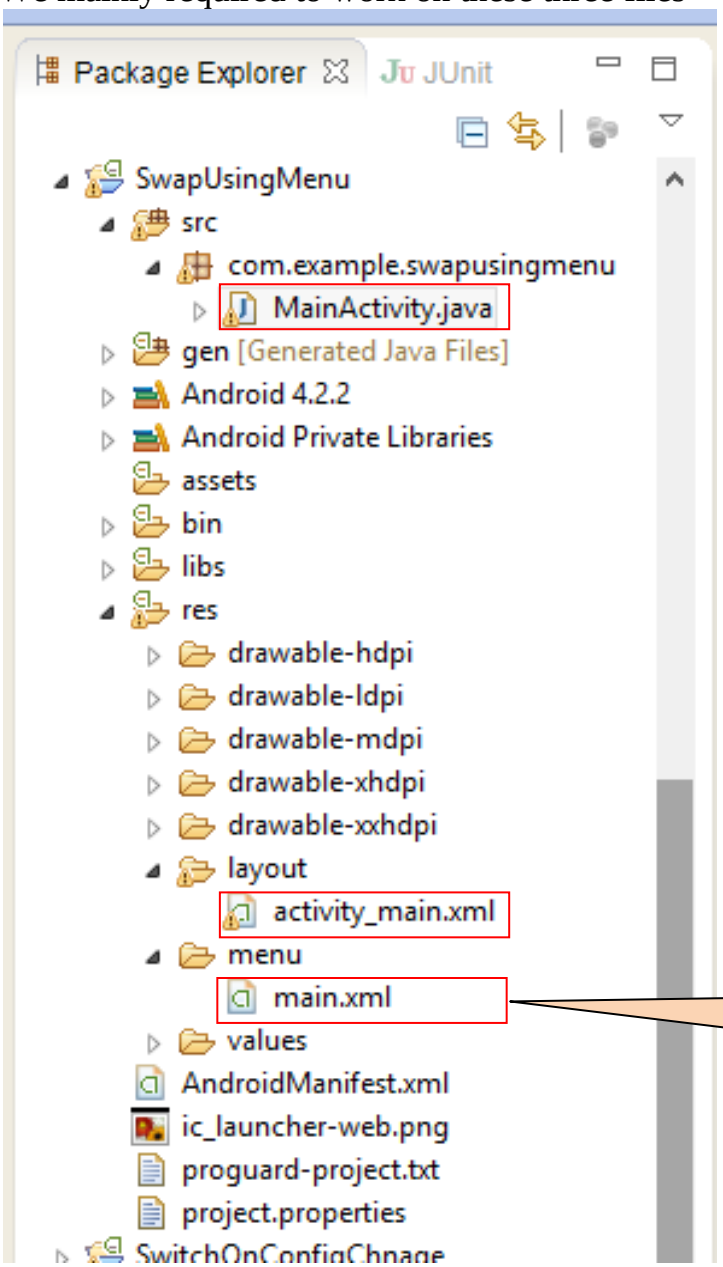
Manisha Vyas

Question.

