CSE 162 Mobile Computing

Lab5a Camera Programming

Goal: Use Camera to take a photo.

- Use built-in camera to take a photo
- Return the full-size photo
- Display it on the screen

Use Camera to take a photo.

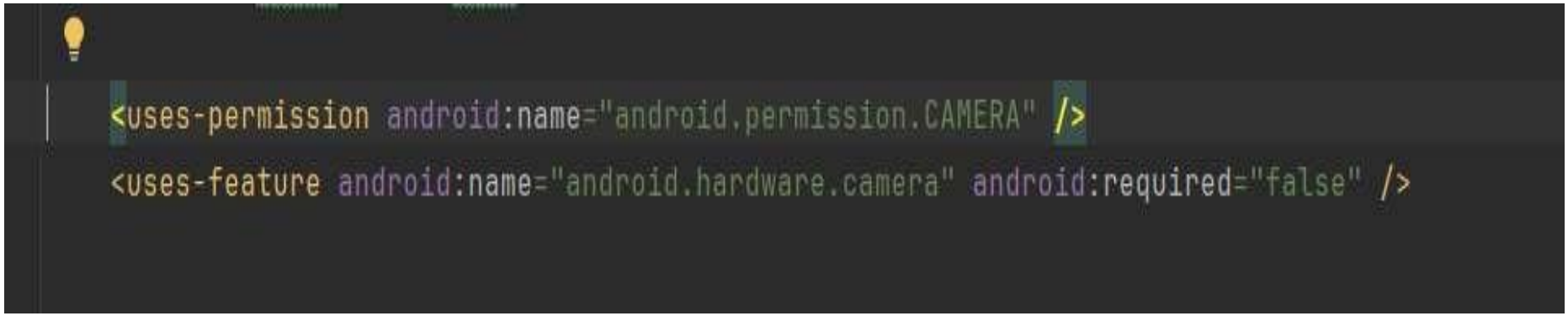
- Create Intent to existing Camera App--- just to take quick picture and get file returning to your app…
- https://developer.android.com/training/camera/photobasics.html

Some Classes involved…..

- Option 1: via Intent to existing Camera App
  - Intent An intent action type of MediaStore.ACTION_IMAGE_CAPTURE or MediaStore.ACTION_VIDEO_CAPTURE can be used to capture

Request Camera Permission

- We create a TakingPhoto project.
- To advertise that your application depends on having a camera, put a <user-feature> tag in your manifest file:



Take a Photo with the Camera App

- The Android way of delegating actions to other applications is to invoke an Intent that describes what you want done. This process involves three pieces: The Intent itself, a call to start the external Activity, and some code to handle the image data when focus returns to your activity.

```
private void openCamera() {  2 usages
    Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    cameraLauncher.launch(cameraIntent);
}
```

Permissions

- Remember to grant the permission
- A simple if-else statement will provide the log on the terminal to help you debug

```
    private final ActivityResultLauncher<String> requestPermissionLauncher =  1 usage
            registerForActivityResult(new ActivityResultContracts.RequestPermission(), isGranted -> {
                if (isGranted) {
                    openCamera();
                } else {
                    Log.e( tag: "Permission",  msg: "Camera permission denied");
                }
            });

}
```

Set a bitmap on the result

```
cameraLauncher = registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(),
        result -> {
            if (result.getResultCode() == RESULT_OK && result.getData() != null) {
                Bitmap picture = (Bitmap) result.getData().getExtras().get("data");
                image.setImageBitmap(picture);
            }
        }
);
```

Taking a picture is using .ACTION_IMAGE_CAPTURE

```
private void openCamera() {  2 usages
    Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    cameraLauncher.launch(cameraIntent);
}
```

Update your layout xml

- Create a xml layout directory under /res
- Create a activity_main.xml file

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/image"
        android:layout_width="match_parent"
        android:layout_height="600sp" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/image"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20sp"
        android:text="Click Picture"
        android:id="@+id/picture"/>
</RelativeLayout>
```

AndroidManifest.xml

```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-permission android:name="android.permission.CAMERA" />
    <uses-feature android:name="android.hardware.camera" android:required="false" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="camera"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Camera">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Newer api versions (import)

```
1   package rahul.cse_162_2l.camera;
2
3   import android.content.Intent;
4   import android.content.pm.PackageManager;
5   import android.graphics.Bitmap;
6   import android.os.Bundle;
7   import android.provider.MediaStore;
8   import android.util.Log;
9   import android.widget.Button;
10  import android.widget.ImageView;
11
12  import androidx.activity.result.ActivityResultLauncher;
13  import androidx.activity.result.contract.ActivityResultContracts;
14  import androidx.appcompat.app.AppCompatActivity;
15  import androidx.core.content.ContextCompat;
16
17  import android.Manifest;
```

MainActivity.java

```java
public class MainActivity extends AppCompatActivity {

    2 usages
    ImageView image;
    2 usages
    Button click_picture;

    2 usages
    private ActivityResultLauncher<Intent> cameraLauncher;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        image = findViewById(R.id.image);
        click_picture = findViewById(R.id.picture);

        cameraLauncher = registerForActivityResult(
                new ActivityResultContracts.StartActivityForResult(),
                ActivityResult result -> {
                    if (result.getResultCode() == RESULT_OK && result.getData() != null) {
                        Bitmap picture = (Bitmap) result.getData().getExtras().get("data");
                        image.setImageBitmap(picture);
                    }
                }
        );

        click_picture.setOnClickListener( View view -> {
            if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.CAMERA) == PackageManager.PERMISSION_GRANTED) {
                openCamera();
            } else {
                requestPermissionLauncher.launch(Manifest.permission.CAMERA);
            }
        });
    }
```

MainActivity.java cont.

```
    2 usages
    private void openCamera() {
        Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        cameraLauncher.launch(cameraIntent);
    }

    1 usage
    private final ActivityResultLauncher<String> requestPermissionLauncher =
            registerForActivityResult(new ActivityResultContracts.RequestPermission(), Boolean isGranted -> {
                if (isGranted) {
                    openCamera();
                } else {
                    Log.e( tag: "Permission", msg: "Camera permission denied");
                }
            });
}
```

Build:grade (app)

```
plugins {
    alias(libs.plugins.android.application)
}

android {
    namespace 'rahul.cse_162_2l.camera'
    compileSdk 36

    defaultConfig { DefaultConfig it ->
        applicationId "rahul.cse_162_2l.camera"
        minSdk 24
        targetSdk 36
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes { NamedDomainObjectContainer<BuildType> it ->
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions { CompileOptions it ->
        sourceCompatibility JavaVersion.VERSION_11
        targetCompatibility JavaVersion.VERSION_11
    }
}

dependencies {

    implementation libs.appcompat
    implementation libs.material
    testImplementation libs.junit
    androidTestImplementation libs.ext.junit
    androidTestImplementation libs.espresso.core
}
```

Reference of the toml

```
[versions]
agp = "8.6.0"
junit = "4.13.2"
junitVersion = "1.2.1"
espressoCore = "3.6.1"
appcompat = "1.7.0"
material = "1.12.0"

[libraries]
junit = { group = "junit", name = "junit", version.ref = "junit" }
ext-junit = { group = "androidx.test.ext", name = "junit", version.ref = "junitVersion" }
espresso-core = { group = "androidx.test.espresso", name = "espresso-core", version.ref = "espressoCore" }
appcompat = { group = "androidx.appcompat", name = "appcompat", version.ref = "appcompat" }
material = { group = "com.google.android.material", name = "material", version.ref = "material" }

[plugins]
android-application = { id = "com.android.application", version.ref = "agp" }
```