Date

Android Layout

Linear Layout

Android linear layout is a group in view class that align all children in either vertical or horizontal order

Some attributes of linear layout are

1) Android :id this is the id which uniquely identifies the layout

2) android : baseline-Aligned this must be a boolean value either true or false and prevents the layout from aligning its children baselines.

3) android : baseline Aligned Child Index When a linear layout is a part of another layout that is baseline align, it can specify which of its children to baseline Align.

4) android : divider this is a drawable to use a vertical dividers between buttons. It takes a color value as an input in the form of rgb.

5) android : gravity this specifies how an object should position its content on both x and y axis. Possible values are top, bottom, left, right, center, center-vertical, center-horizontal etc

Date

1) android : orientation this specifies how an object should be positioned on both x and y axis. Posible values are vertical, for column and horizontal for row. Default value is horizontal

2) android : weightSum it sums up all the childs weight

Relative Layout

Relative layout enables the developer to specify how child use our position relative to each other. The position of each view can be specified as relative to sibling element or relative to the parent

Some attributes of relative layout are

1) android : id it specifies a unique id to uniquely identify the layout.

2) android : gravity same as linear layout

3) android : ignoreGravity this indicates what view should not be effected by the gravity.

3) android : layout_above it positions the bottom edge of this view above the given anchor view id. and must be a reference to another resource

4) android : layout_alignBottom : it makes the bottom

Spiral

Date

edge of this view to match the bottom edge of the given anchor view id and it must be a reference to another resource.

5) android : layout_alignLeft It makes the left edge of this view to match the left edge of the given anchor view id. and must be a reference to another resource

6) android : layout_alignParentBottom if true, it makes the bottom edge of this view to match bottom edge of the parent. It must be a boolean value either true or false

7) android : layout_alignParentEnd if true it makes the end edge of the view to match the end edge of the parent. Must be a boolean value either true or false

8) android : layout_alignParentLeft if true it makes the left edge of the view to match the left edge of the parent. must be a boolean value either true or false

9) android : layout_alignParentRight if true it makes the right edge of the view to match the right edge of the parent. must be a boolean value either true or false

10) android : layout_alignParentStart

11) android : layout_alignParentTop

12) android : layout_alignRight it makes the right edge of the view to match right edge of the given anchor view id and must be a ref

Spiral

Date

13) android: layout_alignStart: it makes the start edge of this view matches the start edge of the given anchor id and it must be a reference to another resource

14) android: layout_alignTop: it makes the top edge of this view matches the top edge of the given anchor view id and it must be a reference to another resource

15) android: layout_below: it positions the top edge of this view below the given anchor view id and it must be a reference to another resource

16) android: layout_centerHorizontal: if true, it centers this horizontally within it's parent. It must be a boolean value either true or false.

17) android: layout_centerInParent: if true, it centers this child horizontally and vertically with it's parent. It must be a boolean value either true or false

18) android: layout_centerVertical: if true, it center this child vertically within it's parent. It must be a boolean value either true or false.

19) android: layout_toEndOf: it positions the start edge of this view to the end of the given anchor view id and it must be a reference to another resource

20) android: layout_toLeftOf: it positions the ~~left~~ right edge of this view to the left of the given anchor view id and it must be a reference to another resource

Spiral

Date

21) android: layout_toRightOf: it positions the left edge of this view to the right of the given anchor view id and it must be a reference to another resource

22) android: layout_toStartOf: it positions the end edge of this view to the start of the given anchor view id and it must be a reference to another resource.

Frame Layout

It is a view group subclass which is use to specify the position of multiple views placed on the top of each other to represent a single view screen. Generally frame layout simply blocks a particular area on the screen to display a single view. Under this all the child views or elements are added in stack format that means the most recently added child will be shown on the top of the screen. Some important attributes under frame layout are

1) android: id: this is the id which uniquely identifies the layout.

2) android: foreground: this defines the drawable to draw over the content and possible value may be a color value in RGB format.

