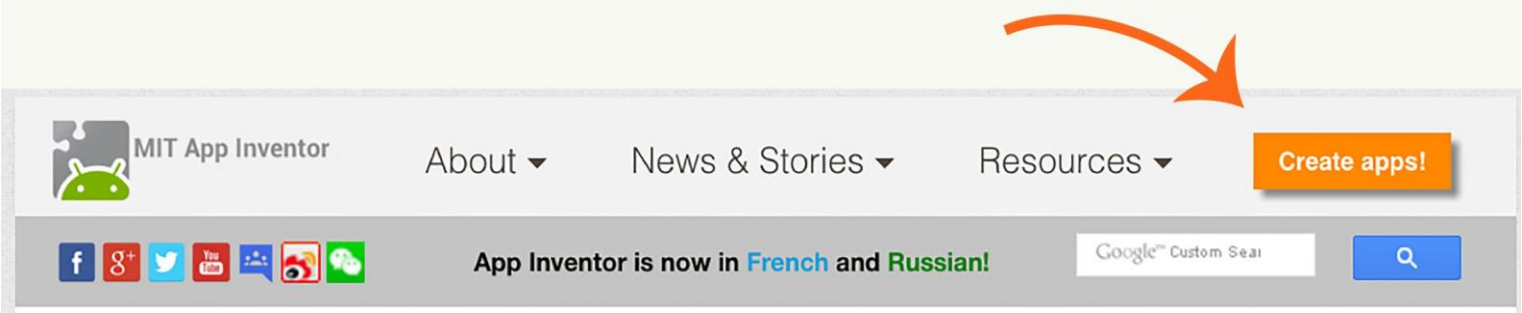
TalkToMe: A beginner App Inventor app

This step-by-step picture tutorial will guide you through making a talking app.

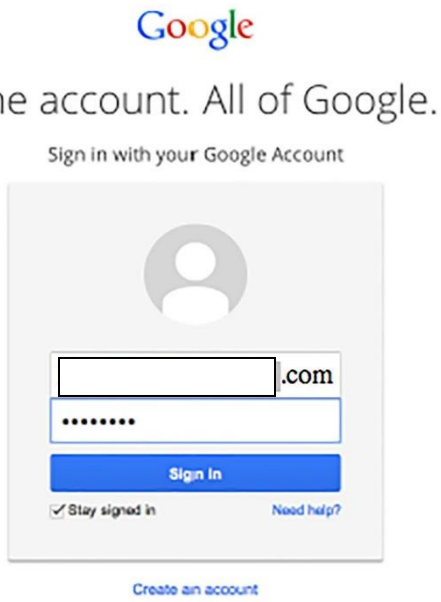
1. Go to the App Inventor home page: www.appinventor.mit.edu

click the orange “Create apps” Button in the menu bar.

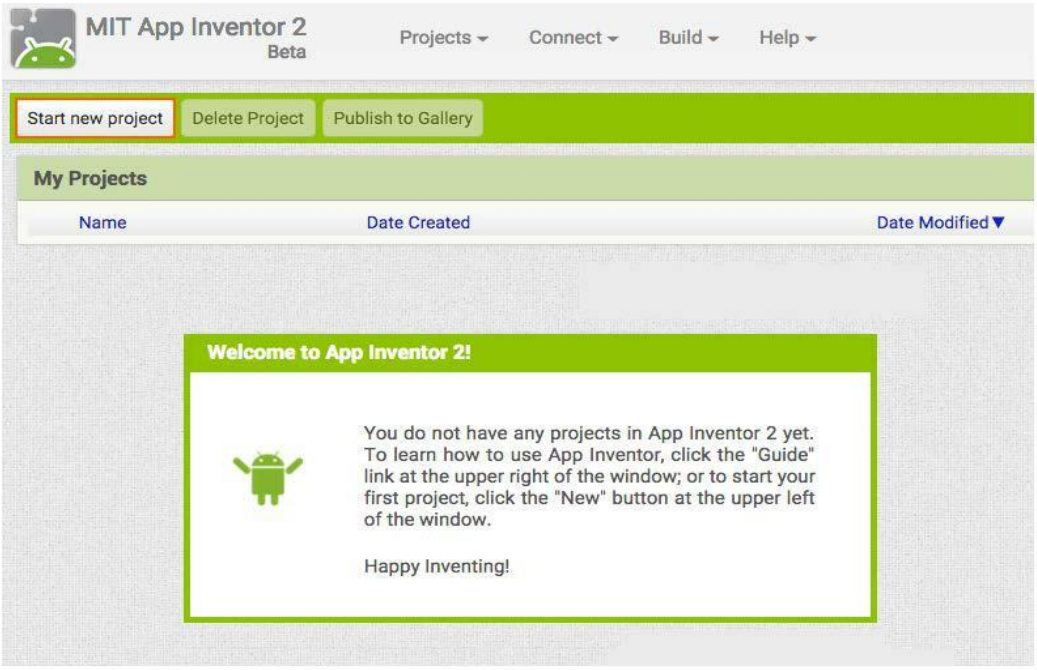


2. Log in to App Inventor with your Gmail (or Google) username and password.

Use an existing g mail account or school-based google account to log in to ai2.appinventor.mit.edu (Example Gmail shown below.)



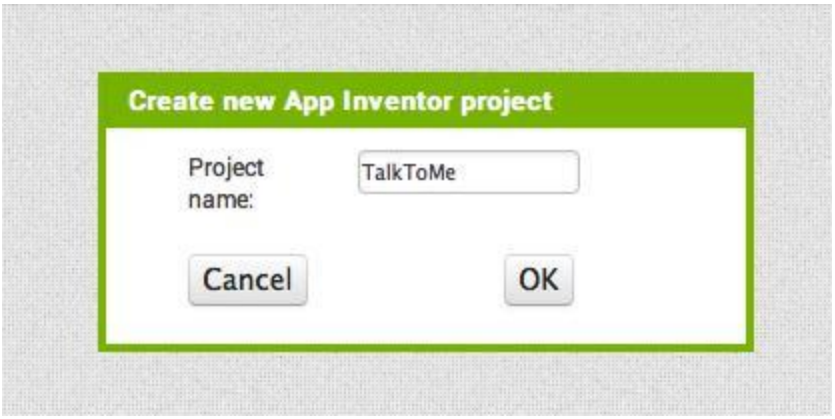
3.Start a new project by clicking the "Start new project" button.



