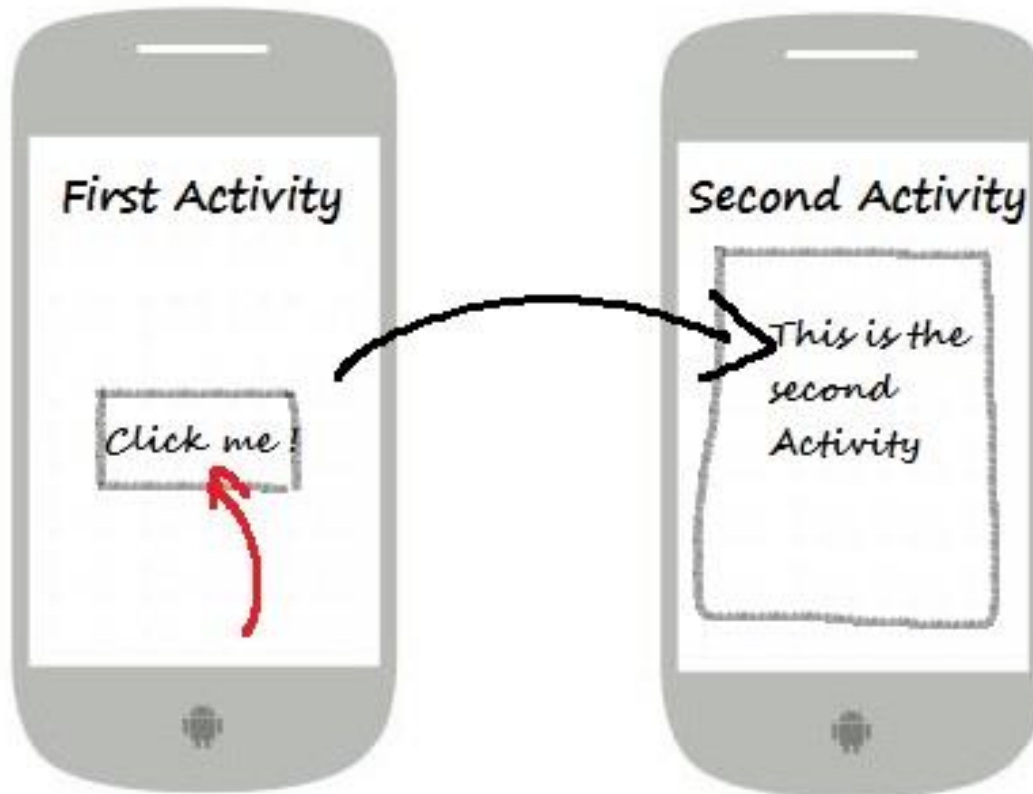
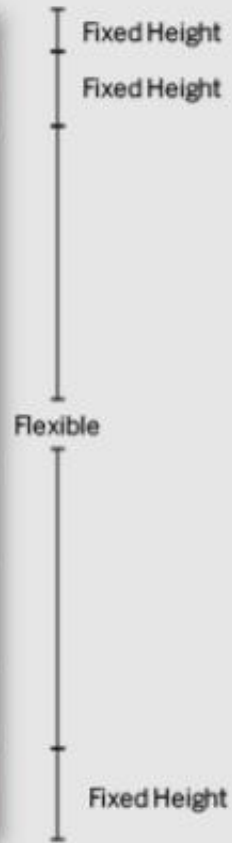
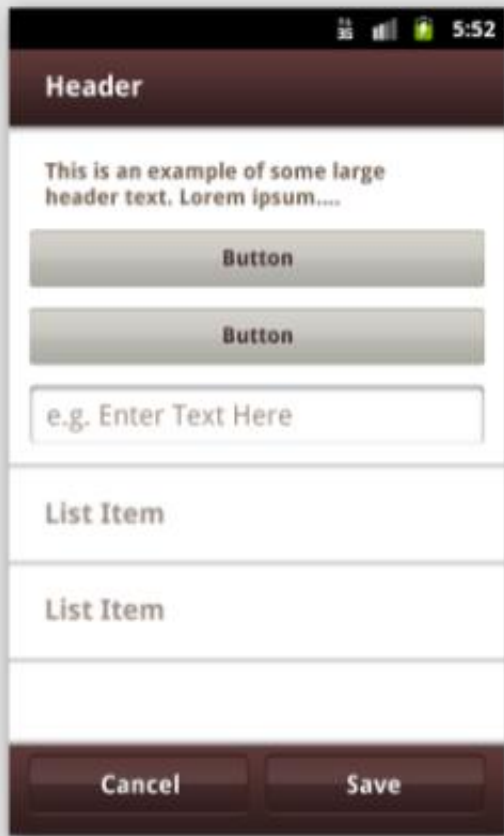


Android Programming 3

Adding new Activity





Activity: Layout + Java code

Presentation

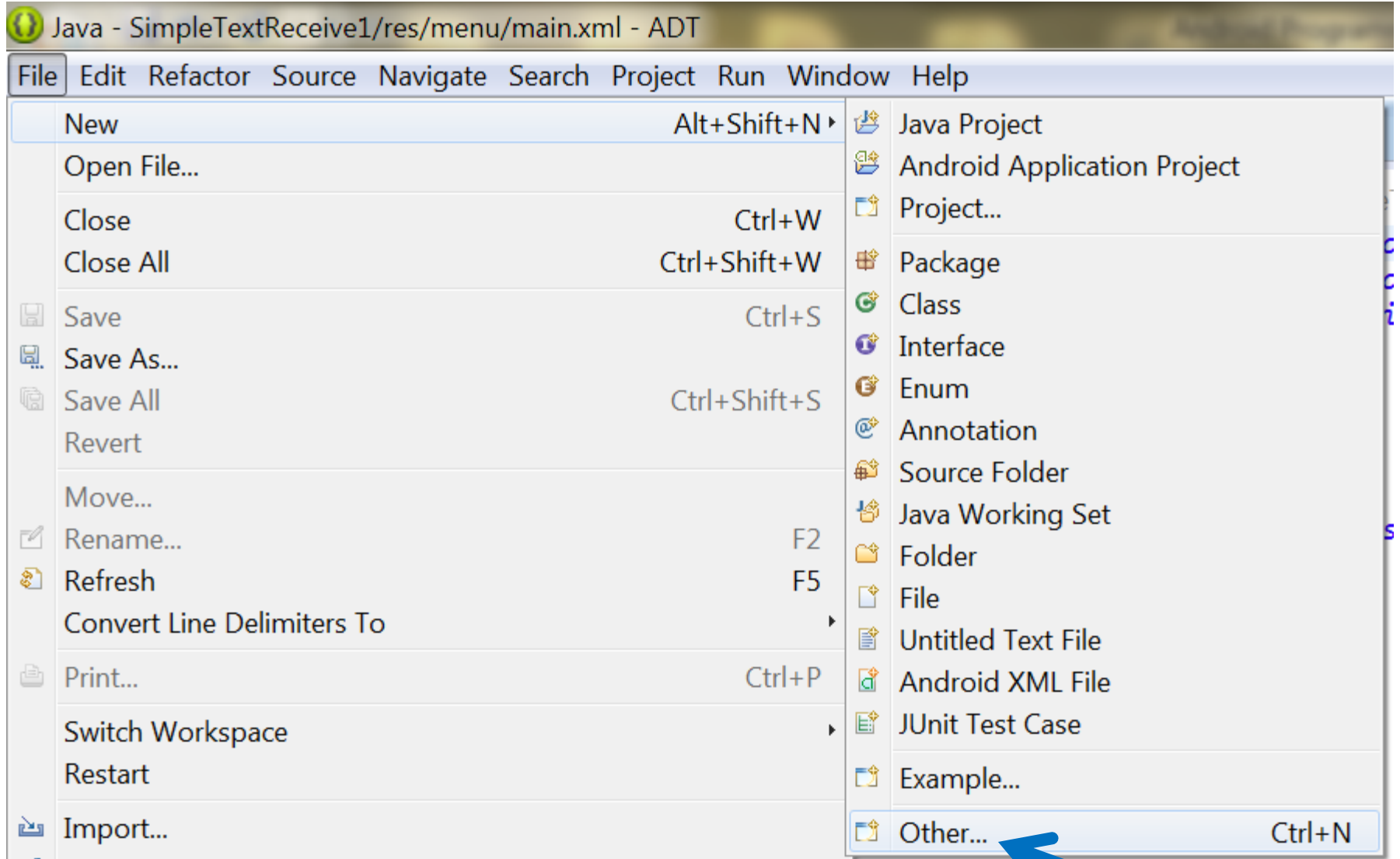
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Hello World"
    android:gravity="center"
    android:background="#cccccc"
    />
<ImageView
    android:src="@drawable/ic_launcher"
    android:id="@+id/imageView1"
    android:layout_height="wrap_content"
    android:scaleType="centerCrop"
    android:layout_width="match_parent"
    android:layout_height="11"
    android:background="#ff00ff"
    />
```

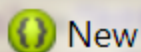
**UI Layout Definition
(XML File)**

Functionality

```
public class ImageActivity extends Activity
{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Java Code





New

Select a wizard

Create an Android Activity



Wizards:

- Android
 - Android Activity
 - Android Application Project
 - Android Icon Set
 - Android Object
 - Android Project from Existing Code



< Back

Next >

Finish

Cancel

New Activity

Blank Activity

Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.