3) android: foregroundGravity: it defines the gravity to apply the foreground drawable. The gravity default is fill. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc

4) android: measureAllChildren: it determines whether to measure all children or just those in the visible or invisible state when measuring. the default value is false

Spiral

Date

Absolute Layout

The absolute layout allows you to specify the exact location i.e x and y coordinates of its children with respect to the origin at the top left corner of the layout. The absolute layout is less flexible and harder to maintain for varing sizes of screen. That is the reason it is not recommended. Absolute layout is depriciated now.

Some imp attributes are

1) android:id : this is the id that uniquely specify the absolute layout.

2) android: layout_x : it specifies the x coordinate of the view. Possible value is in density pixel or pixel.

3) android: layout_y : it specifies y coordinate of the view Possible value is in density pixel or pixel.

~~4) android:id~~

Table Layout.

A table layout is a view group that arranges its children ie views and other layouts in a table form with rows and columns. To define a row table row tag is use. There is no need to mention no of columns in table layout as android automatically adds column as per the no of views and other layout added in the table rows.

There is no need to provide layout_width and layout_height for table rows because by default its width is match_parent and height is wrap_content

Spiral

Date

The width of the column automatically adjust based on the size of the column with max width.
Some imp attributes are

1) android:id : this is the id that uniquely identifies the table layout.

2) android: ~~shrink~~ stretch Columns: when a column width is less and you need to stretch it or expand it you will use this attribute

3) android: shrinkColumns: when a column width is high and you do need extra space on the column, this attribute can be use to shrink or to remove extra spaces.

4) android: collapseColumns: it hides the column of the given index in the table layout

5) android: layout_span: if a view takes only one column width but you want your view to take more than one column space, this attribute can be use.

6) android: layout_column: if you want your view is present in the first table row to appear below the other table rows, this attribute can be used

Constraint layout

constraint layout is similar to other view groups such as relative layout, linear layout etc.
It provides the ability to completely design the UI by drag and drop feature provided by android studio

Date

Advantages

design editor. It helps to improve the UI components over other layouts. With the help of constraint layout, the developers can easily add animations to the UI components deployed in any applications. Under constraint layout we can control the group of widgets through a single line of code.

Disadvantages

While constraint layout is used, the xml code generated automatically becomes a bit difficult to understand.

In most of the cases the result obtained will not be the same as we got to see in the design editor.

Sometimes the developers need to a separate layout file to handle the UI for the landscape mode.

Some imp attributes of the layout are

1) android:id this is the id which is used to give a unique id to the layout.

2) app:layout_constraintBottom_toBottomOf: this is used to constraint the view with respect to bottom position

3) app:layout_constraintLeft_toLeftOf: this is used to constraint the view with respect to left position

4) app:layout_constraintRight_toRightOf: this is used to constraint the view with respect to right position

Spiral

Date

5) app: layout_constraintTop_toTopOf: this is used to constraint the views with respect to top position.

Events

Events are the useful way to collect data about user's interaction with the application like button click, screen touch, etc.

The Android framework maintains an event queue as first in first out basis. You can capture these events in your application and take appropriate actions as per user requirements.

Three things related to android event management are

1) Event Listener: It is an interface in the view class that contains a single call back method. These methods are called by the android framework when the view to which the listener has been registered is triggered by the user interaction with the components of UI.

2) Event Listener Registration: Event Listener registeration is the process by which an event handler

Spiral

Date

registered with an event Listener so that event handler is called when event listener fires an event

3) Event Handlers: When an event happens and the programmer has to register an event for the event occurrence, Event Listener calls the event handler which is the method that actually handles the event

Some popular Event Listeners and Handlers are as follows

Handler	Listener
onClick()	onClickListener() — This method is called when the user either clicks or touches or focuses on any widget like button, text, image etc. You can use onClick Event handler to handle such events
onLongClick()	onLongClickListener — This method is called when the user either clicks or touches or focuses on any widget like text, button, images, etc. for a longer duration of time. You can use onL
onTouch()	onTouchListener() — This method is called when the user press any key releases the key or any movement gestures on the screen

Spiral

Date _____

onTouchEventHandler is used to handle such events.