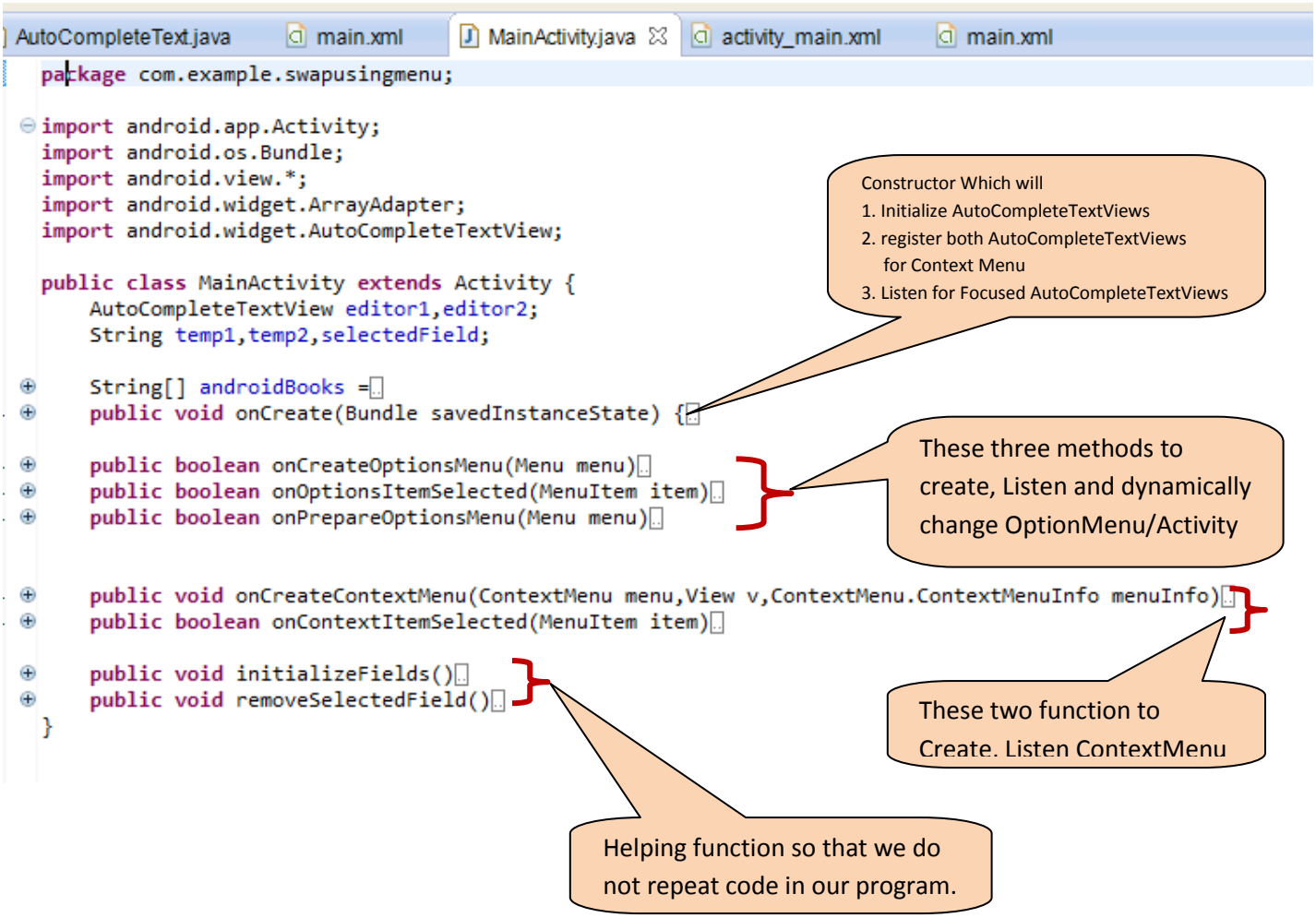
- Activity Menu
  - Adding the menu item "Swap Item". Pressing this menu item will swap the text of the two AutoCompleteTextViews.
  - Adding the menu item "Remove Item". Pressing this menu item will remove the text in the focused AutoCompleteTextView.
  - This exercise allows to swap text by pressing the Swap menu item. When the user clicks the "Swap Item" menu item, the text in the two AutoCompleteTextViews are swapped.
  - If a focused AutoCompleteTextView is empty, the "Remove Item" menu item should be disabled.

- Context Menu
  - When the user uses D-Pad Up or D-Pad Down keys to move the focus to one of the two AutoCompleteTextView fileds, and then hold D-Pad Center, a context menu pops up. The user presses the menu and the focused AutoCompleteTextView field is removed.

