



Mobile Apps Development

ActionBar & Menus

(Lesson 08)

Lecturer: Vang Le

Faculty of Information Technology

Ton Duc Thang University

Overview

- The most common actions for an activity live on the default Action Bar
- But actions that are more specific to an item or element can be displayed contextually using menus and popups.
- In addition to menus, apps often need to display popup overlays that contain informational content.

ActionBar: Examples



Changing the ActionBar Title

```
<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme">
    <activity
        android:name="vn.edu.tdt.it.actionbar.MainActivity"
        android:label="Your ActionBar Title" >
    </activity>
</application>
```

AndroidManifest.xml

```
ActionBar actionBar = getSupportActionBar();
actionBar.setTitle("Your ActionBar Title");
```

MainActivity.java

Displaying ActionBar Icon



SimpleApp

```
getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
getSupportActionBar().setLogo(R.mipmap.ic_launcher);  
getSupportActionBar().setDisplayUseLogoEnabled(true);
```

MainActivity.java

Adding Action Items

SimpleApp



Material App



```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
  <item
    android:id="@+id/miCompose"
    android:icon="@drawable/ic_compose"
    app:showAsAction="ifRoom"
    android:title="Compose">
  </item>
  <item
    android:id="@+id/miProfile"
    android:icon="@drawable/ic_profile"
    app:showAsAction="ifRoom"
    android:title="Profile">
  </item>
</menu>
```

res/menu/menu_main.xml

Adding Action Items

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}
```

MainActivity.java

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.miCompose:
            //do something
            return true;
        case R.id.miProfile:
            //do something
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

MainActivity.java

Custom ActionBar Layout



Custom Title

```
<?xml version="1.0" encoding="utf-8"?>    res/layout/actionbar_layout.xml
<LinearLayout ...>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/app_title"
        android:textColor="#ffffff"
        android:id="@+id/mytext"
        android:textSize="18dp" />
</LinearLayout>
```

MainActivity.java

```
getSupportActionBar().setDisplayOptions(
    ActionBar.DISPLAY_SHOW_CUSTOM);
getSupportActionBar().setCustomView(R.layout.actionbar_title);
```


Adding SearchView to ActionBar

SimpleApp



```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto" >
    <item android:id="@+id/action_search"
        android:orderInCategory="5"
        android:title="Search"
        android:icon="@android:drawable/ic_menu_search"
        app:showAsAction="always|collapseActionView"
        app:actionViewClass="android.support.v7.widget.SearchView"
    />
</menu>
```

Notice that the **app:showAsAction** attribute also includes the "**collapseActionView**" value which collapses the search into an icon until clicked.

Adding SearchView to ActionBar

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.main, menu);
    MenuItem searchItem = menu.findItem(R.id.action_search);
    final SearchView searchView;
    searchView = (SearchView) MenuItemCompat.getActionView(
                                                                    searchItem);
    searchView.setOnQueryTextListener(new OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String query) {
            ...
        }
        @Override
        public boolean onQueryTextChange(String newText) {
            ...
        }
    });
    return super.onCreateOptionsMenu(menu);
}
```

Extending SearchView

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // ... Lookup the search view
    MenuItem searchItem = menu.findItem(R.id.action_search);
    final SearchView searchView;
    searchView = (SearchView) MenuItemCompat.getActionView(
        searchItem);

    int searchIconId =
        android.support.v7.appcompat.R.id.search_button;
    ImageView v = (ImageView) searchView.findViewById(searchIconId);
    v.setImageResource(R.drawable.search_btn);

    // Customize searchview text and hint colors
    int searchEditTextId =
        android.support.v7.appcompat.R.id.search_src_text;
    EditText et = (EditText) searchView.findViewById(
        searchEditTextId);

