Multimedia , Animation & Android - Google Maps

Animation is the process of adding a motion effect to any view, image, or text. With the help of an animation, you can add motion or can change the shape of a specific view. Animation in Android is generally used to give your UI a rich look and feel.

With parameters like start value, end value, size, time duration, rotation angle, etc., Animation performs the desired animation on an object. Animation in Android is provided by a class called Animation, to be applied to any type of object. In Android, various classes and interfaces are present in the android. animation package, for animation development. Animation in Android can be simply understood as the phenomenon of changing the object property and behavior at run time.

To do animation in android, we can use various ways. To start and end the animation, the methods of the AnimationDrawable class are used.

Animations:

● Animations can add visual cues that notify users about what's going on in your app. They are especially useful when the UI changes state, such as when new content loads or new actions become available.

● Animations also add a polished look to your app, which gives it a higher quality look and feel.

● Android includes different animation APIs depending on what type of animation you want, so this page provides an overview of the different ways you can add motion to your UI.

● In order to perform animation in android , we are going to call a static function loadAnimation() of the class AnimationUtils. We are going to receive the result in an instance of Animation Object. Its syntax is as follows –

```
Animation animation = AnimationUtils.loadAnimation(getApplicationContext(),

R.anim.myanimation);
```

```
Animation animation = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.myanimation);
```

Second parameter is the name of the our animation xml file. You have to create a new folder called anim under res directory and make an xml file under anim folder.

- startAnimation(Animation animation)
    - o load the animation from XML file.
- AnimationUtils
    - o Defines common utilities for working with animations.

Taking Photos:

● The Android way of delegating actions to other applications is to invoke an Intent that describes what you want done. This process involves three pieces: The Intent itself, a call to start the external Activity, and some code to handle the image data when focus returns to your activity.

● The Android Camera application encodes the photo in the return Intent delivered to onActivityResult() as a small Bitmap in the extras, under the key "data".

Animation class has many useful functions which are listed below

Sr.No	Method & Description
1	**start()** This method starts the animation.
2	**setDuration(long duration)** This method sets the duration of an animation.
3	**getDuration()** This method gets the duration which is set by above method
4	**end()** This method ends the animation.
5	**cancel()** This method cancels the animation.

In order to apply this animation to an object , we will just call the startAnimation() method of the object. Its syntax is –

```
ImageView image1 = (ImageView)findViewById(R.id.imageView1);
image.startAnimation(animation);
```

Media Player

https://www.youtube.com/watch?v=MG21u819x6U&t=11s

Android Animation tutorial

https://www.youtube.com/watch?v=LoluU2Jg0Es

https://www.w3schools.blog/android-animation

How to Record Audio or Voice in Android Studio

https://www.youtube.com/watch?v=3ffs2VbJ9JY

Android - Audio Capture

Android has a built in microphone through which you can capture audio and store it , or play it in your phone. There are many ways to do that but the most common way is through MediaRecorder class.

Android provides MediaRecorder class to record audio or video. In order to use MediaRecorder class ,you will first create an instance of MediaRecorder class. Its syntax is given below.

```
MediaRecorder myAudioRecorder = new MediaRecorder();
```

Now you will set the source , output and encoding format and output file. Their syntax is given below.

```
myAudioRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
myAudioRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
myAudioRecorder.setAudioEncoder(MediaRecorder.OutputFormat.AMR_NB);
myAudioRecorder.setOutputFile(outputFile);
```

After specifying the audio source and format and its output file, we can then call the two basic methods prepare and start to start recording the audio.

```
myAudioRecorder.prepare();
myAudioRecorder.start();
```

Apart from these methods , there are other methods listed in the MediaRecorder class that allows you more control over audio and video recording.

Sr.No	Method & description
1	setAudioSource() This method specifies the source of audio to be recorded
2	setVideoSource() This method specifies the source of video to be recorded
3	setOutputFormat() This method specifies the audio format in which audio to be stored
4	setAudioEncoder() This method specifies the audio encoder to be used
5	setOutputFile() This method configures the path to the file into which the recorded audio is to be stored
6	stop() This method stops the recording process.
7	release() This method should be called when the recorder instance is needed.

Android - Google Maps

Android allows us to integrate google maps in our application. You can show any location on the map , or can show different routes on the map e.t.c. You can also customize the map according to your choices.

Google Map - Layout file

Now you have to add the map fragment into xml layout file. Its syntax is given below −

```
<fragment
   android:id="@+id/map"
   android:name="com.google.android.gms.maps.MapFragment"
   android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

Google Map - AndroidManifest file

You need to add some permissions along with the Google Map API key in the AndroidManifest.XML file. Its syntax is given below −

```
<!--Permissions-->

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="com.google.android.providers.gsf.permission.
   READ_GSERVICES" />

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>

<!--Google MAP API key-->

<meta-data
   android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyDKymeBXNeiFWY5jRUejv6zItpmr2MVyQ0" />
```

Adding Marker

You can place a maker with some text over it displaying your location on the map. It can be done by via **addMarker()** method. Its syntax is given below –

```
final LatLng T = new LatLng(21 , 57);

Marker TP = googleMap.addMarker(new MarkerOptions()
    .position(T).title("Place Marker"));
```

Changing Map Type

You can also change the type of the MAP. There are four different types of map and each give a different view of the map. These types are Normal,Hybrid,Satellite and terrain. You can use them as below

```
googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);

googleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);

googleMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);

googleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
```

How to Get Google Maps API Key for FREE

https://www.youtube.com/watch?v=7LcYpdxAXRA