MAD

Assignment-6

1812401160001
Vishwas Acharya

Page No.:
Date: / /

1) What is Graphics ? How to use Drawable object in Android?

Ans - Graphics are visual images or designs on some surface, such as a wall, canvas, screen, paper, or stone to inform, illustrate, or entertain.

- Images that are generated by a computer are called compture graphics.

- Examples:- photographs, drawings, line art, graphs, diagrams, typography, number, symbols, etc.,

- Graphics are used in application when we add a menu icon like navigation bar or bottom navigation.

- How to use Drawable object

- A drawable resource is a general concept for a graphic that can be drawn to the screen and which you can retrive with APIs such as getDrawable(int)

- To use drawable object we must right click on drawble folder and select vector asset.

- The click on icon and you'll get

many different in-built icon and select whatever you want and add that into the drawable folder.

- Then you can use it in your ~~app~~ android application.

- **Types of Drawable**

  - ↳ Bitmap File
  - ↳ Nine-Patch File
  - ↳ Layer List
  - ↳ State List
  - ↳ Level List
  - ↳ Transistion Drawable
  - ↳ Inset Drawable
  - ↳ Clip Drawable
  - ↳ Scale Drawable
  - ↳ Shape Drawable

=> Hence, this is how you can use drawable object in android.

2) What is Hardware Acceleration?

Ans - In Android 3.0 (API level 11), the Android 2D rendering pipeline supports hardware acceleration, meaning that all drawing operations that are performed on a View's canvas use the GPU.

- Because of the increased resources

required to enable hardware acceleration, your app will consume more RAM.
- Hardware acceleration is enabled by default if your Target API level is >=14, but can also be explicitly enabled.
- If your application uses only standard views and Drawables, turning it on globally should not cause any adverse drawing effects.

- Control hardware acceleration

↳ We can control hardware acceleration at the following levels.

↳ Application
↳ Activity
↳ Window
↳ View

=> Hence, this is how a hardware acceleration works.

3) What is Animation? Explain how we can use it in Android.

Ans - 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.

- Animation 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.

- Animation in android is possible from many ways and the one and widely ~~to~~ used way of making animation called tweened animation.

- **Tween Animation**

↳ It takes some parameters such as start value, end value, size, time duration, rotation angle e.t.c. and perform the required animation on that object.

↳ It can be applied to any type of object. So in order to use this, android has provided us a class called Animation.

Page No.:
Date: / /

↳ 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.

- Syntax :-

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

↳ xml file.

- This animation class has many useful functions which are listed below.

- start()
- setDuration(long duration)
- getDuration()
- end()
- cancel()

- Hence, this is how animation is used and it's meaning in android.

4) Explain How to use Media Player and explain recording and playing sound in android.

Page No.:
Date: / /

Ans
- Android provides many ways to control playback of audio/video files and streams.
- One of this way is through a class called MediaPlayer.
- Android is providing mediaplayer class to access built-in mediaplayer services like playing audio, video etc.
- In order to use MediaPlayer, we have to call a static Method create() of this class.
- This method returns an instance of mediaplayer class.

syntax:
```
MediaPlayer mediaPlayer = MediaPlayer.create(this, R.raw.song);
```

- The second parameter is the name of the song that you want to play.
- You have to make a new folder under your project with name raw and place the music file into it.
- Once you have created the mediaplayer object you can call some methods to start or stop the music.
- These method's are :

```
mediaPlayer.start();
mediaPlayer.pause();
```

- On call to start() method, the music will start playing from the begining.

- If this method is called again after the pause() method, the music would start playing from where it is left and not from the beginning.

- In order to start music from the beginning, you have to call reset() method.

```
mediaPlayer.reset();
```

- Apart from the start and pause method, there are other methods provided by this class for better dealing with audio/video files.

  - isPlaying()
  - seekTo(position)
  - getCurrentPosition()
  - getDuration()
  - reset()
  - release()
  - setVolume(float leftVolume, float rightVolume)
  - setDataSource(FileDescriptor fd)
  - selectTrack(int index)
  - getTrackInfo()

=> Hence, this is the required explaination on how to use media player in Android

Page No.:
Date: / /

5) Explain How to use Camera and Record Video in Android.

Ans - To use camera and record video in android, we have two ways.

↳ Using existing android camera application in our application.
↳ Directly using Camera API provided by android in our application.

- **Using existing camera**

- We use MediaStore.ACTION_IMAGE_CAPTURE to launch an existing camera application installed on your phone.

```
Intent intent = new Intent(android.provider.
MediaStore.ACTION_IMAGE_CAPTURE);
```

- Apart from this, there are more available Intents provided by MediaStore.