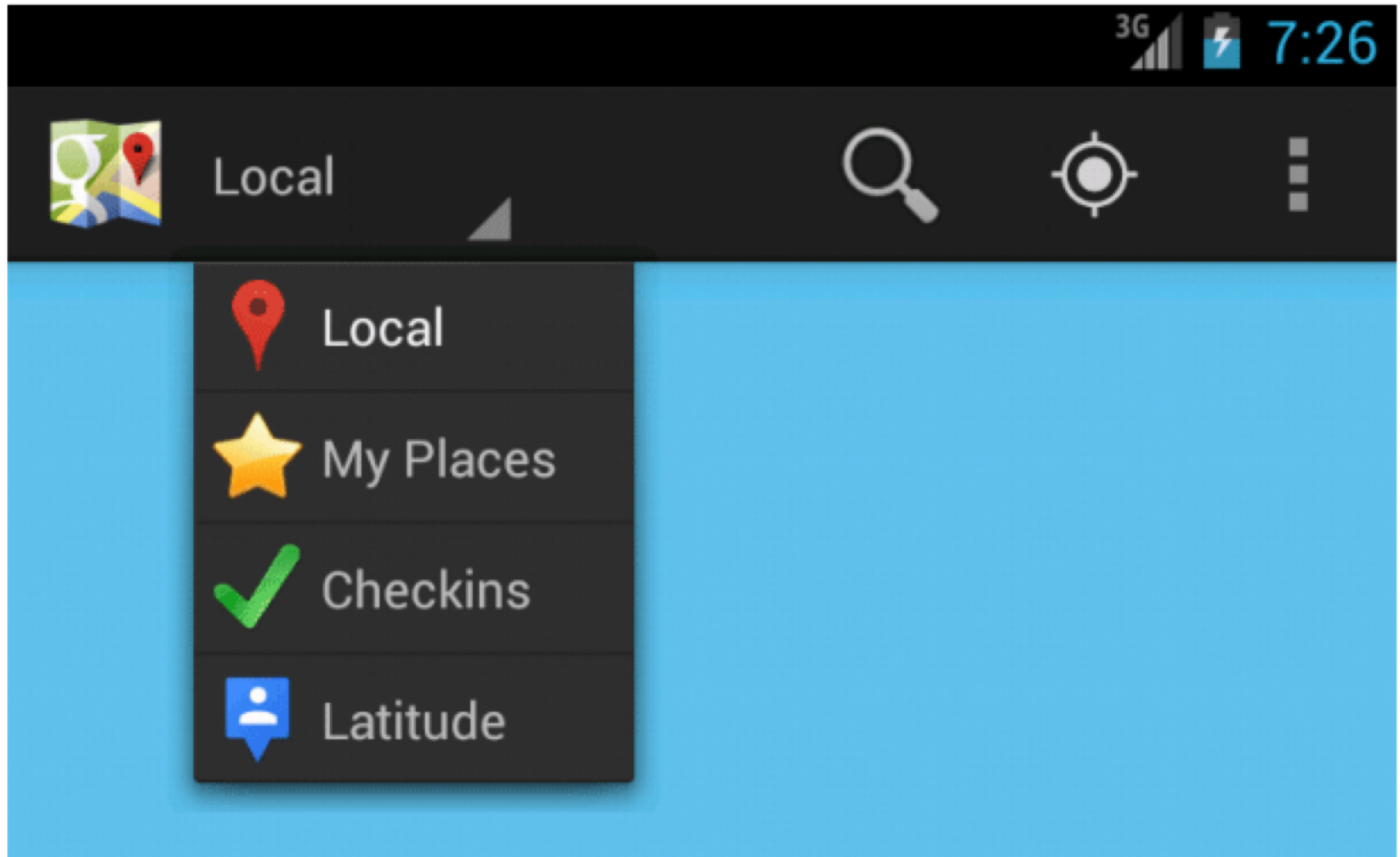
    et.setTextColor(Color.BLACK);
    et.setHintTextColor(Color.BLACK);
}
```

Adding an Up Action

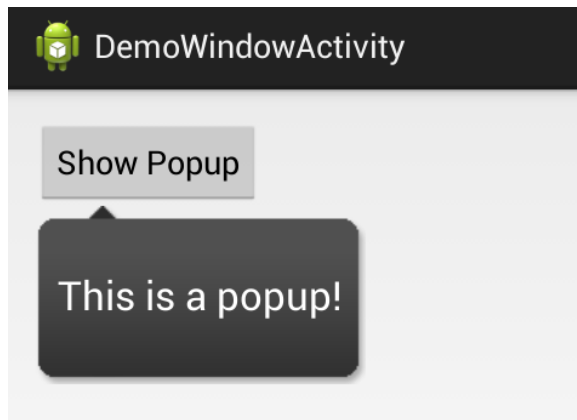
```
<application ... >                                AndroidManifest.xml
    ...
    <activity
        android:name="com.example.myapp.MainActivity" ...>
        ...
    </activity>
    <activity
        android:name="com.example.myfirstapp.MyChildActivity"
        android:label="@string/title_activity_child"
        android:parentActivityName="com.example.myapp.MainActivity">
    </activity>
</application>
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    ActionBar ab = getSupportActionBar();
    ab.setDisplayHomeAsUpEnabled(true);
}
```

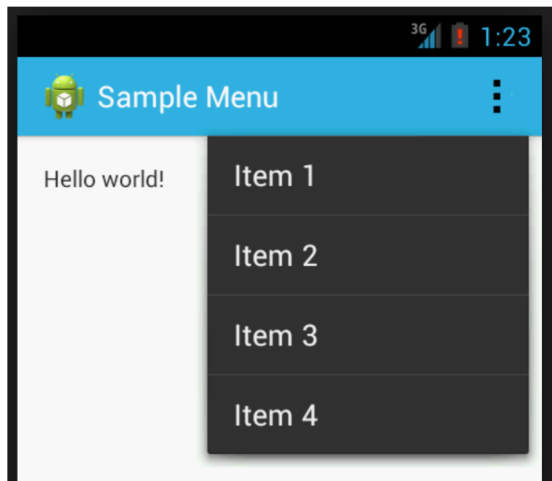
Exercise 01



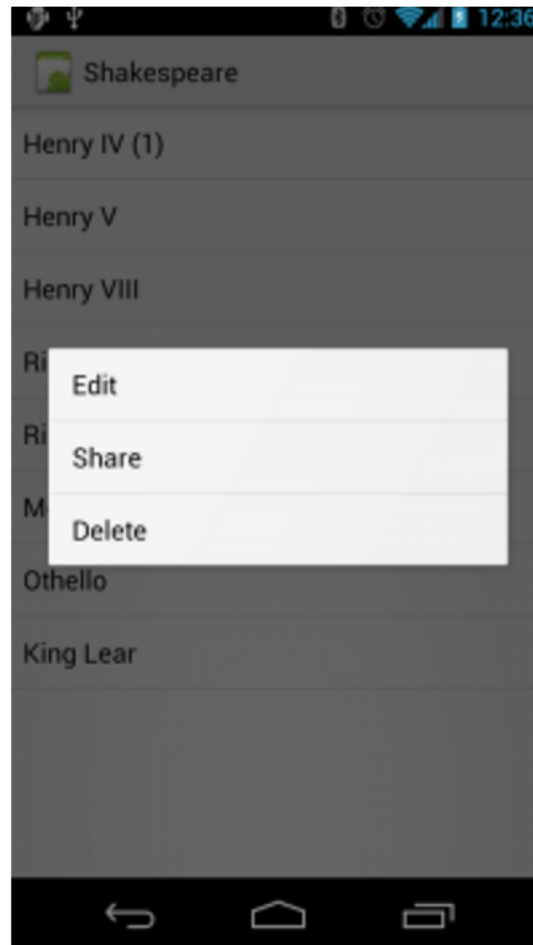
Menus: Examples



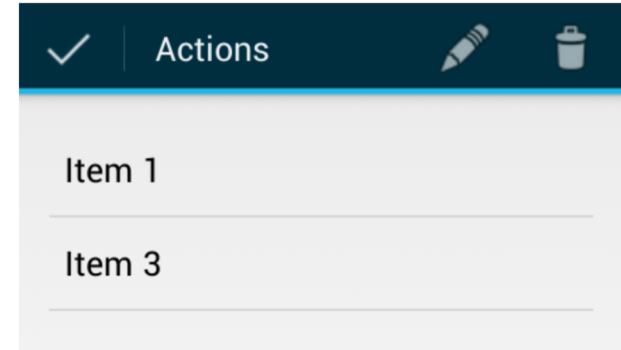
Popup Window



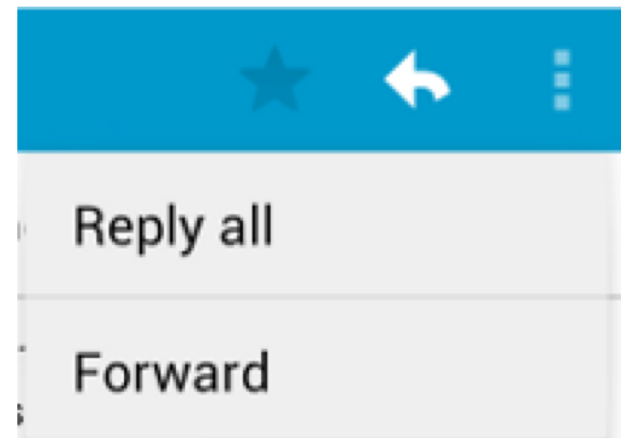
Options menu



Floating context menu



Contextual action bar



Popup Menu

Menu: Contextual Action Modes



