

## ✓ Introdução à Programação Android

### Tutorial: Câmera

**Objetivo:** Trabalhar com as classes e intents relacionados à câmera do dispositivo Android.

A integração de um aplicativo com a câmera de um dispositivo pode ser feita de duas formas: utilizando-se a aplicação de câmera nativa que já existe no Android; e a acessando diretamente as classes que controlam o hardware da câmera.

### Acessando a aplicação nativa de câmera

- 1) Abra o Eclipse e crie um novo projeto: File->New->Project->Android Application Project.
- 2) Modifique arquivo de layout da tela principal, conforme código abaixo:

```
<?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="vertical" >

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:onClick="takePicture"
        android:src="@android:drawable/ic_menu_camera" />

    <ImageView
        android:id="@+id/imagen"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:scaleType="centerInside" />

</LinearLayout>
```

- 3) Adicione, no arquivo AndroidManifest.xml, as permissões para gravar arquivos no SD Card e para acessar a câmera do dispositivo:

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

- 4) Modifique a classe da *activity* principal, segundo código abaixo:

```
public class MainActivity extends Activity {

    /**
```

```

* Código de requisição para poder identificar quando a activity da câmera é
finalizada
*/
private static final int REQUEST_PICTURE = 1000;

/**
* View onde a foto tirada será exibida
*/
private ImageView imageView;

/**
* Local de armazenamento da foto tirada
*/
private File imageFile;

/**
* Invocado quando a activity é criada
* @see android.app.Activity#onCreate(android.os.Bundle)
*/
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    this.imageView = (ImageView) findViewById(R.id.imagem);

    // Obtém o local onde as fotos são armazenadas na memória externa do
dispositivo
    // geralmente /mnt/sdcard/Pictures
    File picsDir =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);

    // Toast.makeText(getApplicationContext(), picsDir.getAbsolutePath(),
Toast.LENGTH_SHORT).show();

    // Define o local completo onde a foto será armazenada (diretório +
arquivo)
    this.imageFile = new File(picsDir, "foto.jpg");
}

/**
* Método que tira uma foto
* @param v
*/
public void takePicture(View v) {
    // Cria uma intent que será usada para abrir a aplicação nativa de câmera
    Intent i = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

    // Indica na intent o local onde a foto tirada deve ser armazenada
    i.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(imageFile));

    // Abre a aplicação de câmera
    startActivityForResult(i, REQUEST_PICTURE);
}

/**
* Método chamado quando a aplicação nativa da câmera é finalizada
* @see android.app.Activity#onActivityResult(int, int, android.content.Intent)
*/
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // Verifica o código de requisição e se o resultado é OK (outro resultado
indica que

```

```

// o usuário cancelou a tirada da foto)
if (requestCode == REQUEST_PICTURE && resultCode == RESULT_OK) {

    FileInputStream fis = null;

    try {
        try {
            // Cria um FileInputStream para ler a foto tirada pela
            câmera

            fis = new FileInputStream(imageFile);

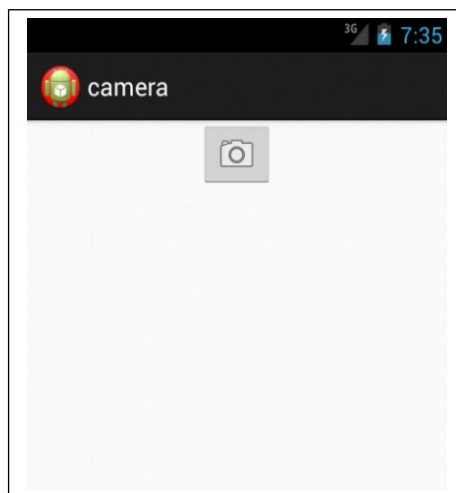
            // Converte a stream em um objeto Bitmap
            Bitmap picture = BitmapFactory.decodeStream(fis);

            // Mostra o arquivo bitmap na view, para que o
            usuário veja a foto tirada

            imageView.setRotation(180);
            imageView.setImageBitmap(picture);
        } finally {
            if (fis != null) {
                fis.close();
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

- 5) Inicie um emulador. Após este ter iniciado, clique com o botão direito na pasta raiz do projeto e escolha “Run as” – “Android Application”. Verifique se a aplicação foi executada com sucesso no emulador.



- 6) Agora conecte o celular no computador e instale o programa no dispositivo. Observe seu funcionamento.

## Controlando a câmera via classes de acesso

1. Abra o Eclipse e crie um novo projeto: File->New->Project->Android Application Project.
2. Modifique arquivo de layout da tela principal, conforme código abaixo:

```
<?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="vertical" >

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:onClick="takePicture"
        android:src="@android:drawable/ic_menu_camera" />

    <SurfaceView
        android:id="@+id/preview"
        android:layout_width="150pt"
        android:layout_height="150pt"
        android:layout_gravity="center_horizontal" />

</LinearLayout>
```

3. Adicione, no arquivo AndroidManifest.xml, as permissões para gravar arquivos no SD Card e para acessar a câmera do dispositivo:

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

4. Modifique a classe da activity principal, segundo código abaixo:

```
public class MainActivity extends Activity implements SurfaceHolder.Callback {

    private Camera camera;

    //Local de armazenamento da foto
    private File imageFile;

    //View para exibição do preview da imagem (obrigatório)
    private SurfaceView surfaceView;

    private int old_orientation=0;

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

        this.setOrientation();

        // Obtém uma instância da câmera
        camera = Camera.open();
        surfaceView = (SurfaceView) findViewById(R.id.preview);
        // Adiciona o callback à SurfaceView,
        surfaceView.getHolder().addCallback(this);
```

```

    // Obtém o local onde as fotos são armazenadas na memória externa do
    dispositivo
    File picsDir =
    Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);

    // Define o local completo onde a foto será armazenada (diretório +
    arquivo)
    this.imageFile = new File(picsDir, "foto.jpg");
}

public void takePicture(View v) {

    // Cria uma classe que implementa a interface PictureCallback, que será
    usada como callback ao tirar uma foto
    PictureCallback jpeg = new PictureCallback() {

        @Override
        public void onPictureTaken(byte[] data, Camera camera) {
            FileOutputStream fos = null;
            try {
                try {
                    // Grava os bytes da imagem no arquivo onde
                    a foto deve ser armazenada
                    fos = new FileOutputStream(imageFile);
                    fos.write(data);
                } finally {
                    if (fos != null) {
                        fos.close();
                    }
                }
            } catch (IOException e) {
                e.printStackTrace();
            }

            // Inicia novamente o preview
            camera.startPreview();
        }
    };

    // Tira uma foto. O callback fornecido é chamado assim que a imagem JPEG
    estiver OK
    camera.takePicture(null, null, jpeg);

    // Mostra uma mensagem indicando que a foto foi tirada
    Toast.makeText(this, "Foto gravada em " + imageFile.getPath(),
    Toast.LENGTH_LONG).show();
}

protected void onDestroy() {
    super.onDestroy();
    if (camera != null) {
        // Libera o uso da câmera
        camera.release();
    }
    setRequestedOrientation(old_orientation);
}

protected void onPause() {
    super.onPause();
    if (camera != null) {
        // Para a visualização do preview
        camera.stopPreview();
    }
}

```

```

    }
}

public void surfaceChanged(SurfaceHolder holder, int format, int width, int
height) {
    // Só atua se a superfície não for nula
    if (holder.getSurface() != null) {
        try {
            // Para o preview
            camera.stopPreview();
        } catch (Exception e) {
        }

        camera.setDisplayOrientation(90);

        try {
            // Faz a ligação do preview da câmera com a SurfaceView
            camera.setPreviewDisplay(holder);
            // Inicia o preview
            camera.startPreview();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

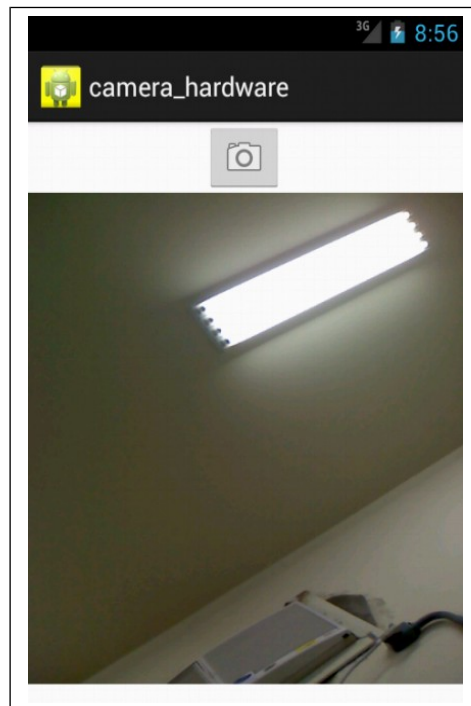
@Override
public void surfaceCreated(SurfaceHolder holder) {
    try {
        // Faz a ligação do preview da câmera com a SurfaceView presente
        camera.setPreviewDisplay(holder);
        // Inicia o preview
        camera.startPreview();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

@Override
public void surfaceDestroyed(SurfaceHolder holder) {
    // Não é necessário fazer nada aqui
}

protected void setOrientation() {
    old_orientation = getRequestedOrientation();
    // only switch the orientation if not in portrait
    if ( old_orientation != ActivityInfo.SCREEN_ORIENTATION_PORTRAIT ) {
        setRequestedOrientation( ActivityInfo.SCREEN_ORIENTATION_PORTRAIT );
    }
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_NOSENSOR);
}
}

```

5. Inicie um emulador. Após este ter iniciado, clique com o botão direito na pasta raiz do projeto e escolha “Run as” – “Android Application”. Verifique se a aplicação foi executada com sucesso no emulador.



6. Agora conecte o celular no computador e instale o programa no dispositivo. Observe seu funcionamento.

## TRABALHO

Faça um aplicativo que acesse a câmera do dispositivo e permita tirar somente um foto por vez ou 3 fotos em sequência, com intervalo de 1 segundo entre elas. DICA: use 2 botões para que o usuário escolha o tipo de foto.