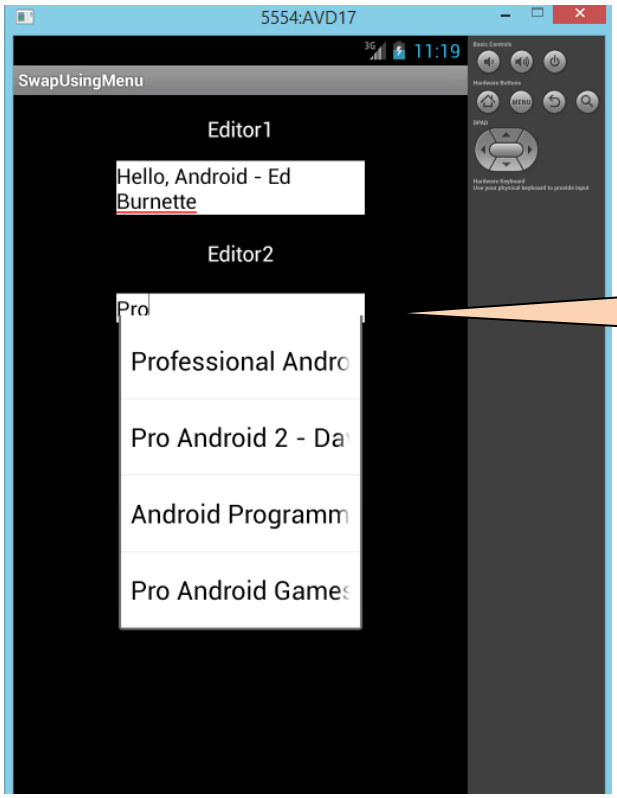
We mainly required to work on these three files



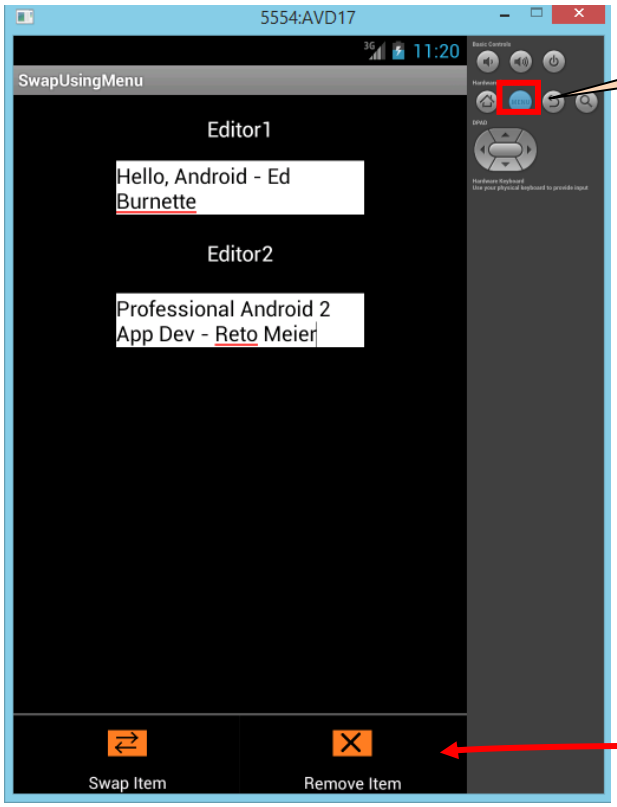
In MainActivity.java We will implement Following methods...