---

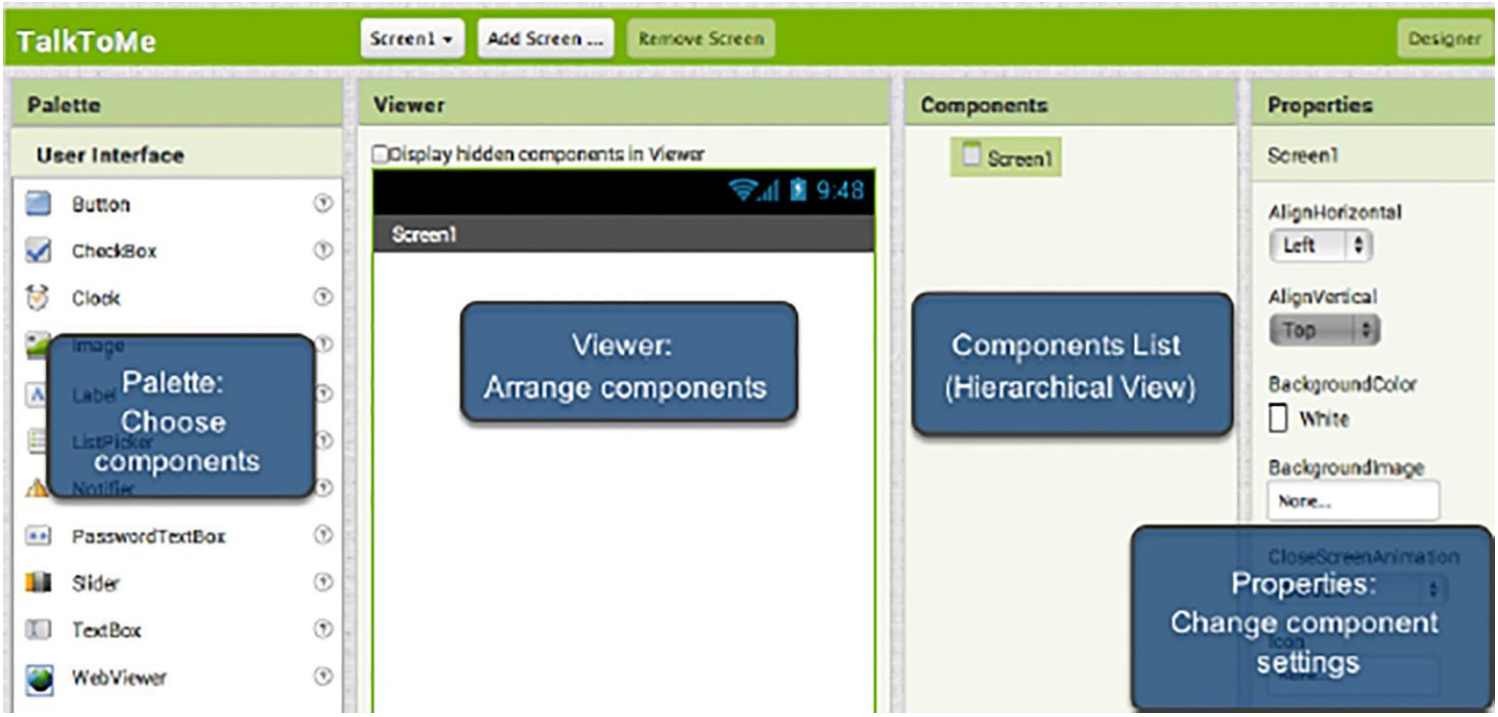
4. Name the project "TalkToMe" (no spaces)

Type in the project name (underscores are allowed, spaces are not) and click OK.



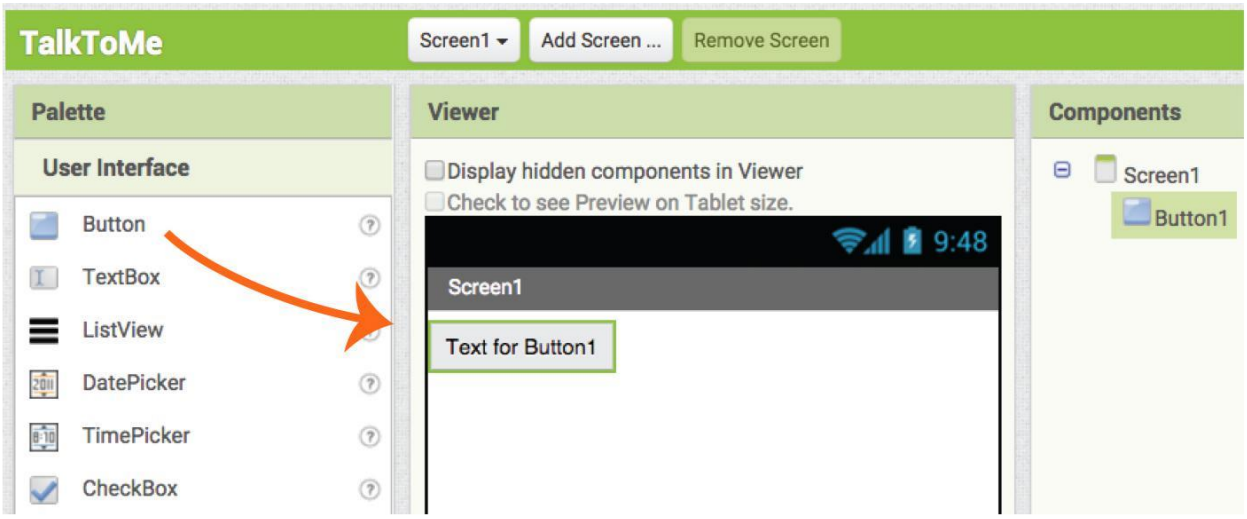
5. App Inventor opens the Designer window

The "Designer" is where you create the Graphical User Interface (GUI) or the look and feel of your app. You choose components like Buttons, Images, and Text boxes, and functionalities like Text-to-Speech, Sensors, and GPS.



6.Add a Button

Click and hold on the word "Button" in the Palette. Drag your mouse over to the Viewer. Release the mouse. A new button will appear on the Viewer.



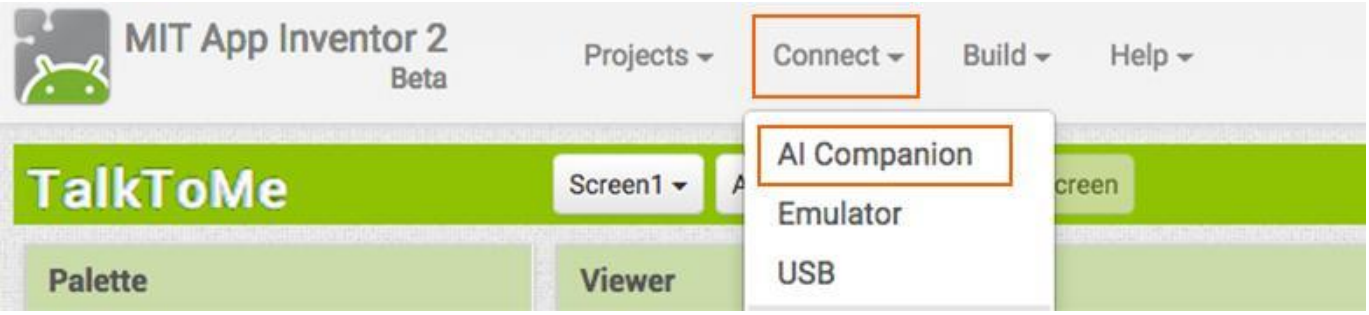
7.Download the MIT AI2 Companion App from the Google Play Store and install it on your phone or tablet.

8.Once you've installed the AI2 Companion app, you can connect your App Inventor project to your phone or tablet for live testing

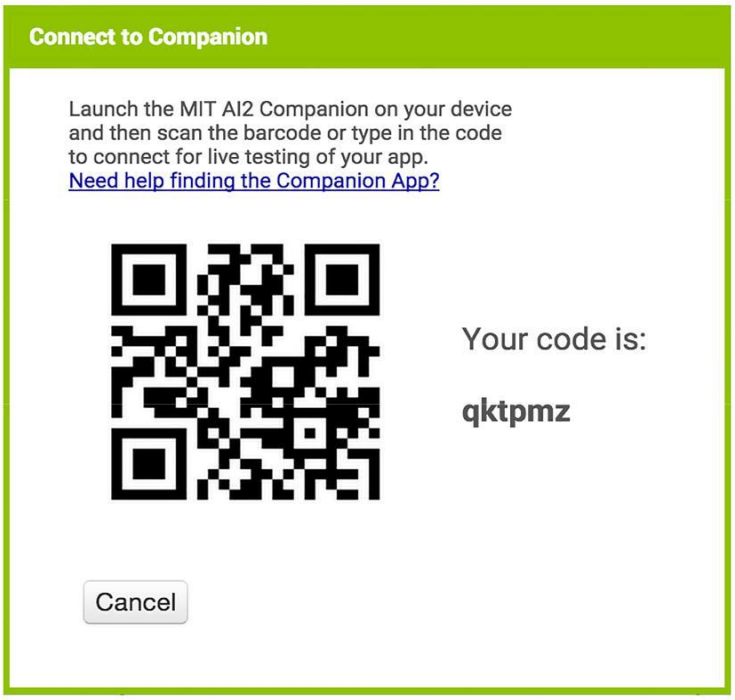
While you're building an app on your computer, you can test it on a connected Android phone or tablet**.**

Be sure your computer and mobile device are connected to the same WiFi network.

Return to the Designer Window on your computer.
Click Connect and choose AI Companion from the dropdown menu.



A QR code and a 6character code will appear on the screen of your computer screen.

