

Android Application Second Session

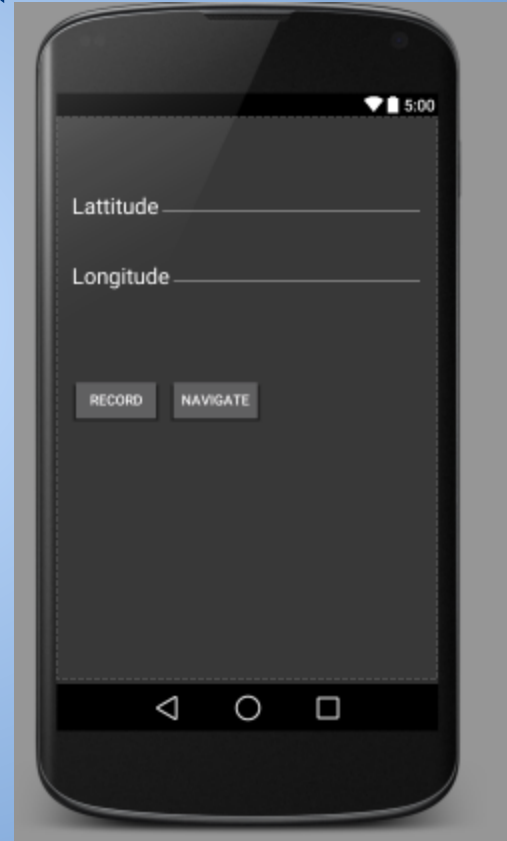
Swathi Matsa

Application use case

- Often times we park our car and walk some distance to do things
- It is very easy to forget exactly where we parked the car
- MyGuide is an application that would remember the location and takes us back to the location when needed

Application Screen (Wireframing)

- We have two text boxes displaying current GPS location (Longitude, Latitude)
- We have a record button and Navigate button



Application Behavior

- The current GPS location would always be automatically updated as the user moves
- When the user clicks record button the application remembers the location
- When the user clicks Navigate the application brings up navigation screen

Step 1

- Create a new project.
- Name the project as *MyGuide*.
- Create two text boxes and two buttons in the `activity_main.xml` file.
- Run the application and test it in the device.

Step 1 continued...

```
<EditText
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:inputType="numberDecimal"  
    android:id="@+id/latitude"  
    android:ems="10"  
    android:layout_marginTop="54dp"  
    android:layout_alignParentStart="true"  
    android:layout_alignParentLeft="true"  
/>
```

```
<EditText
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:inputType="numberDecimal"  
    android:ems="10"  
    android:id="@+id/longitude"  
    android:layout_below="@+id/latitude"  
/>
```

Step 1 continued...

<Button

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Record"  
    android:id="@+id/recordBtn"  
    android:clickable="true"  
    android:layout_centerVertical="true"  
    android:layout_alignEnd="@+id/latitude"  
    android:layout_alignRight="@+id/latitude"/>
```

<Button

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Navigate"  
    android:id="@+id/navigateBtn"  
    android:layout_alignLeft="@+id/recordBtn"  
    android:layout_below="@+id/recordBtn"/>
```

Step 2

- Goal of this step is to display some fixed numbers in the text boxes
- To do this we need to change code in `MainActivity.java` as shown in the next slide

Adding Text

```
public class MainActivity extends ActionBarActivity {
```

```
    EditText latitudeEt;  
    EditText longitudeEt;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);
```

```
        latitudeEt = (EditText) findViewById(R.id.lattitudeEt);  
        longitudeEt = (EditText) findViewById(R.id.longitudeEt);  
        latitudeEt.setText("123");  
        longitudeEt.setText("567");
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
```

Step 2 continued...

```
public class MainActivity extends ActionBarActivity {  
    EditText latitudeEt;  
    EditText longitudeEt;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        latitudeEt = (EditText) findViewById (R.id.latitude);  
        longitudeEt = (EditText) findViewById (R.id.longitude);  
        latitudeEt.setText("123");  
        longitudeEt.setText("567");  
    }  
}
```

What is happening here

- We are declaring two variables of type 'EditText' in the class 'MainActivity'
- onCreate method would be executed when the application opens. In this method we are getting references to the text boxes and setting some numbers.
- findViewById is an android method that returns the view object for a given Id

Step 3

- Goal of this step is to change the numbers in GPS text boxes when the user clicks “Record button”
- To this we need to change code in MainActivity.java as shown in the next slide

