Practical No. 9

Title: Android program to work with images and videos

Aim: Create an application to demonstrate images and videos components

Introduction

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 media player 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. Its syntax is as follows −

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 methods are listed below.

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

On call to **start()** method, the music will start playing from the beginning. 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. Its syntax is given below.

mediaPlayer.reset();

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

Sr.No	Method & description
1	**isPlaying()** This method just returns true/false indicating the song is playing or not

2	**seekTo(position)** This method takes an integer, and move song to that particular position millisecond
3	**getCurrentPosition()** This method returns the current position of song in milliseconds
4	**getDuration()** This method returns the total time duration of song in milliseconds
5	**reset()** This method resets the media player
6	**release()** This method releases any resource attached with MediaPlayer object
7	**setVolume(float leftVolume, float rightVolume)** This method sets the up down volume for this player
8	**setDataSource(FileDescriptor fd)** This method sets the data source of audio/video file
9	**selectTrack(int index)** This method takes an integer, and select the track from the list on that particular index
10	**getTrackInfo()**

	This method returns an array of track information

Exercise - Create an application to demonstrate example of image and video elements

Implementation:

Program:

1)activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <SurfaceView
        android:id="@+id/surfaceView"
        android:layout_width="match_parent"
        android:layout_height="248dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0"/>

    <SeekBar
        android:id="@+id/seekBar"
        android:layout_width="291dp"
        android:layout_height="22dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.451" />

    <Button
        android:id="@+id/btnPlay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/play"
        android:textSize="25sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.632" />

    <Button
        android:id="@+id/btnVidVw"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/video_view"
        android:textSize="25sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.75" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

2)activity_video_view.xml :-

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/main"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainVideoViewActivity">
  <VideoView
    android:id="@+id/videoView"
    android:layout_width="match_parent"
    android:layout_height="225dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.4"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.35" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

3) MainActivity.java :-

```
package com.example.demoimagevideo;

import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import android.widget.Button;
import android.widget.SeekBar;

import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.Timer;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity {

    Button button,button2;
    SurfaceView surfaceView;
    MediaPlayer mediaPlayer;
    SeekBar seekBar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });
        button=findViewById(R.id.btnPlay);
        button2=findViewById(R.id.btnVidVw);
        surfaceView=findViewById(R.id.surfaceView);
        seekBar=findViewById(R.id.seekBar);

        mediaPlayer=MediaPlayer.create(this,R.raw.nature);
        surfaceView.setKeepScreenOn(true);
        SurfaceHolder surfaceHolder=surfaceView.getHolder();
        surfaceHolder.addCallback(new SurfaceHolder.Callback() {
```

```
      @Override
      public void surfaceCreated(@NonNull SurfaceHolder holder) {
        mediaPlayer.setDisplay(surfaceHolder);
      }

      @Override
      public void surfaceChanged(@NonNull SurfaceHolder holder, int format, int width,
int height) {}

      @Override
      public void surfaceDestroyed(@NonNull SurfaceHolder holder) {}

    });
    seekBar.setMax(mediaPlayer.getDuration());
    new Timer().schedule(new TimerTask() {
      @Override
      public void run() {
        try{
          seekBar.setProgress(mediaPlayer.getCurrentPosition());

        }catch (Exception e){
          e.printStackTrace();
          }
        }
      },0,800);
    seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
      @Override
      public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
        if(fromUser){
          mediaPlayer.seekTo(progress);
        }
      }
      @Override
      public void onStartTrackingTouch(SeekBar seekBar) {}
```

```
            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {}
        });
        button.setOnClickListener(v->{
            if(mediaPlayer.isPlaying()){
                mediaPlayer.pause();
                button.setText(R.string.play);
            }
            else {
                mediaPlayer.start();
                button.setText(R.string.pause);
            }
        });

        button2.setOnClickListener(v->{
            Intent i=new Intent(this,MainVideoViewActivity.class);
            startActivity(i);
        });
    }
}
```

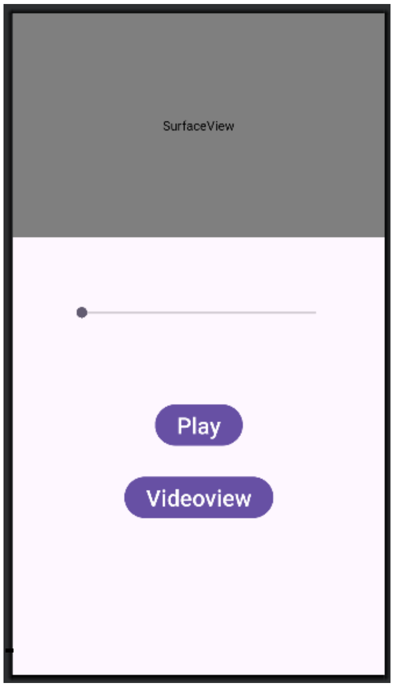
4) MainVideoViewActivity.java

```
package com.example.demoimagevideo;

import android.net.Uri;
import android.os.Bundle;
import android.widget.MediaController;
import android.widget.VideoView;
import androidx.appcompat.app.AppCompatActivity;


public class MainVideoViewActivity extends AppCompatActivity {
    VideoView videoView;
    String videoURL="https://paglasongs.com/files/download/id/588";
```