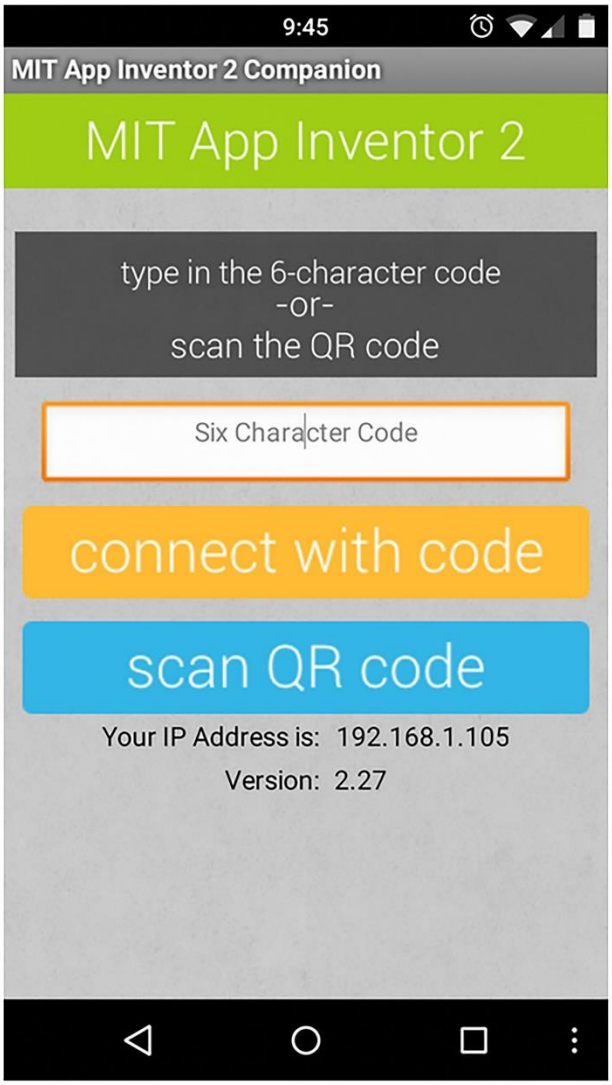


Open the AI2 Companion app on your device by clicking on the app icon.

A screen (like the one shown below) will appear with the option to scan the QR code or type in the six character code.

If you choose to scan the code, press the blue "scan QR code" button for the scanner to launch. Scan the QR code. Wait a few seconds for your app to open on your mobile device.

If you choose to use the code, type it into the white text box, click the orange button afterwards.

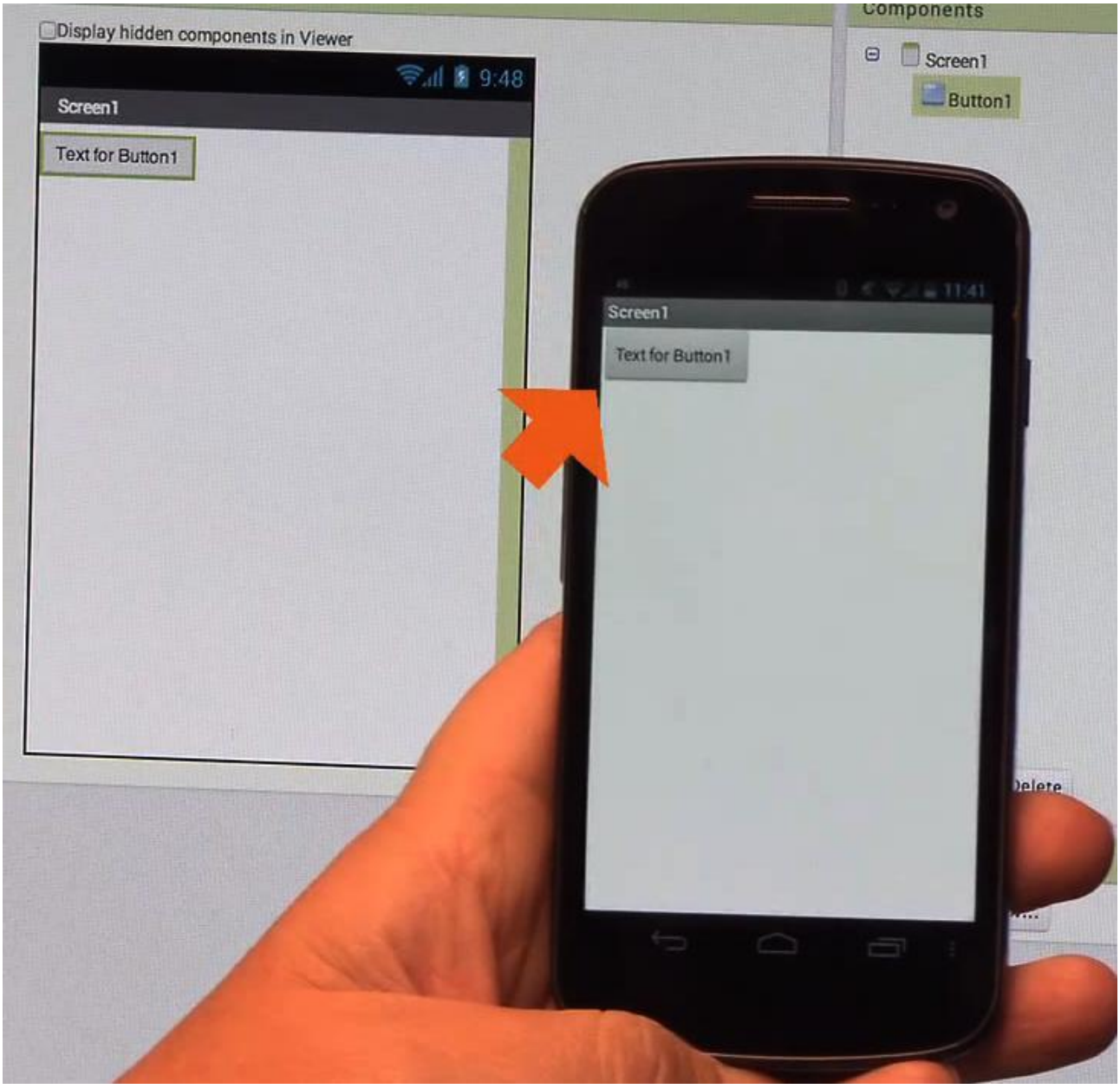


9.See your app on your connected device

You will know that your connection is successful when you see your app on the connected device.

Since our app only has a button, that is what you will see on your mobile device.

As you add more components to the project, your app will update on your computer and your phone.