OnMenuItemClick ()

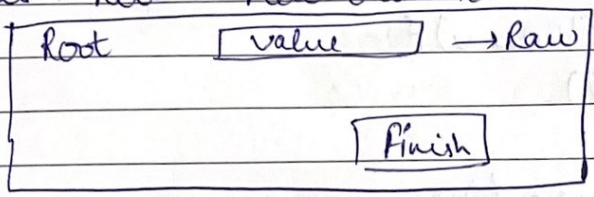
onMenuItemClickListener (.) =
This method is called when the user selects a menu item. onMenuItemClick event handler is used to handle such events.

There are many more eventlisteners and eventHandlers as a part of View class like onHoverListener, onDragListeners etc.

Date

Media Player

res → new → new resource directory



Android studio facilitates many ways to control playback of audios and videos. One of the way is to call an inbuilt media player class. Media player class provides services like playing audio/video, pausing, running video, restarting resuming and terminating the running process. To run a media player class we need to call an inbuilt function create

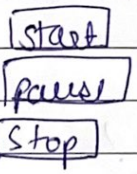
```
MediaPlayer mp =
    MediaPlayer.create (this, R.raw.filename);
```

Q Develop an android application to create an audio media player

XML

```
<LinearLayout
    orientation = "vertical"
    gravity = "center" />

<Button
    height =
    width =
    onClick = "start" />
```



Java

```
MediaPlayer player;
= onCreat
= ——— }
public void start(View v) {
    if (player == null)
    {
        player = MediaPlayer.create (this, R.raw.audio);
```

filename

```
        PlayerStart();
    }

    public void pause(View v){
        if(Player!=null)
        {
            Player.pause();
        }
    }

    public void stop(View v){


    }

    private void MediaRelease()
    {
        if(Player!=null)
        {
            Player.release();
            Player=null;
        }
        Toast.makeToast(this, "msg", Toast.LENGTH_LONG);
    }
}
```

Some methods related to media player class are

1) isPlaying() - this method will return true or false value indicating that audio is playing or not

2) SeekTo() - this method returns true or false value to move the location of song on to the particular position on the screen

3) getCurrentPosition() - this method returns the current position of the song in millisecords.

4) getDuration() - this method returns the total time consumed by an audio in millisecords

5) reset() - this method resets the media player

6) release() - this method releases any resource occupied the media player.

7) setVolume() - this method sets the volume for the current running media player

8) selectTrack() - this method take an integer value and selects the track from the list on the particular index.

9) getTrackInfo() - this method returns an array of track info.

10) setDataSource() - this method sets the data of the current running audio/video file.

Video View Class

Date

xml

<LinearLayout
---/>

<VideoView
=
=
id= @+id/v1 />

Java

VideoView vv1;

Button v1 = findViewById(R.id.v1);
String path = "android.resource://" + getPackageName() + "/" + R.raw.video_name;

Uri u = Uri.parse(path);

vv1.setVideoURI(u);

MediaController m = new MediaController(this);
vv1.setMediaController(m);
m.setAnchorView(vv1);

VideoView class in android provides a container to display video files. It can load images from various sources such as resources or content providers taking care of computing its measurement from the video so that it can be used for any layout manager providing display options.

Date

scaling and tempting. VideoView class does not retain its full state when going into background. In particular it does restore it's current state. Applications should save and restore its current ongoing states.

Methods in VideoView class

1) setVideoURI(URI)
this method is used to set the absolute path of the video file which is going to be played this method takes URI as an argument.

2) setMediaController
this method of video view class is used to set the controls of view playback.

3) start()
this method is used to start the playback of a particular view file.

4) pause()
this method of video view class is used to pause the current ongoing file.

5) canPause()
this method of video view class tells whether the video view can be paused with the ongoing running file. This method returns a boolean value either false or true.

6) canSeekForward()
this method of video view class

Date

The video is able to seek forward. It returns a boolean value either true or false.

7) getDuration()
This method is use to get the total duration of video view. It returns an integer value

8) getCurrentPosition()
This method is used to fetch the current position of playback. It returns an integer value.

9) isPlaying()
this method tells whether the video is currently playing or not. It returns a boolean value either true or false

10) stopPlayback()
this method of video view is use to stop the video playback

11) setOnPrepareListener()
This method is a listener which allows a callback method to call the video to be ready to played.

12) setOnErrorListener()
This method allows a callback () to be called when an error occurs during the video playback.

