Reproductir de musica

como usar la librería media player

Configuración del XML

comenzaremos configurando el xml con las siguientes paletas:

-un linearLayout

-4 botones para grabar el audio, detener la grabación de audio, reproducir el audio y detener el audio

-un surface que se utilizará para visualizar la grabación de video

- y otros 4 botones para grabar video, detener la grabación de video, reproducir el video y detener el video

configuración del proyecto

luego necesitaremos configurando los permisos del archivo *AndroidManifest.xml*

que servirán para escribir y leer en el almacenamiento externo y para grabar audio y usar la cámara

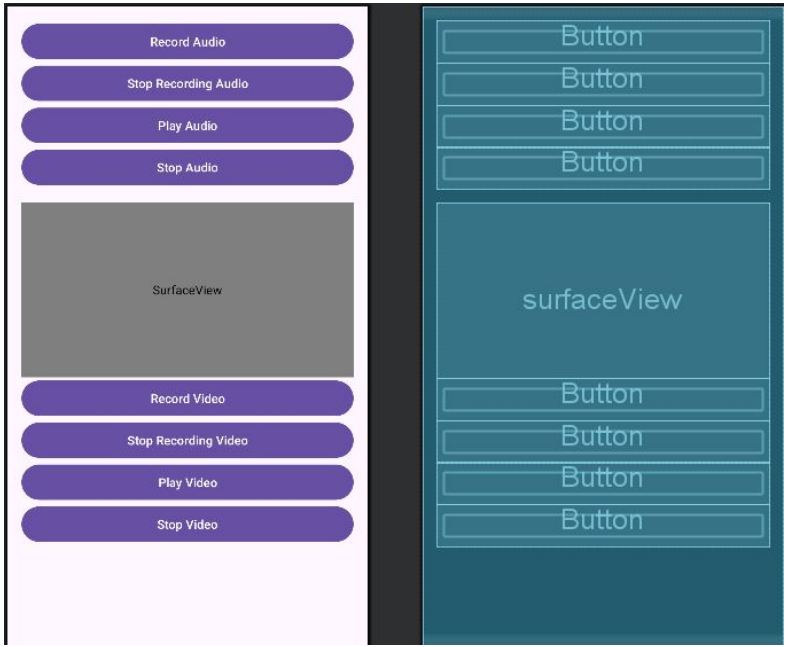
```
<uses-permission android:name="android.permission.RECORD_AUDIO" />

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

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

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

quedaria asi:



grabar audio

creamos una clase llamada MediaRecorder que lo utilizaremos para enlazarlo con el MainActivity, con configuramos para que haga las siguiente funciones:

-StartRecording()

-StopRecording()

StartRecording()

```
fun startRecording() {
    fileName = "${externalCacheDir?.absolutePath}/audiorecordtest.3gp"

    mediaRecorder = MediaRecorder().apply {
        setAudioSource(MediaRecorder.AudioSource.MIC)
        setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP)
        setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB)
        setOutputFile(fileName)

        try {
            prepare()
        } catch (e: IOException) {
            Log.e("AudioRecorder", "prepare() failed")
        }

        start()
    }
}
```

StopRecording()

```
fun stopRecording() {
    mediaRecorder?.apply {
        stop()
        release()
    }
    mediaRecorder = null
}
```

Reproducir Vídeo

Ahora creamos una clase llamada **mediaPlayer** y hacemos lo mismo que el script anterior con estos métodos:

-StartPlaying()

-StopPlaying()

StartPlaying()

```
fun startPlaying(fileName: String) {
    mediaPlayer = MediaPlayer().apply {
        setDataSource(fileName)
        prepare()
        start()
    }
}
```

StopPlaying()

```
fun stopPlaying() {
    mediaPlayer?.release()
    mediaPlayer = null
}
```

uso de los scripts anteriores en el mainActivity.kt

```kotlin
// Declaración de las variables para grabación y reproducción de media y el SurfaceHolder
private lateinit var mediaRecorder: MediaRecorder
private lateinit var mediaPlayer: MediaPlayer
private lateinit var surfaceHolder: SurfaceHolder
private var audioFileName: String = ""
private var videoFileName: String = ""

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    // Solicitar los permisos necesarios
    requestPermissions()

    // Definir los nombres de archivo para las grabaciones de audio y video
    audioFileName = "${externalCacheDir?.absolutePath}/audiorecordtest.3gp"
    videoFileName = "${externalCacheDir?.absolutePath}/videorecordtest.mp4"

