MIT App Inventor

Creating Text to Speech Application in MIT App Inventor

Talk to Me

Aim:

Text-to-Speech is a screen reader application. It powers applications to read aloud (speak) the text on the screen with support for many languages. Users must install voice data for each language.

Components:

- Screen
- Button
- Textbox
- Text to Speech

Screen:

Top-level component containing all other components in the program.

Button:

Buttons are components that users touch to perform some action in your app. Buttons detect when users tap them. Many aspects of a button's appearance can be changed. You can use the Enabled property to choose whether a button can be tapped.

Textbox:

Users enter text in a text box component. The initial or user-entered text value in a text box component is in the Text property. If Text is blank, you can use the Hint property to provide the user with a suggestion of what to type. The Hint appears as faint text in the box.

TextToSpeech:

TextToSpeech. The TextToSpeech component speaks a given text aloud. You can set the pitch and the rate of speech.

Designer Part:

If you don't have any projects created in App Inventor, you will land in the Projects View. Start a new project by clicking the "Start new project" button.
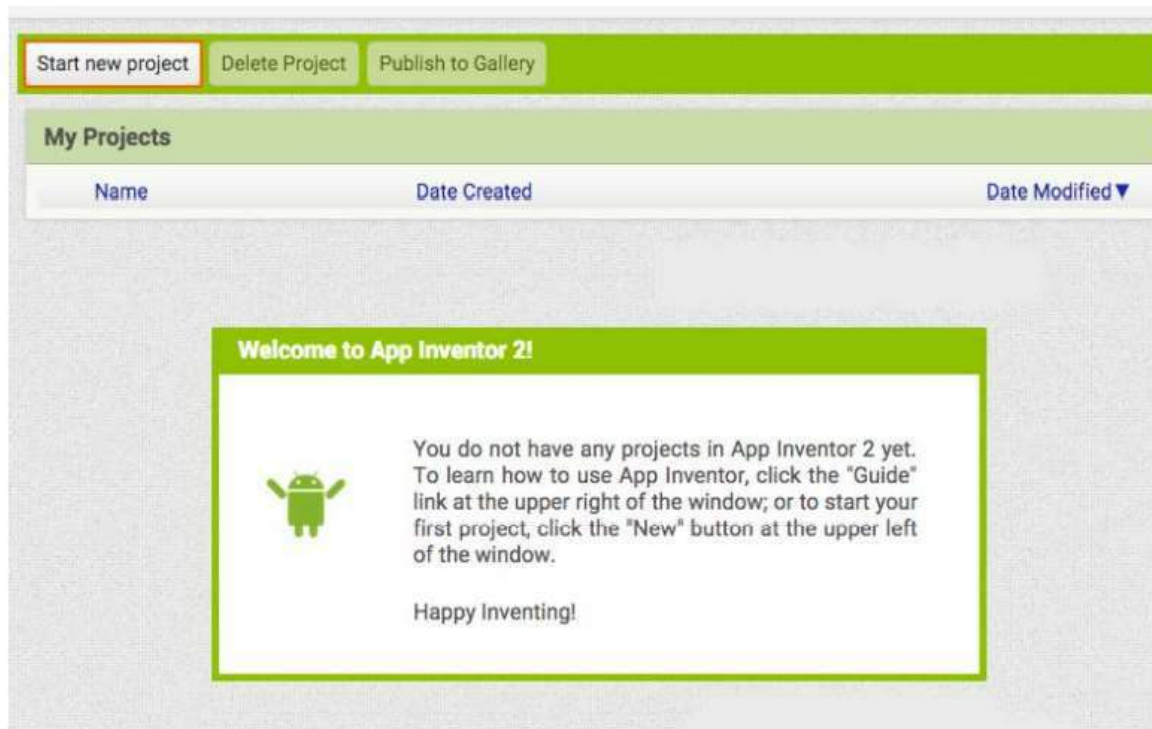
Name the project "TalkToMe" (no spaces)
Type in the project name (underscores are allowed, spaces are not) and click OK.

