Session 9: Dialogs

Assignment: Dialogs Basic.

Problem Statement

a. What is dialog?

b. How to create custom dialog?

c. How to use existing dialogs?

Solution:

1. A dialog is a small window that prompts the user to make a decision or enter additional information. A dialog does not fill the screen and is normally used for modal events that require users to take an action before they can proceed.

The Dialog class is the base class for dialogs, but should avoid instantiating Dialog directly. Instead, use one of the following subclasses:

* AlertDialog

A dialog that can show a title, up to three buttons, a list of selectable items, or a custom layout.

* DatePickerDialog or TimePickerDialog

A dialog with a pre-defined UI that allows the user to select a date or time.

These classes define the style and structure for the dialog, but one should use a DialogFragment as a container for the dialog. The DialogFragment class provides all the controls need to create the dialog and manage its appearance, instead of calling methods on the Dialog object.

Using DialogFragment to manage the dialog ensures that it correctly handles lifecycle events such as when the user presses the Back button or rotates the screen. The DialogFragment class also allows to reuse the dialog's UI as an embeddable component in a larger UI, just like a traditional Fragment (such as when you want the dialog UI to appear differently on large and small screens).

For example, here's a basic AlertDialog that's managed within a DialogFragment:

public class FireMissilesDialogFragment extends DialogFragment {

@Override

public Dialog onCreateDialog(Bundle savedInstanceState) {

// Use the Builder class for convenient dialog construction

AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());

builder.setMessage(R.string.dialog\_fire\_missiles)

.setPositiveButton(R.string.fire, new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

// FIRE ZE MISSILES!

}

})

.setNegativeButton(R.string.cancel, new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

// User cancelled the dialog

}

});

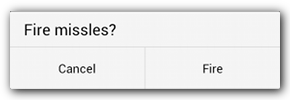
// Create the AlertDialog object and return it

return builder.create();

}

}

Output will be:



1. In Android one can use the Dialog component, and can customize it to own User Interface so it can have any use we want and it is custom dialog. To do this, we will:

Create a new xml layout file

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical"

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/button\_label" />

</LinearLayout>

A Custom.xml

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent" >

<ImageView

android:id="@+id/image"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/ic\_launcher"

android:contentDescription="@drawable/ic\_launcher" />

<TextView

android:id="@+id/txt"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_toRightOf="@+id/image"/>

<Button

android:id="@+id/dialogButton"

android:layout\_width="100dp"

android:layout\_height="wrap\_content"

android:text="@string/dialog\_button\_label"

android:layout\_marginTop="5dp"

android:layout\_marginRight="5dp"

android:layout\_below="@+id/image"

/>

and Bundle the new layout file with the View of the Dialog Box.

import android.app.Activity;

import android.app.Dialog;

import android.content.Context;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.TextView;

public class MainActivity extends Activity {

final Context context = this;

private Button button;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

button = (Button) findViewById(R.id.button);

button.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View view) {

final Dialog dialog = new Dialog(context);

dialog.setContentView(R.layout.dialog);

dialog.setTitle("Android Custom Dialog Box");

TextView txt = (TextView) dialog.findViewById(R.id.txt);

txt.setText("This is an Android custom Dialog Box Example! Enjoy!");

Button dialogButton = (Button) dialog.findViewById(R.id.dialogButton);

dialogButton.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

dialog.dismiss();

}

});

dialog.show();

}

});

}

}

1. DialogFragment can implement the onCreateDialog method and return an existing dialog. The Dialog class is the base class for implementing a dialog. Typically, you use one of its subclasses, e.g., AlertDialog, ProgressDialog, DatePickerDialog or TimePickerDialog.

Android also provides a ProgressDialog, which can be opened via a ProgressDialog.open() method call.

Implementation of DialogFragment communicates similar to other fragment to the activity. A good practice is that the fragment defines an interface which the activity must implement so that the fragment can call back into the activity without knowing the implementation details of the activity.

Example:

import android.app.AlertDialog;

import android.app.Dialog;

import android.app.DialogFragment;

import android.content.DialogInterface;

import android.os.Bundle;

import android.widget.Toast;

public class MyAlertDialogFragment extends DialogFragment {

@Override

public Dialog onCreateDialog(Bundle savedInstanceState) {

return new AlertDialog.Builder(getActivity())

// set dialog icon

.setIcon(android.R.drawable.stat\_notify\_error)

// set Dialog Title

.setTitle("Alert dialog fragment example")

// Set Dialog Message

.setMessage("This is a message")

// positive button

.setPositiveButton("OK", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

Toast.makeText(getActivity(),"Pressed OK",Toast.LENGTH\_SHORT).show();

}

})

// negative button

.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

Toast.makeText(getActivity(),"Cancel",Toast.LENGTH\_SHORT).show();

}

}).create();

}

}

import android.app.DialogFragment;

import android.os.Bundle;

import android.view.KeyEvent;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.view.WindowManager;

import android.widget.EditText;

import android.widget.TextView;

public class MyDialogFragment extends DialogFragment implements TextView.OnEditorActionListener {

private EditText mEditText;

public interface UserNameListener {

void onFinishUserDialog(String user);

}

// Empty constructor required for DialogFragment

public MyDialogFragment() {

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_username, container);

mEditText = (EditText) view.findViewById(R.id.username);

// set this instance as callback for editor action

mEditText.setOnEditorActionListener(this);

mEditText.requestFocus();

getDialog().getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT\_INPUT\_STATE\_VISIBLE);

getDialog().setTitle("Please enter username");

return view;

}

@Override

public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {

// Return input text to activity

UserNameListener activity = (UserNameListener) getActivity();

activity.onFinishUserDialog(mEditText.getText().toString());

this.dismiss();

return true;

}

}

Change the code of your activity to the following listing.

package dialogfragmentexample.android.vogella.com.dialogfragmentexample;

import android.app.Activity;

import android.app.Fragment;

import android.app.FragmentManager;

import android.os.Bundle;

import android.view.View;

import android.widget.Toast;

public class MainActivity extends Activity implements MyDialogFragment.UserNameListener {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

@Override

public void onFinishUserDialog(String user) {

Toast.makeText(this, "Hello, " + user, Toast.LENGTH\_SHORT).show();

}

public void onClick(View view) {

// close existing dialog fragments

FragmentManager manager = getFragmentManager();

Fragment frag = manager.findFragmentByTag("fragment\_edit\_name");

if (frag != null) {

manager.beginTransaction().remove(frag).commit();

}

switch (view.getId()) {

case R.id.showCustomFragment:

MyDialogFragment editNameDialog = new MyDialogFragment();

editNameDialog.show(manager, "fragment\_edit\_name");

break;

case R.id.showAlertDialogFragment:

MyAlertDialogFragment alertDialogFragment = new MyAlertDialogFragment();

alertDialogFragment.show(manager, "fragment\_edit\_name");

break;

}

}

}