Spiral

Date

13) setOnCompletionListener()
This listener allows a callback method to be called when the end of the video is reached

Methods of Media Controller class in Android

1) setAnchorView()
this method is use to designate the view to which the controller is to be anchored. This controls the location of the controllers on the screen

2) show()
this method is use to show the controllers on the screen

3) show(int timeout)
this method is use to set the time to show the controllers on the screen

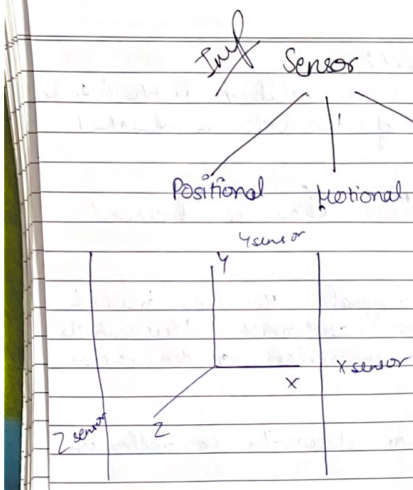
4) hide()
this method is used to hide the video controller from the screen

5) isShowing()
this method returns a boolean value indicating whether the controls are visible or not.

6 Develop an android application to include Video view class and implement video playback controls on it.

Spiral

Date



Q Develop an android app to enlist all the inbuild sensors in a particular device.

xml `<textview id="@+id/tv" />`

Java
```
TextView tv;
SensorManager sm
tv = findViewById(R.id.tv);
sm = (SensorManager) getSystemService(SENSOR_SYSTEM);
List<Sensor> ml = sm.getSensorList(Sensor.TYPE_ALL);
```

Spiral

Date

```
for (int i=1; i<ml.size(); i++)
{
    tv.append("\n" + ml.get(i).getName());
}
}
```

Sensor Managers
Sensors can be used to monitor three dimensional device movement or change in the environment of the device. Android provides APIs to work with different types of sensors.

Android supports three types of sensors
Motion sensors
These sensors measures acceleration forces and rotational forces along three axis.
This category include gravity sensors, rotation vector sensors etc.

Position sensors
These sensors measures the physical position of a device
This category includes orientation sensors and magneto meters.

Environmental sensors
These sensors measures various environmental parameters such as ambient air temperature and pressure, humidity
This category includes barometers, photometers & thermometers

Android Sensor APIs

Spiral

Date

The raw data can be collected by using android sensor API. It provides many classes and interfaces some of the imp classes and interfaces are

SensorManager class
It is used to access various sensors present in the device

Sensor class
This class is used to get information about the sensor such as sensor name, sensor type, sensor resolution etc.

SensorEvent class
This class is used to find information about the sensor

SensorEventListener ~~Class~~ interface
This is used to perform some actions when sensor accuracy changes

Spiral

Date

Android Layout

Linear Layout
Android linear layout is a group in view class that aligns all children in either vertical or horizontal order

Some attributes of linear layout are
1) Android:id this is the id which uniquely identifies the layout

2) android: baseline-Aligned this must be a boolean value either true or false and prevents the layout from aligning its children baselines.

3) android: baselineAlignedChildIndex When a linear layout is a part of another layout that is baseline align, it can specify which of its children to baseline Align.

4) android: divider this is a drawable to use a vertical dividers between buttons. It takes a color value as an input in the form of rgb

5) android: gravity this specifies how an object should position its content on both x and y axis. Possible values are top, bottom, left, right, center, center-vertical center-horizontal etc

Spiral

Date

6> android : orientation this specifies how an object should be positioned (x or both x and y axis. Posible values are vertical for column and horizontal for row. Default value is horizontal

7> android : weightSum it sums up all the childs weight

Relative Layout

Relative layout enables the developer to specify how child use are position relative to each other. The position of each view can be specified as relative to sibling element or relative to the parent

Some attributes of relative layout are

1> android : id it specifies a unique id to uniquely identify the layout.

2> android : gravity same as linear layout

3> android : ignoreGravity this indicates what view should not be effected by the gravity.