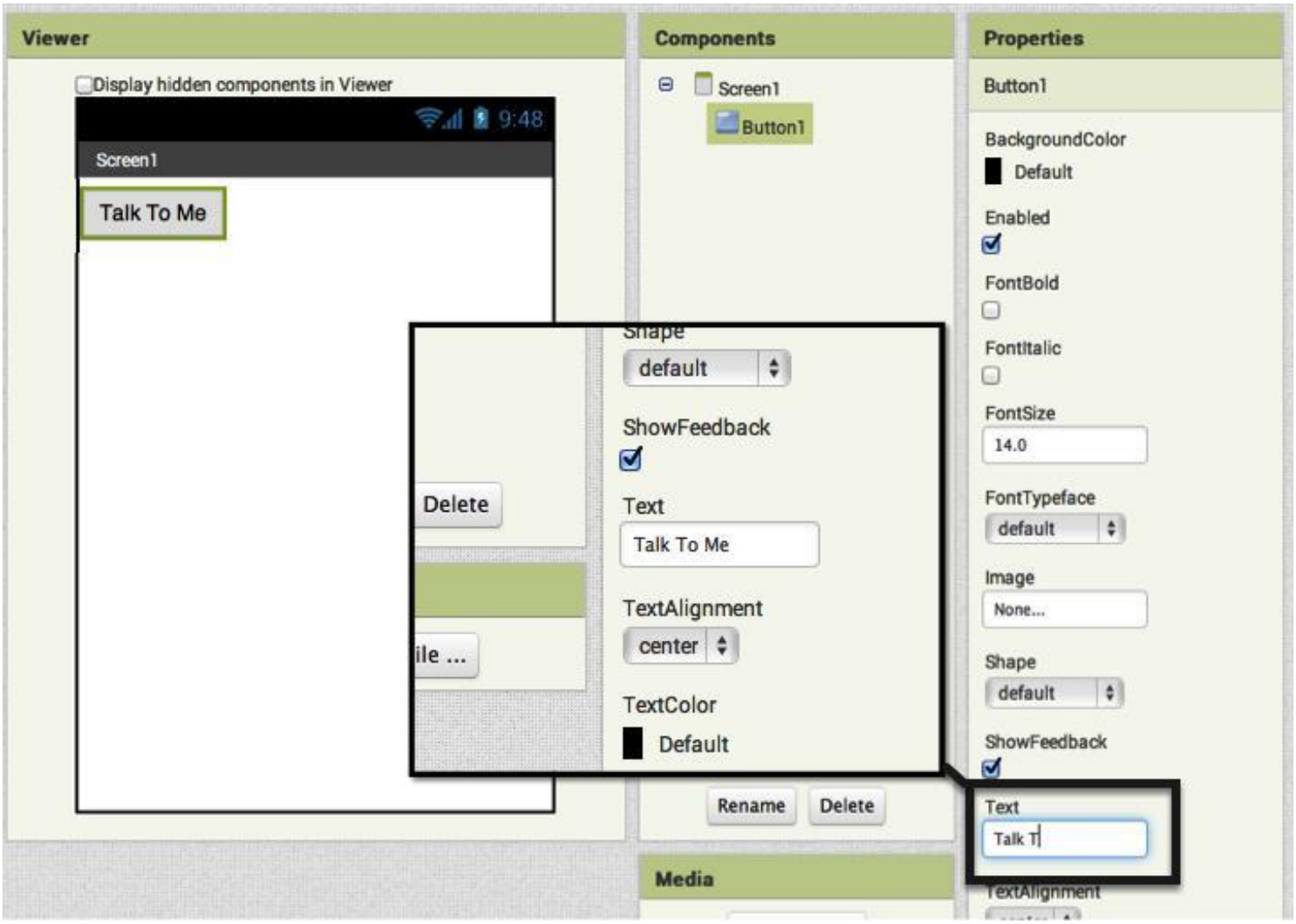


10.Change the Text on the Button

In the properties panel, change the text for the Button.

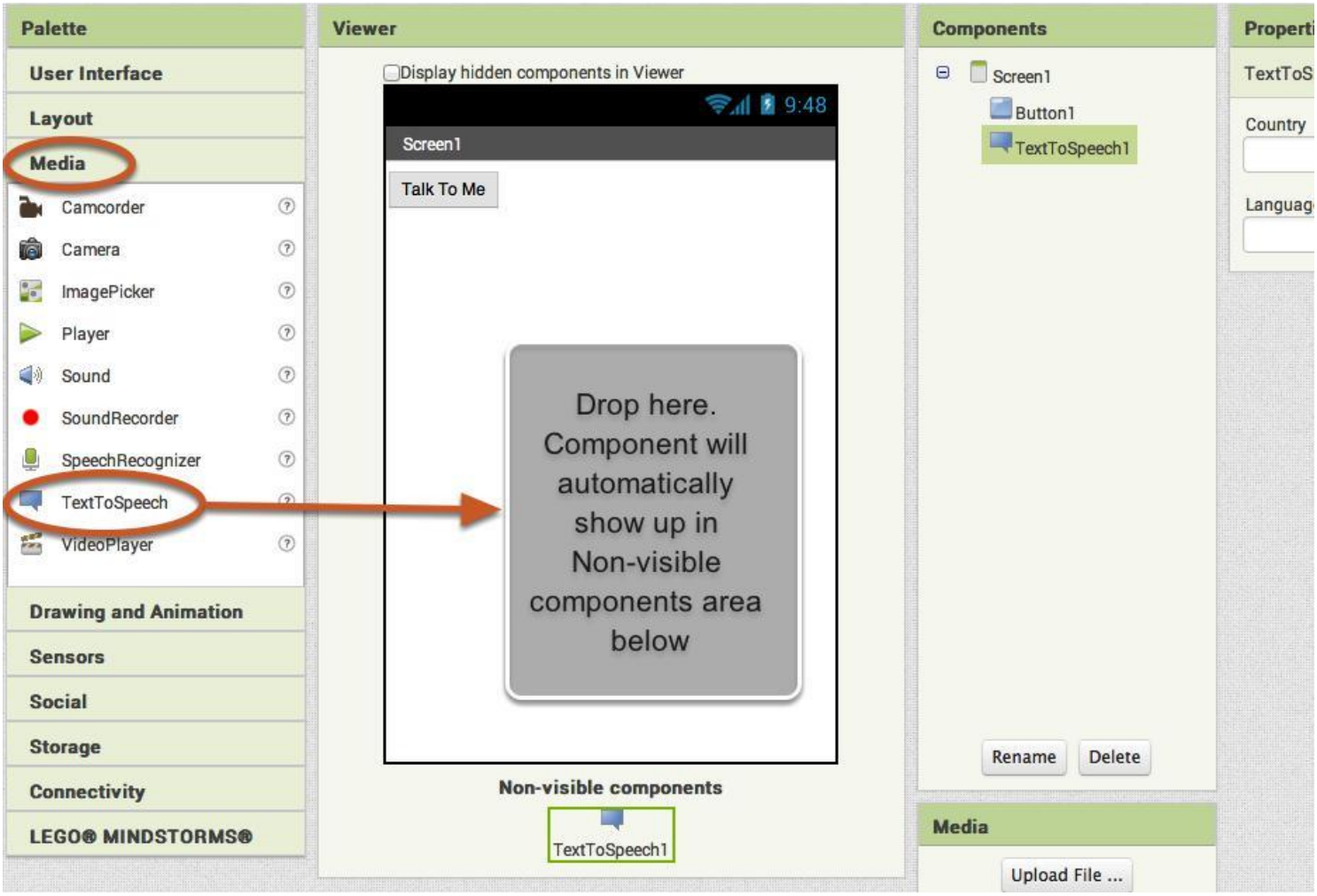
Under the Text property, select "Text for Button 1", delete it and type in "Talk To Me".

Notice that the text on your app's button changes right away too.