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener{
```

```
    EditText latitudeEt;  
    EditText longitudeEt;
```

```
    Button recordBtn;  
    Button navigateBtn;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);
```

```
        latitudeEt = (EditText) findViewById(R.id.latitudeEt);  
        longitudeEt = (EditText) findViewById(R.id.longitudeEt);  
        latitudeEt.setText("123");  
        longitudeEt.setText("567");
```

```
        recordBtn = (Button) findViewById(R.id.recordBtn);  
        navigateBtn = (Button) findViewById(R.id.navigateBtn);  
        recordBtn.setOnClickListener(this);  
        navigateBtn.setOnClickListener(this);
```

```
    @Override
```

```
    public void onClick(View v) {
```

```
    }
```

Step 3 continued...

```
public class MainActivity extends ActionBarActivity implements View.OnClickListener {  
    EditText latitudeEt;  
    EditText longitudeEt;  
  
    Button recordBtn;  
    Button navigateBtn;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        latitudeEt = (EditText) findViewById (R.id.latitude);  
        longitudeEt = (EditText) findViewById (R.id.longitude);  
        latitudeEt.setText("123");  
        longitudeEt.setText("567");  
  
        recordBtn = (Button) findViewById(R.id.recordBtn);  
        navigateBtn = (Button) findViewById(R.id.navigateBtn);  
        recordBtn.setOnClickListener(this);  
        navigateBtn.setOnClickListener(this);  
    }  
}
```

What is happening here

- Like we got the reference for 'EditText' variables we get reference to Buttons
- We need to implement 'View.OnClickListener' for us to be able to tap into button clicks.
- We call 'setOnClickListener' method on the buttons so that we can get notified

- We need to implement 'onClick' method.
- Skeleton of this method is generated by the IDE
- 'implements' is a way to implement interfaces in java. These are similar to abstract methods in other programming languages
- Now lets add logic to 'onClick' method as shown in the next slide


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

    latitudeEt = (EditText) findViewById(R.id.LattitudeEt);
    longitudeEt = (EditText) findViewById(R.id.LongitudeEt);
    latitudeEt.setText("123");
    longitudeEt.setText("567");

    recordBtn = (Button) findViewById(R.id.recordBtn);
    navigateBtn = (Button) findViewById(R.id.navigateBtn);
    recordBtn.setOnClickListener(this);
    navigateBtn.setOnClickListener(this);
}
```

```
@Override
public void onClick(View v) {

    if (v.getId() == R.id.recordBtn) {
        latitudeEt.setText("400");
        longitudeEt.setText("500");
    }
    else if (v.getId() == R.id.navigateBtn) {

    }
}
```

Step 3 continued ...

```
public void onClick(View v) {  
    if (v.getId() == R.id.recordBtn) {  
        latitudeEt.setText("400");  
        longitudeEt.setText("500");  
    }  
    else if (v.getId() == R.id.navigateBtn) {  
  
    }  
}
```

Step 4

- Goal of this step is to display actual GPS location in the Text boxes.
- The location text boxes should keep changing when the user moves from one location to another.
- Follow the code in the next few slides

```
public class MainActivity extends ActionBarActivity implements View.OnClickListener,
    LocationListener {

    EditText latitudeEt;
    EditText longitudeEt;

    Button recordBtn;
    Button navigateBtn;

    LocationManager locationManager;

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

        latitudeEt = (EditText) findViewById(R.id.latitudeEt);
        longitudeEt = (EditText) findViewById(R.id.longitudeEt);
        latitudeEt.setText("123");
        longitudeEt.setText("567");
        recordBtn = (Button) findViewById(R.id.recordBtn);
        navigateBtn = (Button) findViewById(R.id.navigateBtn);
        recordBtn.setOnClickListener(this);
        navigateBtn.setOnClickListener(this);

        locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
        locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 60, 0, this);
    }
}
```

Step 4 continued ...

```
public class MainActivity extends ActionBarActivity implements View.OnClickListener, LocationListener {
    EditText latitudeEt;
    EditText longitudeEt;
    Button recordBtn;
    Button navigateBtn;
    LocationManager locationManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        latitudeEt = (EditText) findViewById(R.id.latitude);
        longitudeEt = (EditText) findViewById(R.id.longitude);
        latitudeEt.setText("123");
        longitudeEt.setText("567");

        recordBtn = (Button) findViewById(R.id.recordBtn);
        navigateBtn = (Button) findViewById(R.id.navigateBtn);
        recordBtn.setOnClickListener(this);
        navigateBtn.setOnClickListener(this);

        locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
        locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 60, 0, this);
    }
}
```