4> android : layout_above it positions the bottom edge of this view above the given anchor view id. and must be a reference to another resource

4> android : layout_alignBottom : it makes the bottom

Spiral

Date

edge of this view to match the bottom edge of the given anchor view id and it must be a reference to another resource.

5> android : layout_alignLeft It makes the left edge of this view to match the left edge of the given anchor view id. and must be a reference to another resource.

6> android : layout_alignParentBottom if true, it makes the bottom edge of this view to match bottom edge of the parent. It must be a boolean value either true or false

7> android : layout_alignParentEnd if true it makes the end edge of the view to match the end edge of the parent. Must be a boolean value either true or false

8> android : layout_alignParentLeft if true it makes the left edge of the view to match the left edge of the parent. must be a boolean value either true or false

9> android : layout_alignParentRight if true it makes the right edge of the view to match the right edge of the parent. must be a boolean value either true or false

10> android : layout_alignParentStart

11> android : layout_alignParentTop

12> android : layout_alignRight it makes the right edge of the view to match right edge of the given anchor view id and must be a ref

Spiral

Date

13) android:layout_alignStart: it makes the start edge of this view to matches the start edge of the given anchor id and it must be a reference to another resource

14) android:layout_alignTop: it makes the top edge of this view to matches the top edge of the given anchor view id and it must be a reference to another resource

15) android:layout_below: it positions the top edge of this view below to the given anchor view id and it must be a reference to another resource

16) android:layout_centerHorizontal: if true, it centers this horizontally within it's parent. It must be a boolean value either true or false.

17) android:layout_centerInParent: if true, it centers this child horizontally and vertically with it's parent. It must be a boolean value either true or false

18) android:layout_centerVertical: if true, it center this child vertically within it's parent. It must be a boolean value either true or false.

19) android:layout_toEndOf: it positions the start edge of this view to the end of the given anchor view id and it must be a reference to another resource

20) android:layout_toLeftOf: it positions the right edge of this view to the left of the given anchor view id and it must be a reference to another resource

Spiral

Date

21) android:layout_toRightOf: it positions the left edge of this view to the right of the given anchor view id and it must be a reference to another resource

22) android:layout_toStartOf: it positions the end edge of this view to the start of the given anchor view id and it must be a reference to another resource.

Frame Layout

It is a view group subclass which is use to specify the position of multiple views placed on the top of each other to represent a single view screen. Generally frame layout simply blocks a particular area on the screen to display a single view. Under this all the child view or elements are added in stack format that means the most recently added child will be shown on the top of the screen. Some important attributes under frame layout are

1) android:id: this is the id which uniquely identifies the layout.

2) android:foreground: this defines the drawable to draw over the content and possible value may be a color value in RGB format.

3) android:foregroundGravity: it defines the gravity to apply the foreground drawable. The gravity default is fill. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc

4) android:measureAllChildren: it determines whether to measure all children or just those in the visible or invisible state

Spiral

Date

Absolute Layout

The absolute layout allows you to specify the exact location i.e x and y coordinates of its children with respect to the origin at the top left corner of the layout. The absolute layout is less flexible and harder to maintain for varing sizes of screen. That is the reason it is not recommended. Absolute layout is depriciated now.
Some imp attributes are

1) android:id : this is the id that uniquely specify the absolute layout.

2) android: layout_x : it specifies the x coordinate of the view. Possible value is in density pixel or pixel.

3) android: layout_y : it specifies y coordinate of the view Possible value is in density pixel or pixel.

Table Layout.

A table layout is a view group that arranges its children if views and other layouts in a table form with rows and columns. To define a row table row tag is use. There is no need to mention no of columns in table layout as android automatically adds column as per the no of views and other layout added in the table rows.
There is no need to provide layout_width and layout_height for table rows because by default its width is match_parent and height is wrap_content

Spiral

Date

The width of the column automatically adjust based on the size of the column with max width.
Some imp attributes are

1) android:id : this is the id that uniquely identifies the table layout.

2) android: ~~shrink~~ stretch Columns: when a column width is less and you need to stretch it or expand it you will use this attribute.

3) android: shrinkColumns: when a column width is high and you do need extra space on the column, this attribute can be use to shrink or to remove extra spaces.