11.Add a Text-to-Speech component to your app

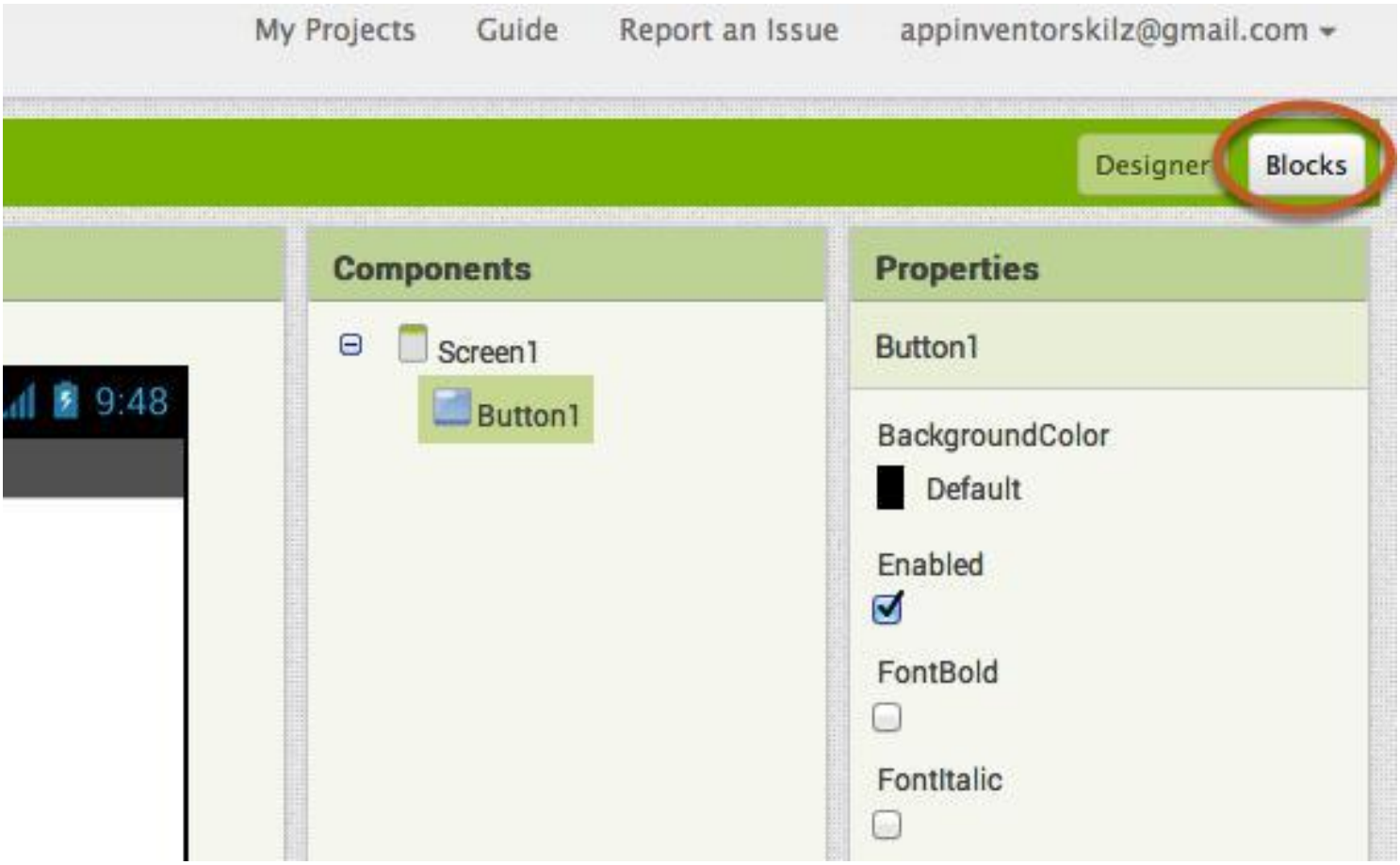
Go to the Media drawer in the Palette and drag out a TextToSpeech component.
Drag and drop it onto the Viewer.
Notice that it drops down under "Non-visible components" because it is not something that will show up on the app's user interface. It's more like a tool that is available to the app.



12.Switch over to the Blocks Editor

It's time to tell your app what to do. The Blocks Editor is where you program the behaviour of your app.

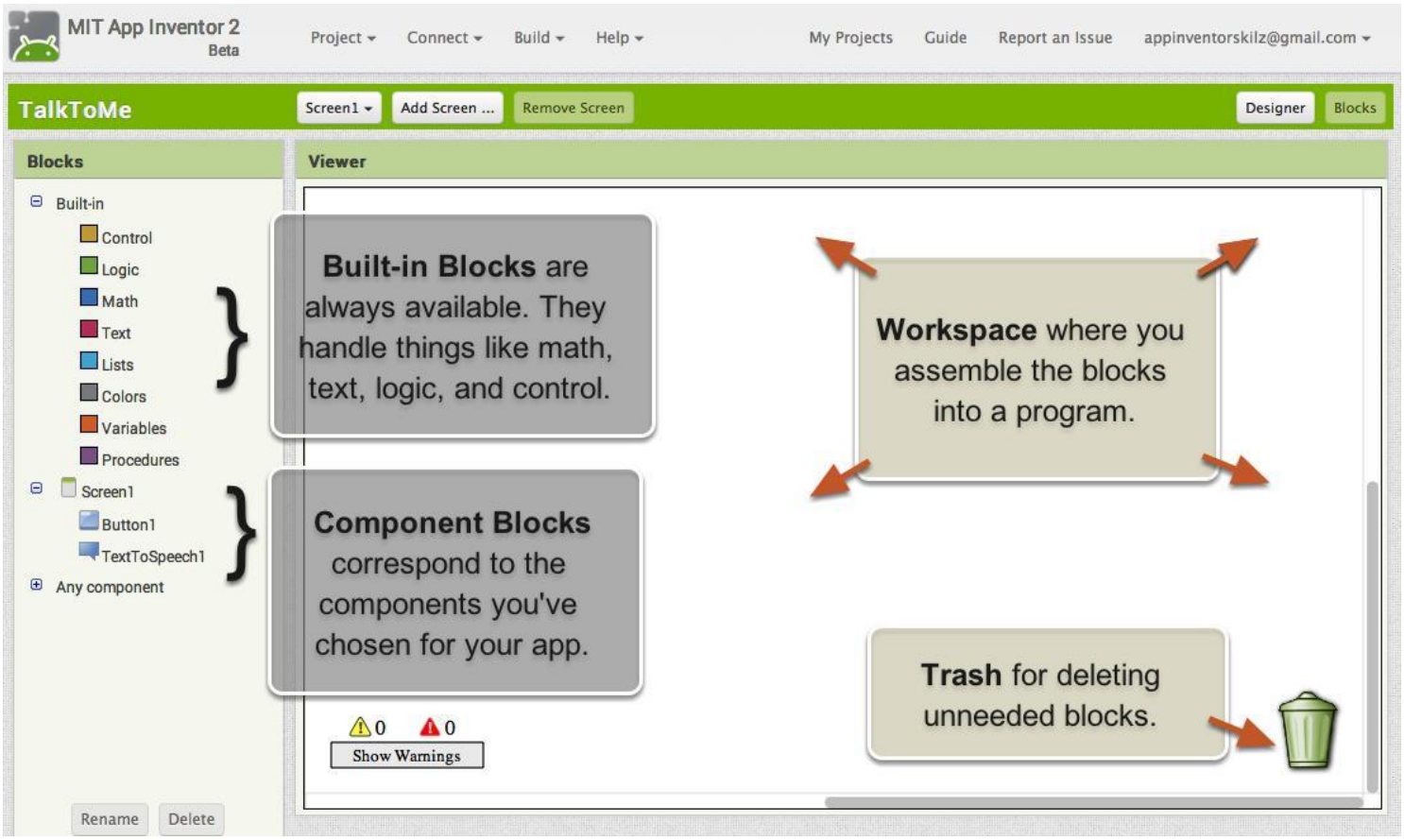
Click the button "Blocks" to move over to the Blocks Editor.
You will often toggle between the Designer and Blocks Editor as you develop apps.



13.The Blocks Editor

There are Built-in blocks that handle things like math, logic, and text. Below that are the blocks that go with each of the components you add to your app.

(*In order to get the blocks for a certain component to show up in the Blocks Editor, you first add that component to your app in the Designer.)*



14.Make a button click event

Click on the Button1 drawer.
Click and hold the ***when Button1.Click do*** event block.
Drag it over to the Viewer and drop it there.
This block will launch when the button on your app is clicked.
It is called an "Event Handler".