Project: test2

Activity Name[Ⓢ] Activity2

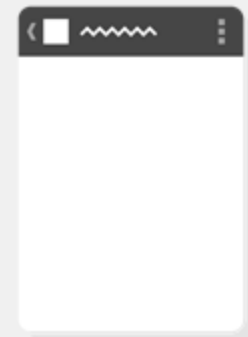
Layout Name[Ⓢ] activity_activity2

Fragment Layout Name[Ⓢ] fragment_activity2

Title[Ⓢ] Activity2

Launcher Activity

Hierarchical Parent[Ⓢ] Optional



< Back

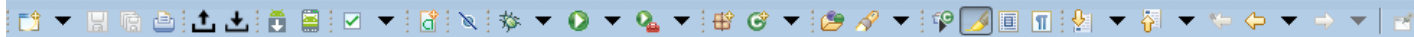
Next >

Finish

Cancel

Java - test2/src/com/example/test2/Activity2.java - ADT

File Edit Refactor Source Navigate Search Project Run Window Help



Package Explorer

- > events1
- > events2
- > events3
- > jsonparser2
- > Location1
- > ShowLocation2
- > SimpleTextReceive1
- > SimpleTextShare1
- test2
 - src
 - com.example.test2
 - Activity2.java
 - MainActivity.java

Activity2.java

```
package com.example.test2;

import android.support.v7.app.AppCompatActivity;

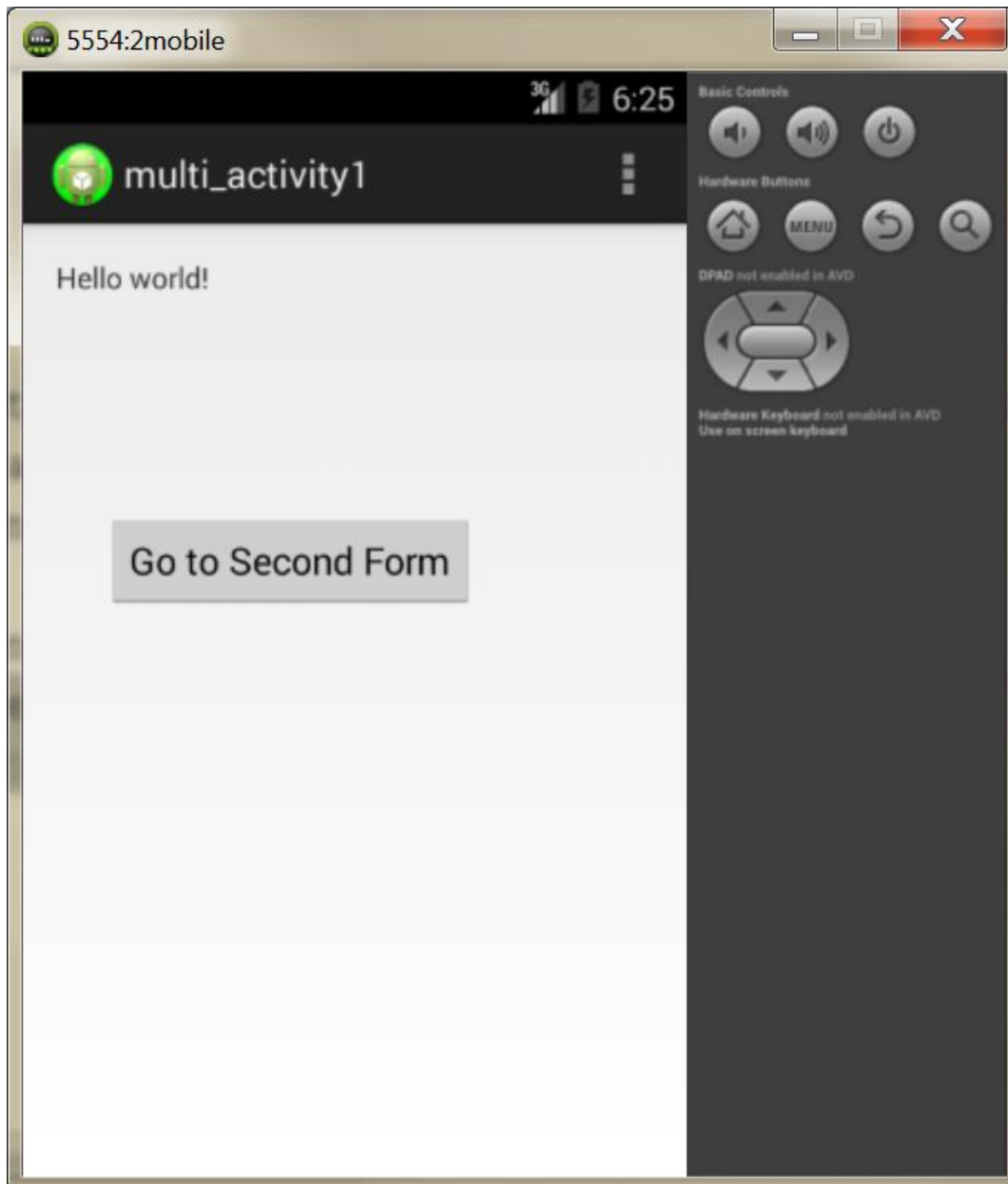
public class Activity2 extends AppCompatActivity {

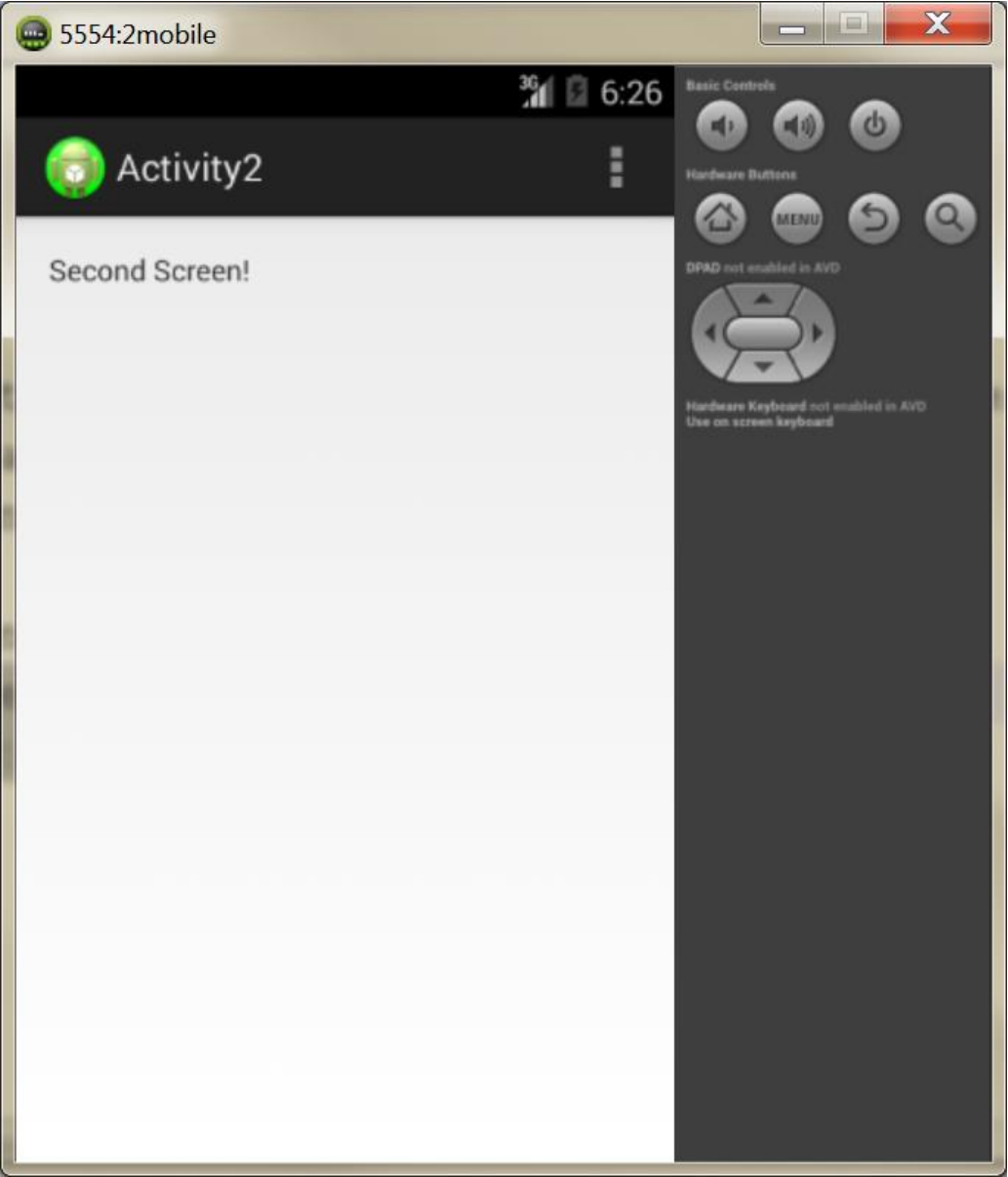
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_activity2);

        if (savedInstanceState == null) {
            getSupportFragmentManager().beginTransaction()
                .add(R.id.container, new PlaceholderFragment()).commit();
        }
    }
}
```

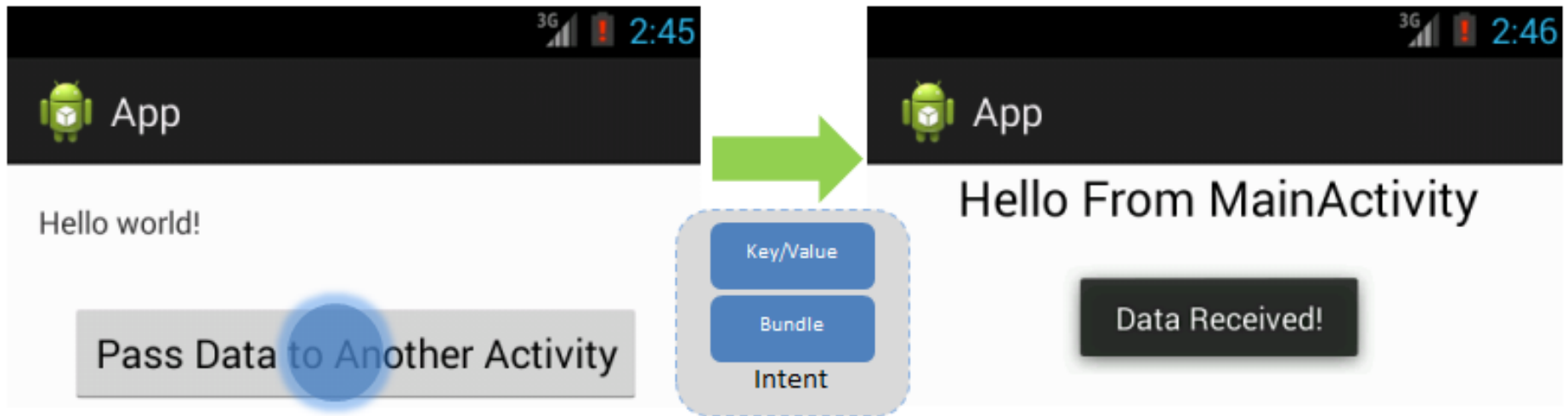

Java snippet to start new Activity

```
public void secondActivity(View view) {  
    Intent intent = new Intent(this, Activity2.class);  
    startActivity(intent);  
}
```





Start new Activity and pass value!

