

Desenvolvimento de Aplicativos Android

Aula prática 3 - Utilizando sensores

Os aparelhos de hoje possuem vários tipos de sensores que podem ser acessados por meio da API do Android SDK. Nesse exercício vamos explorar alguns deles:

Sensor de luminosidade

Na classe MainActivity do arquivo MainActivity.java vamos instanciar o sensor, o gerenciador do sensor e a referência para o texto na tela como atributos:

```
public class MainActivity extends Activity implements SensorEventListener {  
  
    private SensorManager sensorManager;  
  
    private Sensor light;  
  
    TextView lightValue;
```

Também precisamos declarar os métodos `onSensorChanged()` e `onAccuracyChanged()` para implementar o acesso ao sensor. No método `onSensorChanged()` vamos detectar as mudanças de valores. Inclua o código abaixo na classe MainActivity:

```
public void onAccuracyChanged(Sensor sensor, int accuracy) {  
  
}  
  
public void onSensorChanged(SensorEvent event) {  
    Sensor sensor = event.sensor;  
    if(sensor.getType() == Sensor.TYPE_LIGHT)  
    {  
        lightValue.setText("Light Intensity: " + event.values[0]);  
    }  
}
```

O método onCreate deve ser modificado para se conectar com o sensor dessa forma:

```
protected void onCreate(Bundle savedInstanceState) {  
  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    lightValue = (TextView) findViewById(R.id.light);  
  
    sensorManager = (SensorManager)  
getSystemService(Context.SENSOR_SERVICE);  
    light = sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);  
    if(light != null)  
    {  
        sensorManager.registerListener(MainActivity.this, light,  
SensorManager.SENSOR_DELAY_NORMAL);  
  
    }else  
    {  
        lightValue.setText("Light sensor not supported");  
    }  
}
```

Quando a comunicação é estabelecida com sucesso, a variável é associada ao sensor e, em caso contrário, ela é definida com **null**. Isso permite que seja feita uma verificação caso o dispositivo tenha ou não o sensor que estamos tentando usar.

Já no arquivo activity_main.xml vamos criar o campo para mostrar o valor que veio do sensor. Substitua o conteúdo do arquivo por esse trecho:

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

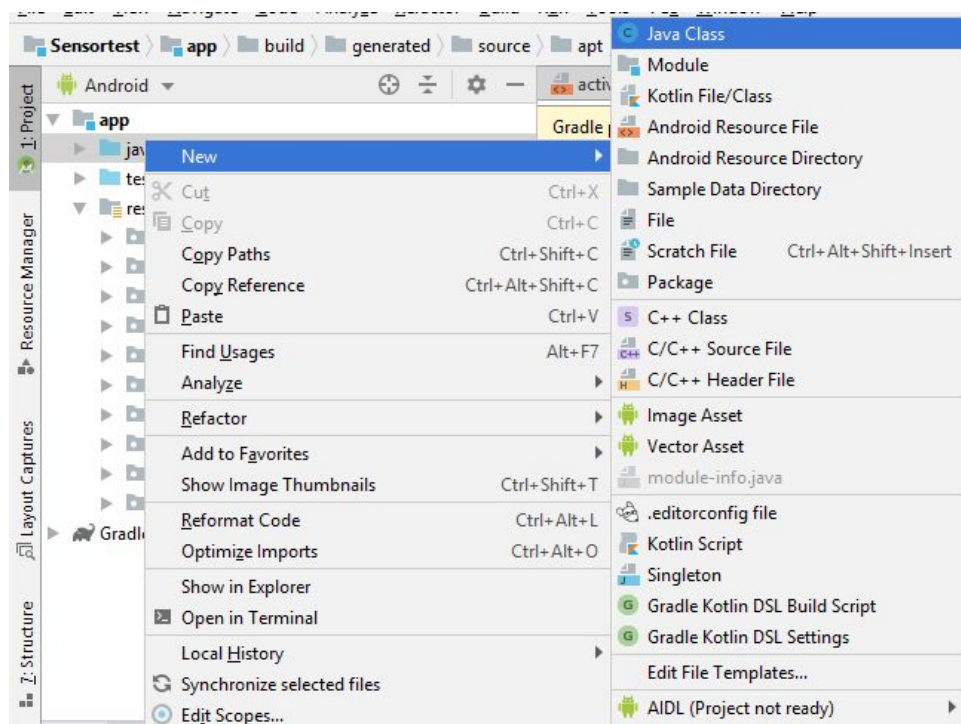
```
<TextView
    android:layout_width="100sp"
    android:layout_height="wrap_content"
    android:id="@+id/light"
/>

</RelativeLayout>
```