- We are implementing an interface defined by 'LocationListener'
- We are declaring a variable of type 'LocationManager'
- This class 'LocationManager' is defined by the Android SDK
- We need to get reference to the location manager using 'getSystemService' API
- Using 'requestLocationUpdates' API we are requesting updates for ever change

- Since we are implementing ‘LocationListener’ the required methods are auto generate as shown here

```
@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) {

}
```

- We need to implement the ‘onLocationChanged’ method to get the location and set it on the text boxes. The code below does that

```
@Override
public void onLocationChanged(Location location) {
    Log.e("GPS APP", "The user has moved 10 meters");
    latitude = location.getLatitude();
    longitude = location.getLongitude();
    latitudeEt.setText(String.valueOf(latitude));
    longitudeEt.setText(String.valueOf(longitude));
}
```


Step 4 continued ...

```
@Override
public void onLocationChanged(Location location) {
    Log.e("GPS APP", "The user has moved 10 meters");
    latitude = location.getLatitude();
    longitude = location.getLongitude();
    latitudeEt.setText(String.valueOf(latitude));
    longitudeEt.setText(String.valueOf(longitude));
}
```

One Important thing to remember

- Whenever we need to use system services we need to explicitly request permission from Android
- This can be done by placing permissions in **AndroidManifest.xml**. Add the below line just before the last line

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

Step 5: Stitching it all together

- Goal of this step is to store the actual GPS coordinates when the user presses Record button
- Change the code in the method onClick as shown in the next slide

```
public class MainActivity extends ActionBarActivity implements View.OnClickListener,
                                                                    LocationListener {

    EditText latitudeEt;
    EditText longitudeEt;
    Button recordBtn;
    Button navigateBtn;
    LocationManager locationManager;
    Double latitude;
    Double longitude;
    Double storedLatitude;
    Double storedLongitude;

    @Override
    protected void onCreate(Bundle savedInstanceState) {...}

    @Override
    public void onClick(View v) {

        if (v.getId() == R.id.recordBtn) {

            storedLatitude = latitude;
            storedLongitude = longitude;

        }

        else if (v.getId() == R.id.navigateBtn) {

        }

    }
}
```

Step 5 continued ...

```
public class MainActivity extends ActionBarActivity implements View.OnClickListener, LocationListener {  
    EditText latitudeEt;  
    EditText longitudeEt;  
  
    Button recordBtn;  
    Button navigateBtn;  
    LocationManager locationManager;  
    Double latitude;  
    Double longitude;  
    Double storedLatitude;  
    Double storedLongitude;  
  
    public void onClick(View v) {  
        if (v.getId() == R.id.recordBtn) {  
            storedLatitude = latitude;  
            storedLongitude = longitude;  
        }  
        else if (v.getId() == R.id.navigateBtn) {  
  
        }  
    }  
}
```

- We just store the location when the user presses the Record button.
- We also got rid of showing some meaningless coordinates in the text boxes.
- Now we have all the ingredients except the navigation.

Step 6: Navigation

- Goal of this final step is to take the user to navigation application when the user presses Navigate button. We do not write our own Navigation screens.
- It is very common in Android programming to invoke standard system applications to achieve something
- Make the following changes in the code

```
@Override
public void onClick(View v) {

    if (v.getId() == R.id.recordBtn) {

        storedLatitude = latitude;
        storedLongitude = longitude;

    }
    else if (v.getId() == R.id.navigateBtn) {

        Intent intent;
        intent = new Intent(android.content.Intent.ACTION_VIEW,
            Uri.parse("google.navigation:q=" + String.valueOf(storedLatitude) + "," + String.valueOf(storedLongitude)));
        startActivity(intent);

    }
}
```


Step 6 continued ...

@Override

```
public void onClick(View v) {  
    if (v.getId() == R.id.recordBtn) {  
        storedLatitude = latitude;  
        storedLongitude = longitude;  
    }  
    else if (v.getId() == R.id.navigateBtn) {  
        Intent intent;  
        Log.e("I am in navigate condition.", "Do something");  
        intent = new Intent(android.content.Intent.ACTION_VIEW,  
            Uri.parse("google.navigation:q=" + String.valueOf(storedLatitude) + "," + String.valueOf(storedLongitude)));  
        startActivity(intent);  
    }  
}
```

Intents

- Intent is a means to signal to android to invoke another application
- We create an intent to invoke Navigation. The arguments to this intent is an URL that contains the longitude and latitude of the location we need to go.

Thank You