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.
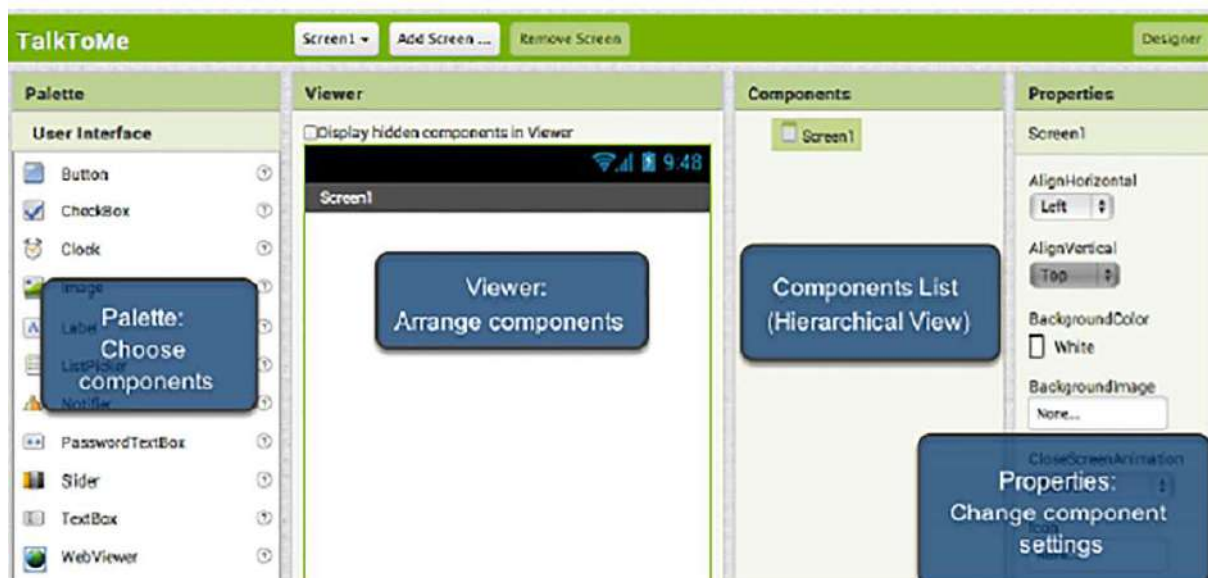
After taking a screen from User Interface here we have kept a picture like a speech representation.

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.
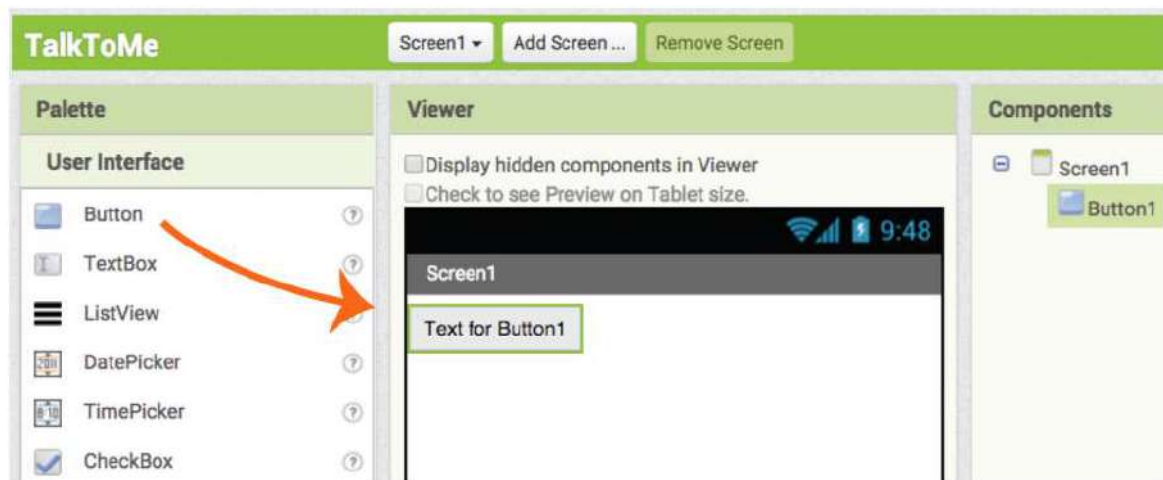
And we can rename the text on the button. In this project we are changing that name to Talk.

