ICP11

Keenan Flynn – kpfxn8@umsystem.edu – https://github.com/kfly2fly/Web-Mobile-Spring-2022
Jasmine Naraine- jnytc@umstyem.edu -https://github.com/JNaraine/490-Mobile

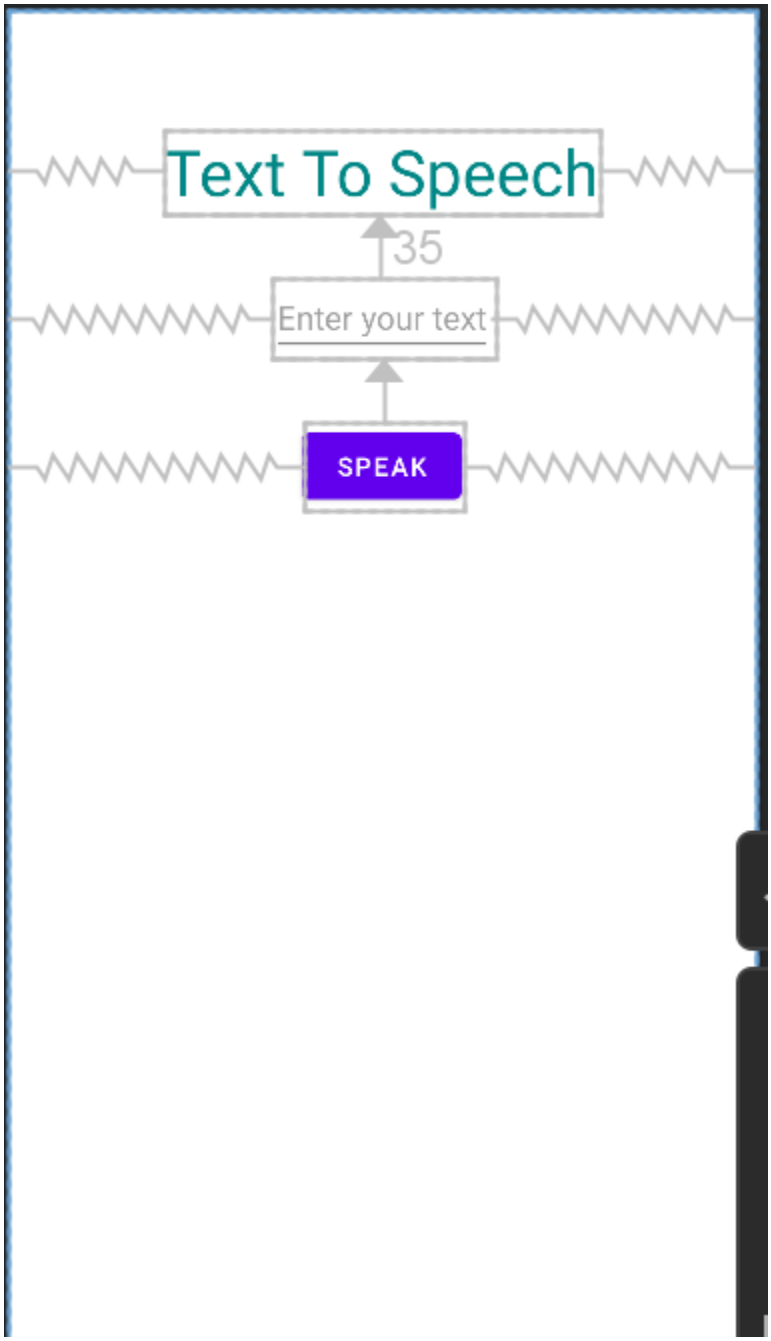
In this ICP, we followed along in class to create a simple app that allows the user to use the Text to Speech function of Android, which comes natively to the OS. The Text-To-Speech function sends text to the audio output. There is only one TTS engine per device so we need to make sure that permissions are being asked for and given up correctly.