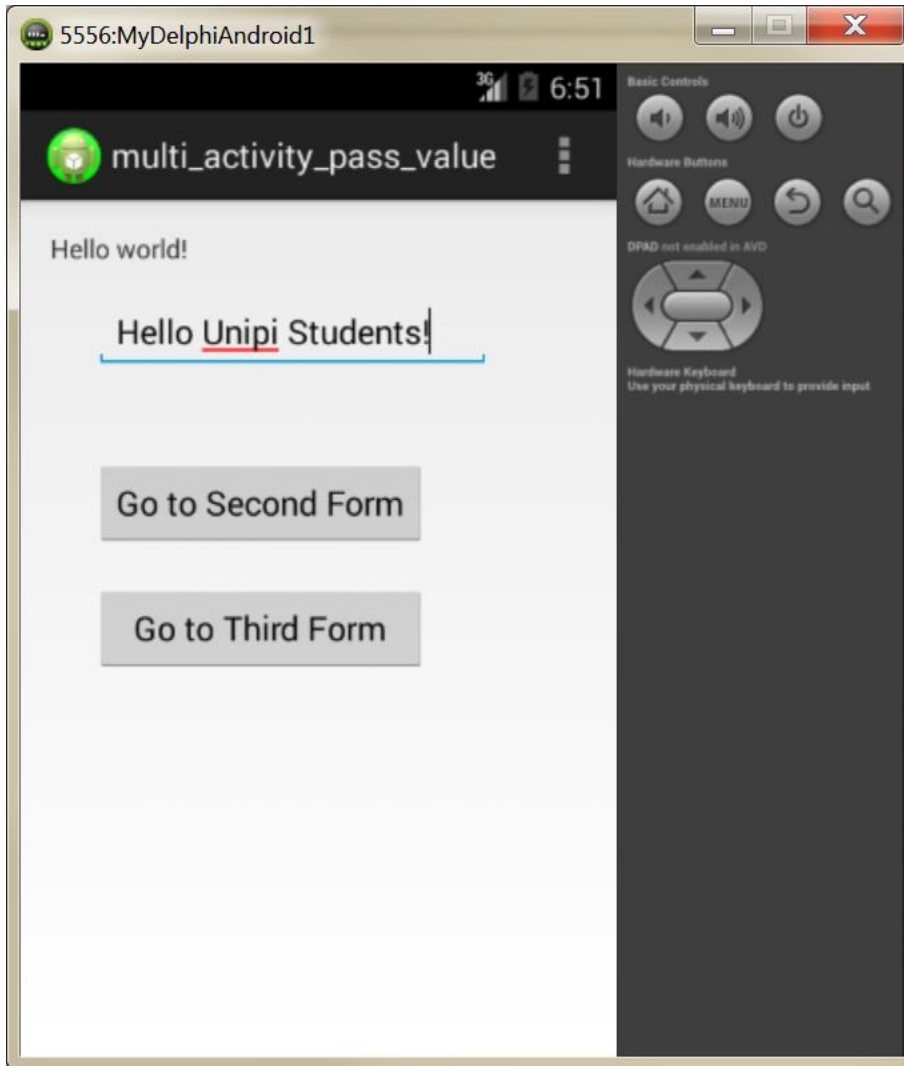


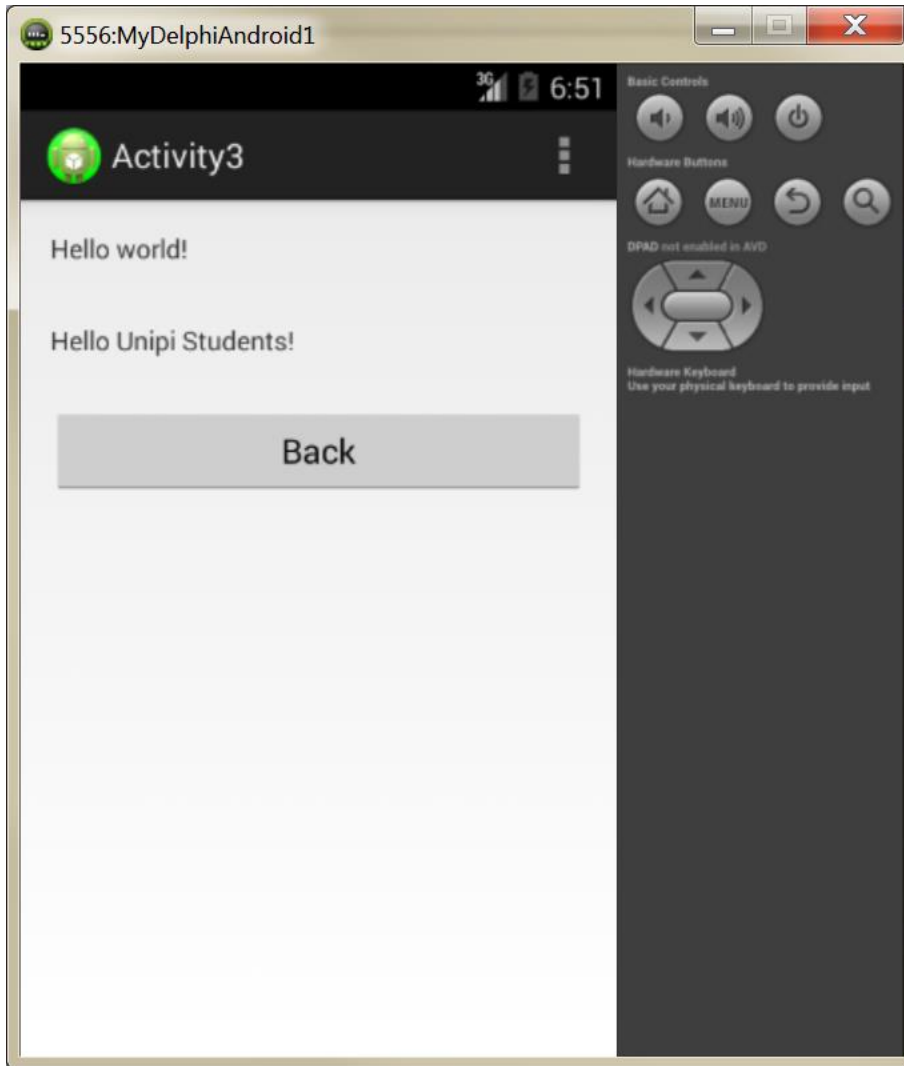
Send a **bundle** with data

```
public void thirdActivity(View view) {  
    EditText myedittext=(EditText)findViewById(R.id.editText1);  
    Intent intent = new Intent(this, Activity3.class);  
    intent.putExtra("str1", myedittext.getText().toString());  
    startActivity(intent);  
}
```

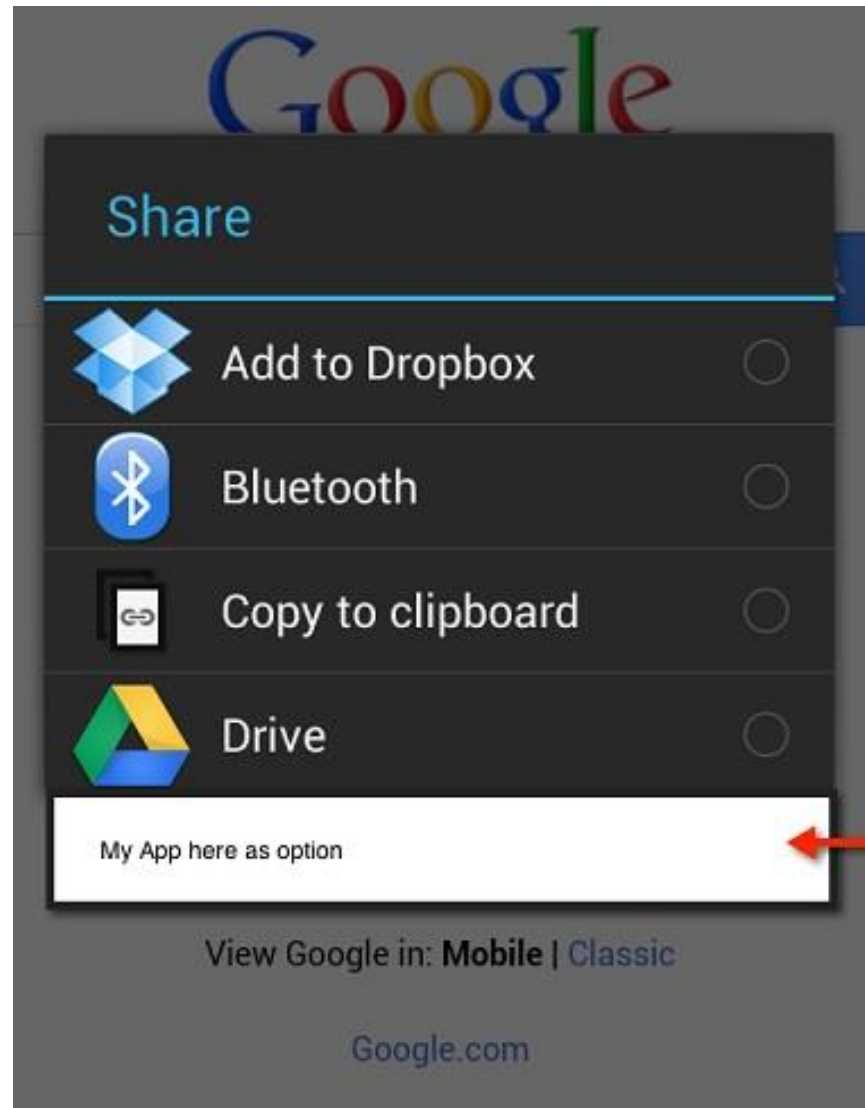
Receive the bundle from the new Activity

