

write an android program to play audio and video:

Code:

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- Button to play/pause audio -->
    <Button
        android:id="@+id/playPauseAudioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Play Audio"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"/>

    <!-- Button to play/pause video -->
    <Button
        android:id="@+id/playPauseVideoButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Play Video"
        android:layout_below="@id/playPauseAudioButton"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"/>

    <!-- SeekBar to control audio progress -->
    <SeekBar
        android:id="@+id/audioSeekBar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/playPauseVideoButton"
        android:layout_marginTop="30dp"/>

    <!-- VideoView to display the video -->
    <VideoView
        android:id="@+id/videoView"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:layout_below="@id/audioSeekBar"
        android:layout_marginTop="20dp"/>

</RelativeLayout>
```

MainActivity.java:

```
package com.example.audiovideoplayer;

import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.SeekBar;
import android.widget.VideoView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private Button playPauseAudioButton, playPauseVideoButton;
    private MediaPlayer mediaPlayer;
    private VideoView videoView;
    private SeekBar audioSeekBar;
    private boolean isAudioPlaying = false, isVideoPlaying = false;
    private Handler handler = new Handler();

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

        // Initialize UI components
        playPauseAudioButton = findViewById(R.id.playPauseAudioButton);
        playPauseVideoButton = findViewById(R.id.playPauseVideoButton);
        audioSeekBar = findViewById(R.id.audioSeekBar);
        videoView = findViewById(R.id.videoView);

        // Play/Pause Audio Button
        playPauseAudioButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (isAudioPlaying) {
                    pauseAudio();
                } else {
                    playAudio();
                }
            }
        });

        // Play/Pause Video Button
        playPauseVideoButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (isVideoPlaying) {
                    pauseVideo();
                } else {
                    playVideo();
                }
            }
        });
    }
}
```

```

    }
    });

    // SeekBar for Audio
    audioSeekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
            if (fromUser && mediaPlayer != null) {
                mediaPlayer.seekTo(progress);
            }
        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {}

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {}
    });
}

private void playAudio() {
    if (mediaPlayer == null) {
        mediaPlayer = MediaPlayer.create(this, R.raw.sample_audio); // Use your audio file here
        mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
            @Override
            public void onCompletion(MediaPlayer mp) {
                resetAudioControls();
            }
        });
        audioSeekBar.setMax(mediaPlayer.getDuration());
    }
    mediaPlayer.start();
    isAudioPlaying = true;
    playPauseAudioButton.setText("Pause Audio");
    updateAudioSeekBar();
}

private void pauseAudio() {
    if (mediaPlayer != null && mediaPlayer.isPlaying()) {
        mediaPlayer.pause();
        isAudioPlaying = false;
        playPauseAudioButton.setText("Play Audio");
    }
}

private void playVideo() {
    Uri videoUri = Uri.parse("android.resource://" + getPackageName() + "/" +
R.raw.sample_video); // Use your video file here
    videoView.setVideoURI(videoUri);
    videoView.start();
    isVideoPlaying = true;
    playPauseVideoButton.setText("Pause Video");

    // Update video SeekBar while the video is playing
    audioSeekBar.setMax(videoView.getDuration());
}

```

```

        videoView.setOnPreparedListener(mp -> updateVideoSeekBar());
    }

    private void pauseVideo() {
        if (videoView.isPlaying()) {
            videoView.pause();
            isVideoPlaying = false;
            playPauseVideoButton.setText("Play Video");
        }
    }

    // Method to update the audio seek bar while the audio is playing
    private void updateAudioSeekBar() {
        if (mediaPlayer != null && mediaPlayer.isPlaying()) {
            audioSeekBar.setProgress(mediaPlayer.getCurrentPosition());
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    updateAudioSeekBar();
                }
            }, 1000);
        }
    }

    // Method to update the video seek bar while the video is playing
    private void updateVideoSeekBar() {
        if (videoView.isPlaying()) {
            audioSeekBar.setProgress(videoView.getCurrentPosition());
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    updateVideoSeekBar();
                }
            }, 1000);
        }
    }

    @Override
    protected void onPause() {
        super.onPause();
        if (mediaPlayer != null) {
            mediaPlayer.release();
            mediaPlayer = null;
        }
        if (videoView.isPlaying()) {
            videoView.stopPlayback();
        }
    }

    // Reset audio controls when audio finishes
    private void resetAudioControls() {
        isAudioPlaying = false;
        playPauseAudioButton.setText("Play Audio");
        audioSeekBar.setProgress(0);
    }

```

```
}  
}
```

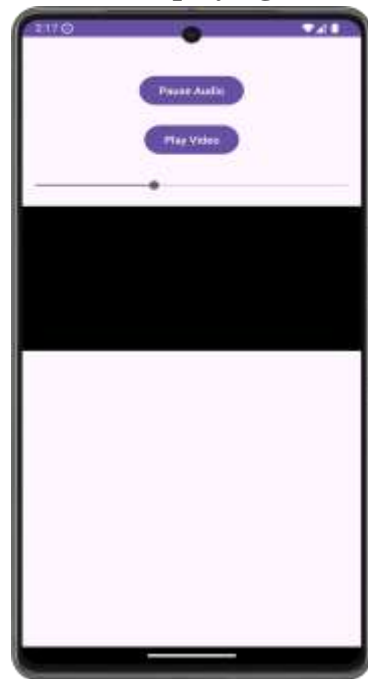
AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools">  
  
    <uses-permission android:name="android.permission.INTERNET"/>  
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>  
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>  
  
    <application  
        android:allowBackup="true"  
        android:dataExtractionRules="@xml/data_extraction_rules"  
        android:fullBackupContent="@xml/backup_rules"  
        android:icon="@mipmap/ic_launcher"  
        android:label="@string/app_name"  
        android:roundIcon="@mipmap/ic_launcher_round"  
        android:supportRtl="true"  
        android:theme="@style/Theme.Audiovideoplayer"  
        tools:targetApi="31">  
        <activity  
            android:name=".MainActivity"  
            android:exported="true">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
    </application>  
  
</manifest>
```

OUTPUT:



Audio playing



Video playing:

