

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.

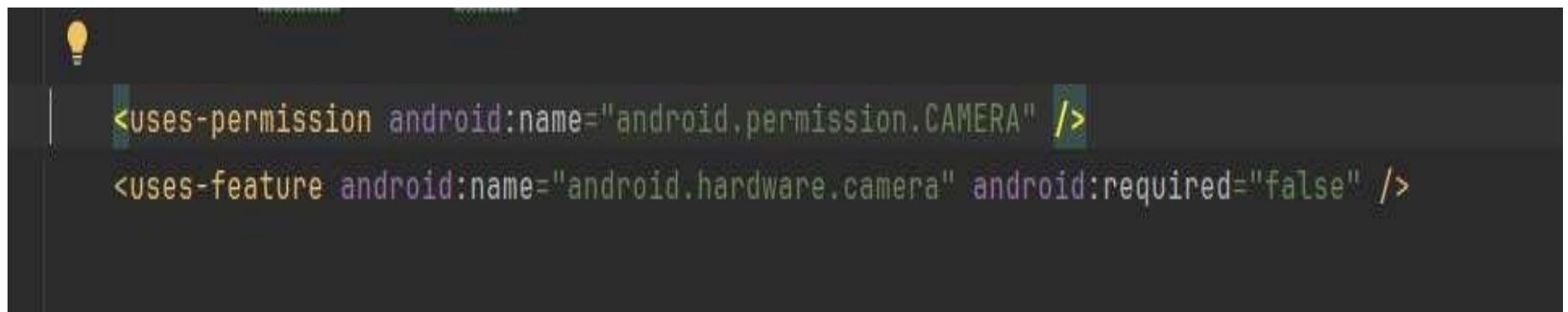
- Create Intent to existing Camera App--- just to take quick picture and get file returning to your app...
- <https://developer.android.com/training/camera/photobasics.html>

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:



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

A screenshot of an AndroidManifest.xml file in an IDE. The code shows two declarations: a permission declaration for the CAMERA permission and a feature declaration for the android.hardware.camera feature, which is marked as optional (android:required="false").

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 = 1usage
    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

```
1  <?xml version="1.0" encoding="utf-8"?>
2  © <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      tools:context=".MainActivity">
7
8      <ImageView
9          android:id="@+id/image"
10         android:layout_width="match_parent"
11         android:layout_height="600sp" />
12
13      <Button
14          android:layout_width="wrap_content"
15          android:layout_height="wrap_content"
16          android:layout_below="@+id/image"
17          android:layout_centerHorizontal="true"
18          android:layout_marginTop="20sp"
19          android:text="Click Picture"
20          android:id="@+id/picture"/>
21
22  </RelativeLayout>
```

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools">
4          <uses-permission android:name="android.permission.CAMERA" />
5          <uses-feature android:name="android.hardware.camera" android:required="false" />
6          <application
7              android:allowBackup="true"
8              android:dataExtractionRules="@xml/data_extraction_rules"
9              android:fullBackupContent="@xml/backup_rules"
10             android:icon="@mipmap/ic_launcher"
11             android:label="camera"
12             android:roundIcon="@mipmap/ic_launcher_round"
13             android:supportsRtl="true"
14             android:theme="@style/Theme.Camera">
15                 <activity
16                     android:name=".MainActivity"
17                     android:exported="true">
18                     <intent-filter>
19                         <action android:name="android.intent.action.MAIN" />
20
21                         <category android:name="android.intent.category.LAUNCHER" />
22                     </intent-filter>
23                 </activity>
24             </application>
25
26     </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

```
17  
20 ></> public class MainActivity extends AppCompatActivity {  
21  
22     2 usages  
23     ImageView image;  
24     2 usages  
25     Button click_picture;  
26  
27     2 usages  
28     @Override  
29     public void onCreate(Bundle savedInstanceState) {  
30         super.onCreate(savedInstanceState);  
31         setContentView(R.layout.activity_main);  
32  
33         image = findViewById(R.id.image);  
34         click_picture = findViewById(R.id.picture);  
35  
36         cameraLauncher = registerForActivityResult(  
37             new ActivityResultContracts.StartActivityForResult(),  
38             ActivityResult result -> {  
39                 if (result.getResultCode() == RESULT_OK && result.getData() != null) {  
40                     Bitmap picture = (Bitmap) result.getData().getExtras().get("data");  
41                     image.setImageBitmap(picture);  
42                 }  
43             }  
44         );  
45  
46         click_picture.setOnClickListener( View view -> {  
47             if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.CAMERA) == PackageManager.PERMISSION_GRANTED) {  
48                 openCamera();  
49             } else {  
50                 requestPermissionLauncher.launch(Manifest.permission.CAMERA);  
51             }  
52         });  
53     }  
54 }
```

MainActivity.java cont.

```
2 usages
54     private void openCamera() {
55         Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
56         cameraLauncher.launch(cameraIntent);
57     }
58
59     1 usage
60     private final ActivityResultLauncher<String> requestPermissionLauncher =
61         registerForActivityResult(new ActivityResultContracts.RequestPermission(), Boolean isGranted -> {
62             if (isGranted) {
63                 openCamera();
64             } else {
65                 Log.e( tag: "Permission", msg: "Camera permission denied");
66             }
67         });
68 }
```

Build:grade (app)

```
1  plugins {
2      alias(libs.plugins.android.application)
3  }
4
5  android {
6      namespace 'rahul.cse_162_2l.camera'
7      compileSdk 36
8
9      defaultConfig { DefaultConfig it ->
10         applicationId "rahul.cse_162_2l.camera"
11         minSdk 24
12         targetSdk 36
13         versionCode 1
14         versionName "1.0"
15
16         testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
17     }
18
19     buildTypes { NamedDomainObjectContainer<BuildType> it ->
20         release {
21             minifyEnabled false
22             proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
23         }
24     }
25     compileOptions { CompileOptions it ->
26         sourceCompatibility JavaVersion.VERSION_11
27         targetCompatibility JavaVersion.VERSION_11
28     }
29 }
30
31 dependencies {
32
33     implementation libs.appcompat
34     implementation libs.material
35     testImplementation libs.junit
36     androidTestImplementation libs.ext.junit
37     androidTestImplementation libs.espresso.core
38 }
```

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" }
```