    // Obtener SurfaceHolder para la vista previa del video
    surfaceHolder = findViewById<SurfaceView>(R.id.surfaceView).holder

    // Configurar listeners para los botones de la interfaz de usuario
    setupButtonListeners()
}
```

```kotlin
// Función para solicitar permisos en tiempo de ejecución
private fun requestPermissions() {
    val permissions = arrayOf(
        Manifest.permission.RECORD_AUDIO,
        Manifest.permission.WRITE_EXTERNAL_STORAGE,
        Manifest.permission.READ_EXTERNAL_STORAGE,
        Manifest.permission.CAMERA
    )
    // Comprobar si todos los permisos han sido concedidos, si no, solicitarlos
    if (!permissions.all { ContextCompat.checkSelfPermission(this, it) == PackageManager.PERMISSION_GRANTED }) {
        ActivityCompat.requestPermissions(this, permissions, 200)
    }
}

// Manejar el resultado de la solicitud de permisos
override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    if (requestCode == 200 && grantResults.isNotEmpty() && grantResults.all { it == PackageManager.PERMISSION_GRANTED }) {
        // Permisos concedidos
        Log.d("MainActivity", "Todos los permisos han sido concedidos.")
    } else {
        // Permisos denegados
        Log.d("MainActivity", "No se concedieron todos los permisos.")
    }
}

// Configurar listeners para los botones
private fun setupButtonListeners() {
    findViewById<Button>(R.id.recordAudioButton).setOnClickListener { startRecording(audioFileName, true) }
    findViewById<Button>(R.id.stopRecordAudioButton).setOnClickListener { stopRecording() }
    findViewById<Button>(R.id.playAudioButton).setOnClickListener { startPlaying(audioFileName, true) }
    findViewById<Button>(R.id.stopAudioButton).setOnClickListener { stopPlaying() }
    findViewById<Button>(R.id.recordVideoButton).setOnClickListener { startRecording(videoFileName, false) }
    findViewById<Button>(R.id.stopRecordVideoButton).setOnClickListener { stopRecording() }
    findViewById<Button>(R.id.playVideoButton).setOnClickListener { startPlaying(videoFileName, false) }
    findViewById<Button>(R.id.stopVideoButton).setOnClickListener { stopPlaying() }
}
```

```kotlin
// Función para iniciar la grabación de audio o video
private fun startRecording(fileName: String, isAudio: Boolean) {
    mediaRecorder = MediaRecorder().apply {
        if (isAudio) {
            // Configuración para grabar audio
            setAudioSource(MediaRecorder.AudioSource.MIC)
            setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP)
            setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB)
        } else {
            // Configuración para grabar video
            setAudioSource(MediaRecorder.AudioSource.CAMCORDER)
            setVideoSource(MediaRecorder.VideoSource.CAMERA)
            setOutputFormat(MediaRecorder.OutputFormat.MPEG_4)
            setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB)
            setVideoEncoder(MediaRecorder.VideoEncoder.MPEG_4_SP)
            setPreviewDisplay(surfaceHolder.surface)
        }
        setOutputFile(fileName)
        try {
            prepare()
            start()
        } catch (e: IOException) {
            Log.e("MediaRecorder", "prepare() failed")
        }
    }
}
```

```kotlin
// Función para detener la grabación de audio o video
private fun stopRecording() {
    mediaRecorder.apply {
        stop()
        release()
    }
}
```

```kotlin
// Función para iniciar la reproducción de audio o video
private fun startPlaying(fileName: String, isAudio: Boolean) {
    mediaPlayer = MediaPlayer().apply {
        setDataSource(fileName)
        if (!isAudio) {
            setDisplay(surfaceHolder)
        }
        try {
            prepare()
            start()
        } catch (e: IOException) {
            Log.e("MediaPlayer", "prepare() failed")
        }
    }
}
```

```kotlin
// Función para detener la reproducción de audio o video
private fun stopPlaying() {
    mediaPlayer.release()
}
```

```kotlin
// Liberar los recursos cuando la actividad se destruye
override fun onDestroy() {
    super.onDestroy()
    if (::mediaRecorder.isInitialized) mediaRecorder.release()
    if (::mediaPlayer.isInitialized) mediaPlayer.release()
}
```