```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.c
    <item android:id="@+id/menu_edit"
        android:title="Edit"
        android:icon="@android:drawable/ic_
    <item android:id="@+id/menu_delete"
        android:title="Delete"
        android:icon="@android:drawable/ic_menu_delete"/>
</menu>
```

res/menu/action_mode_menu.xml

```
<style name="MyCustomTheme" parent="AppTheme">
    <item name="android:actionModeCloseDrawable">
        @android:drawable/ic_menu_send</item>
</style>
```

res/value/styles.xml

```
<activity android:name=".MainActivity"
    android:theme="@style/MyCustomTheme">
```

AndroidManifest.xml

Menu: Contextual Action Modes

```
final EditText editText = (EditText)findViewById(R.id.editText);
editText.setOnLongClickListener(new View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View v) {
        ActionMode.Callback callback = buildActionMode();
        startActionMode(callback);
        return true;
    }
});
```

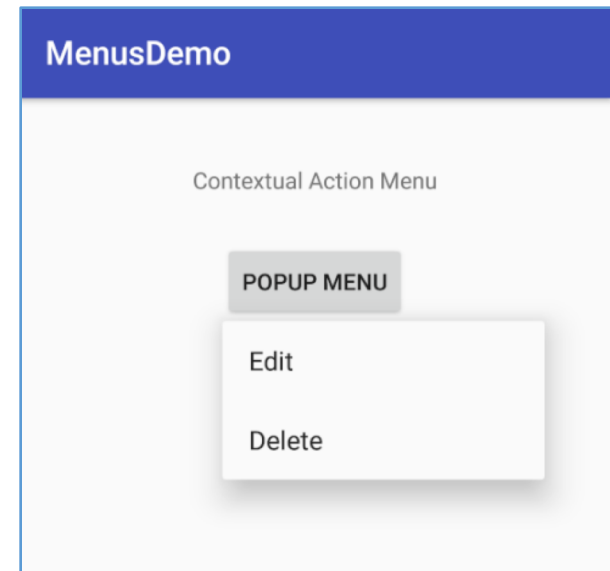
MainActivity.java

Menu: Contextual Action Modes

