ICFAI University, Dehradun

ASSIGNMENT – 2

Android App Development

By:-

Ashirbad Sarangi

18STCUDDN01008

Qus1- Describe drawable resources in android app development

Ans- A drawable resource is a general concept for a graphic that can be drawn to the screen and which you can retrieve with APIs such as getDrawable(int) or apply to another XML resource with attributes such as android:drawable and android:icon. There are several different types of drawables:

Bitmap File:- A bitmap graphic file (.png, .jpg, or .gif). Creates a BitmapDrawable.

Nine-Patch File:- A PNG file with stretchable regions to allow image resizing based on content (.9.png). Creates a NinePatchDrawable.

Layer List:- A Drawable that manages an array of other Drawables. These are drawn in array order, so the element with the largest index is be drawn on top. Creates a LayerDrawable.

State List:- An XML file that references different bitmap graphics for different states (for example, to use a different image when a button is pressed). Creates a StateListDrawable.

Level List:- An XML file that defines a drawable that manages a number of alternate Drawables, each assigned a maximum numerical value. Creates a LevelListDrawable.

Transition Drawable:- An XML file that defines a drawable that can cross-fade between two drawable resources. Creates a TransitionDrawable.

Inset Drawable:- An XML file that defines a drawable that insets another drawable by a specified distance. This is useful when a View needs a background drawable that is smaller than the View's actual bounds.

Clip Drawable:-An XML file that defines a drawable that clips another Drawable based on this Drawable's current level value. Creates a ClipDrawable.

Scale Drawable:- An XML file that defines a drawable that changes the size of another Drawable based on its current level value. Creates a ScaleDrawable

Shape Drawable:- An XML file that defines a geometric shape, including colors and gradients. Creates a GradientDrawable.

Qus2-Explain the procedure how to create drawable resources

Ans-Step 1: Go to the **app > res > drawable** and **right-click** on it. Please refer to the screenshot below to get a clear cut view of the steps.

Step 2: After right-clicking on the drawable file go to **New > Drawable resource.**.

Step 3: When you click on the **Drawable resource file** a dialog box will open on your computer screen. Enter the **file name** in the text box and then click on **OK.**

Note: File names must start with a lowercase letter.

Step 4: After that the drawable resource XML file will be created and one can find the file in **app > res > drawable** as shown in the figure below. In this case, we have named the file as **round_button**.

Step 5: Now click on the file name and one can customize the views by writing the necessary codes inside this file.

Qus3-Describe the process of handling soft keyboard states

Ans-The Android system shows an on-screen keyboard, known as a soft input method, when a text field in your UI receives focus. To provide the best user experience, you can specify characteristics about the type of input you expect (such as whether it's a phone number or email address) and how the input method should behave (such as whether it performs auto-correct for spelling mistakes).

Displaying the Soft Keyboard

AVD Manager:- By default, the soft keyboard may not appear on the emulator. If you want to test with the soft keyboard, be sure to open up the Android Virtual Device Manager (Tools => Android => AVD Manager) and uncheck "Enable Keyboard Input" for your emulator.

Showing Soft Keyboard Programmatically

The following code will reveal the soft keyboard focused on a specified view:

Java:-

```
public void showSoftKeyboard(View view){
```

```
    if(view.requestFocus()){
        InputMethodManager imm = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
        imm.showSoftInput(view,InputMethodManager.SHOW_IMPLICIT);
    }}
```

Kotlin:-

```
fun showSoftKeyboard(view: View) {
    if (view.requestFocus()) {
        val imm: InputMethodManager =
            getSystemService(Context.INPUT_METHOD_SERVICE) as InputMethodManager
        imm.showSoftInput(view, InputMethodManager.SHOW_IMPLICIT)
    }
}
```

Hiding the Soft Keyboard Programmatically

You can force Android to hide the virtual keyboard using the InputMethodManager, calling hideSoftInputFromWindow, passing in the token of the window containing your edit field.