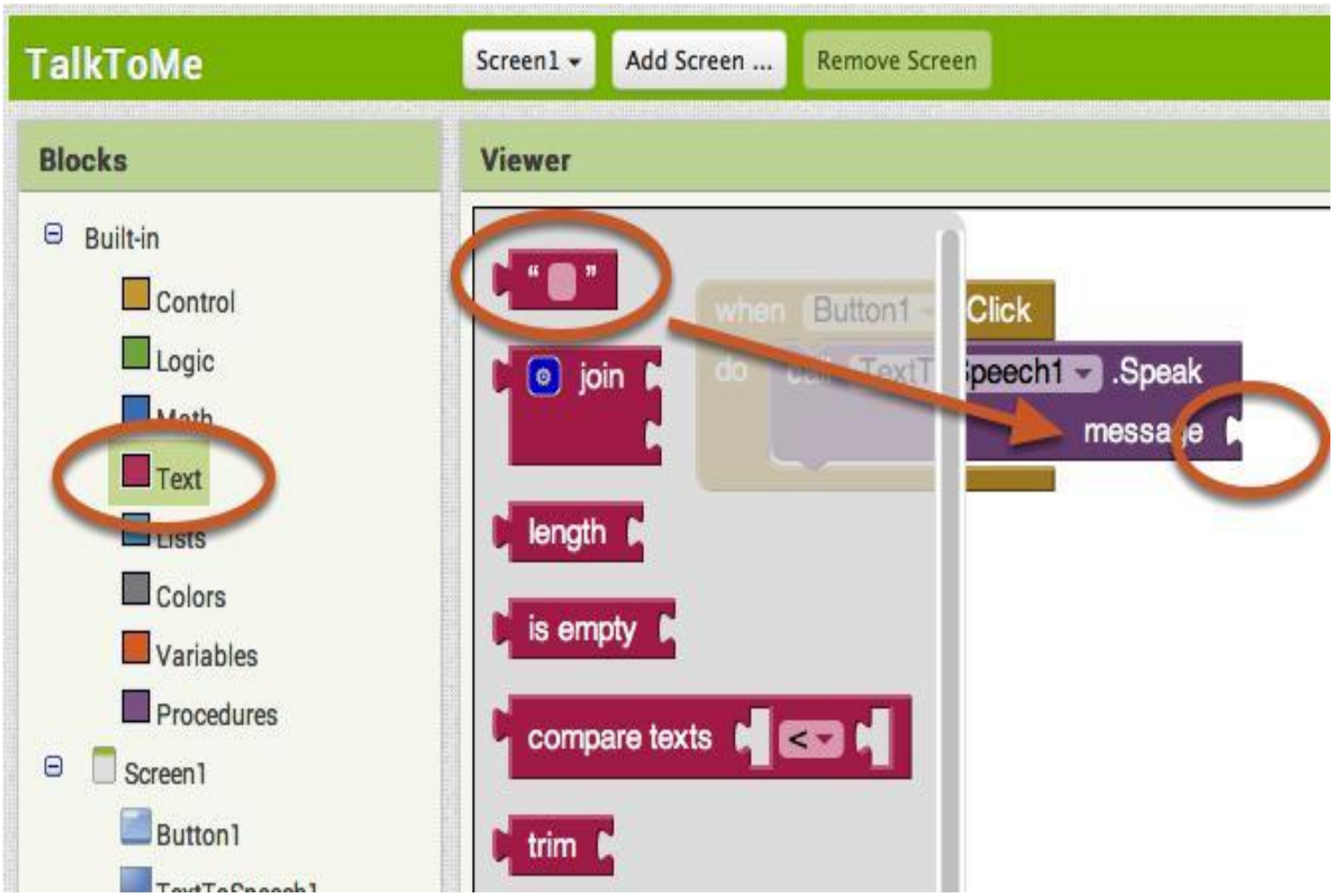
15.Program the TextToSpeech action

Click on the TextToSpeech drawer.
Click and hold the ***call TextToSpeech1.Speak*** block.
Drag it over to the Viewer and drop it there.
This is the block that will make the phone speak.
Because it is inside the Button Click, it will run when the button on your app is clicked.

16.Fill in the message socket on TextToSpeech.Speak Block

Now you need to tell the TextToSpeech.Speak block what to say.

Click on the Text drawer, drag out a ***text*** block and plug it into the socket labelled "message".



17.Specify what the app should say when the button is clicked

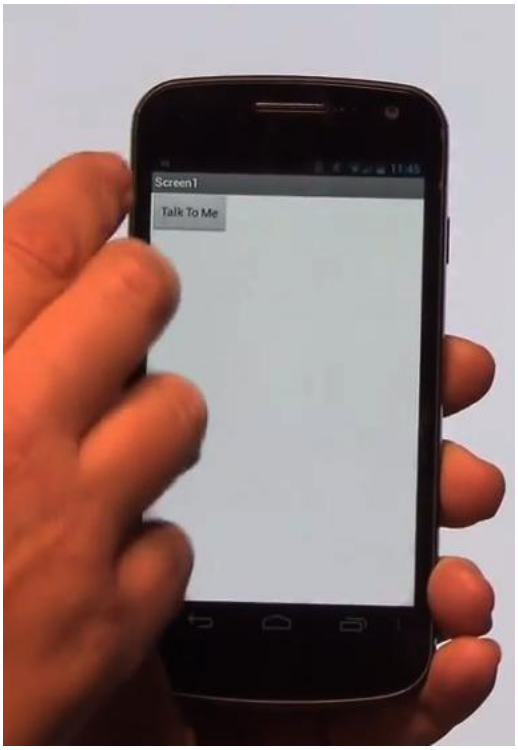
Click on the text block and type in "Congratulations! You've made your first app."

(Feel free to use any phrase you like.)

18. Now test it out!

Go to your connected device and click the button. Make sure your volume is up! You should hear the phone speak the phrase out loud.

(This works even with the emulator.)

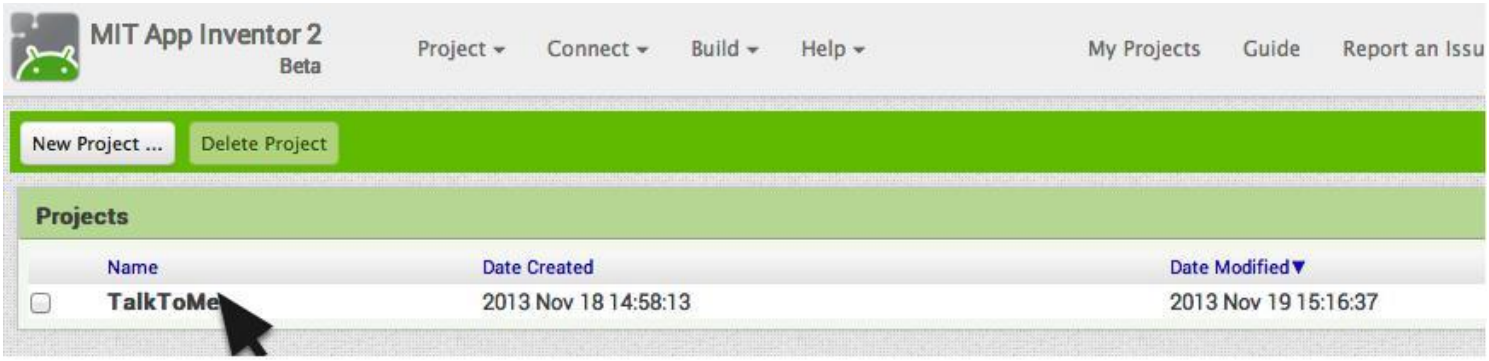


TalkToMe Part 2: Shaking and User Input

This tutorial shows you how to extend the basic TalkToMe app so that it responds to shaking, and so that the user can make the phone say any phrase s/he types in.

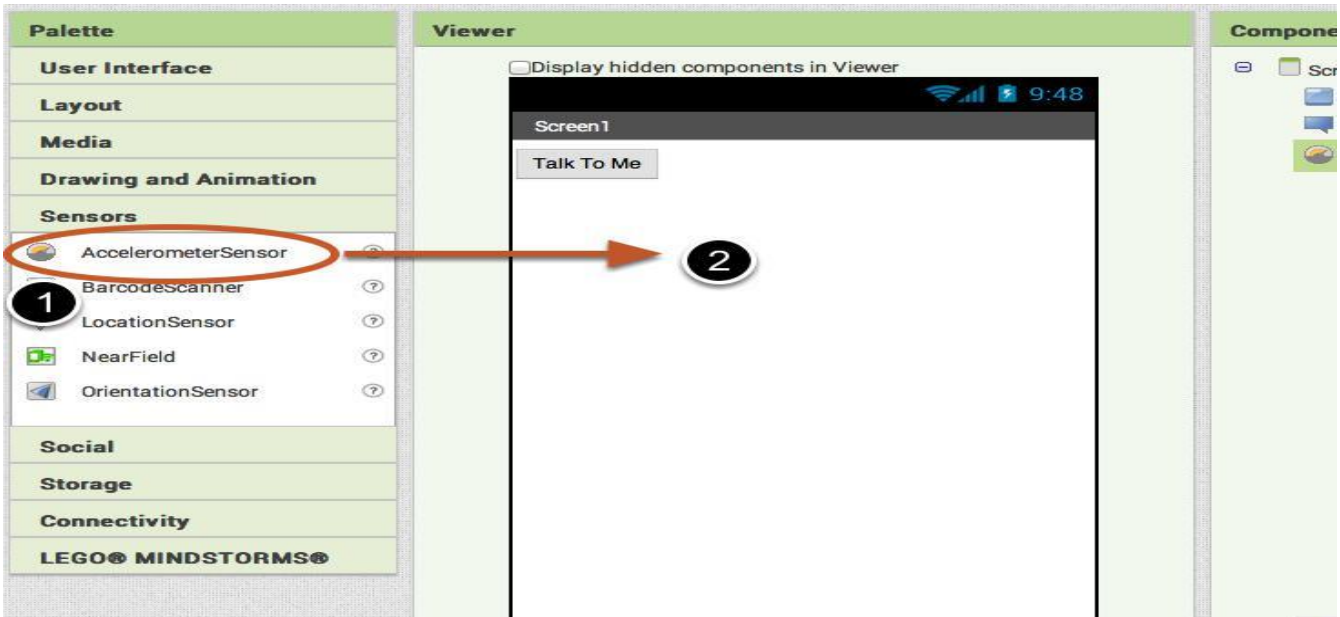
1. Go to App Inventor on the web and log in.

