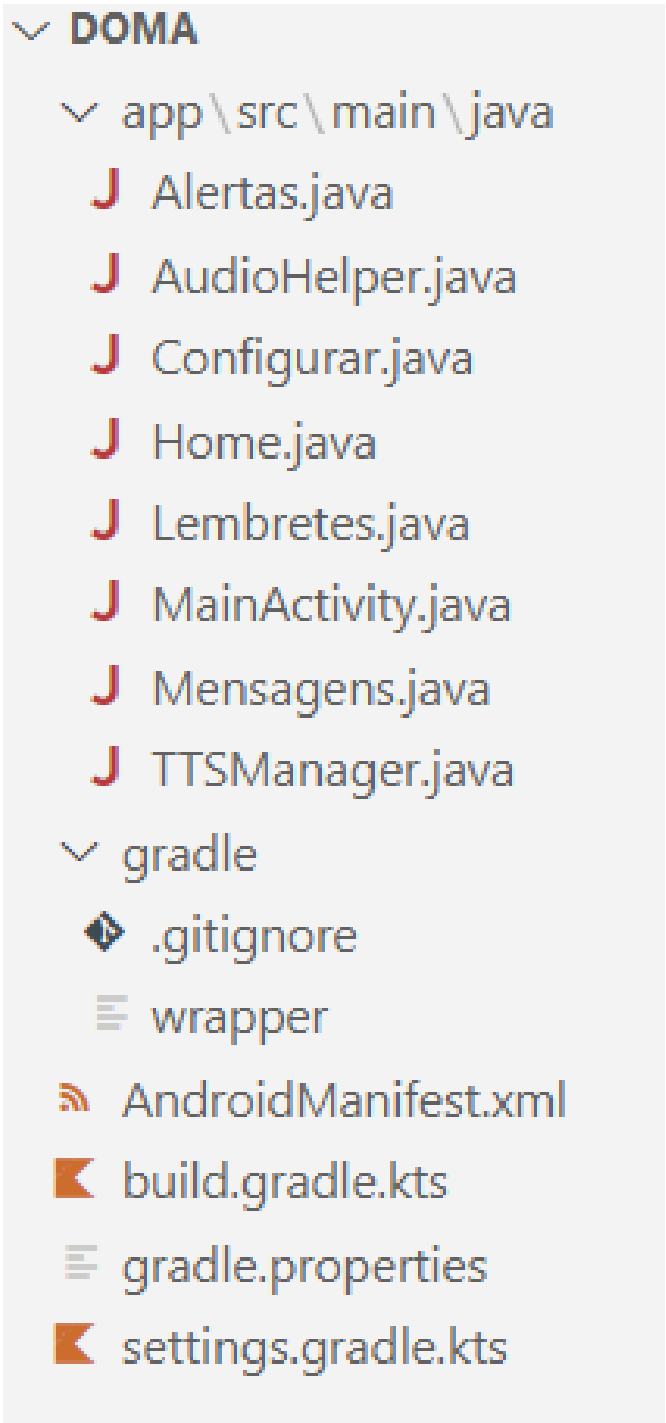
Disciplina: Missão Prática | Nível 3 | Mundo 4
Semestre: 2º - 2024
Aluno: DAVIDSON PEREIRA DE OLIVEIRA
Matrícula: 2022.11.54585-5
Repositório no GIT: dev-davidson/Missao-Pratica-Nivel-3-Mundo-4 (github.com)

Relatório discente de acompanhamento

RPG0025 - Lidando com sensores em dispositivos móveis

Estrutura do código:



Códigos:

Alertas.java

```java
package com.example.doma;

import android.content.Context;
import android.speech.tts.TextToSpeech;
import java.util.Locale;

public class Alertas {
    private TextToSpeech tts;

    public Alertas(Context context) {
        tts = new TextToSpeech(context, status -> {
            if (status != TextToSpeech.ERROR) {
                tts.setLanguage(Locale.US);
            }
        });
    }

    public void speak(String text) {
        tts.speak(text, TextToSpeech.QUEUE_FLUSH, null, null);
    }
}
```