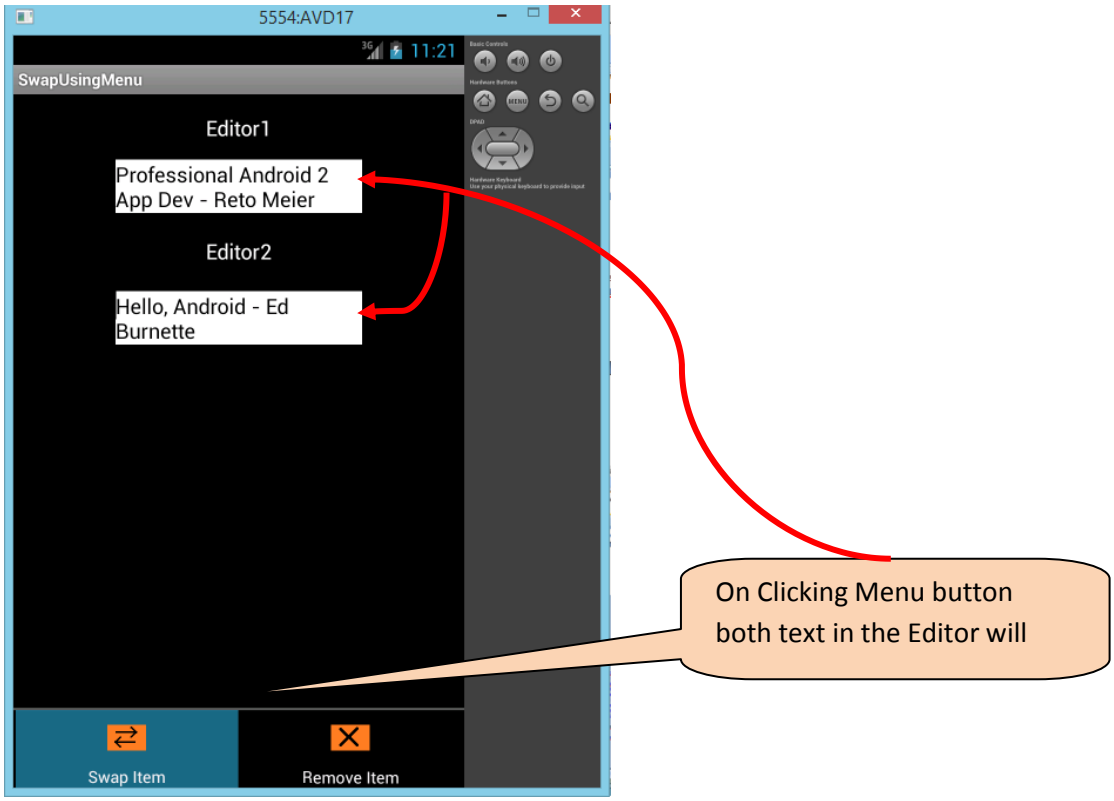
Complete code is given below

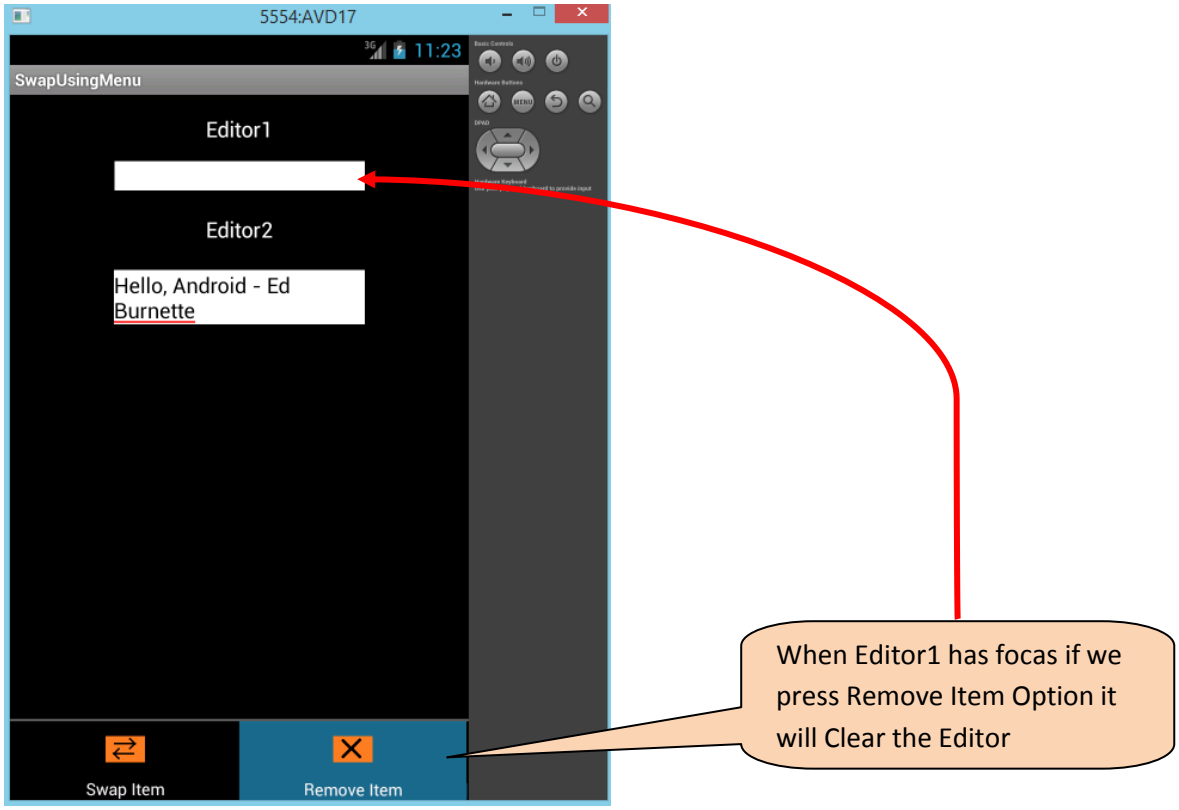


