

توسعه برنامه های

موبایل

جلسه سیزدهم مجازی

بخش سوم

سحر صادقی

استفاده مستقیم از دوربین API ارائه شده در برنامه ی اندروید.

ما از دوربین API برای کامل کردن برنامه ی دوربین استفاده خواهیم کرد.

ابتدا نیاز خواهید داشت تا با استفاده از روش استاتیک ارائه شده به وسیله api مقداردهی آبجکت دوربین را آغاز کنید. ترکیب آن به شکل زیر می باشد.

[?](#)

```
1 Camera object = null;
2 object = Camera.open();
```

علاوه بر عملکرد بالا، عملکردهای دیگری توسط گروه camera ارائه می شوند که در زیر بیان شده اند.

روشها و توضیحات

مراحل

`getCameraInfo(int cameraId, Camera.CameraInfo cameraInfo)` اطلاعات مربوط به یک دوربین خاص را باز می گرداند.

1

`getNumberOfCameras()` عدد صحیح که دوربین های حاضر روی دستگاه را تعریف می کند.

2

`lock()` برای قفل کردن دوربین استفاده می شود، بنابراین هیچ برنامه ی دیگری نمی تواند به آن دسترسی داشته باشد.

3

`release()` برای آزاد کردن قفل دوربین استفاده می شود، بنابراین دیگر برنامه ها می توانند به آن دسترسی داشته باشند.

4

`open(int cameraId)` برای باز کردن یک دوربین خاص، زمانیکه چند دوربین پشتیبانی می شوند، استفاده می شود.

5

`enableShutterSound(boolean enabled)` برای فعال یا غیر فعال کردن صدای پیش فرض شاتر دوربین در هنگام گرفتن عکس، استفاده می شود.

اکنون نیاز دارید که یک گروه مجزا بسازید و آن را با `SurfaceView` گسترش دهید و اینترفیس `SurfaceHolder` را اجرا کنید. دو گروهی که استفاده می شوند دارای اهداف زیر می باشند.

روشها و توضیحات

مراحل

برای کنترل دوربین و گرفتن تصاویر یا ویدیو از دوربین استفاده می شود.

Camera

این گروه برای نشان دادن یک پیش نمایش زنده از دوربین برای یوزر استفاده می شود.

SurfaceView

شما باید روش پیش نمایش گروه دوربین را برای آغاز پیش نمایش برای یوزر فرا بخوانید.

?

```

1 public class ShowCamera extends SurfaceView implements
  SurfaceHolder.Callback {
2     private Camera theCamera;
3     public void surfaceCreated(SurfaceHolder holder) {
4         theCamera.setPreviewDisplay(holder);
5         theCamera.startPreview();
6     }
7     public void surfaceChanged(SurfaceHolder arg0, int arg1,
  int arg2, int arg3) {
8     }
9     public void surfaceDestroyed(SurfaceHolder arg0) {
10    }
11 }
12

```

علاوه بر پیش نمایش، آپشن های دیگری برای دوربین وجود دارد که می تواند با استفاده از عملکردهای دیگر که توسط `Camera API` ارائه می شوند، تنظیم شود.

`startFaceDetection()` این عملکرد کشف صورت در دوربین را آغاز می کند.

1

`stopFaceDetection()` برای متوقف کردن عملیات کشف صورت که با عملکرد بالا فعال شده، استفاده می شود.

2

`startSmoothZoom(int value)` مقدار صحیح را می گیرد و دوربین به آرامی روی آن مقدار زوم می کند.

3

`stopSmoothZoom()` برای متوقف کردن زوم دوربین استفاده می شود.

4

`stopPreview()` برای متوقف کردن پیش نمایش برای یوزر استفاده می شود.

5

`takePicture(Camera.ShutterCallback shutter, Camera.PictureCallback raw, Camera.PictureCallback jpeg)` برای فعال یا غیرفعال کردن صدای پیش فرض شاتل در هنگام گرفتن عکس استفاده می شود.

6

مثال:

مثال زیر استفاده ی camera API را در برنامه توضیح می دهد.

برای آزمایش با این مثال به یک دستگاه موبایل مجهز شده با آخرین ورژن Android OS نیاز دارید، زیرا مقلد دوربین را پشتیبانی نمی کند.

Description

مراحل

برای ایجاد یک برنامه ی اندروید از Eclipse IDE استفاده خواهید کرد و آن را با عنوان Camera تحت بسته ی `com.example.camera1` نام گذاری کنید. زمان ایجاد این برنامه مطمئن شوید که Target SDK و Compile With در آخرین ورژن Android SDK هستند تا از سطوح بالاتر API استفاده کنید.

