



Centro Universitário Estácio
Polo São Lourenço da Mata – PE

Curso: Desenvolvimento Full Stack

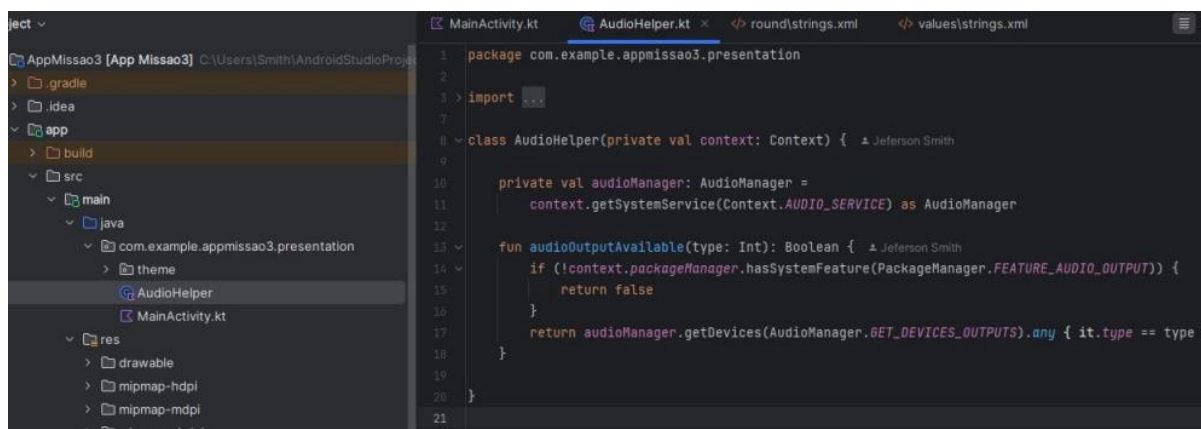
Disciplina: Lidando com sensores em dispositivos móveis - Nível 3

Semestre: 4º

Aluno: Jonnatha Walben Saldanha da Silva

Audio Helper

Essa classe AudioHelper verifica se há uma saída de áudio disponível no dispositivo. Ela utiliza o AudioManager para obter informações sobre os dispositivos de saída de áudio conectados e verifica se o tipo de saída de áudio especificado está presente. Além disso, faz uma checagem inicial se o dispositivo suporta a funcionalidade de saída de áudio.



MainActivity

Essa classe MainActivity de um aplicativo Wear OS gerencia conexões de áudio, detectando dispositivos Bluetooth A2DP. Ao conectar ou desconectar um fone de ouvido Bluetooth, ela exibe um Toast para o usuário e, ao desconectar, também abre as configurações de Bluetooth. A interface do aplicativo exibe um texto centralizado e a

hora usando o WearApp com o tema do aplicativo. Este aplicativo Wear OS monitora dispositivos de áudio Bluetooth, especificamente fones de ouvido Bluetooth A2DP. Ele exibe uma notificação (Toast) para o usuário quando um fone de ouvido Bluetooth é conectado ou desconectado. No caso de desconexão, o aplicativo também abre automaticamente as configurações de Bluetooth do dispositivo para facilitar a reconexão. O monitoramento dos dispositivos de áudio é feito por meio de um `AudioDeviceCallback`, que detecta eventos de adição e remoção de dispositivos. A interface do usuário, construída com Jetpack Compose, apresenta um relógio na parte superior da tela e um texto de saudação centralizado, exibindo "Hello, World", estilizado com um tema personalizado. O aplicativo também implementa uma splash screen ao iniciar, e define o tema visual para o padrão do Wear OS.

```
1 package com.example.appmissao3.presentation
2
3 import androidx.activity.compose.setContent
4
5 class MainActivity : AppCompatActivity() {
6     private lateinit var audioManager: AudioManager
7     private lateinit var audioHelper: AudioHelper
8
9     override fun onCreate(savedInstanceState: Bundle?) {
10         installSplashScreen()
11         super.onCreate(savedInstanceState)
12
13         setTheme(android.R.style.Theme_DeviceDefault)
14
15         audioManager = getSystemService(Context.AUDIO_SERVICE) as AudioManager
16         audioHelper = AudioHelper(context = this)
17
18         val audioDeviceCallback = object : AudioDeviceCallback() {
19             override fun onAudioDevicesAdded(addedDevices: Array<out AudioDeviceInfo>?) {
20                 super.onAudioDevicesAdded(addedDevices)
21                 if (audioHelper.audioOutputAvailable(AudioDeviceInfo.TYPE_BLUETOOTH_A2DP)) {
22                     showToast("Fone de ouvido Bluetooth conectado")
23                 }
24             }
25
26             override fun onAudioDevicesRemoved(removedDevices: Array<out AudioDeviceInfo>?) {
27                 super.onAudioDevicesRemoved(removedDevices)
28                 if (removedDevices != null) {
29                     for (device in removedDevices) {
30                         if (device.type == AudioDeviceInfo.TYPE_BLUETOOTH_A2DP) {
31                             promptBluetoothConnection()
32                             showToast("Fone de ouvido Bluetooth desconectado")
33                         }
34                     }
35                 }
36             }
37         }
38
39         audioManager.registerAudioDeviceCallback(audioDeviceCallback, handler = null)
40     }
41 }
```

```

setContent {
    WearApp(greetingName: "Android")
}

private fun showToast(message: String) {
    Toast.makeText(context, message, Toast.LENGTH_SHORT).show()
}

private fun promptBluetoothConnection() {
    val intent = Intent(Settings.ACTION_BLUETOOTH_SETTINGS)
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TASK)
    startActivity(intent)
}

@Composable
fun WearApp(greetingName: String) {
    AppMissaoTheme {
        Box(
            modifier = Modifier
                .fillMaxSize()
                .background(MaterialTheme.colors.background),
            contentAlignment = Alignment.Center
        ) {
            TimeText()
            Greeting(greetingName = greetingName)
        }
    }
}

@Composable
fun Greeting(greetingName: String) {
    Text(
        modifier = Modifier.fillMaxSize(),
        textAlign = TextAlign.Center,
        color = MaterialTheme.colors.primary,
        text = stringResource(id = R.string.conecte_seu_audio_via_bluetooth)
    )
}

@Preview(device = Devices.WEAR_OS_SMALL_ROUND, showSystemUI = true)
@Composable
fun DefaultPreview() {
    WearApp(greetingName: "Preview Android")
}

```

Executando o aplicativo