The User Interface consists of 3 views. There is a TextView that is static and acts as the title of the app. There is an EditText view. This view allows the user to enter a string which will be then spoken through the speaker. Finally there is a Button View that when pressed, passes the text in the EditText view to the Java code which then speaks the entered text.

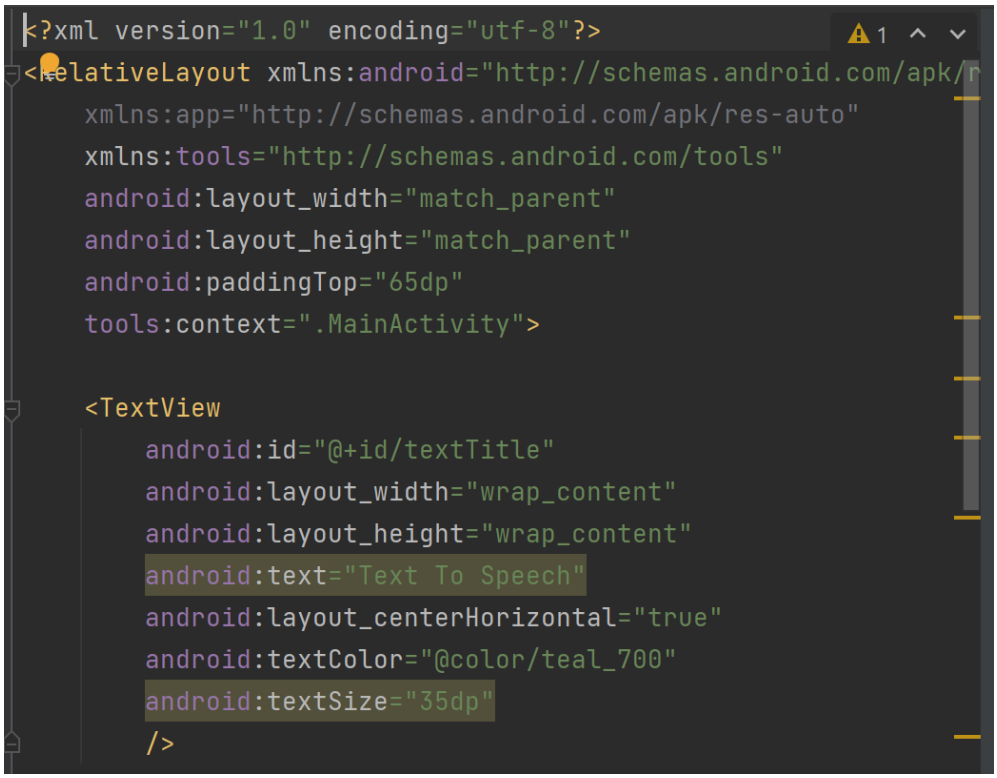
The Java backend is where the Text-to-Speech instance is created. When the button is pressed, the app creates a new TTS object using the MainActivity context. This new object has a method called onInit() which passes a success variable that we can use to troubleshoot the application. We saw this during class as the Initialization was failing until we properly connected the button. In the onInit() method, we set the Language to English and catch any errors with an if-else branch. If there are no errors, the application will connect to the speaker and output as audio the user's text input.

Finally we have to clean up our TTS object so that other applications can use this function. We override the onPause() and onDestroy() methods so that we can stop the TTS object.

User view



activity_main.xml


<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/r
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingTop="65dp"
tools:context=".MainActivity">
<TextView
android:id="@+id/textTitle"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Text To Speech"
android:layout_centerHorizontal="true"
android:textColor="@color/teal_700"
android:textSize="35dp"
/>