4) android: collapseColumns: it hides the column of the given index in the table layout

5) android: layout_span: if a view takes only one column width but you want your view to take more than one column space, this attribute can be use.

6) android: layout_column: if you want your view to present in the first table row to appear below the other table rows, this attribute can be used

Constraint layout

constraint layout is similar to other view groups such as relative layout, linear layout etc.
It provides the ability to completely design the UI by drag and drop feature provided by android studio

Date

Advantages

design editor. It helps to improve the UI components over other layouts. With the help of constraint layout, the developers can easily add animations to the UI components deployed in any applications. Under constraint layout we can control the group of widgets through a single line of code.

Disadvantages.

While constraint layout is used, the xml code generated automatically becomes a bit difficult to understand.

In most of the cases the result obtained will not be the same as we got to see in the design editor.

Sometimes the developers need to a separate layout file to handle the UI for the landscape mode.

Some imp attributes of the layout are

1) android:id this is the id which is used to give a unique id to the layout.

2) app: layout_constraintBottom_toBottomOf: this is used to constraint the view with respect to bottom position

3) app: layout_constraintLeft_toLeftOf: this is used to constraint the view with respect to left position

4) app: layout_constraintRight_toRightOf: this is used to constraint the view with respect to right position

Spiral

Date

SQLITE

is another data storage available in android where we can store data in the user's device and can use it any time when required.

SQlite Databases in android application are used to perform 4 basic operation known as CRUD where
C - create R - read U - update D - delete

SQlite Database is an open source database which stores data inside user's device in form of text file. It's an offline database that is locally stored in the user's device and there is no need to create any connection to connect database.

Data is stored in SQLite database in the form of tables. The tables are arranged similar to as excel sheets.

Some important methods are

1) getColumnName():- This method is used to get the of column name of SQLite tables.

2) getCount():- This method will return the no of rows on the cursor

3) isClosed():- This method returns a boolean value when the cursor is closed

4) getColumnCount():- This method returns the total number of column present in the table.

Spiral

5) This method will return the name of the column when a particular index is passed inside this method.

Date

6) getColumnIndex :- This method will return the index of the column when a particular index is passed from the name of the column

7) getPosition():- This method will return the current position of cursor in the table.

Database Helper class

This class is used for managing all the operations related to database, a helper class has been given in android studio and is known as SQLiteOpenHelper() it automatically manages the creation and updation of database.

Some methods of SQLiteOpenHelper class are

1) public void onCreate(SQLiteDatabase db):-
This method is called on once when the database is created for the first time.

2) public void onUpgrade():- This method is called when the database is needed to be upgraded.

3) public void synchronized close():- This method is called to close the database object

4) public void onDowngrade():- The method is called when database is required to be downgraded

5) void execSQL():- This method is used to execute SQL queries

Spiral

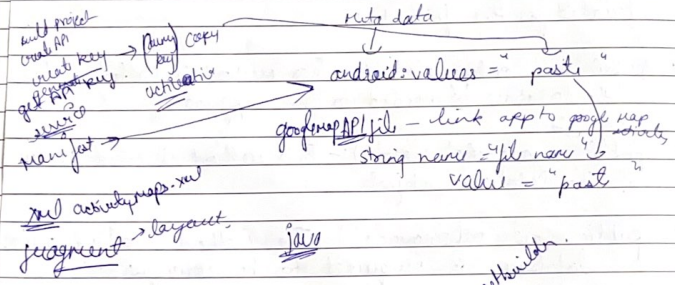
Date

6) Long Insert ():- This method is used to insert records in the database. The table specifies the tablename and it doesn't allow a complete null value. If the second argument is null Android will store a null but if all the values are null android will encounter an exception.

7) Int Update ():- This method is used to update is the rows of the table

8) Cursor query ():- It returns the cursor value over the result set

Q Develop an android app to implement google maps into your current running activities



Q Develop an android app to enable current location into your google maps activity.

Spiral

Date

Google Maps

Android provides the facility to integrate google maps in your application. Google maps are helpful in displaying your current location, navigate location directions, search a particular location etc.

Types of Google maps

There are four different types of google map as well as an optional to no map at all. each of them gives a different view these maps are as follows.

1) Normal Google Map

Spiral