Java:

```
public void hideSoftKeyboard(View view){
  InputMethodManager imm =(InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
  imm.hideSoftInputFromWindow(view.getWindowToken(), 0);
}
```

Kotlin

```
fun hideSoftKeyboard(view: View) {
    val imm =
```

```
        getSystemService(Context.INPUT_METHOD_SERVICE) as InputMethodManager

    imm.hideSoftInputFromWindow(view.windowToken, 0)

}
```

Adding a "Done" Key

In the keyboard, you can hide the "Next" key and add "Done" instead by adding the following to the imeOptions for the EditText view:

```
<EditText

  android:imeOptions="actionDone">

</EditText>
```

or in Java:

Java

```
myEditText.setImeOptions(EditorInfo.IME_ACTION_DONE);
```

Kotlin

```
myEditText.setImeOptions(EditorInfo.IME_ACTION_DONE)
```

Configuring the Soft Keyboard Mode

The soft keyboard can be configured for each activity within the AndroidManifest.xml file using the android:windowSoftInputMode attribute to adjust both default visibility and also how the keyboard affects the UI when displayed.

Showing the Keyboard when Activity Starts

Although Android gives focus to the first text field in your layout when the activity starts, it does not show the soft keyboard. To show the keyboard when your activity starts, add the android:windowSoftInputMode attribute to the <activity> element with the "stateVisible" value within the Android manifest. Check out this guide for more details. Within the AndroidManifest.xml file:

```
<activity
```

```
    android:name="com.example.myactivity"
    android:windowSoftInputMode="stateVisible" />
```

The options for the mode include two aspects: visibility of the keyboard and adjustment of the UI. Visibility options include stateUnchanged, stateHidden, stateVisible and several others listed here.

Changing UI Reaction

The virtual keyboard reduces the amount of space available for your app's UI. We can also use this same android:windowSoftInputMode property within the <activity> node to change the way that the soft keyboard displays the view elements when appearing within the AndroidManifest.xml file:

```
<!-- Configures the UI to be resized to make room for the keyboard -->
<activity
    android:name="com.example.myactivity"
    android:windowSoftInputMode="adjustResize" />
```

The options for the mode include two aspects: visibility and adjustment. Adjustment options include adjustResize, adjustPan, and adjustUnspecified and are listed in full here. Both visibility and adjustment can be combined with:

```
<!-- Configures the keyboard to be visible right away and for UI to be resized when shown -->
<activity
    android:name="com.example.myactivity"
    android:windowSoftInputMode="stateVisible|adjustResize" />
```

See the guide on keyboard visibility for more details.

Troubleshooting

Toolbar Height Expands on UI Resize

To avoid incorrect Toolbar height calculations, you can add android:fitsSystemWindows="true" (learn more) to the parent layout of the Toolbar. In many cases, this should resolve the issue.

Qus 4- Explain different string resources

A string resource provides text strings for your application with optional text styling and formatting. There are three types of resources that can provide your application with strings:

String

XML resource that provides a single string.

String Array

XML resource that provides an array of strings.

Quantity Strings (Plurals)

XML resource that carries different strings for pluralization.

All strings are capable of applying some styling markup and formatting arguments. For information about styling and formatting strings, see the section about Formatting and Styling.

Qus 5-Explain different view group properties

The ViewGroup is a subclass of View and it will act as a base class for layouts and layouts parameters. The ViewGroup will provide an invisible containers to hold other Views or ViewGroups and to define the layout properties.

For example, Linear Layout is the ViewGroup that contains a UI controls like button, textview, etc. and other layouts also.

Following are the commonly used ViewGroup subclasses in android applications.

- **Linear Layout:-**In android, LinearLayout is a ViewGroup subclass which is used to render all child View instances one by one either in Horizontal direction or Vertical direction based on the orientation property.
- **Relative Layout:-**In android, RelativeLayout is a ViewGroup which is used to specify the position of child View instances relative to each other (Child A to the left of Child B) or relative to the parent (Aligned to the top of parent).

- **Table Layout:**-In android, TableLayout will position its children elements into rows and columns and it won't display any border lines for rows, columns or cells.

  The TableLayout in android will work the same as the HTML table and the table will have as many columns as the row with the most cells. The TableLayout can be explained as <table> and TableRow is like a <tr> element.

- **Frame Layout:-**In android, Framelayout is a ViewGroup subclass that is used to specify the position of View instances it contains on the top of each other to display only single View inside the FrameLayout.

  In simple manner, we can say FrameLayout is designed to block out an area on the screen to display a single item

- **Web View:-I**n android, WebView is an extension of View class and it is used to show the static HTML web pages content or remote web pages content with URL in android applications as a part of our activity layout.

  Generally, in android, WebView will act as an embedded browser to include the web pages content in our activity layout and it won't contain any features of normal browsers, such as address bar, navigation controls, etc.