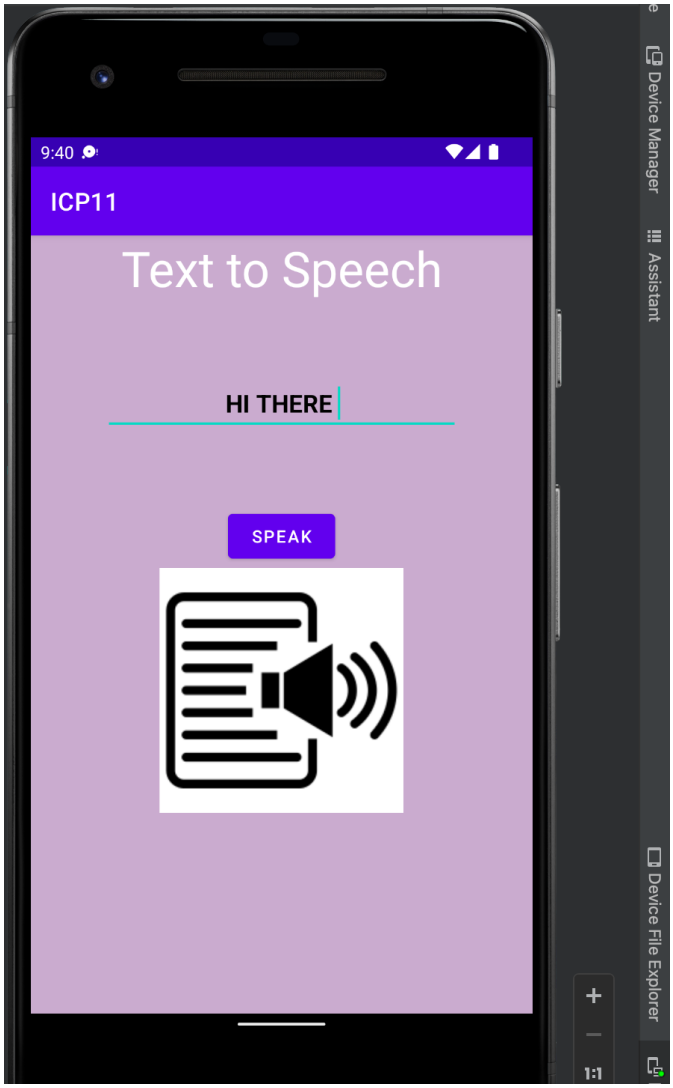
```
    <EditText
        android:id="@+id/editText"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:hint="Enter your text"
        android:layout_below="@id/textTitle"
        android:layout_margin="35dp"
        android:layout_centerHorizontal="true"
        android:inputType="text"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/editText"
        android:id="@+id/btn"
        android:text="Speak"
        android:layout_centerHorizontal="true"
        />
```

MainActivity.java

```java
package com.example.icp11_inclass;
import ...

public class MainActivity extends AppCompatActivity {

    EditText message;
    Button btn;
    TextToSpeech tts;

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

        message = findViewById(R.id.editText);
        btn = findViewById(R.id.btn);
```

```java
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                tts = new TextToSpeech( context: MainActivity.this, new TextToSpeech.OnInitListener() {
                    @Override
                    public void onInit(int i) {
                        if (i == TextToSpeech.SUCCESS) {
                            int result = tts.setLanguage(Locale.US);
                            if (result == TextToSpeech.LANG_NOT_SUPPORTED || result == TextToSpeech.LANG
                                Log.e( tag: "message",  msg: "Language is not supported");
                            } else {
                                tts.speak( text: "Test Text to Speech", TextToSpeech.QUEUE_ADD,  params: nul
                            }
                        } else {
                            Log.e( tag: "message",  msg: "TTS is not supported");
                        }
                    }});
            }
        });
    }
```

```
    void speak() {
        String text = message.getText().toString();
        tts.setSpeechRate(0.5f);
        tts.speak(text, TextToSpeech.QUEUE_ADD,  params: null);
    }

    @Override
    protected void onPause() {
        super.onPause();
        tts.stop();
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        tts.shutdown();
    }
}
```



Final output

Whatever the user puts in the bar and clicks speak, through the speaker the user will hear the same thing that was typed in.