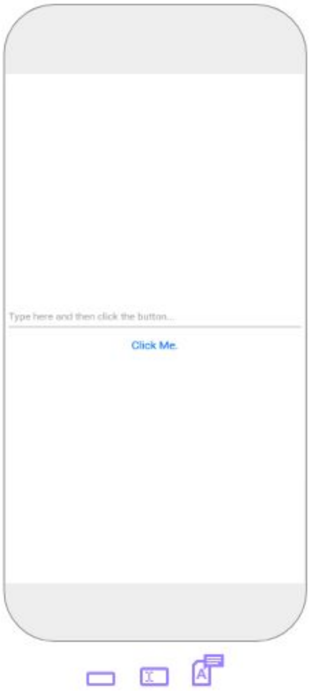
1. My Device Can Talk



Objective:
Students start off with an app that uses the `Text to Speech` component to make their phone or tablet talk to them.
Once this is completed the `TextInput` is added to make the phone say whatever we choose.

Set Up:
In order to get the very most out of your class time we recommended that students install the [live testing app] on their devices ahead of time. If you are using a shared set of iPads or tablets then this is something that you may have to organise too.

Design Concepts:
Visible & Invisible Components.
The notion of visible and invisible components are introduced. User interface, or UI, elements such as the TextInput and the Button are visible (or can be made visible) while something like the Text to Speech makes used of the device's speaker.

Button [docs]
The button is a visible UI component that has several events associated with it, the most popular of which is the "click" event. Buttons can be used in lots of scenarios like start/stop or navigating forwards/backwards.

TextInput [docs]
The textInput is a visible UI component that shows the user a keyboard and allows them to type anything the like. As the app developer you can choose what sort of information you would like them to enter, such as words, numbers, email addresses etc.

TextToSpeech [docs]
This is an invisible components that can use the speaker of your device to output a spoken message.

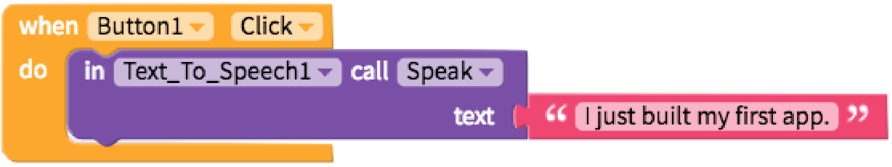
Blocks:
Events. [example]
The first programming concept we look at is event handling. Any time that something happens in your app this is known as an event. By adding an event handler to your app you can deal with, or handle, things happening in your app like buttons being pressed, something starting or stopping

Procedures:
The first procedure we encounter in this course is “in Text to Speech1 call Speak”. A procedure is a set of well-defined instructions that can be repeated, or called, as often as you like in your app.

Instructions:

1. From the User Interface section, drag and drop a `Button` component into the phone. Since there can be many buttons in a single app, by default new components are also given a number. Note that your first button is called **`Button1.`**
2. From the Voice section, drag and drop a `Text to Speech` component into the phone. This is an example of an invisible component and so it appears in the Invisible Components tree/
3. Note that the “Design” tab is already active. Click the “Blocks” tab to switch to a new view. This is where you will program your app.
4. Click to open the **`Button1`** drawer. Drag and drop the “when Button1 Click” block into the blocks editor. In your app, when the button is clicked, it will run the blocks next to the word 'do'.
5. Click to open Text_to_Speech1's drawer. Drag and drop the "in Text_to_Speech1 call Speak" block into the "when Button1 Click" block. When the blocks are connected, they will make a "thunk" noise!
6. Open the Text drawer and select an empty text block. Drag and drop the empty text block into the opening of the “in Text_to_Speech1 call Speak” block. Now write something! Anything you write in this block will be read by your phone when Button1 is clicked.
7. Congratulations! You just built your first app. Open the Thunkable Live App on your phone, and then click the Live Test button on your computer.

For the first app your blocks should look something like this:



```
when Button1 Click
do  in Text_To_Speech1 call Speak
                          text “ I just built my first app. ”
```

Sample App:

My Device Can Talk http://bit.ly/thunk101

Progression:

- Add in an additional Visible Component. Allow users to type in text that the phone reads aloud.
- Add in an additional Invisible Component. Have the phone say back what you say to it.
- Change accents
- Add more buttons for a phrase finder/catchphrase board
- Use the `Clock` and make the app tell jokes (set up + punchline)