- **List View:-**In android, ListView is a ViewGroup that is used to display the list of scrollable items in multiple rows and the list items are automatically inserted to the list using an adapter.

  Generally, the adapter pulls data from sources such as an array or database and converts each item into a result view and that's placed into the list

- **Grid View:-**In android, Grid View is a ViewGroup that is used to display items in a two dimensional, scrollable grid and grid items are automatically inserted to the gridview layout using a list adapter.

  Generally, the adapter pulls data from sources such as an array or database and converts each item into a result view and that's placed into the list.

Qus6-Describe pop-up menu & Pop-up windows

PopupMenu

public class PopupMenu

extends Object

java.lang.Object

android.widget.PopupMenu

A PopupMenu displays a Menu in a modal popup window anchored to a View. The popup will appear below the anchor view if there is room, or above it if there is not. If the IME is visible the popup will not overlap it until it is touched. Touching outside of the popup will dismiss it.

PopupWindow

Java

```
public class PopupWindow


extends Object
```

java.lang.Object

android.widget.PopupWindow

This class represents a popup window that can be used to display an arbitrary view. The popup window is a floating container that appears on top of the current activity.

Qus-7 Explain toast and customizing toast

Ans- In Android, Toast is used to display information for a period of time. It contains a message to be displayed quickly and disappears after a specified period of time. It does not block the user interaction. Toast is a subclass of Object class. In this we use two constants for setting the duration for the Toast. Toast notification in android always appears near the bottom of the screen. We can also create our custom toast by using custom layout(xml file).

In Android, Sometimes simple Toast may not be satisfactory, and then we can go for customizing a Toast. For creating a custom layout, define a View layout, in XML and pass the

root View object to the setView(View) method.

Qus 8- What is intent? Also describe types of intent

Ans- Intent is to perform an action. It is mostly used to start activity, send broadcast receiver, start services and send message between two activities. There are two intents available in android as Implicit Intents and Explicit Intents.

Explicit Intent − It going to connect the internal world of an application such as start activity or send data between two activities. To start new activity we have to create Intent object and pass source activity and destination activity as shown below −

```
Intent send = new Intent(MainActivity.this, SecondActivity.class);

startActivity(send);
```

And we should declare about second activity in Manifest.xml file or else it going to show run time exception. sample declaration is as shown below.

```
<activity android:name = ".SecondActivity"></activity>
```

Implicit Intents − It going to connect with out side application such as call, mail, phone,see any website ..etc. In implicit intent we have to pass an action using setAction() as shown below example.

```
Intent i = new Intent();

i.setAction(Intent.ACTION_VIEW);

i.setData(Uri.parse("www.tutorialspoint.com"));

startActivity(i);
```

In the above example we are giving action as view. so it going to show something which we have given in setData method.

setData() - This method is only to specifies a URI.

setType()- This method specifies a MIME type.

setDataAndType()- This method i specifies both a URI and a MIME type.

Qus 9-Describe intent filter with the process of defining your own intent

Ans-An Intent is a messaging object you can use to request an action from another app component. Although intents facilitate communication between components in several ways, there are three fundamental use cases:

● Starting an activity

An Activity represents a single screen in an app. You can start a new instance of an Activity by passing an

Intent to startActivity(). The Intent describes the activity to start and carries any necessary data.

If you want to receive a result from the activity when it finishes, call startActivityForResult(). Your activity

receives the result as a separate Intent object in your activity's onActivityResult() callback.

● Starting a service

A Service is a component that performs operations in the background without a user interface. With Android

5.0 (API level 21) and later, you can start a service with JobScheduler.

For versions earlier than Android 5.0 (API level 21), you can start a service by using methods of the Service

class. You can start a service to perform a one-time operation (such as downloading a file) by passing an

Intent to startService(). The Intent describes the service to start and carries any necessary data.

If the service is designed with a client-server int

Qus 10- Describe XML or Jason-parsing

Ans-JSON-JSON is used to store information in an organized, and easy-to-access manner. Its full form is JavaScript Object Notation. It offers a human-readable collection of data which can be accessed logically.