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.
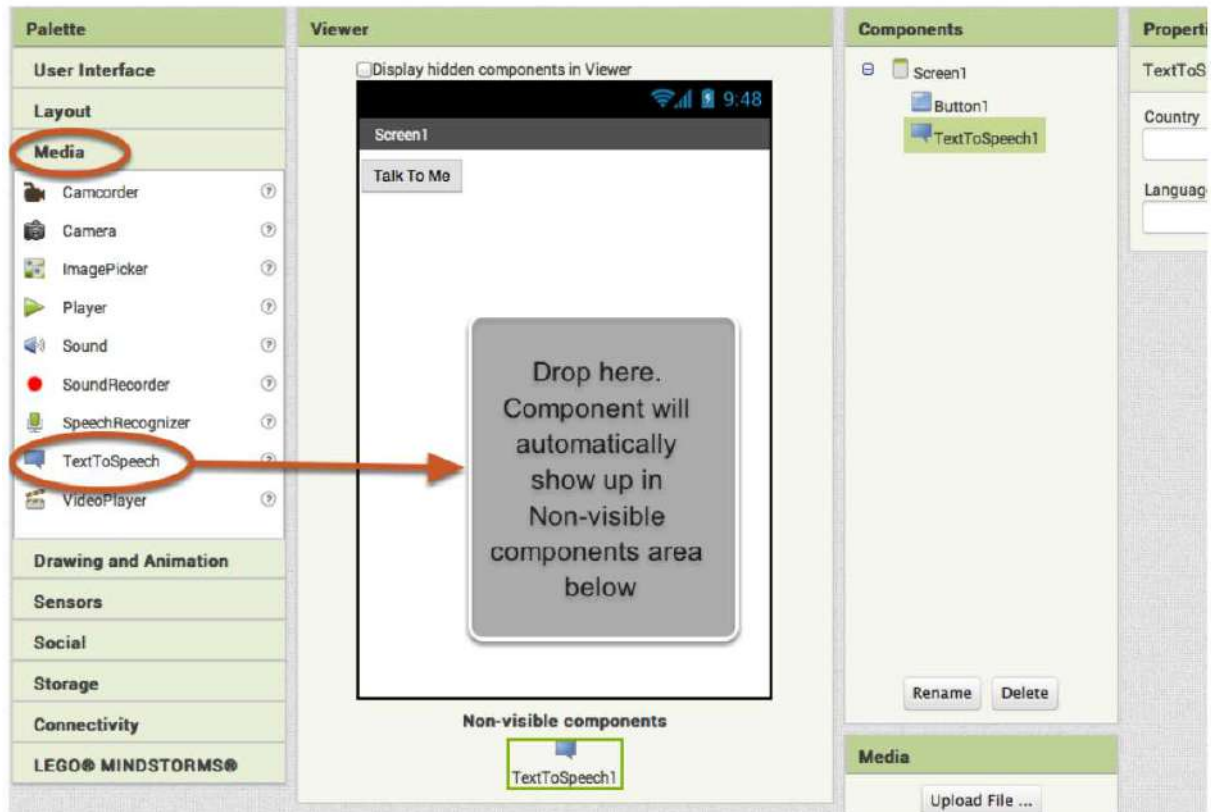
Blocks Part:

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.
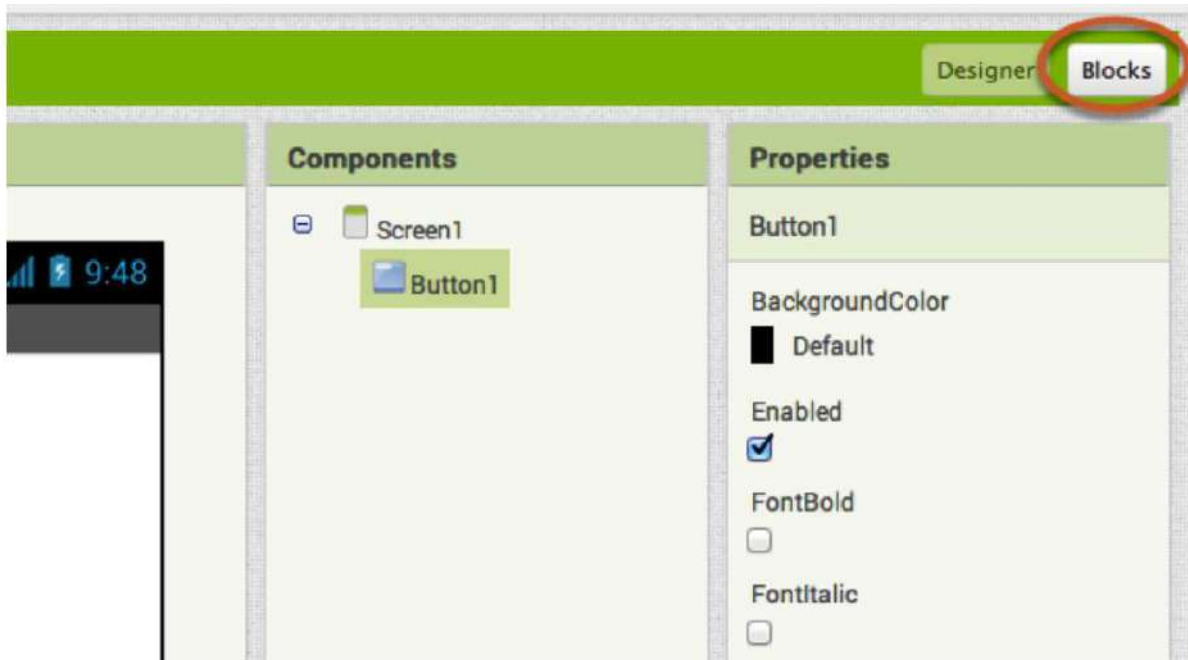
Blocks Editor: -

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.