```
private ActionMode.Callback buildActionMode() {
    ActionMode.Callback callback = new ActionMode.Callback() {
        @Override
        public boolean onCreateActionMode(ActionMode mode,
                                           Menu menu) {
            getMenuInflater().inflate(R.menu.action_mode_menu,
                                     menu);
            mode.setTitle("Actions");
            return true;
        }
        @Override
        public boolean onActionItemClicked(ActionMode mode,
                                           MenuItem item) {
            mode.finish();
            return true;
        }
    };
    return callback;
}
```

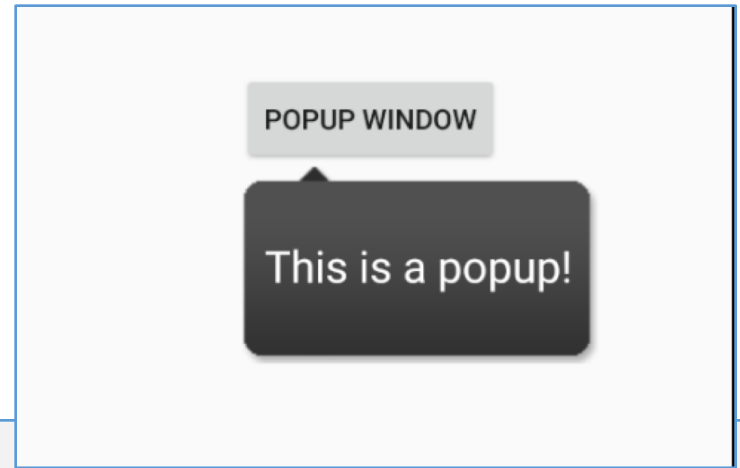
MainActivity.java

Menu: Popup Menu



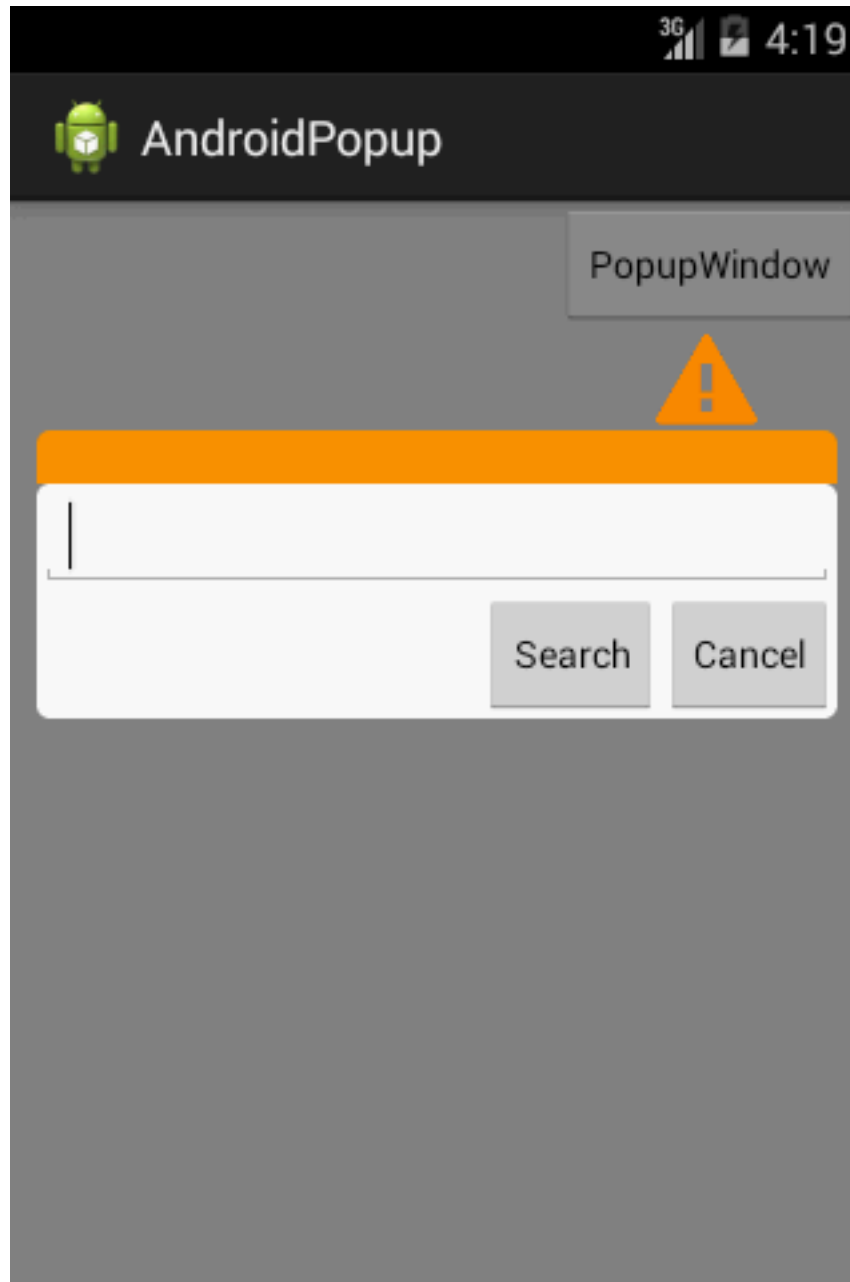
```
final Button button1 = (Button)findViewById(R.id.button1);
button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        PopupMenu popupMenu = new PopupMenu(
                                MainActivity.this, button1);
        popupMenu.getMenuInflater().inflate(
                                R.menu.popup_menu,
                                popupMenu.getMenu());
        popupMenu.show();
    }
});
```

Menu: Popup Window



```
final Button btnPopupWindow =  
    (Button)findViewById(R.id.btnPopupWindow);  
btnPopupWindow.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        PopupWindow popupWindow = new PopupWindow(MainActivity.this);  
        View popup_content = getLayoutInflater().inflate(  
            R.layout.popup_window, null);  
        popupWindow.setContentView(popup_content);  
        Drawable transparentDrawable = new ColorDrawable(  
            Color.TRANSPARENT);  
        popupWindow.setBackgroundDrawable(transparentDrawable);  
        popupWindow.showAsDropDown(btnPopupWindow);  
    }  
});
```

Exercise 02



References

- <https://guides.codepath.com/android/Menus-and-Popups>
- <https://developer.android.com/reference/android/support/v7/app/ActionBar.html>