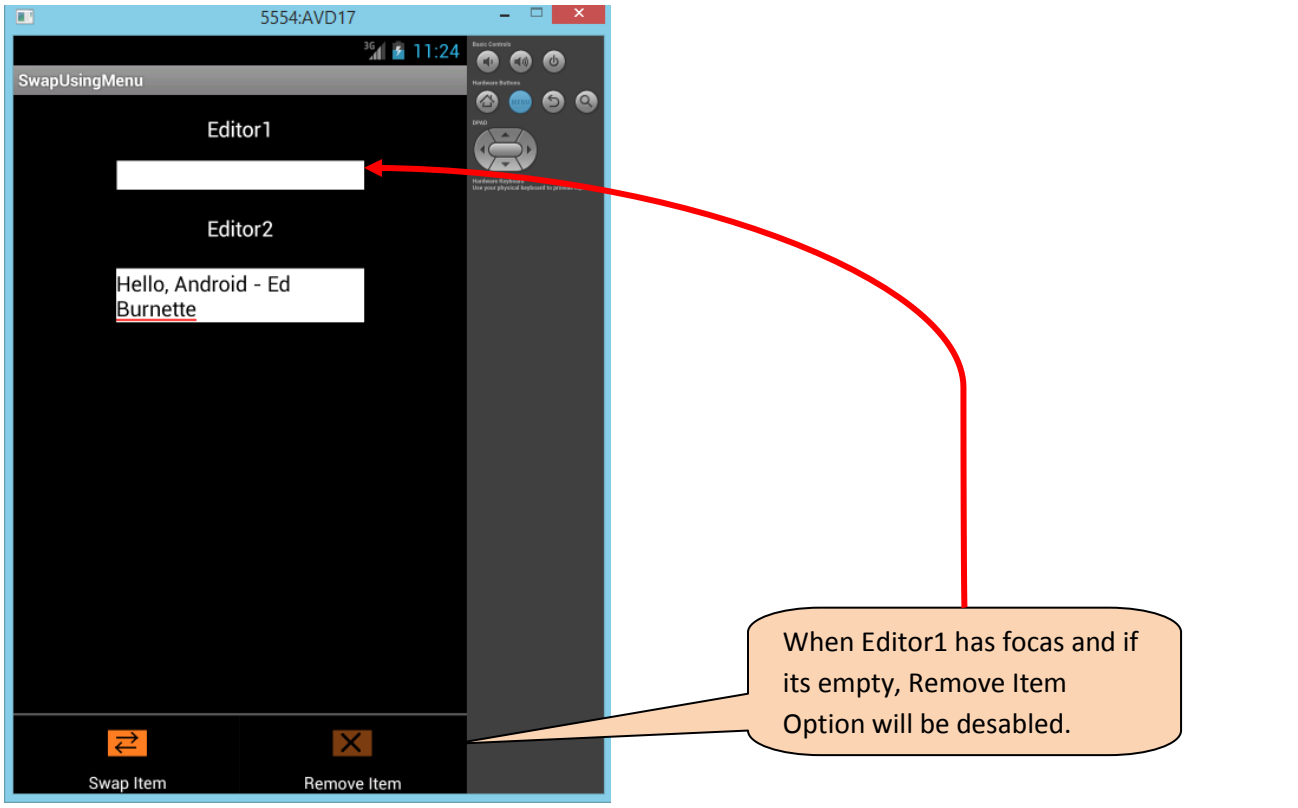
Autocomplete gives suggestion after typeing 3 characters