- ACTTON_IMAGE_CAPTURE_SECURE
- ACTTON_VIDEO_CAPTURE
- EXTRA_SCREEN_ORIENTATION
- EXTRA_FULL_SCREEN
- INTENT_ACTION_VIDEO_CAMERA
- EXTRA SIZE_LIMIT

- Now we use the function startActivityForResult() to launch this activity and wait for its result.

startActivityForResult(intent, 0)

- This method has been defined in the activity class. we are calling it from main activity.

6) Explain How to publishing and distributing android application.

Ans - Android application publishing is a process that makes your Android applications available to users.
- It is the last phase of the Android application development process.
- Once you developed & fully tested your android app, you can start selling or distributing free using google play.
- Here is a simplified check list which will help you in launching your android application.

- Regression Testing
- Application Rating
- Targeted Regions
- Application size
- SDK & Screen compatibility
- Application Pricing
- Promotional content
- Build & upload release-ready APK
- Finalize application detail.

- documentry the process of publishing an app on google play store for the first time.

step-1] Sign-Up :- Sign up for an account on the android developer console. creating an account costs $25

step-2] Create a New Application :- On the developer console, select the publish an android application option.
Fill out the details : Title, short/Full discription

step-3] Prepare multimedia :- Screenshots, Hi-res icon, Feature graphic

step-4] Prepare Code for release :- Remove log statements remove the android: debuggable attribute from your manifest file.
set the android: version Code attribute in the manifest tag in manifest.xml

step-5] Build a release-ready, APK

step-6] Upload APK

step-7] Complete the checklist on the left until all the items have a green checkmarks.

-> APK Distribution through app marketplace
-> Distribute android app through website
-> Distributing your app by email
CA quick & easy way to release your apps is to send them to users by email

7) Explain how to versioning the android application.

Ans- Versioning is a critical component of your app upgrade & maintenance strategy.

- Versioning is important because:

↳ Users need to have specific information about the app version that is installed on their devices & the upgrade versions available for installation.

↳ Other apps: to including other apps that you publish as a suite- need to query the system for your app's version, to determine compatibility & identify dependencies

↳ Services through which you will publish your app may also need to query your app for its version, so that they can display the versions to users.

- The Android system user your app's

version information to protect against downgrades.

- The system does not use app version information to enforce restrictions on upgrades or compatibility of third-party apps.
- The android system does enforce system version compatibility as expressed by the minsdk version setting in the build files

- To define the version information for your app set values for the version setting in the gradle build files.

- VersionCode :- A positive integer used as an internal version number

- This number is used only to determine whether one versions is more recent than mother, with higher number indicating more versions

- VersionName :- A string used as the version number shown to users. This setting can be specified as a raw string or as a references to a string resource
- The value is a string so that you can describe the app version as a <major><minor><paint> string or as any other type of absolute or relative version identifier.

- There are two API level setting available:

1) minSdkVersion :- The minimum version of the android platform on which the app will run, specified by the platform's API level identifier.

2) targetSdkVersion :- Specifies the API level on which the app is designed to run.

- Example: How to get android application version name.

step-1] Create a new project in android studio go to file => New project & fill all required details to create a new project.

step-2] Modify res > layout > activity_main.xml

step-3] Modify src > MainActivity.java

8) Explain how to publishing the android application.

Ans - Android application publishing is a process that makes your android applications available to user.
- It is the last phase of the android application development process.

Page No.:
Date: / /

- Once you developed & fully tested your android application, you can start selling or distributing free using google play.
- Here is a simplified checklist which will help you in launching your android application.

- Regression Testing
- Application Rating
- Targeted Regions
- Application size
- SDK & screen compatibility
- Application Pricing
- Promotional content
- Build & upload release-ready
- Finalize application detail.

- Documenting the process of publishing an app on the google play store for the first time.

Step-1] Sign up :- Sign up for an account on the android developer console, creating an account costs

Step-2] Create a new application : On the developer console select the publish an android application option.

Fill out the details : Title, short/Full description

step-3] Prepare multimedia :- Screenshots, hires icon, feature graphics

step-4] Prepare code for release :- Remove log statements
Remove the android: debuggable attribute from your manifest file.

step-5] Build a release-ready APK :- The release-ready APK is different from the debug APK in that it is signed with certificate that is owned by the developer

step-6] Upload APK :- Go back to developer console & click on manage releases. Then create a production release & upload your signed APK.

step-7] Complete the checklist on the left until all the items have green checkmarks.
The console re-evaluates the checklist every time you click save draft in the top right.