreen {}
    }
}
```

4

# Warm-up




MainActivity.kt | AndroidManifest.xml

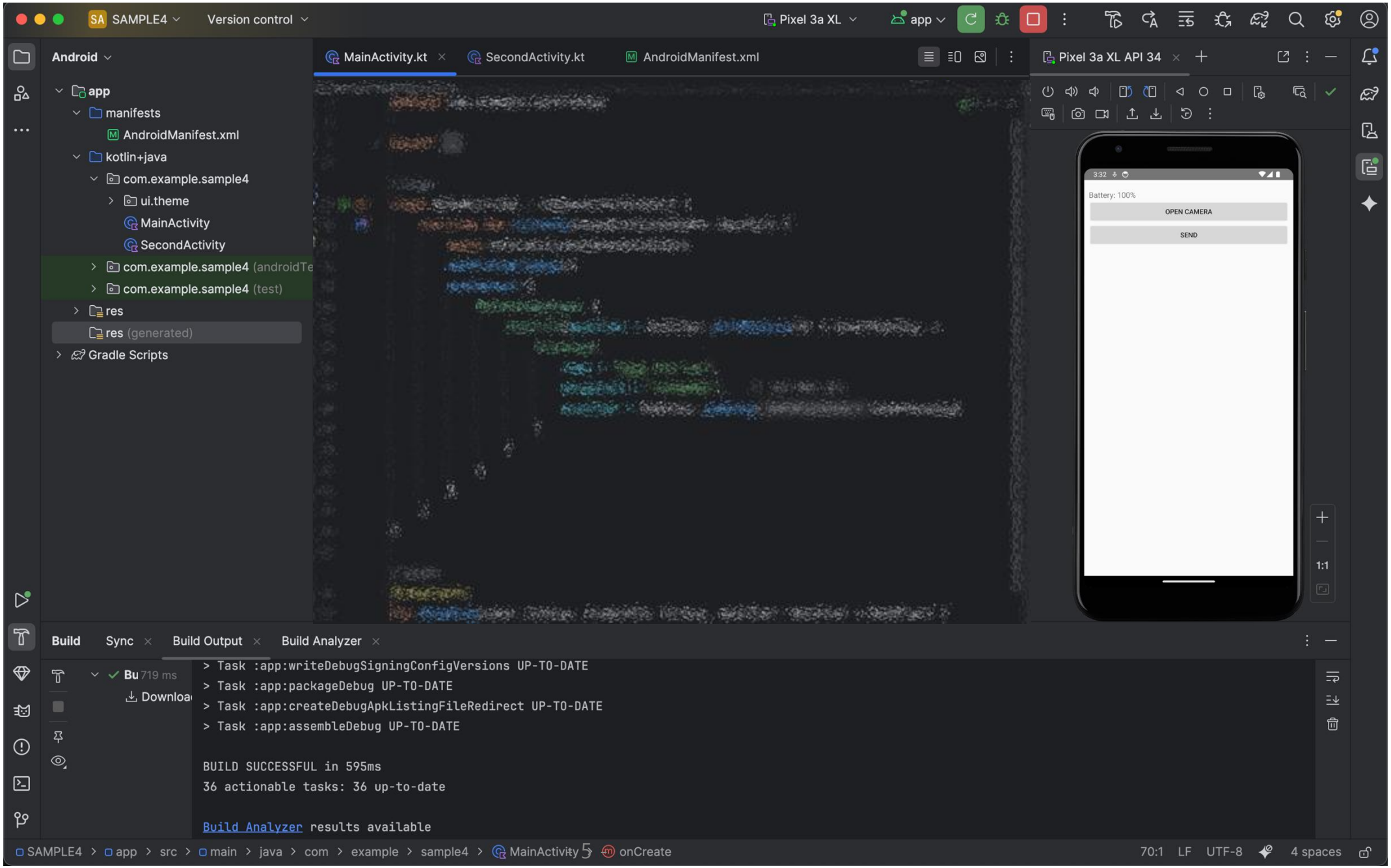
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-feature
        android:name="android.hardware.camera"
        android:required="false" />
    <uses-permission android:name="android.permission.CAMERA" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupCont
        android:icon="@mipmap/
        android:label="CAMERA"
        android:roundIcon="@mi
        android:supportsRtl="t
        android:theme="@style/
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="CAMERA"