1

فایل `src/MainActivity.java` را برای افزودن کد مربوط به دوربین و به دست آوردن مراجع مولفه های XML، تغییر دهید

2

یک فایل جدید `ShowCamera.java` برای گسترش آن با `SurfaceView` و اجرای اینترفیس `SurfaceHolder`، ایجاد کنید.

3

فایل لی اوت XML مربوط به `res/layout/activity_main.xml` را تغییر دهید، اگر لازم است مولفه ی GUI به آن اضافه کنید. در اینجا فقط یک `FrameView`، یک دکمه و یک `ImageView` اضافه می کنیم.

4

`res/values/strings.xml` را تغییر دهید تا مقادیر ثابت لازم را تعریف کنید.

5

`AndroidManifest.xml` را همانطور که در زیر نمایش داده شده، تغییر دهید تا دستورات لازم برای دوربین را اضافه کنید.

6

برنامه را اجرا کنید و یک دستگاه اجرایی اندروید انتخاب کنید و برنامه را روی آن نصب کرده و نتایج را بررسی کنید.

7

در زیر محتوای فایل تغییر یافته ی فعالیت اصلی را مشاهده می کنید.

[?](#)

```
1 src/com.example.camera1/MainActivity.java.  
2 package com.example.camera1;  
3 import android.graphics.Bitmap;  
4 import android.graphics.BitmapFactory;  
5 import android.hardware.Camera;  
6 import android.hardware.Camera.PictureCallback;  
7 import android.os.Bundle;  
import android.app.Activity;
```

```

8         import android.view.Menu;
9         import android.view.View;
10        import android.widget.FrameLayout;
11        import android.widget.ImageView;
12        import android.widget.Toast;
13    public class MainActivity extends Activity {
14        private Camera cameraObject;
15        private ShowCamera showCamera;
16        private ImageView pic;
17        public static Camera isCameraAvailable() {
18            Camera object = null;
19            try {
20                object = Camera.open();
21            }
22            catch (Exception e) {
23            }
24            return object;
25        }
26        private PictureCallback capturedIt = new
27        PictureCallback() {
28            @Override
29            public void onPictureTaken(byte[] data, Camera camera)
30            {
31                Bitmap bitmap =
32                BitmapFactory.decodeByteArray(data, 0, data.length);
33                if(bitmap==null){
34                    Toast.makeText(getApplicationContext(),
35                    "not taken", Toast.LENGTH_SHORT).show();
36                }
37            }
38            else
39            {

```

```

37         Toast.makeText(getApplicationContext(),
38         "taken", Toast.LENGTH_SHORT).show();
39     }
40     cameraObject.release();
41 }
42 };
43
44     @Override
45     protected void onCreate(Bundle savedInstanceState) {
46         super.onCreate(savedInstanceState);
47         setContentView(R.layout.activity_main);
48         pic = (ImageView) findViewById(R.id.imageView1);
49         cameraObject = isCameraAvailable();
50         showCamera = new ShowCamera(this, cameraObject);
51         FrameLayout preview = (FrameLayout)
52         findViewById(R.id.camera_preview);
53         preview.addView(showCamera);
54     }
55
56     public void snapIt(View view) {
57         cameraObject.takePicture(null, null,
58         capturedIt);
59     }
60
61     @Override
62     public boolean onCreateOptionsMenu(Menu menu) {
63         getMenuInflater().inflate(R.menu.main, menu);
64         return true;
65     }
66 }

```

فایل جدید جاوا با نام src/com.example.camera1/ShowCamera.java را ایجاد کنید و کد زیر را به آن اضافه کنید.

[?](#)

```

1     package com.example.camera1;

```

```

2      import java.io.IOException;
3      import android.content.Context;
4      import android.hardware.Camera;
5      import android.view.SurfaceHolder;
6      import android.view.SurfaceView;
7  public class ShowCamera extends SurfaceView implements
  SurfaceHolder.Callback {
8      private SurfaceHolder holdMe;
9      private Camera theCamera;
10     public ShowCamera(Context context, Camera camera) {
11         super(context);
12         theCamera = camera;
13         holdMe = getHolder();
14         holdMe.addCallback(this);
15     }
16     @Override
17     public void surfaceChanged(SurfaceHolder arg0, int
  arg1, int arg2, int arg3) {
18     }
19     @Override
20     public void surfaceCreated(SurfaceHolder holder) {
21         try {
22             theCamera.setPreviewDisplay(holder);
23             theCamera.startPreview();
24         } catch (IOException e) {
25         }
26     }
27     @Override
28     public void surfaceDestroyed(SurfaceHolder arg0) {
29     }
30 }

```


محتوای res/layout/activity_main.xml را تغییر دهید.