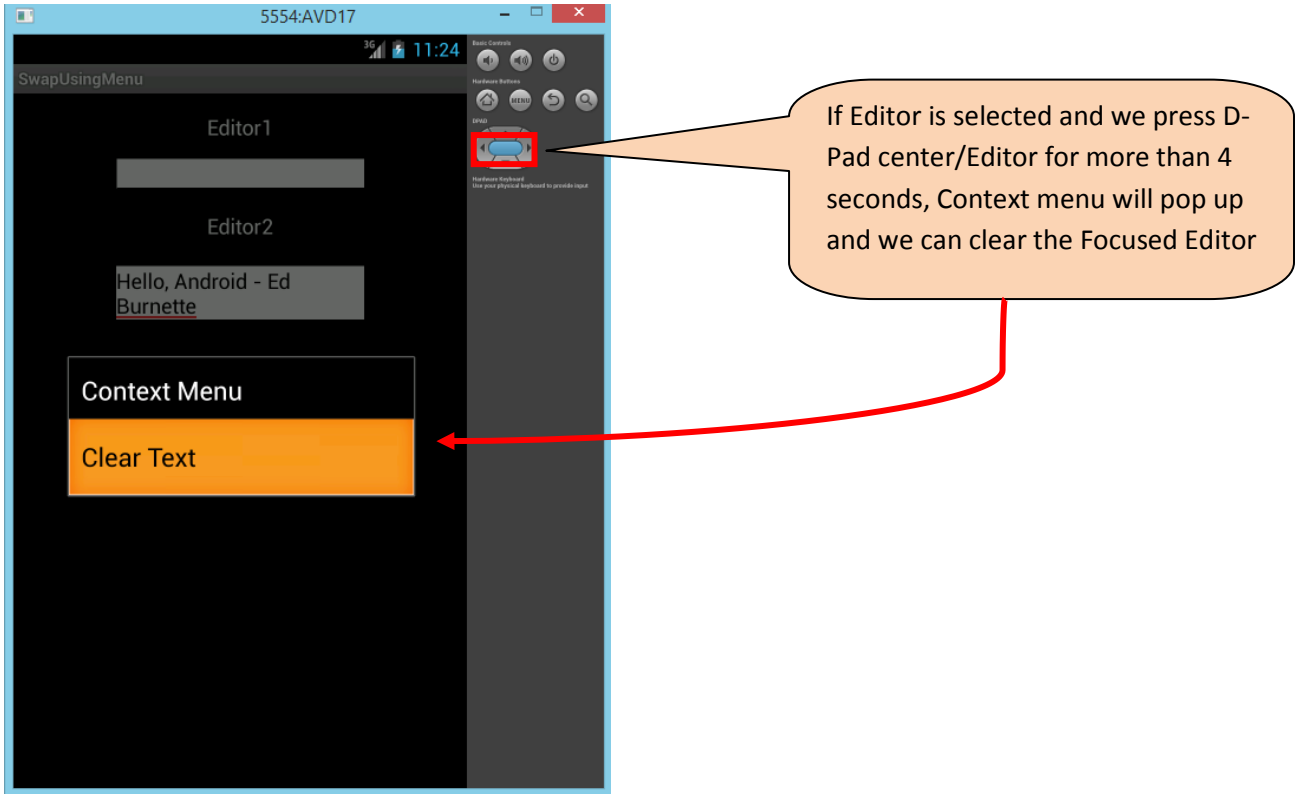


OnClicking Menu button OptionMenu will popup.


5554:AVD17
11:21
SwapUsingMenu
Editor1
Professional Android 2 App Dev - Reto Meier
Editor2
Hello, Android - Ed Burnette
Swap Item
Remove Item
On Clicking Menu button both text in the Editor will