AudioHelper.java

```
package com.example.doma;

import android.content.Context;
import android.media.AudioDeviceInfo;
import android.media.AudioManager;
import android.content.pm.PackageManager;

public class AudioHelper {
    private AudioManager audioManager;

    public AudioHelper(Context context) {
        audioManager = (AudioManager) context.getSystemService(Context.AUDIO_SERVICE);
    }

    public boolean audioOutputAvailable(int type) {
        if (!context.getPackageManager().hasSystemFeature(PackageManager.FEATURE_AUDIO_OUTPUT)) {
            return false;
        }
        for (AudioDeviceInfo device : audioManager.getDevices(AudioManager.GET_DEVICES_OUTPUTS)) {
            if (device.getType() == type) {
                return true;
            }
        }
        return false;
    }
}
```

Configurar.java

```java
package com.example.doma;

import android.content.Intent;
import android.provider.Settings;
import android.content.Context;

public class Configurar {

    public void openBluetoothSettings(Context context) {
        Intent intent = new Intent(Settings.ACTION_BLUETOOTH_SETTINGS);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
        intent.putExtra("EXTRA_CONNECTION_ONLY", true);
        intent.putExtra("EXTRA_CLOSE_ON_CONNECT", true);
        intent.putExtra("android.bluetooth.devicepicker.extra.FILTER_TYPE", 1);
        context.startActivity(intent);
    }
}
```

Home.java

```java
package com.example.doma;

import android.app.Activity;
import android.os.Bundle;

public class Home extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
    }
}
```

Lembretes.java

```java
package com.example.doma;

import android.content.Context;
import android.speech.tts.TextToSpeech;
import java.util.Locale;

public class Lembretes {
    private TextToSpeech tts;

    public Lembretes(Context context) {
        tts = new TextToSpeech(context, status -> {
            if (status != TextToSpeech.ERROR) {
                tts.setLanguage(Locale.US);
            }
        });
    }

    public void speak(String text) {
        tts.speak(text, TextToSpeech.QUEUE_FLUSH, null, null);
    }
}
```

MainActivity.java

```
public package com.example.doma;

import android.app.Activity;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.widget.TextView;
import java.util.Locale;

public class MainActivity extends Activity {
    private TextToSpeech tts;

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

        tts = new TextToSpeech(this, status -> {
            if (status != TextToSpeech.ERROR) {
                tts.setLanguage(Locale.US);
            }
        });

        TextView textView = findViewById(R.id.textView);
        textView.setOnClickListener(v -> speak("Ola, isto e um teste."));
    }
    private void speak(String text) {
        tts.speak(text, TextToSpeech.QUEUE_FLUSH, null, null);
    }
}
 MainActivity {
}
```

Mensagens.java

```java
package com.example.doma;

import android.content.Context;
import android.speech.tts.TextToSpeech;
import java.util.Locale;

public class Mensagens {
    private TextToSpeech tts;

    public Mensagens(Context context) {
        tts = new TextToSpeech(context, status -> {
            if (status != TextToSpeech.ERROR) {
                tts.setLanguage(Locale.US);
            }
        });
    }

    public void speak(String text) {
        tts.speak(text, TextToSpeech.QUEUE_FLUSH, null, null);
    }
}
```

TTSManager.java

```java
package com.example.doma;

import android.content.Context;
import android.speech.tts.TextToSpeech;
import java.util.Locale;

public class TTSManager {
    private TextToSpeech tts;

    public TTSManager(Context context) {
        tts = new TextToSpeech(context, status -> {
            if (status != TextToSpeech.ERROR) {
                tts.setLanguage(Locale.US);
            }
        });
    }

    public void speak(String text) {
        tts.speak(text, TextToSpeech.QUEUE_FLUSH, null, null);
    }
}
```

.gitignore

```
# .gitignore
/.gradle
/build
/.idea
/local.properties
.DS_Store
/*.iml
.gradle/
/captures/
/.externalNativeBuild/
```

gradle-wrapper.properties

```
# gradle-wrapper.properties
distributionBase=GRADLE_USER_HOME
distributionPath=wrapper/dists
zipStoreBase=GRADLE_USER_HOME
zipStorePath=wrapper/dists
distributionUrl=https\://services.gradle.org/distributions/gradle-7.0.2-all.zip
```