Make a button click event: -

Click on the Button1 drawer. Click and hold the when Button1.Click 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".
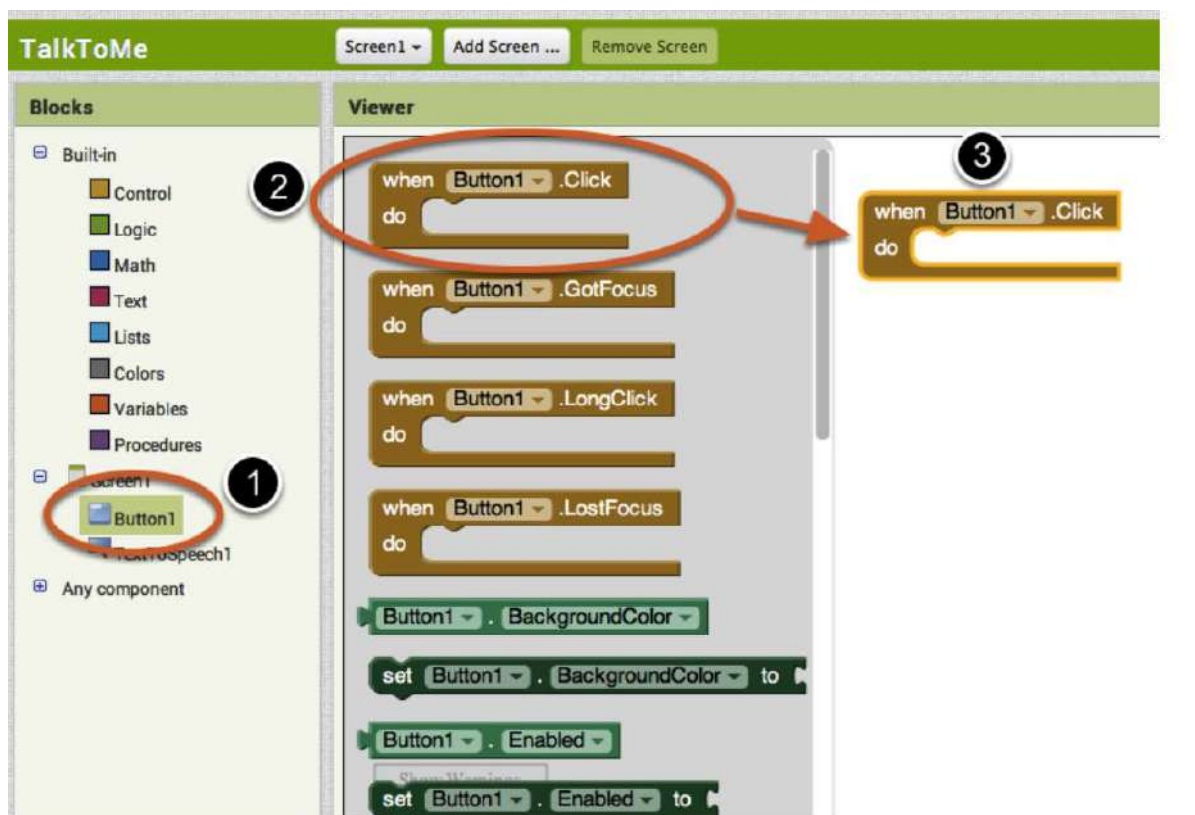

Program the TextToSpeech action: -

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.
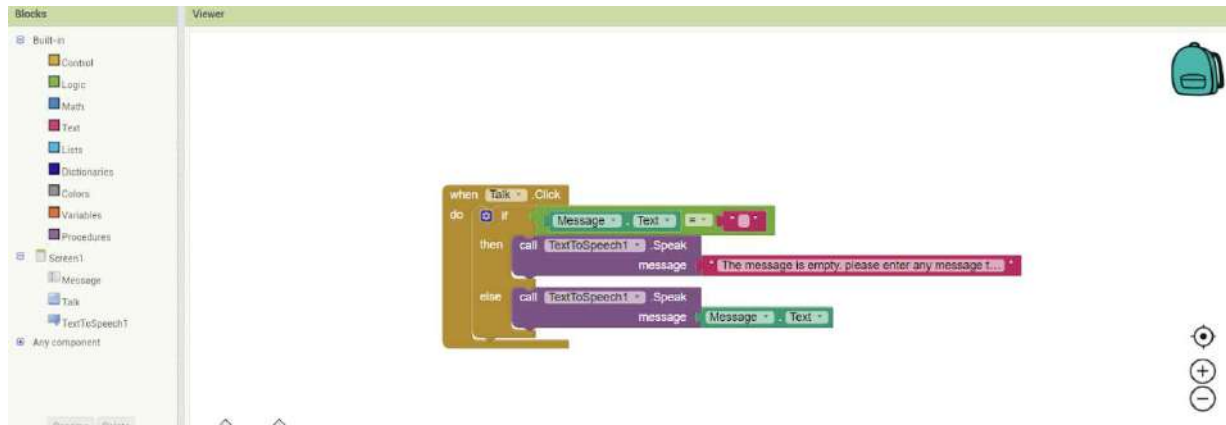
Fill in the message socket on TextToSpeech. Speak Block: -