2. Open the "TalkToMe" project that you worked on in the last tutorial.



3. Add an Accelerometer Sensor

In the **Sensors** drawer, drag out an AccelerometerSensor component and drop it onto the Viewer. (It's a non-visible component, so it drops to the bottom of the screen.) NOTE: emulator users should skip this part and proceed to the next section of this tutorial called "Say Anything". (The emulator can not respond to shaking)

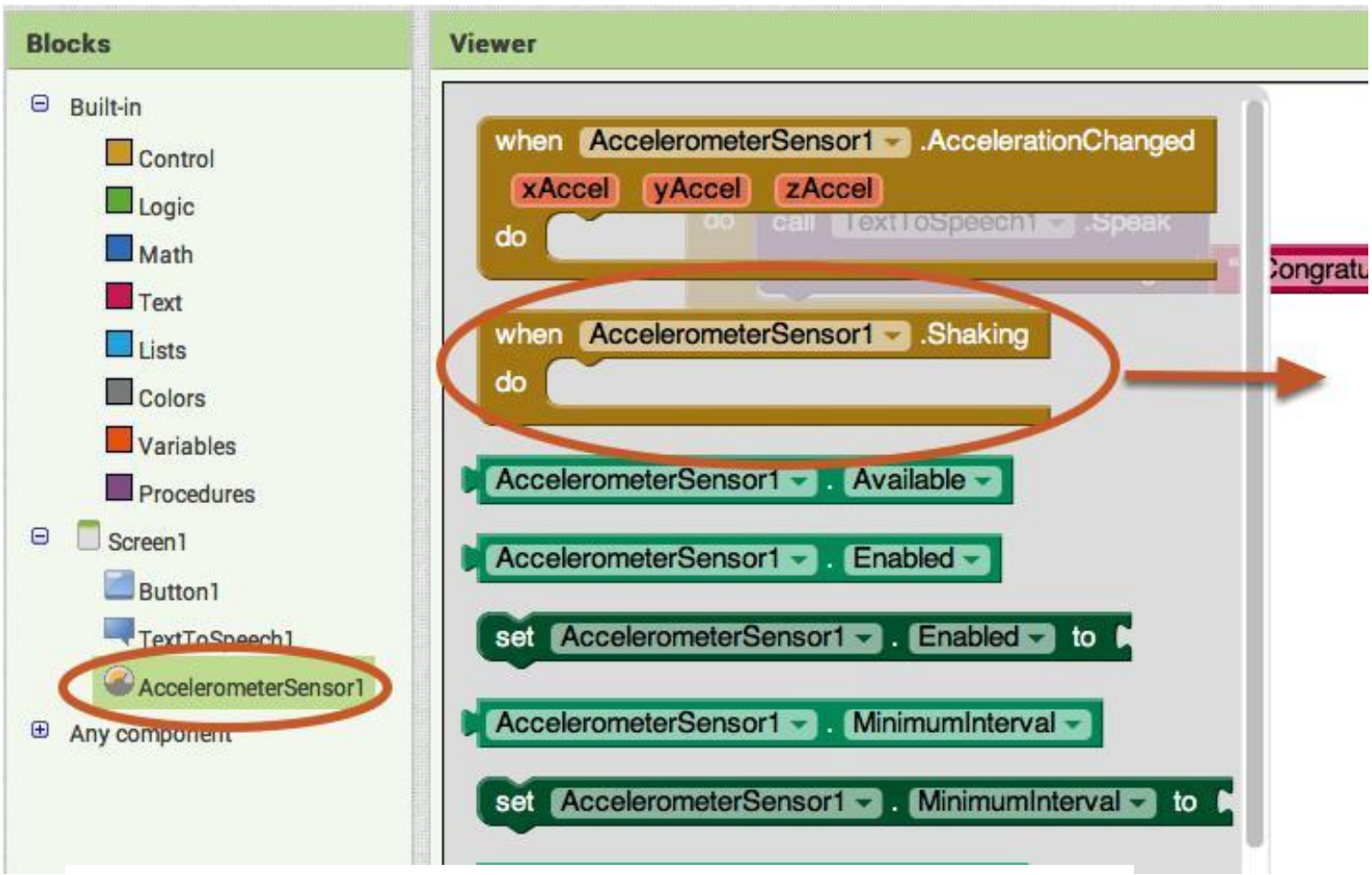


4. Go to the Blocks Editor

Click "Blocks" to program the new Accelerometer Sensor that you just added.

Program the Accelerometer Shaking event

Click the AccelerometerSensor1 drawer to see its blocks. Drag out the ***when AccelerometerSensor1.Shaking do*** block and drop it on the workspace.



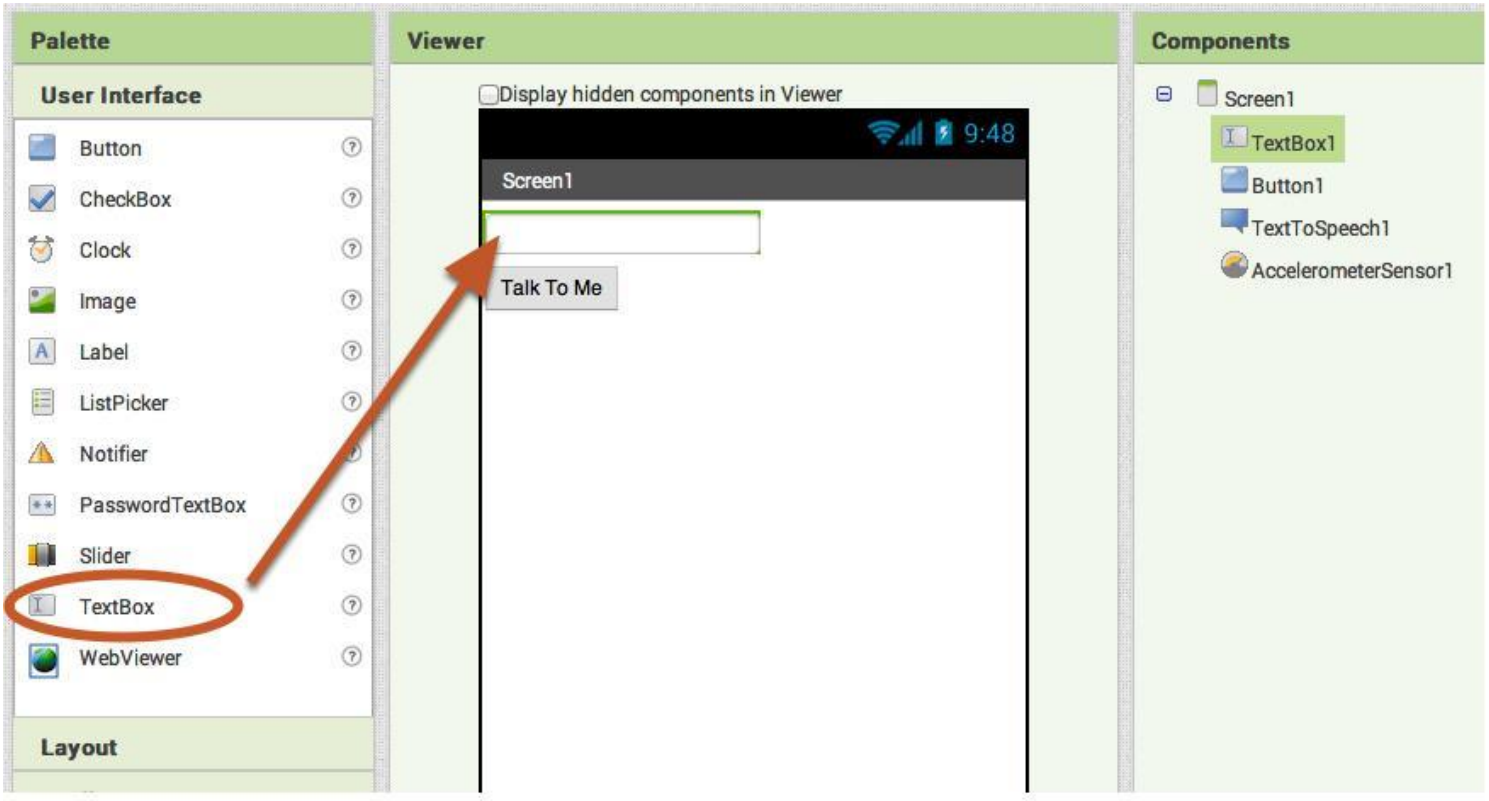
5.Test it out!