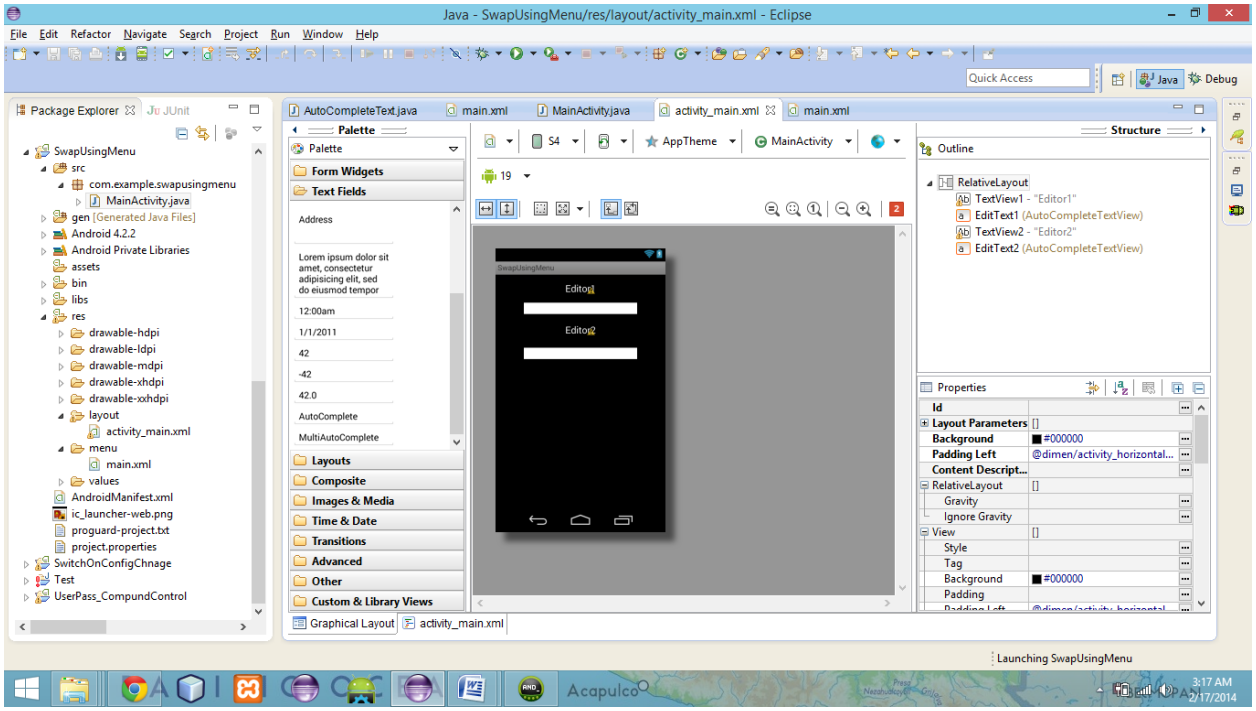

5554:AVD17
11:23
SwapUsingMenu
Editor1
Editor2
Hello, Android - Ed Burnette
Swap Item
Remove Item
When Editor1 has focas if we press Remove Item Option it will Clear the Editor


5554:AVD17
11:24
SwapUsingMenu
Editor1
Editor2
Hello, Android - Ed Burnette
Swap Item
Remove Item
When Editor1 has focas and if its empty, Remove Item Option will be desabled.


5554:AVD17
11:24
SwapUsingMenu
Editor1
Editor2
Hello, Android - Ed Burnette
Context Menu
Clear Text
If Editor is selected and we press D-Pad center/Editor for more than 4 seconds, Context menu will pop up and we can clear the Focused Editor

Step:1 Modify activity_main.xml

activity_main.xml --> Graphical View



Activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#000000"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/TextView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="Editor1"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#FFFFFF" />

    <AutoCompleteTextView
        android:id="@+id/EditText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_below="@+id/TextView1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="15dp"
        android:background="#FFFFFF"
        android:ems="10"/>

    <TextView
        android:id="@+id/TextView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/TextView1"
        android:layout_below="@+id/EditText1"
        android:layout_marginTop="20dp"
        android:text="Editor2"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#FFFFFF" />

    <AutoCompleteTextView
        android:id="@+id/EditText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/EditText1"
        android:layout_below="@+id/TextView2"
        android:layout_marginTop="22dp"
        android:background="#FFFFFF"
        android:ems="10"/>