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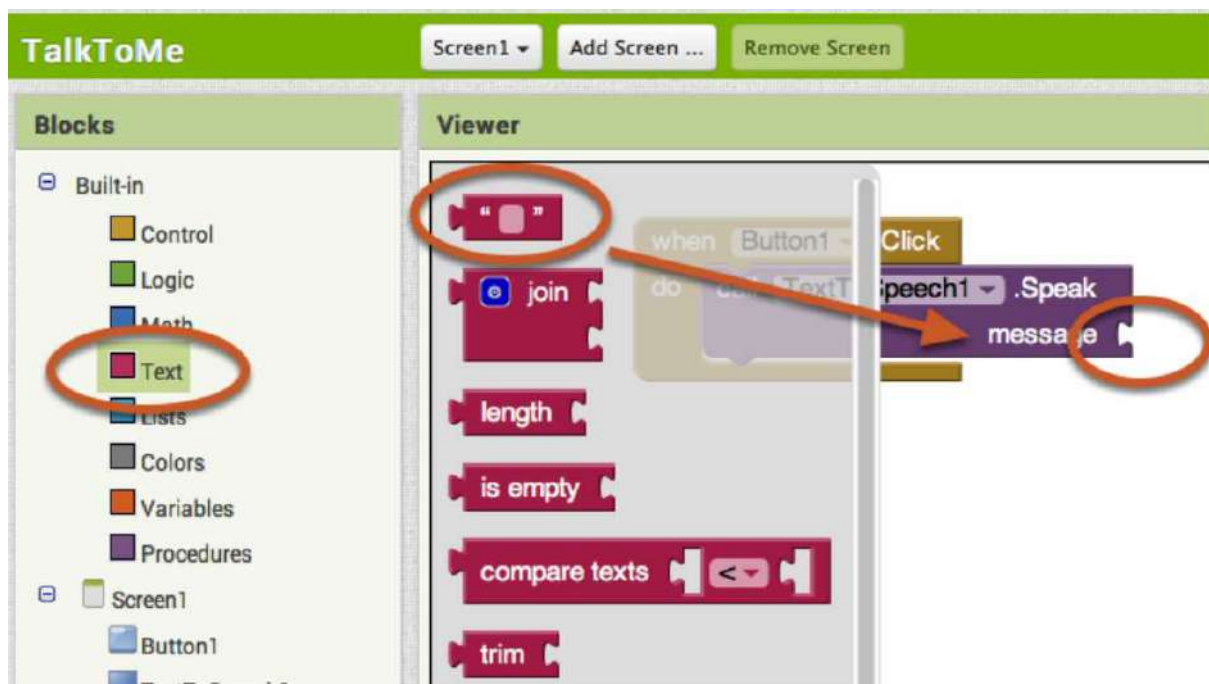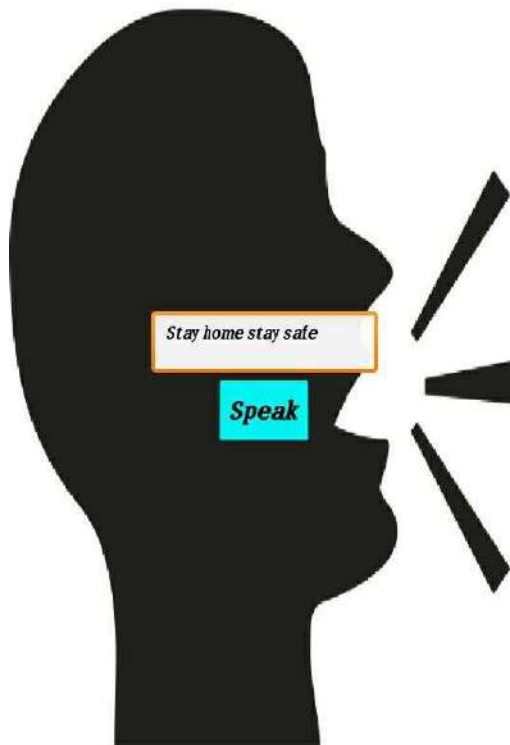
Specify what the app should say when the button is clicked

Output:

Conclusion:

We are entering the text into the text box as you can see in the above picture. After entering the data, you just click the button which is named by Speak then it speaks out.

THANK YOU