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_activity3,
        container, false);
    tv1=(TextView) rootView.findViewById(R.id.textView11);
    Bundle b = getActivity().getIntent().getExtras();
    tv1.setText(b.getString("str1"));
    return rootView;
}
```





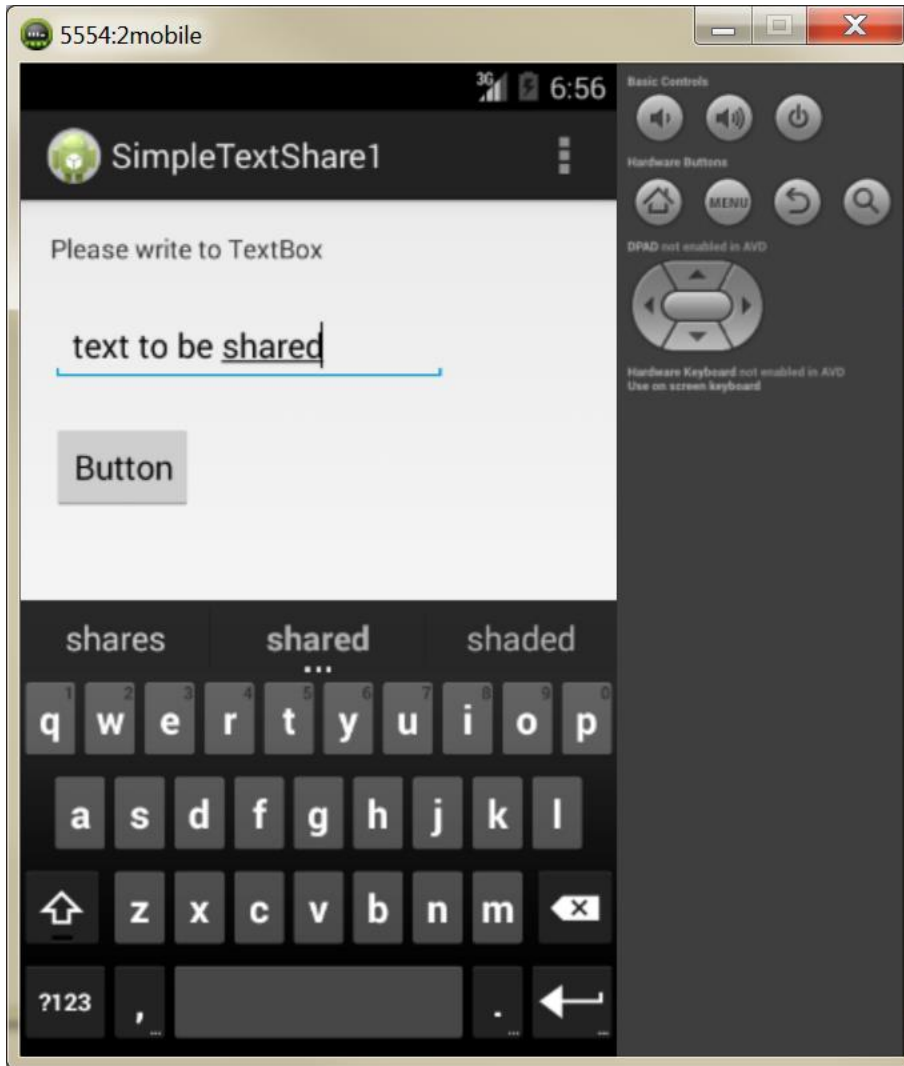
Sharing with other Android Apps (sorry iOS)

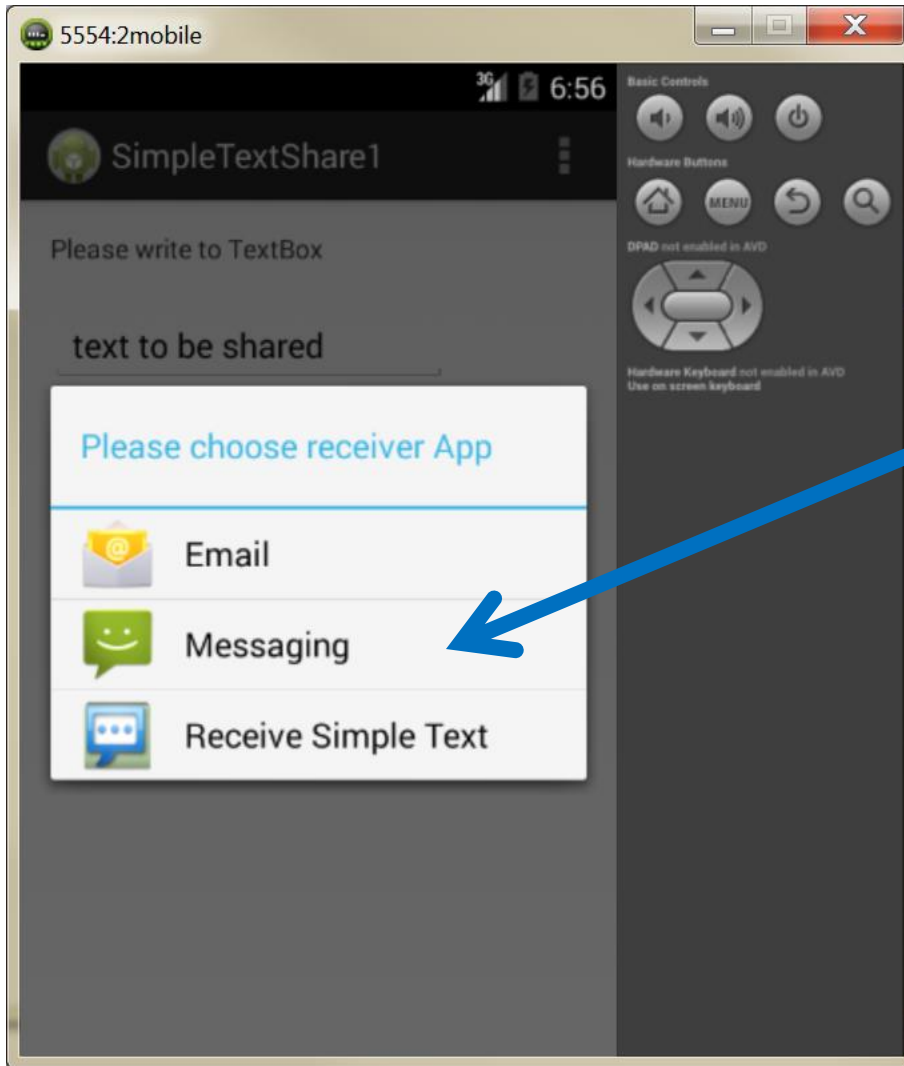


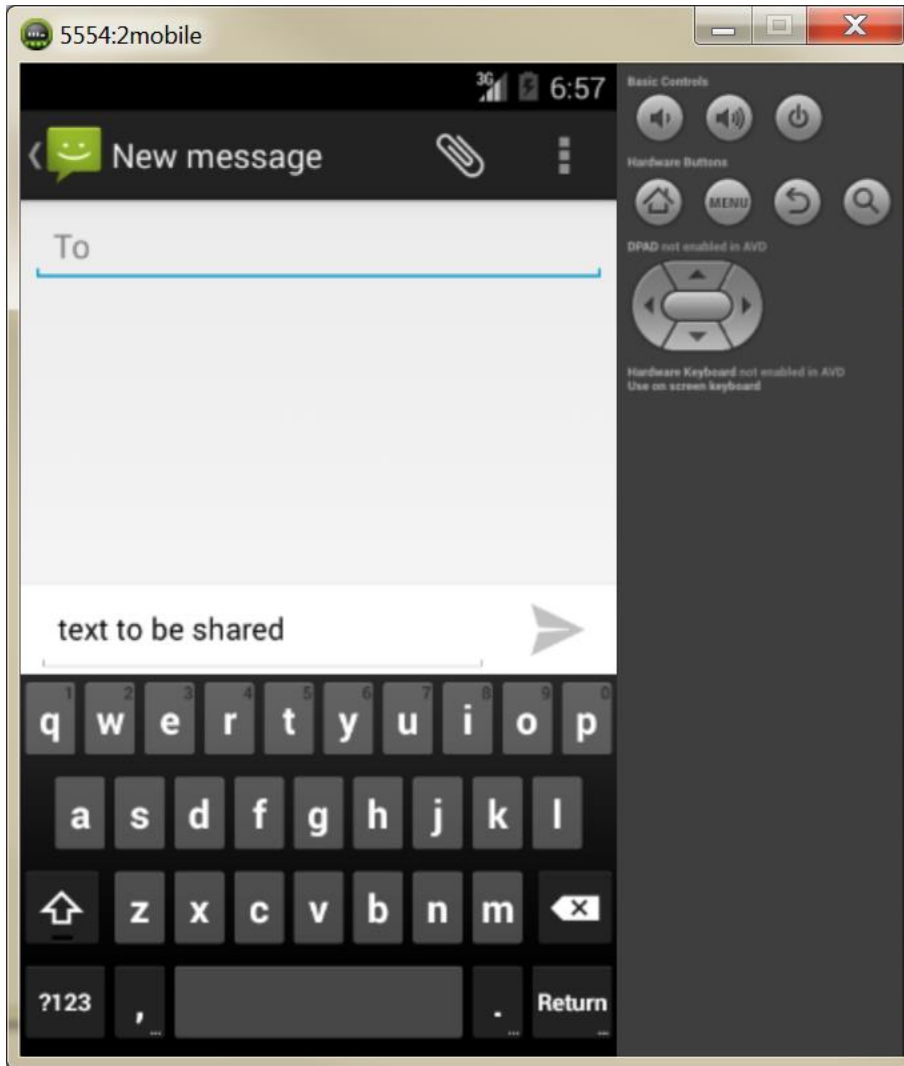
Share Text with other Apps

Source code

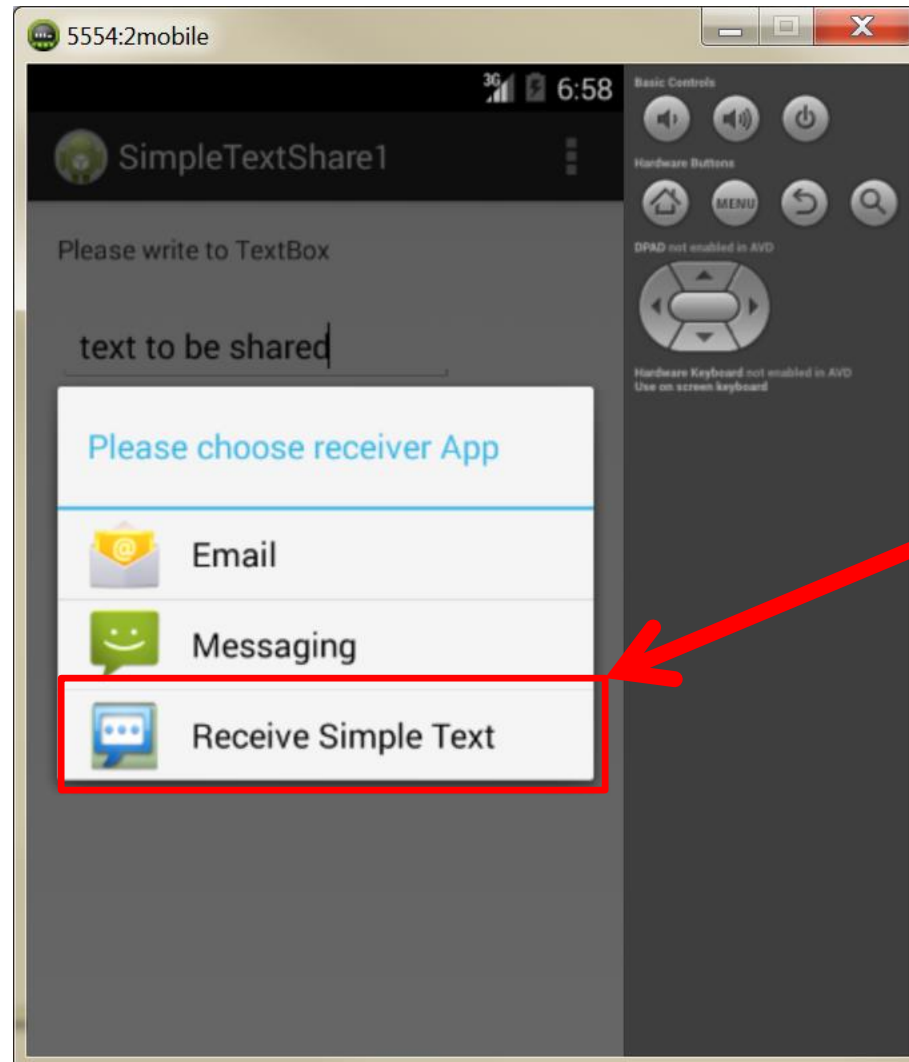
```
public void sendText(View v){
    if (!edit1.getText().toString().matches("")) {
        Intent sendIntent = new Intent();
        sendIntent.setAction(Intent.ACTION_SEND);
        sendIntent.putExtra(Intent.EXTRA_TEXT, edit1.getText().toString());
        sendIntent.setType("text/plain");
        startActivity(Intent.createChooser(sendIntent, getResources().getText(R.string.send_to)));
    }
    else
    {
        Toast.makeText(this, "Please enter some text", Toast.LENGTH_SHORT).show();
    }
}
```







Create an application that can receive data from other apps?



1st Step

Create an intent filter in Manifest