Features of JSON

- **Easy to use** - JSON API offers high-level facade, which helps you to simplify commonly used use-cases.
- **Performance** - JSON is quite fast as it consumes very less memory space, which is especially suitable for large object graphs or systems.
- **Free tool** - JSON library is open source and free to use.
- **Doesn't require to create mapping** - Jackson API provides default mapping for many objects to be serialized.
- **Clean JSON** - Creates clean, and compatible JSON result that is easy to read.
- **Dependency -** JSON library does not require any other library for processing

XML-XML is a markup language which is designed to store data. It's popularly used or transfer of data. It is case sensitive. XML offers you to define markup elements and generate customized markup language. The basic unit in the XML is known as an element. Extension of XML file is .xml

Features of XML

- XML tags are not predefined. You need to define your customized tags.
- XML was designed to carry data, not allows you to display that data.
- Mark-up code of XML is easy to understand for a human.
- Well, the structured format is easy to read and write from programs.
- XML is an extensible markup language like HTML.

Qus 11- Describe the process of ping of database with apps

Ans-- Routinely conducting ping sweeps has a variety of benefits. Pinging informs IT professionals about the state of their network: the availability status of all devices, latency rate of network requests, percentage of data packet loss, and more.

For this reason, it’s important to understand the history of ping and how the ping utility works. I also make my case around the importance of using ping monitor software like Engineer’s Toolset™ to enhance ping capabilities and help support your ability to provide reliable network performance.

Qus 12- What is cursor and process of cursor loader

Ans-Loaders are basically used to provide asynchronous loading of data for an Activity of Fragment on Non-UI thread.While the application should perform any call to a Loader from the main thread, the Loader (or subclasses of Loader) performs their work in a separate thread and delivers its results to the main thread.

● The code implementation should not derive directly from android.content.Loader class but specifically fromandroid.content.CursorLoader class.

● The callbacks of the Loader are invoked at different stages during loading of data in an Activity or Fragment. Inshort, an Activity or a Fragment are required to implement Listeners to use Loaders.

● Loaders internally use AsyncTask to perform the data load. There is no performance gain when Loaders are compared to AsyncTask, provided that the AsyncTask are designed and developed properly.

● Loader, more specifically, CursorLoader queries the Content Resolver in the background thread so that the application#39;s User Interface is not blocked and returns the loaded Cursor to the Activity or Fragment.

● CursorLoader implements the Loader protocol for querying cursors.

● CursorLoader handles the life cycle of the cursor. When using CursorLoader, the developer should never call close() on the cursor.

Qus 13- Describe the Process of using internal storage and external storage for android apps

Ans-Under Android the on disk storage is split into two areas: internal storage and external storage.Often the external storage is physically removable like an SD card, but it need not be. The distinction between internal and external storage is actually about the way access to the files is controlled.

Internal storage-By default any files that your app writes to the internal storage of an Android device are only accessible by your app. This protection is enforced by the Linux kernel through the app sandbox (as described in Week 3), and includes any files created indirectly by SQLite databases etc. If you want to share the content of your files with other apps you should use a Content Provider.

External storage-Files created on external storage are world readable and writeable, i.e. any app can read or write to them. Indeed, since external storage can (often) be removed from the device and connected to any other computer, it is not possible to enforce access control for files on external storage.

Qus 14- Describe Web Chrome Client-

Ans- WebViewClient is an interface for responding to rendering events; WebChromeClient is an event interface for reacting to events that should change elements of chrome around the browser. This includes JavaScript alerts, favicons, and of course updates for loading progress and the title of the current page.

Qus 15- Describe Web view Client

Ans- WebView is a view that displays web pages inside your application. You can also specify an HTML string and can show it inside your application using WebView. WebView turns your application to a web application.

In order to add WebView to your application, you have to add <WebView> element to your xml layout file.

Q.16 Describe using app cache

Ans- When applications are cached, updating the resources (files) that are used in a web page is not enough to update the files that have been cached. You must update the cache manifest file itself before the browser retrieves and uses the updated files. You can do this programmatically using window.applicationCache.swapCache(), though resources that have already been loaded will not be affected. To make sure that resources are loaded from a new version of the application cache, refreshing the page is ideal.

Qus 17- Explain memory management of Bitmap

Ans- Memory is divided up into allocation units using a bitmap.

Corresponding to each allocation unit is a bit in the bitmap, which is zero (0) only if the unit is free andone (1) only if the unit is occupied and vice-versa.

An important design issue is the size of the allocation unit.

The smaller the allocation unit, the larger the bitmap.

A bitmap gives an easy way to keep track of memory words in a fixed amount of memory only because the bitmap size depends on the memory size and the allocation unit size.