?

```
<!--?xml version="1.0" encoding="utf-8" ?-->

<linearlayout
xmlns:android="http://schemas.android.com/apk/res/android"=""
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="horizontal">
1
    <linearlayout =" " android:layout_width="wrap_content"
2 android:layout_height="match_parent" android:layout_weight="0.30"
3 android:orientation="vertical">
4
5        <framelayout =" " android:id="@+id/camera_preview"
        android:layout_width="fill_parent" android:layout_height="199dp">
6
7
8        <button =" " android:id="@+id/button_capture"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" android:onclick="snapIt"
9 android:text="@string/Capture">
10
11        <imageview =" " android:id="@+id/imageView1"
        android:layout_width="fill_parent"
12 android:layout_height="fill_parent" android:scaletype="fitXY"
13 android:src="@drawable/ic_launcher">
14

</imageview></button></framelayout></linearlayout></linearlayout>
```

محتوای res/values/string.xml را تغییر دهید.

?

```
1
2
3
    <!--?xml version="1.0" encoding="utf-8" ?-->
```

```
4
5 <resources>
6     <string name="app_name">Camera1</string>
7     <string name="action_settings">Settings</string>
8     <string name="hello_world">Hello world!</string>
9     <string name="Capture">Capture</string>
10
11 </resources>
```

محتوای AndroidManifest.xml را تغییر دهید و دستورات جدید را به طوری که در زیر می بینید به آن اضافه کنید.

[?](#)