</RelativeLayout>
```

Step:2 Create menu item -main.xml

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- main.xml  This is 2 menu itmes.Swap_item and remove_item with icons -->
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item android:id="@+id/swap_item"
         android:icon="@drawable/swap_item"
          android:title="@string/swap_item" />
    <item android:id="@+id/remove_item"
         android:icon="@drawable/remove_item"
          android:title="@string/remove_item" />
</menu>
```

This XML code generates menu which looks like this-->

Step:1 Create res>xml> userpreference.xml

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen
  xmlns:android="http://schemas.android.com/apk/res/android">


  <ListPreference
    android:key="PREF_NAME_LEN"
    android:title="Name Length"
    android:summary="Select legth of name to display"
    android:entries="@array/name_length"
    android:entryValues="@array/name_length_val"
    android:dialogTitle="NameLength"
    android:defaultValue="5"
  />
</PreferenceScreen>
```

Step:2 Create array in res>values>arrays.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string-array name="name_length">
    <item>All</item>
    <item>2></item>
    <item>3></item>
    <item>4></item>
    <item>5></item>
    <item>6></item>
    <item>7></item>
  </string-array>

    <string-array name="name_length_val">
    <item>1</item>
    <item>2</item>
    <item>3</item>
    <item>4</item>
    <item>5</item>
    <item>6</item>
    <item>7</item>
  </string-array>
</resources>
```

```
////////////////////////////////Option menu/Action menu//////////////////////////////////
    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        MenuInflater menuInflater = getMenuInflater();
        menuInflater.inflate(R.menu.main, menu);
        return true;
    }
    @Override//Option menu selection listener
```

creating Option/Activity Menu which we created in main.xml

```
    public boolean onOptionsItemSelected(MenuItem item)
    {
        switch (item.getItemId())
        {
        case R.id.swap_item:
            temp1 = editor1.getText().toString();
            temp2 = editor2.getText().toString();

            editor1.setText(temp2);
            editor2.setText(temp1);
             return true;
        case R.id.remove_item:
             removeSelectedField();
             return true;
        default:
            return super.onOptionsItemSelected(item);
        }
    }
    @Override //dynamic menu selection changes
    public boolean onPrepareOptionsMenu(Menu menu)
    {
       initializeFields();
       super.onPrepareOptionsMenu(menu);
       if(selectedField.equals("field1") && editor1.getText().toString().equals(""))
       {
             menu.getItem(1).setEnabled(false);
       }
       else if(selectedField.equals("field2")&& editor2.getText().toString().equals(""))
       {
             menu.getItem(1).setEnabled(false);
       }
       else
       {
             menu.getItem(1).setEnabled(true);
       }
       return true;
    }
////////////////////////////Context Menu/////////////////////////////////////////
    @Override
    public void onCreateContextMenu(ContextMenu menu,View v,ContextMenu.ContextMenuInfo
menuInfo)
    {
       super.onCreateContextMenu(menu,v,menuInfo);

       menu.setHeaderTitle("Context Menu");
       menu.add(0,Menu.FIRST,Menu.NONE,"Clear Text");
    }
    public boolean onContextItemSelected(MenuItem item)
    {
       if(item.getItemId() == Menu.FIRST)
       {
             removeSelectedField();
             return true;
       }
             return super.onContextItemSelected(item);
```

If Swap_item is selected then swap both text

If remove_item is selected then Clear focused field using helping function

If focused filed is editor1 and its empty then disable remove_item option

If focused filed is editor2 and its empty then disable remove_item option

or else keep that option enabled

Create Context menu with Title and a menu item

if Context menu's first and only item selected then Clear AutoCompleteTextViews

```
    }
////////////////////////////////Helping Function/////////////////////////////////////
    public void initializeFields()
    {
       ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this,android.R.layout.simple_dropdown_item_1line,androidBooks);

        editor1 = (AutoCompleteTextView)findViewById(R.id.EditText1);
        editor2 = (AutoCompleteTextView)findViewById(R.id.EditText2);

        editor1.setThreshold(3);
        editor1.setAdapter(adapter);
        editor2.setThreshold(3);
        editor2.setAdapter(adapter);

    }
    public void removeSelectedField()
    {
       if(selectedField.equals("field1") )
       {
              editor1.setText("");
       }
       else if(selectedField.equals("field2"))
       {
              editor2.setText("");
       }
    }
}
```