AndroidManifest.xml

```
plugins {
    id("com.android.application")
    id("kotlin-android")
}

android {
    compileSdkVersion 30
    defaultConfig {
        applicationId "com.example.doma"
        minSdkVersion 21
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles(getDefaultProguardFile("proguard-android-optimize.txt"), "proguard-rules.pro")
        }
    }
}

dependencies {
    implementation("org.jetbrains.kotlin:kotlin-stdlib:1.5.21")
    implementation("androidx.core:core-ktx:1.6.0")
    implementation("androidx.appcompat:appcompat:1.3.1")
    implementation("com.google.android.material:material:1.4.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.0")
}
```

build.gradle.kts

```
// build.gradle.kts
plugins {
    id("com.android.application") version "7.0.0" apply false
    id("org.jetbrains.kotlin.android") version "1.5.21" apply false
}

tasks.register("clean", Delete::class) {
    delete(rootProject.buildDir)
}
```

gradle.properties

```
org.gradle.jvmargs=-Xmx1536m
```

settings.gradle.kts

```
// settings.gradle.kts
rootProject.name = "Doma"
include(":app")
```

Resumo do Aplicativo Wear OS da Empresa "Doma"

Contexto e Objetivo:

A empresa "Doma" busca desenvolver um aplicativo Wear OS com o objetivo de melhorar a eficiência e a comunicação interna, oferecendo assistência específica para funcionários com necessidades especiais. Este aplicativo visa promover uma interação mais sólida e inclusiva, utilizando tecnologias de áudio para fornecer informações e suporte em tempo real.

Funcionalidades Principais:

1. **Leitura de Mensagens e Notificações em Voz Alta:**
   - ✓ O aplicativo pode ler mensagens de texto e notificações em voz alta, ajudando funcionários com deficiência visual a se manterem informados sem a necessidade de olhar para a tela.
2. **Respostas a Comandos de Voz:**
   - ✓ Permite que os usuários interajam com o aplicativo usando comandos de voz, facilitando a navegação e execução de tarefas sem precisar usar as mãos.
3. **Alertas de Segurança:**
   - ✓ O aplicativo pode emitir alertas de segurança, como notificações de emergência, alertas de tempestades, notícias importantes ou outras informações críticas que requerem atenção imediata.
4. **Instruções e Feedbacks de Áudio:**
   - ✓ Ideal para treinamentos e educação, o aplicativo fornece instruções, dicas e feedbacks em áudio durante o aprendizado ou a prática de novas habilidades, garantindo que os funcionários possam seguir orientações de forma eficaz.

Estrutura do Código:

O código é organizado em diversas classes Java, cada uma responsável por uma funcionalidade específica:

- ✓ **MainActivity.java:** Ponto de entrada do aplicativo, responsável por inicializar o Text-to-Speech (TTS) e configurar interações básicas de voz.

- ✓ **Home.java:** Tela inicial do aplicativo com uma mensagem de boas-vindas.

- ✓ **AudioHelper.java:** Classe utilitária para gerenciar dispositivos de áudio disponíveis (alto-falantes e fones de ouvido Bluetooth).

- ✓ **TTSManager.java:** Gerencia o serviço de Text-to-Speech para converter texto em fala.

- ✓ **Alertas.java, Mensagens.java, Lembretes.java:** Gerenciam alertas, mensagens e lembretes que o aplicativo fornecerá aos usuários.

- ✓ **Configurar.java:** Classe responsável por configurar o aplicativo, como direcionar usuários para configurações de Bluetooth.

Benefícios:

- **Acessibilidade:**
  - ✓ O aplicativo torna o ambiente de trabalho mais acessível para funcionários com deficiências visuais, promovendo inclusão e igualdade de oportunidades.
- **Eficiência:**
  - ✓ A comunicação interna é aprimorada através de notificações em tempo real e respostas rápidas a comandos de voz.
- **Segurança:**
  - ✓ Alertas de segurança e notificações críticas ajudam a garantir que todos os funcionários estejam cientes de situações de emergência.