

Mobile Development

Lecture 7: Action Bar, Media

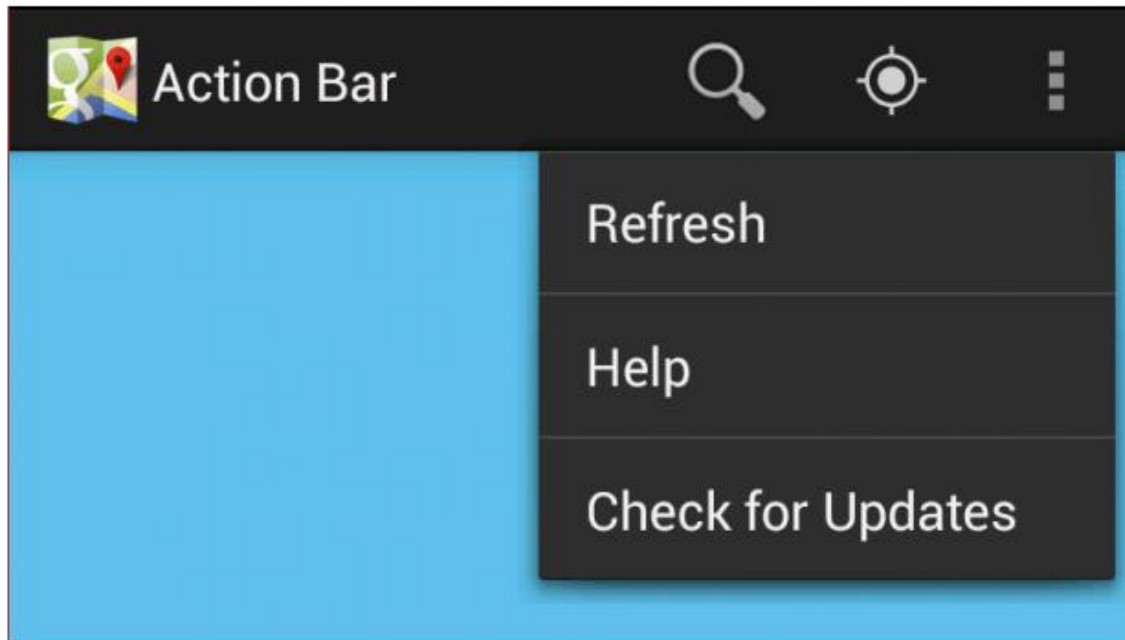
Mahmoud El-Gayyar

elgayyar@ci.suez.edu.eg

Elgayyar.weebly.com



1. The Action Bar



Action Bar

- *action bar: top-level menu of app functions*
 1. identifies current activity/app to user
 2. make common actions prominent and available
 3. make less common actions available through a drop-down menu



Support for Action Bar

- *make activity class extend ActionBarActivity*
 - ◆ Write (Override) methods:
 - ▶ *onCreateOptionsMenu*
 - ▶ *onOptionsItemSelected*
- *declare the menu items in res/menu/menu_activity.xml*
 - ◆ decide which items have icons, which have text,
 - ◆ which should appear in main menu, which in "overflow" submenu
 - ◆ need to place icon image files in **res/drawable** folder
- *handle events*
 - ◆ write code in *onOptionsItemSelected* to check what option was clicked and respond accordingly



ActionBarActivity

```
public class MainActivity extends ActionBarActivity {  
    ...  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        MenuInflater inflater = getMenuInflater(); // reads XML  
        inflater.inflate(R.menu.menu_main, menu); // to create  
        return super.onCreateOptionsMenu(menu); // the menu  
    }  
  
    @Override  
    public boolean onOptionsItemSelected(MenuItem item) {  
        // TODO: handle clicks on the menu items  
        return super.onOptionsItemSelected(item);  
    }  
}
```

Menu XML Data

```
<menu

    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"
    tools:context=".MainActivity">

    <item android:id="@+id/action_send" android:title="Send"
        android:icon="@drawable/iconsend"
        app:showAsAction="always" />
    <item android:id="@+id/action_archive" android:title="Archive"
        android:icon="@drawable/iconarchive"
        app:showAsAction="always" />
    <item android:id="@+id/action_open" android:title="Open"
        android:icon="@drawable/iconopen" />

</menu>
```

showAsAction can be always, never, ifRoom, withText, ...

onOptionsItemSelected

```
public class MainActivity extends ActionBarActivity {  
    ...  
    /* Handles presses on the action bar items. */  
    @Override  
    public boolean onOptionsItemSelected(MenuItem item) {  
        if (item.getItemId() == R.id.action_send) {  
            // do something;  
        } else if (item.getItemId() == R.id.action_archive) {  
            // do something;  
        } else if (item.getItemId() == R.id.action_open) {  
            // do something;  
        }  
        return super.onOptionsItemSelected(item);  
    }  
}
```

2. Media



Playing a Video

- *Video files should be added in a raw folder (video.mp4)*
 - ◆ If not exist, just create inside *res* folder
- *Add VideoView in your layout*
- *Get VideoView in your code*
- *Set Video Path*
- *Start the Video*



```
VideoView videoView=(VideoView)findViewById(R.id.myVideo);  
videoView.setVideoPath("android.resource://" + getPackageName() + "/" + R.raw.video);  
videoView.start();
```

Add Media Controller to Your Video

```
VideoView videoView=(VideoView)findViewById(R.id.myVideo);  
  
videoView.setVideoPath("android.resource://" + getPackageName() + "/" + R.raw.test);  
  
MediaController mediaController=new MediaController(this);  
  
mediaController.setAnchorView(videoView);           //attach video to controller  
  
videoView.setMediaController(mediaController);     //attach controller to video  
  
videoView.start();
```



Controlling Audio

- *Again Audio files should be added in a raw folder (audio.mp3)*
- *Add Media Player*

```
MediaPlayer mPlayer= MediaPlayer.create(this, R.raw.audio);
```

- *Strat the media player*

```
mPlayer.start();
```

- *Control the media player*

```
mPlayer.pause();
```

```
mPlayer.reset();
```

```
mPlayer.seekTo(10);           // move to second 10
```



[More Info](#)