You can now shake your phone and it should respond by saying "Stop shaking me!" (or whatever phrase you put in.)

Is your phone talking to you? Cool! Now let's program the button click so that it causes the phone to speak whatever phrase the user put into the text box. Go back to the Designer.

6.Add a Text Box to your user interface.

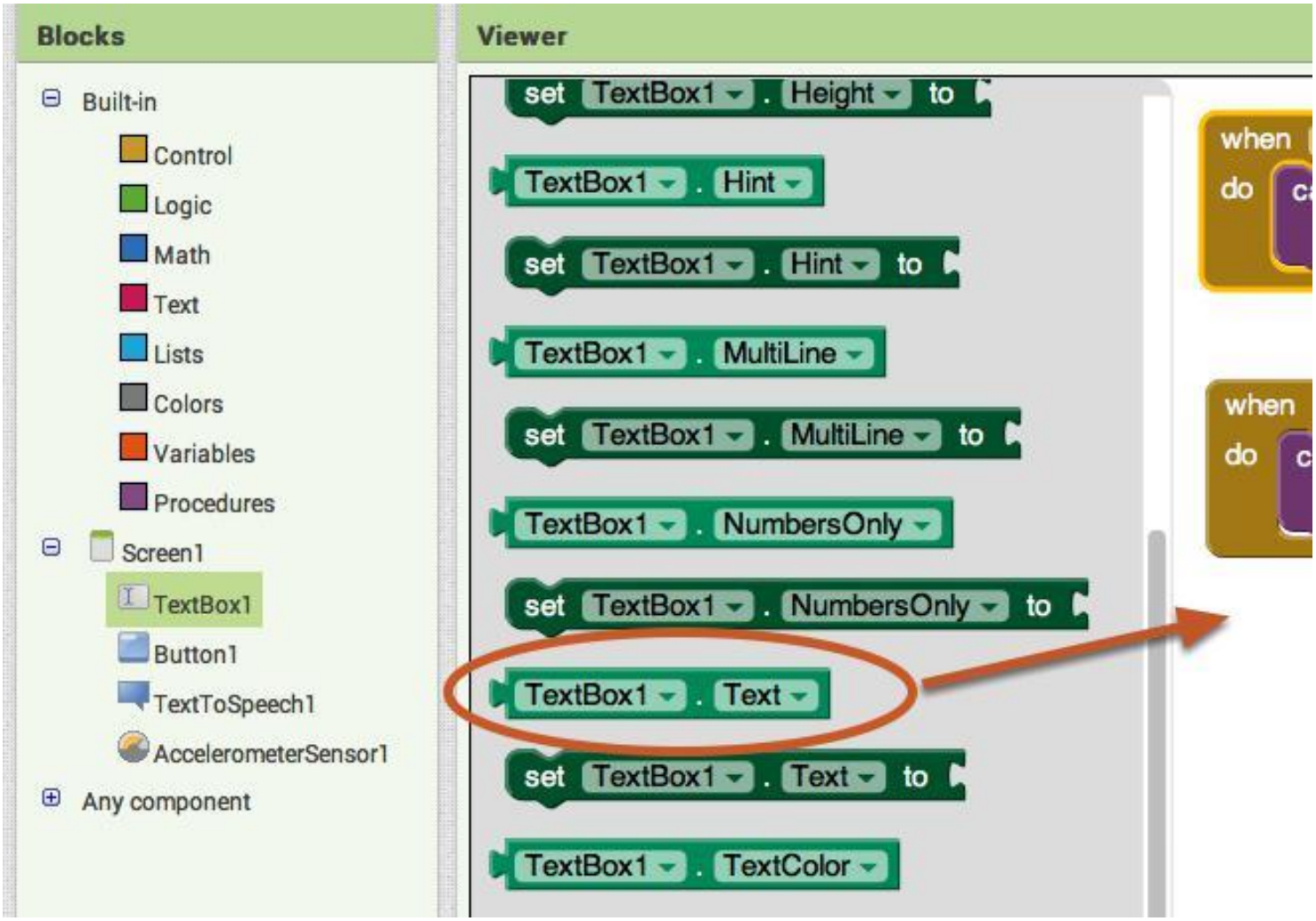
From the User Interface drawer, drag out a TextBox and put it above the Button that is already on the screen.



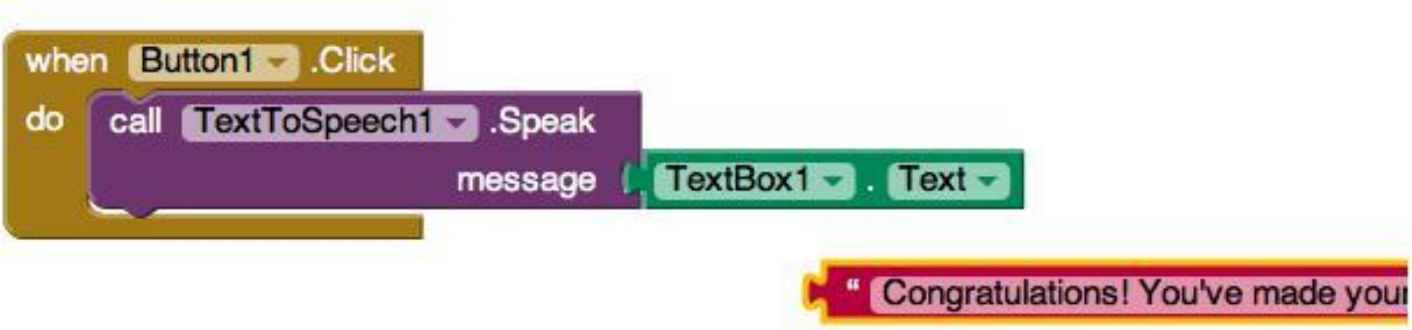
7.Back to the Blocks Editor!

8.Get the text that is typed into the TextBox.

Get the text property of the TextBox1. The green blocks in the TextBox1 drawer are the "getters" and "setters" for the TextBox1 component. You want your app to speak out loud whatever is currently in the TextBox1 Text property (i.e. whatever is typed into the text box). Drag out the ***TextBox1.Text*** getter block.



9.Set the Button Click event to speak the text that is in the Text Box.



10.Test your app!

Now your app has two behaviors: When the button is clicked, it will speak out loud whatever words are currently in the Text Box on the screen. (if nothing is there, it will say nothing.) The app will also say "Stop Shaking Me" when the phone is shaken.