Agora ao compilar o aplicativo, o valor da luminosidade deve ser mostrado na tela caso o sensor esteja disponível.

Acesso ao GPS

Para acessar os dados do GPS, vamos criar uma nova classe chamada GPSTracker. Para isso clique com botão direito na pasta java > New > Java Class



Primeiro criaremos um construtor que recebe o contexto e o salva em um atributo context da classe:

```
public class GPSTracker implements LocationListener {

    Context context;
    public GPSTracker(Context c)
    {
        context = c;
    }

}
```

Depois disso, criaremos o método getLocation que vai retornar os valores da localização. Nesse método, vamos usar um **Toast** para mostrar as informações, que é uma caixa de texto que some depois de algum tempo.

Assim como no uso dos sensores, precisamos de um objeto **LocationManager** que vai ser responsável por gerenciar a comunicação com o componente de localização.

```
public Location getLocation(){
    if(ContextCompat.checkSelfPermission(context,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED){
        Toast.makeText(context, "Permission not granted", Toast.LENGTH_LONG).show();
        return null;
    }
    LocationManager lm = (LocationManager)
context.getSystemService(Context.LOCATION_SERVICE);
    boolean isGPSEnabled = lm.isProviderEnabled(LocationManager.GPS_PROVIDER);
    if(isGPSEnabled)
    {
        lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 6000, 10, this);
        Location l = lm.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        return l;
    }else
    {
        Toast.makeText(context, "Please enable GPS", Toast.LENGTH_LONG).show();
    }
    return null;
}
```

Além disso, após o método getLocation(), precisamos declarar os seguintes métodos que são necessários para implementar a classe **LocationListener**:

```

@Override
public void onLocationChanged(Location location) {

}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {

}

@Override
public void onProviderEnabled(String provider) {

}

@Override
public void onProviderDisabled(String provider) {

}

```

Também precisamos dizer para o sistema que o aplicativo vai usar o componente GPS. Para isso, vá em `app\src\main\AndroidManifest.xml` e adicione a linha **<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>** antes da tag `application`, assim:

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sensortest">
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <application

```

Vamos adicionar um botão na main activity para exibir as informações de latitude e longitude. Para isso, adicione o trecho abaixo no arquivo `activity_main.xml`

```

<Button
    android:id="@+id/getGPSBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="GPS"
    android:layout_centerInParent="true"
/>

```

E finalmente, no arquivo MainActivity.java, vamos adicionar a ação do botão. A parte inicial do método onCreate() fica assim:

```
protected void onCreate(Bundle savedInstanceState) {  
  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    getGPSTbn = (Button) findViewById(R.id.getGPSTbn);  
    ActivityCompat.requestPermissions(MainActivity.this, new  
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 123);  
    getGPSTbn.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            GPSTracker g = new GPSTracker(getApplicationContext());  
            Location l = g.getLocation();  
            if(l!=null)  
            {  
                double lat = l.getLatitude();  
                double longi = l.getLongitude();  
                Toast.makeText(getApplicationContext(), "LAT: "+lat + "LONG: " +  
longi, Toast.LENGTH_LONG).show();  
            }  
        }  
    });  
}
```

Agora as informações de latitude e longitude vão aparecer quando o botão for pressionado.