```
<intent-filter>  
    <action android:name="android.intent.action.SEND" />  
    <category android:name="android.intent.category.DEFAULT" />  
    <data android:mimeType="text/plain" />  
</intent-filter>
```

2nd Step

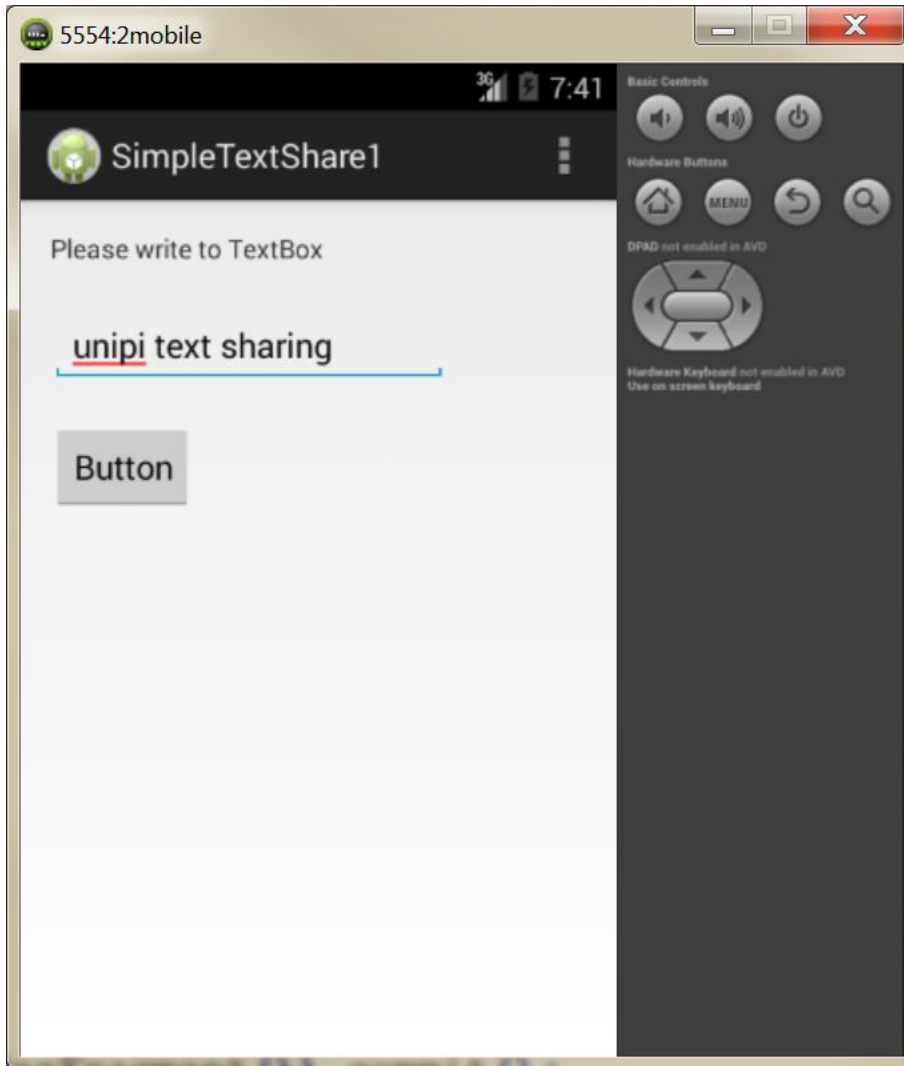
Capture intent in “onCreate” function

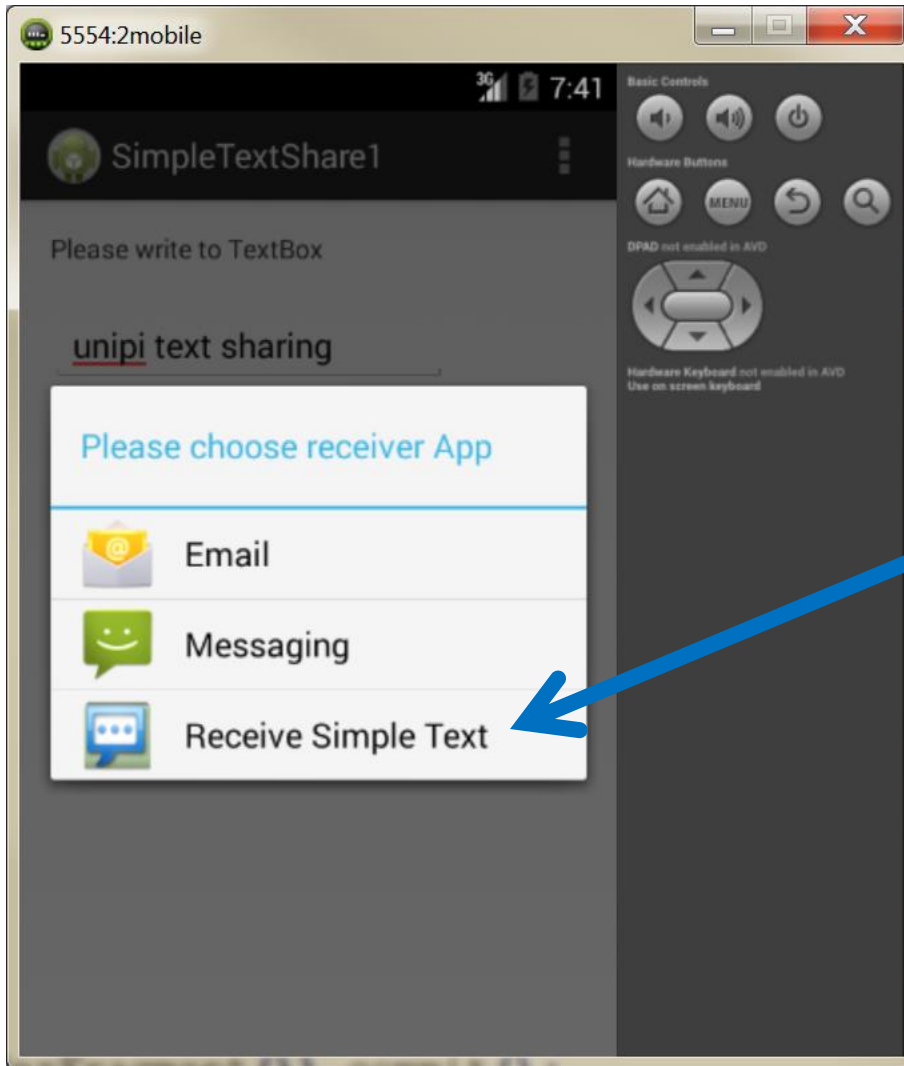
```
protected void onCreate(Bundle savedInstanceState) {  
    Intent intent = getIntent();  
    String action = intent.getAction();  
    String type = intent.getType();  
  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    if (savedInstanceState == null) {  
        getSupportFragmentManager().beginTransaction()  
            .add(R.id.container, new PlaceholderFragment()).commit();  
    }  
  