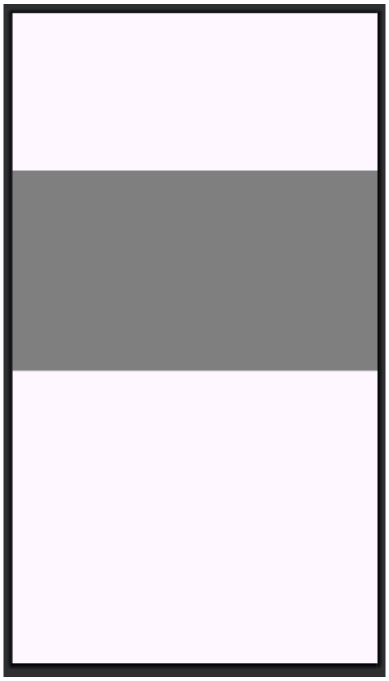
```
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_video_view);

    //video view
    videoView=findViewById(R.id.videoView);
    Uri uri=Uri.parse(videoURL);
    videoView.setVideoURI(uri);
    MediaController mediaController=new MediaController(this);
    mediaController.setAnchorView(videoView);
    mediaController.setMediaPlayer(videoView);
    videoView.setMediaController(mediaController);
    videoView.start();
  }
}
```

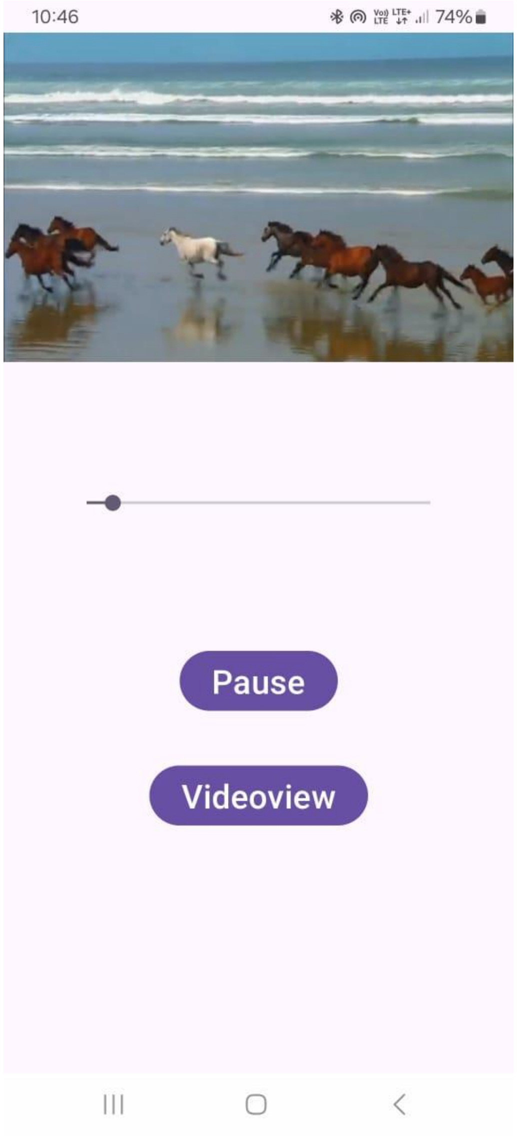
Output:

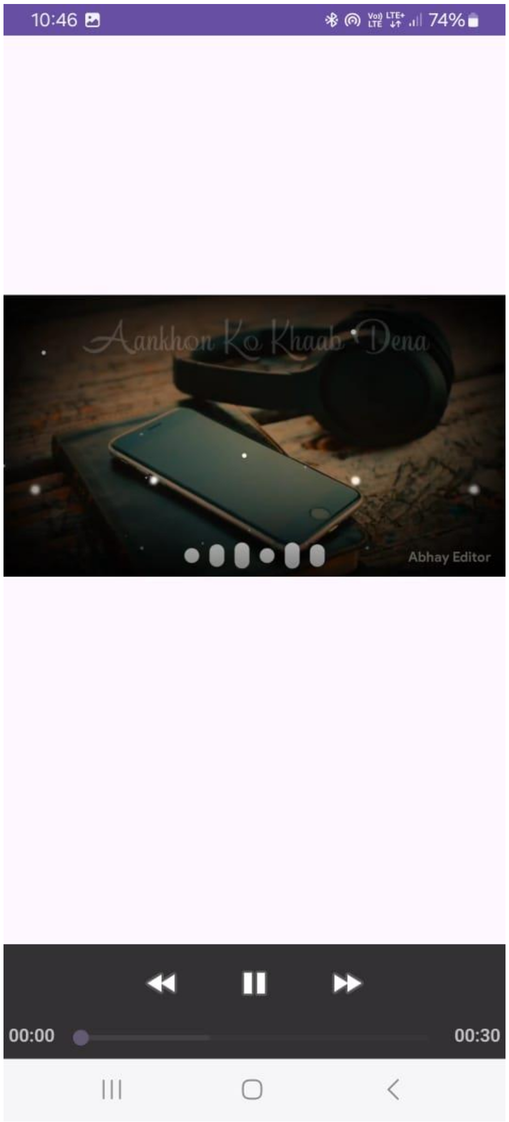
1) activity_main.xml :-



2) activity_video_view.xml :-







Conclusion:-This application successfully demonstrates the integration and manipulation of image and video components within an Android environment. By implementing this project, users gain practical experience in handling multimedia content, enhancing their skills in Android development.