            android:theme="@style/Theme.CAMERA">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

```
<uses-feature
    android:name="android.hardware.camera"
    android:required="false" />
<uses-permission android:name="android.permission.CAMERA" />
```

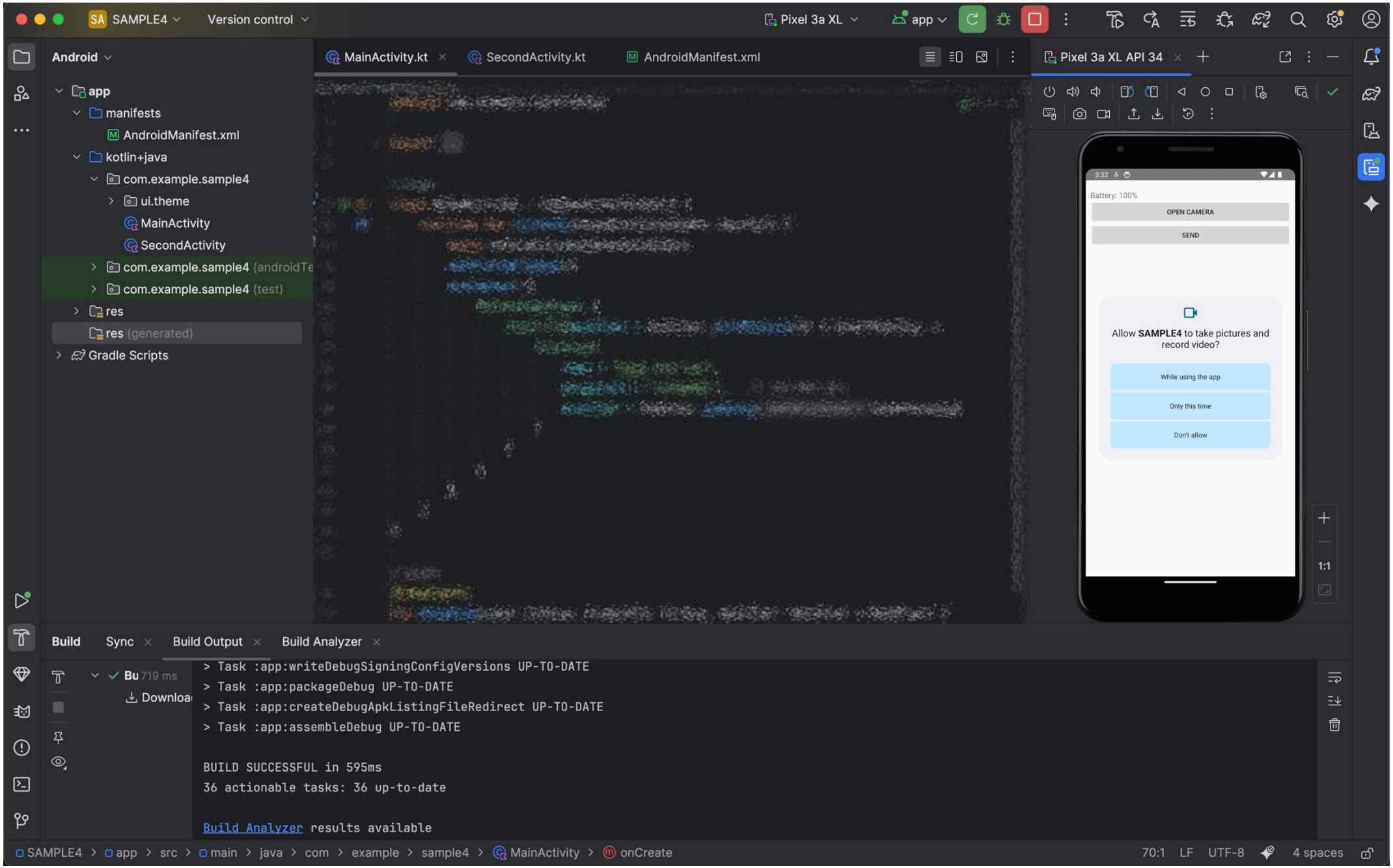
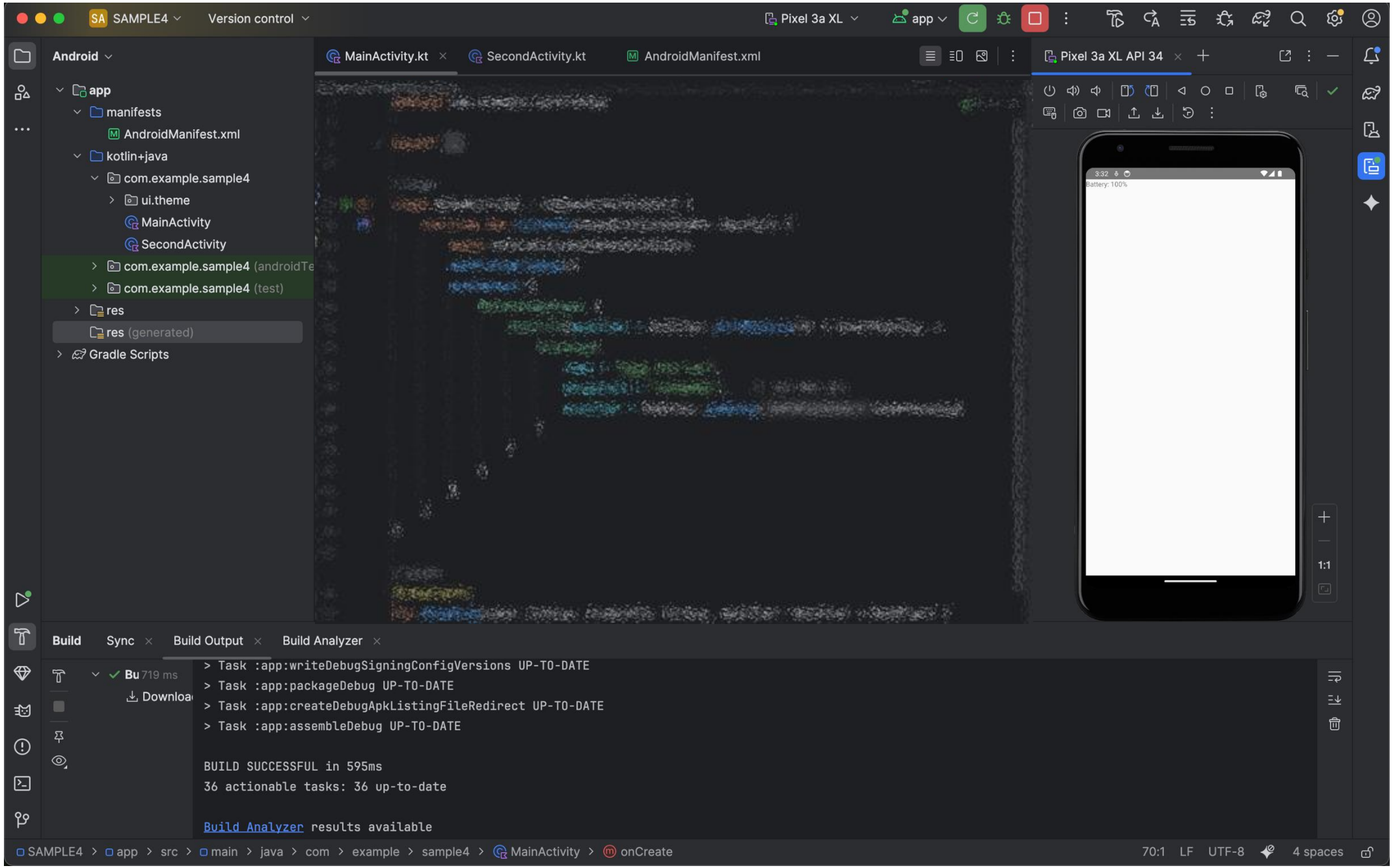
# Assignment#7

- **Task**
  - Create a simple app using Android Studio.
  - MainActivity
    - Display the current battery percentage at the top of the screen.
    - Include two buttons:
      - OPEN CAMERA: Requests camera permission and opens the device camera when permission is granted.
      - SEND: Sends the current battery percentage to SecondActivity using an Intent.
  - SecondActivity
    - Receives the battery percentage value from MainActivity.
    - Displays the received battery level as plain text.
  - Take screenshots of the emulator showing the results (refer to the samples on the next pages).
  - Submit the final APK file together with the screenshots.
- **Submission**
  - **"Three" Emulator screenshots** (e.g., **Lab7_StudentID.png**)
  - **APK file** (e.g., **Lab7_StudentID.apk**)
- **How to Submit**
  - Upload both the screenshot and APK file to e-class
  - Deadline: 25.11.09

# Lab7_StudentID_1.png



# Lab7_StudentID_2.png



# Lab7_StudentID_3.png