    if (Intent.ACTION_SEND.equals(action) && type != null) {  
        if ("text/plain".equals(type)) {  
            handleSendText(intent); // Handle text being sent  
        }  
    }  
}
```

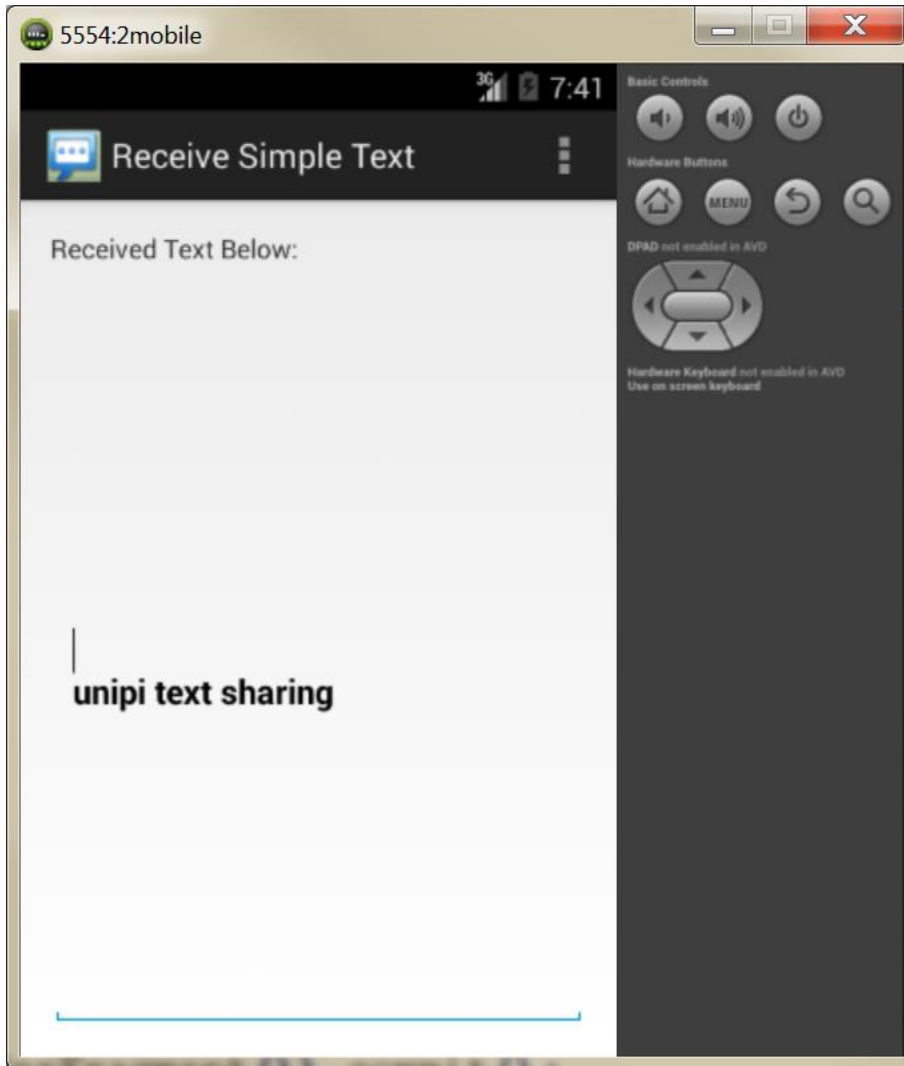

3rd Step

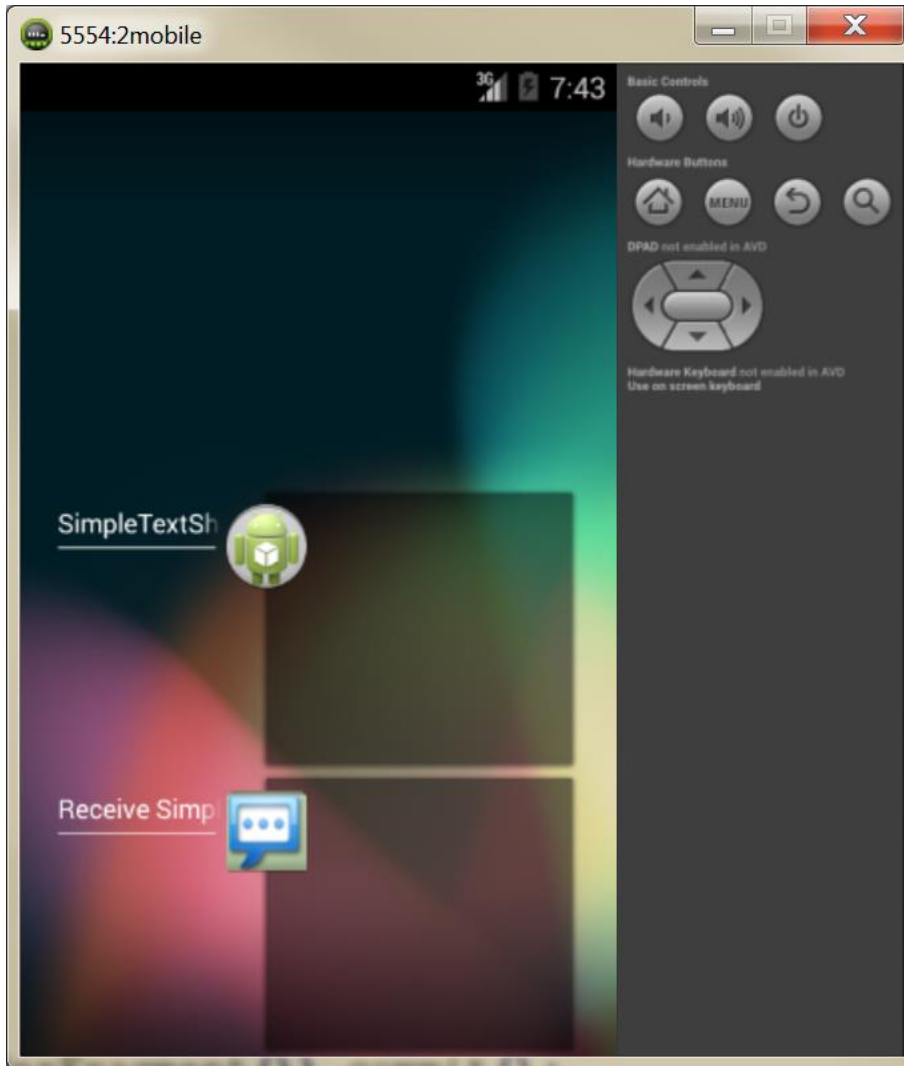
Handle captured data

```
void handleSendText(Intent intent) {  
    String sharedText = intent.getStringExtra(Intent.EXTRA_TEXT);  
    if (sharedText != null) {  
        sent_message=sent_message+"\n"+sharedText;  
    }  
}
```









Android Geolocation



1st Step

Add permissions in Manifest

```
<uses-sdk  
    android:minSdkVersion="16"  
    android:targetSdkVersion="19" />  
<uses-permission android:name="android.permission.INTERNET"/>  
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

2nd Step

Request Location Updates, Implementing “LocationListener”

```
public class MainActivity extends Activity implements LocationListener {  
    private LocationManager locationManager;
```

```
    locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
```

```
    locationManager.requestLocationUpdates( LocationManager.GPS_PROVIDER, 3000, 10, this);
```


Public Interface LocationListener

Public Methods	
abstract void	<code>onLocationChanged(Location location)</code> Called when the location has changed.
abstract void	<code>onProviderDisabled(String provider)</code> Called when the provider is disabled by the user.
abstract void	<code>onProviderEnabled(String provider)</code> Called when the provider is enabled by the user.
abstract void	<code>onStatusChanged(String provider, int status, Bundle extras)</code> Called when the provider status changes.

- **void android.location.LocationManager.requestLocationUpdates(String provider, long minTime, float minDistance, LocationListener listener)**

public void requestLocationUpdates ([String](#) provider, long minTime, float minDistance, [LocationListener](#) listener)

Added in [API level 1](#)

Register for location updates using the named provider, and a pending intent.

See [requestLocationUpdates\(long, float, Criteria, PendingIntent\)](#) for more detail on how to use this method.