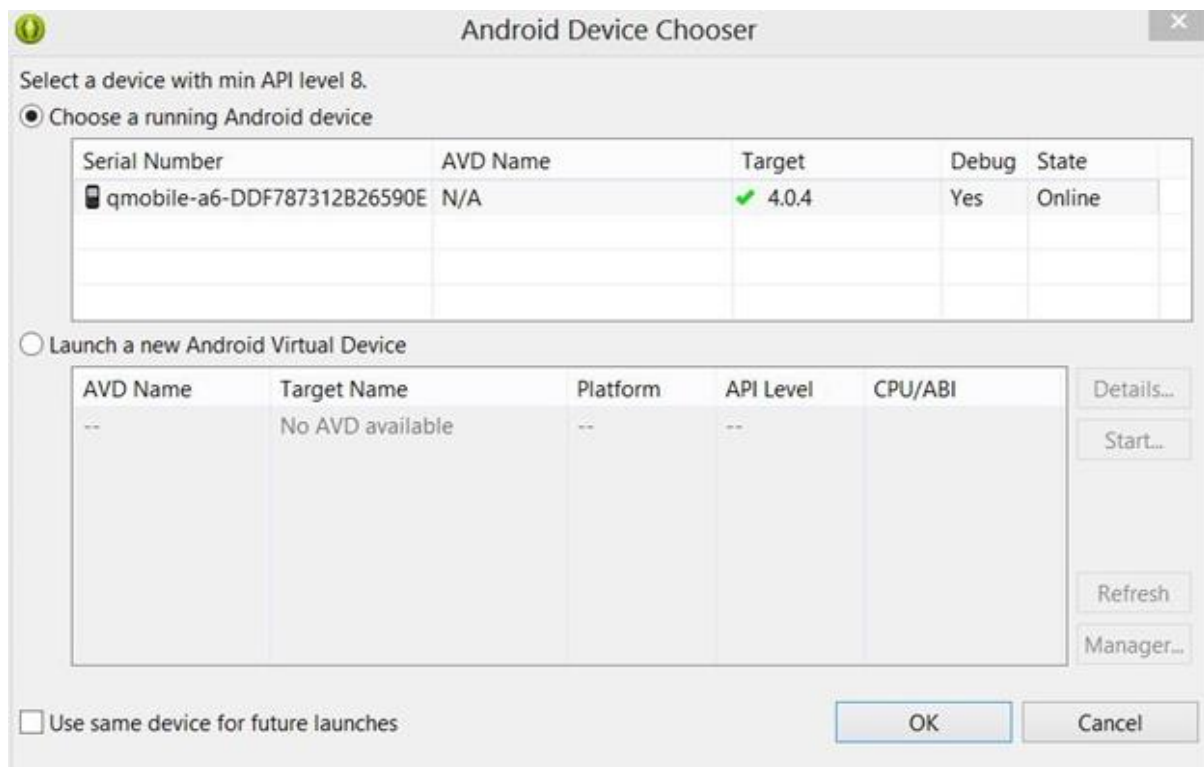
```
1 <!--?xml version="1.0" encoding="utf-8" ?-->
2 <manifest
3     xmlns:android="http://schemas.android.com/apk/res/android"=""
4     package="com.example.camera1" android:versioncode="1"
5     android:versionname="1.0">
6
7     <uses-sdk android:minsdkversion="8"
8     android:targetsdkversion="17">
9
10    <uses-permission
11    android:name="android.permission.CAMERA">
12
13    <uses-feature android:name="android.hardware.camera">
14
15    <uses-feature
16    android:name="android.hardware.camera.autofocus">
17
18    <application android:allowbackup="true"
19    android:icon="@drawable/ic_launcher"
20    android:label="@string/app_name"
21    android:theme="@style/AppTheme">
```

```

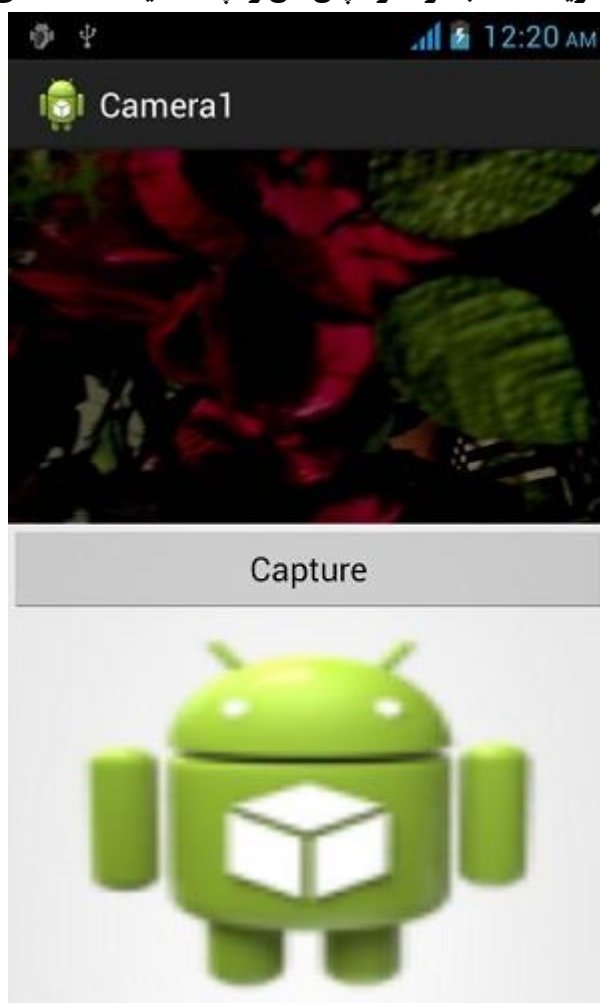
18         <activity =""
19         android:name="com.example.camera1.MainActivity"
20         android:label="@string/app_name">
21
22             <intent-filter>
23
24                 <action
25                 android:name="android.intent.action.MAIN">
26
27                     <category
28                     android:name="android.intent.category.LAUNCHER">
29
30             </category></action></intent-filter>
31
32     </activity>
33
34 </application>
35
36     </uses-feature></uses-feature></uses-
37     permission></uses-sdk></manifest>

```

اجازه بدهید برنامه ی SendSMSDemo را اجرا کنیم. فرض می کنیم که دستگاه موبایل خود را به کامپیوتر متصل کرده اید. برای اجرای برنامه از Eclipse ، یکی از فایل های فعالیت پروژه را باز کنید و روی آیکن Run از توبار کلیک کنید. قبل از شروع برنامه ی شما، Eclipse پنجره ی زیر را برای انتخاب محل اجرای برنامه ی اندروید خود، نمایش خواهد داد.



موبایل خود را به عنوان یک گزینه انتخاب کرده و سپس آن را چک کنید که صفحه ی زیر نمایش خواهد داد.



دروبین شروع به نشان دادن پیش نمایش در قسمت بالایی قاب می کند. روی دکمه ی capture ضربه بزنید. اکنون شما هم می توانید تصویر گرفته شده را ذخیره کنید، در وب آپلود کنید و هم می توانید آن را حذف کنید.