3rd Step

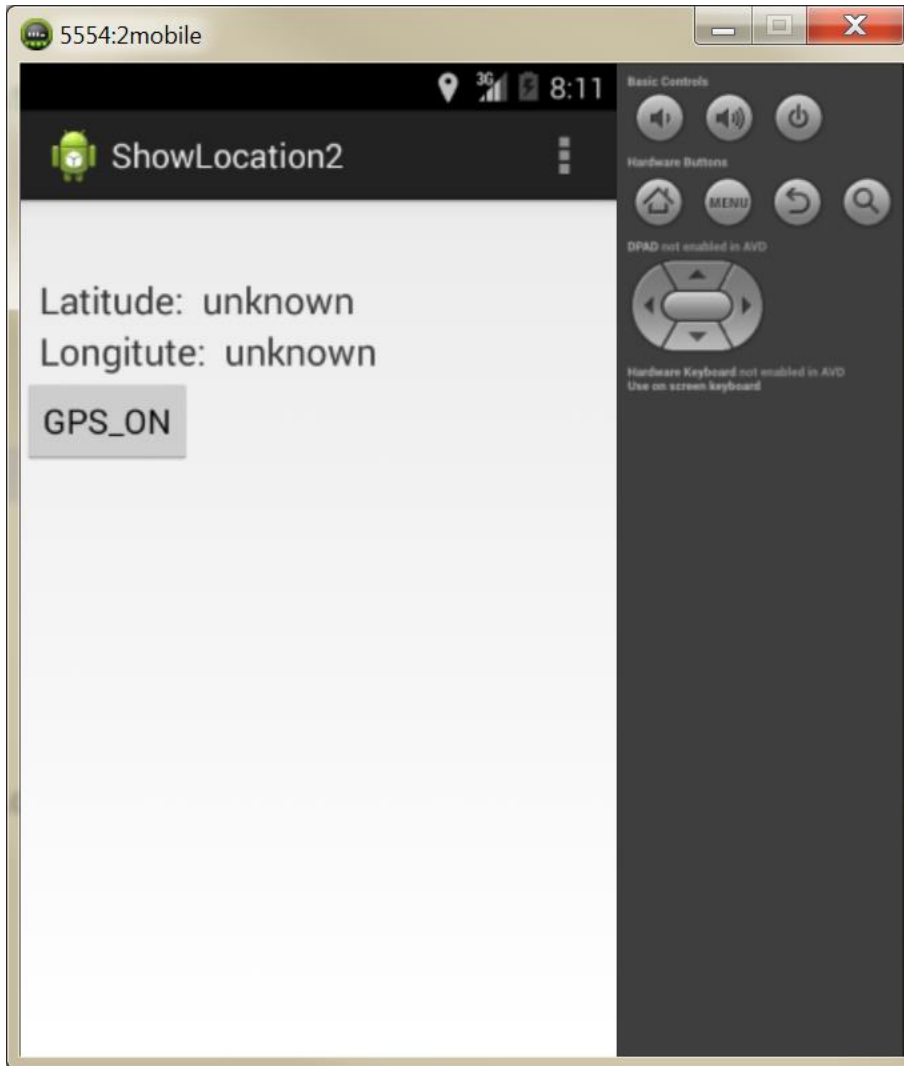
Override “onLocationChanged” function

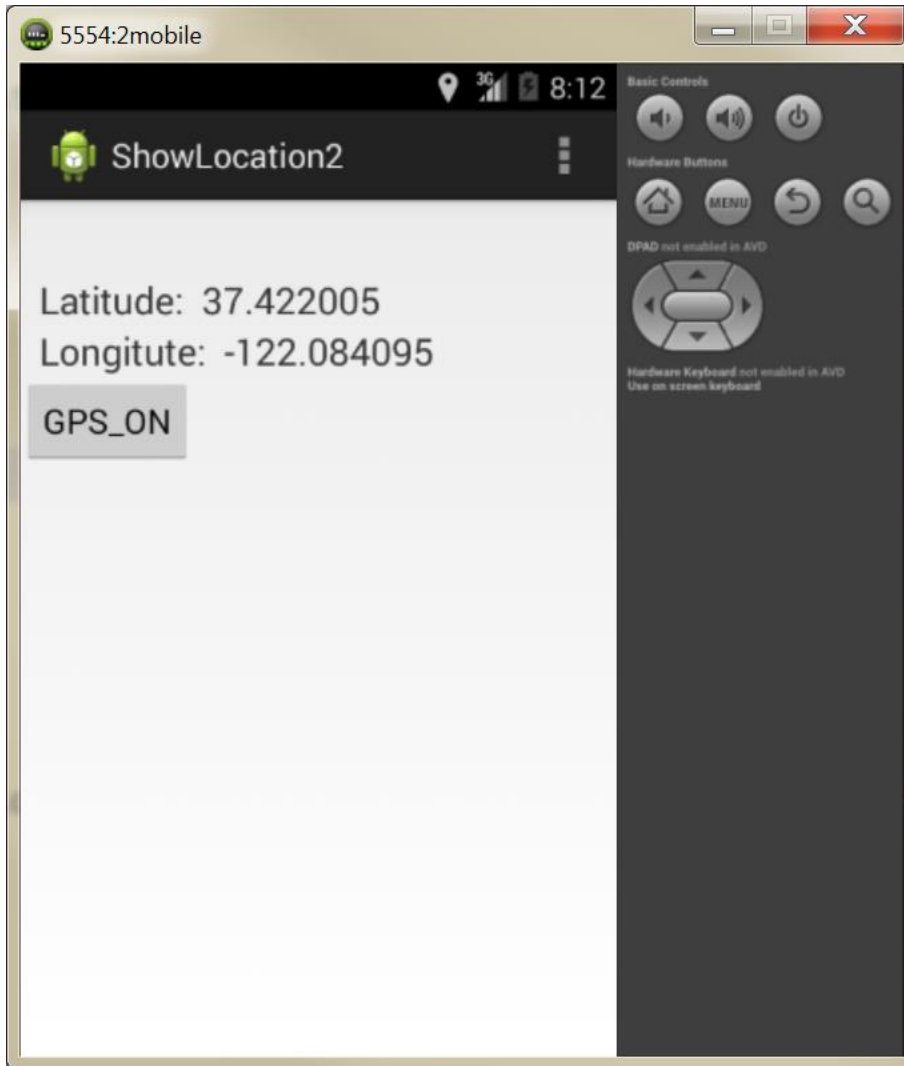
```
@Override
    public void onLocationChanged(Location location) {
        double lat = (double) (location.getLatitude());
        double lng = (double) (location.getLongitude());
        latitudeField.setText(String.valueOf(lat));
        longitudeField.setText(String.valueOf(lng));
    }
```

How to stop location tracking?

```
locationManager.removeUpdates(this);
```





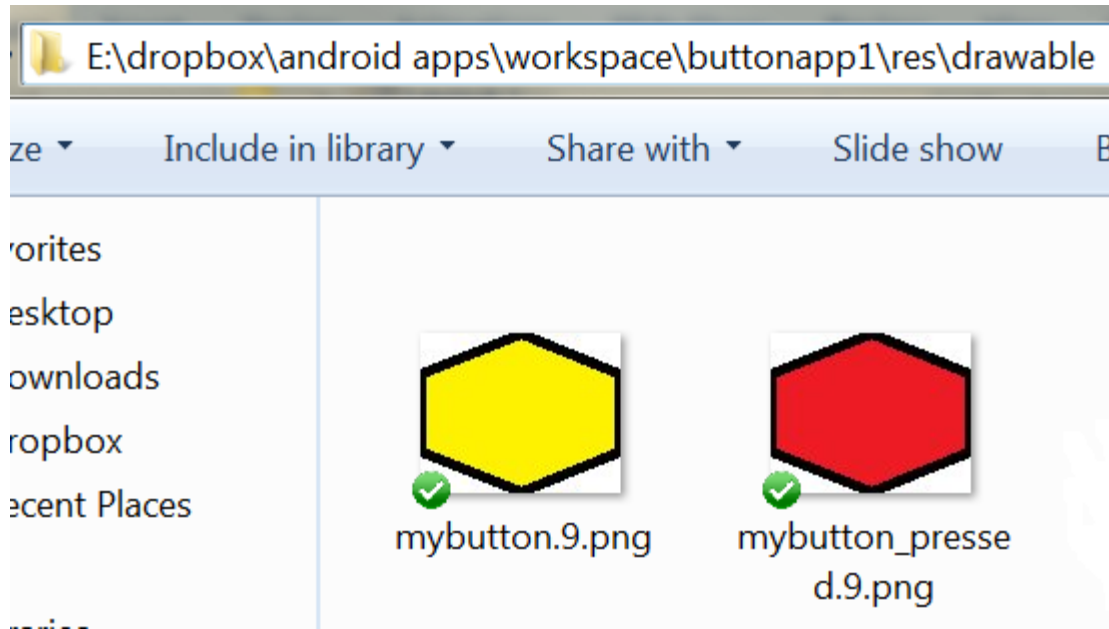


Custom Views



1st Step

Create graphics and save them in
your “res” folder



2nd Step

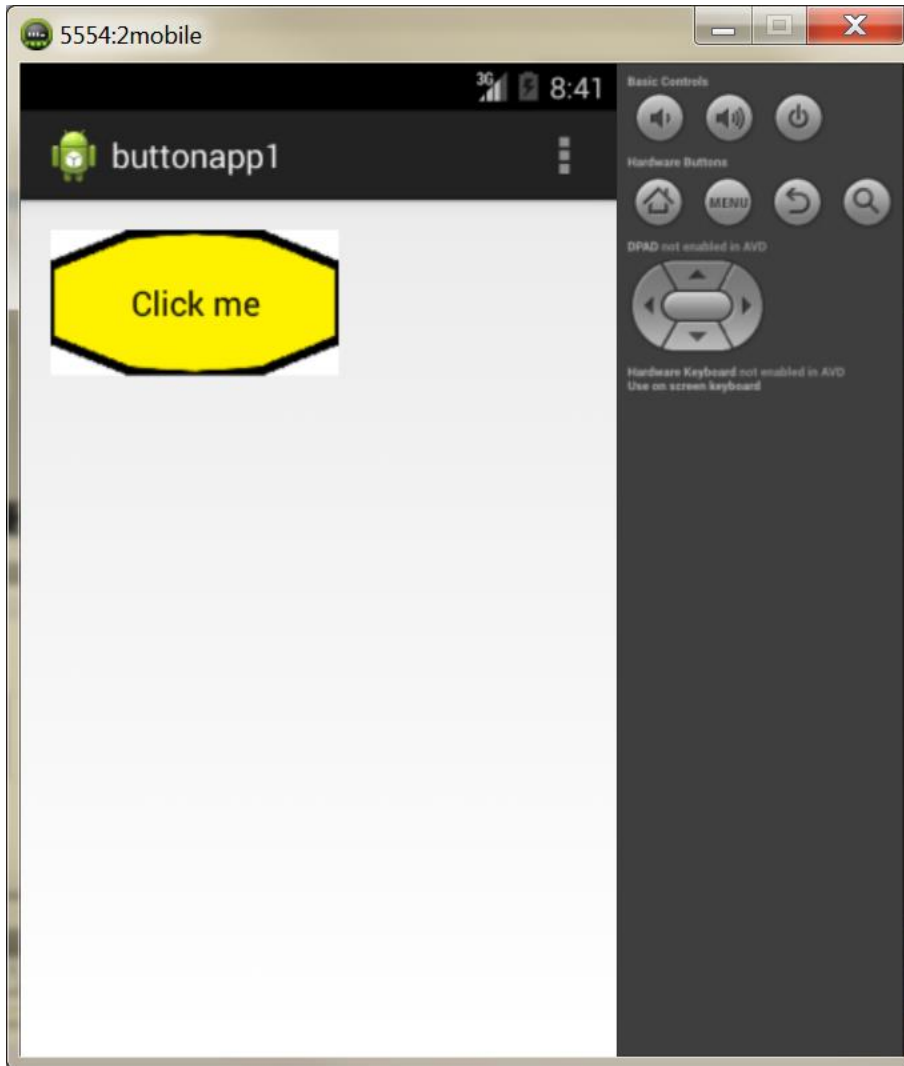
Create the appropriate xml resource file and place it in “res” folder

```
mycustbutton.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <selector xmlns:android="http://schemas.android.com/apk/res/android">
3      <item android:drawable="@drawable/mybutton_pressed"
4          android:state_pressed="true" />
5      <item android:drawable="@drawable/mybutton" />
6  </selector>
```

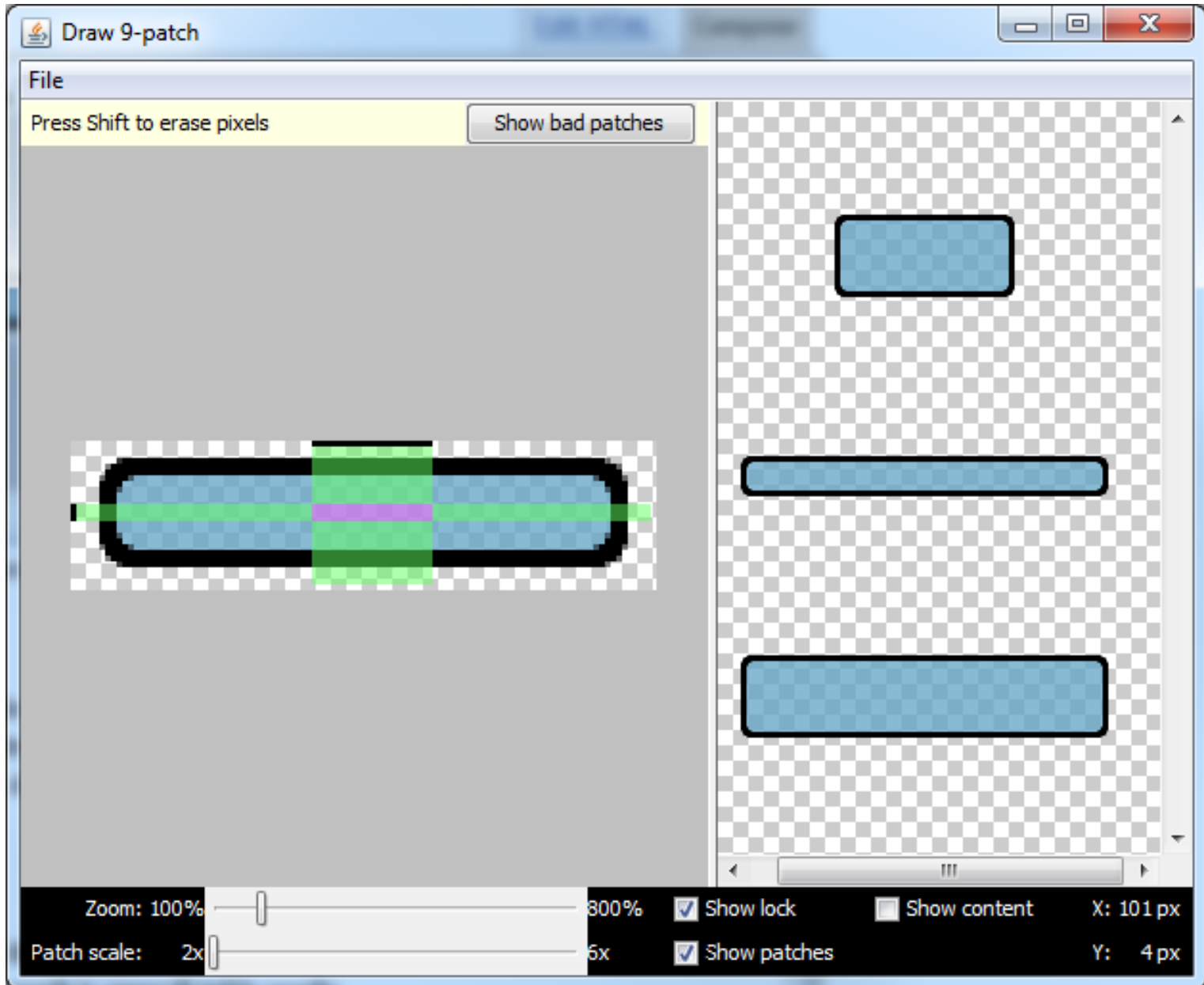
3rd Step

In your main layout file insert the xml file as the button's "background" attribute

```
<Button  
android:id="@+id/button_send"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="@string/button_send"  
android:background="@drawable/mycustbutton" />
```



draw9patch



How to beep and vibrate



Only for vibration: Request permission

```
<uses-sdk  
    android:minSdkVersion="15"  
    android:targetSdkVersion="19" />  
<uses-permission android:name="android.permission.VIBRATE" />
```

Beep through ToneGenerator

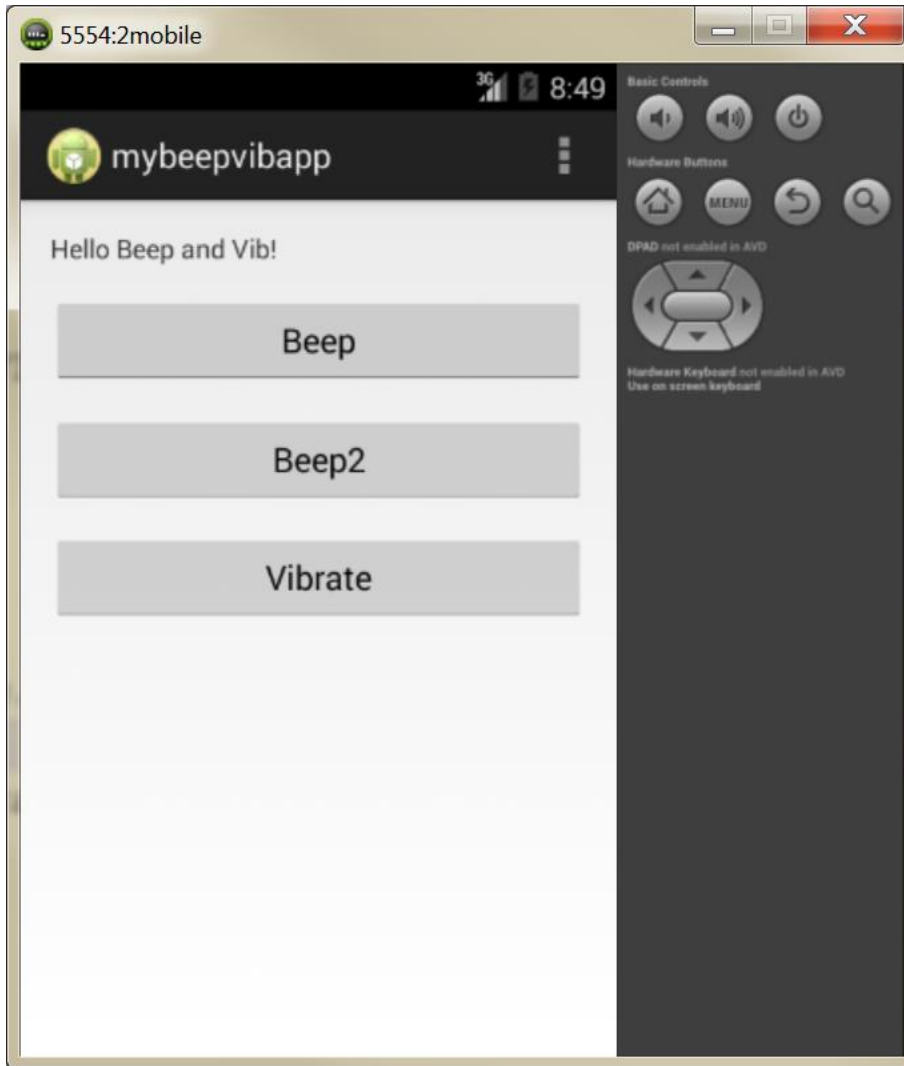
```
public void doBeep(View view){  
    final ToneGenerator tg = new ToneGenerator(AudioManager.STREAM_NOTIFICATION, 100);  
    tg.startTone(ToneGenerator.TONE_PROP_BEEP);  
}
```


Beep through RingtoneManager

```
public void doBeep2(View view){  
    try {  
        Uri notification = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);  
        Ringtone r = RingtoneManager.getRingtone(getApplicationContext(), notification);  
        r.play();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Vibrate!

```
public void doVib(View view) {  
    Vibrator v = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);  
    // Vibrate for 500 milliseconds  
    v.vibrate(500);  
}
```



Android Play a video!



1st Step

Create a VideoView in your layout file

```
<VideoView
```

```
    android:id="@+id/videoView1"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="250dp"
```

```
    android:layout_alignParentLeft="true"
```

```
    android:layout_below="@+id/textView1"
```

```
    android:layout_marginTop="15dp" />
```

2nd Step

If the video is a network resource,
request the appropriate permission

```
<uses-sdk  
    android:minSdkVersion="17"  
    android:targetSdkVersion="19" />  
<uses-permission android:name="android.permission.INTERNET" />
```

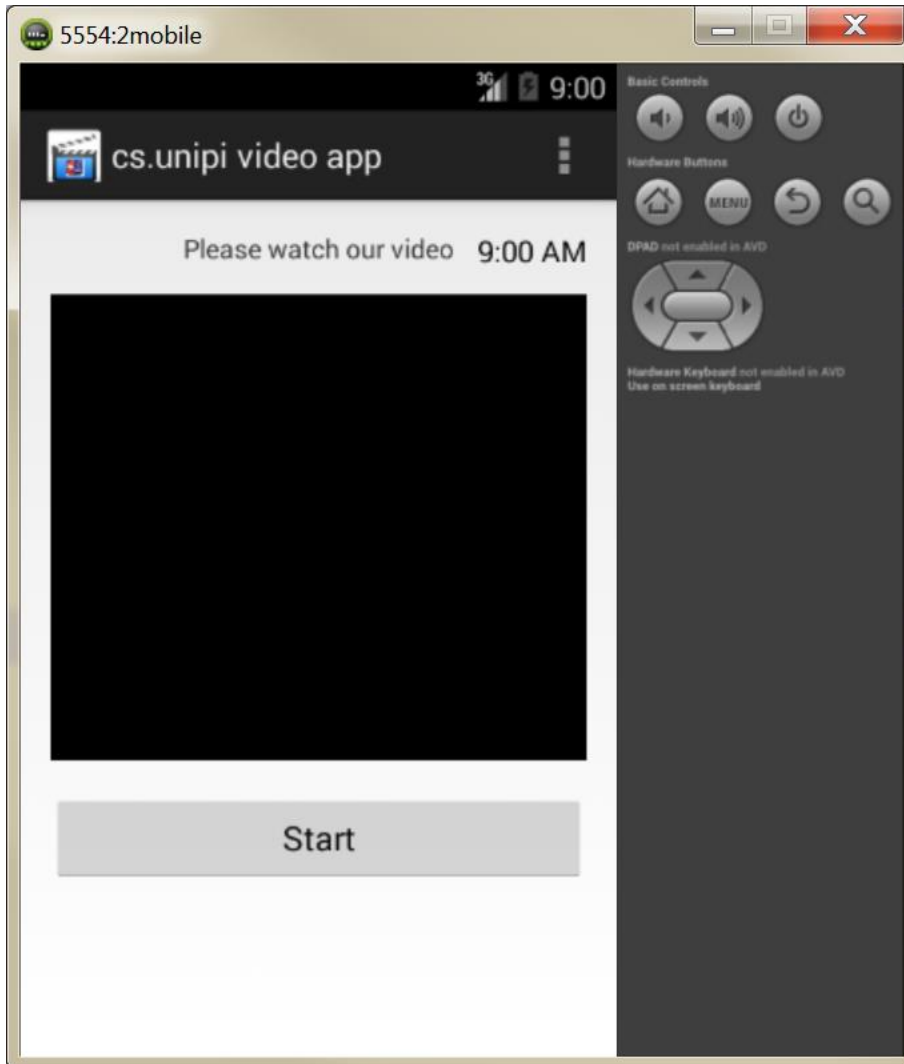
3rd Step

Play the video file inside the VideoView

```
public void startVideo(View v){
    VideoView videoView = (VideoView) findViewById(R.id.videoView1);
    MediaController mc = new MediaController(this);
    videoView.setMediaController(mc);
    String str = "http://www.yoururl.com/yourvideo.mp4";
    Uri uri = Uri.parse(str);
    videoView.setVideoURI(uri);
    videoView.requestFocus();
    videoView.start();
}
```

Stop the video?

```
videoView.stopPlayback();
```



84%



16:02



cs.unipi video app

Please watch our video

16:02



Stop

Play from a local resource?

```
String SrcPath = "/sdcard/Video/myvideo.mp4";
```

