

BRAIN HACKERS APP DEVELOPMENT CURRICULUM

Lesson 1

Question-Answer App – Labels, Buttons and Text-to-Speech

SUMMARY

This section introduces the use of MIT App Inventor to create a simple app that utilizes text-to-speech. For this app, a series of questions will be presented and when a user taps adjacent buttons, answers will be spoken. This exercise demonstrates question asking and self-testing as an aid to learning new information.

DESIGNER WORKSPACE

- In the Palette panel, from the list of User Interface components, drag a Label onto the Viewer - *NOTE: the Label will present the user a question*
 - o In the Components panel, rename the Label, “Question_1_Label”

Helpful Tips

You should always rename components with a name that will help you recognize the component and distinguish it from other components

- o In the Properties panel, change the Text for the Label to, “What is a neuron?”
- From the User Interface components, drag a Button onto the Viewer and place it below the Label – *NOTE: when a user taps the Button, they will trigger the Text-to-Speech component, which will say the answer*
 - o In the Properties panel, change the Text for the Button to, “Answer”
- From the Media components, drag a Text-to-Speech component onto the Viewer – *NOTE: this is a Non-visible component and will appear below the Viewer*

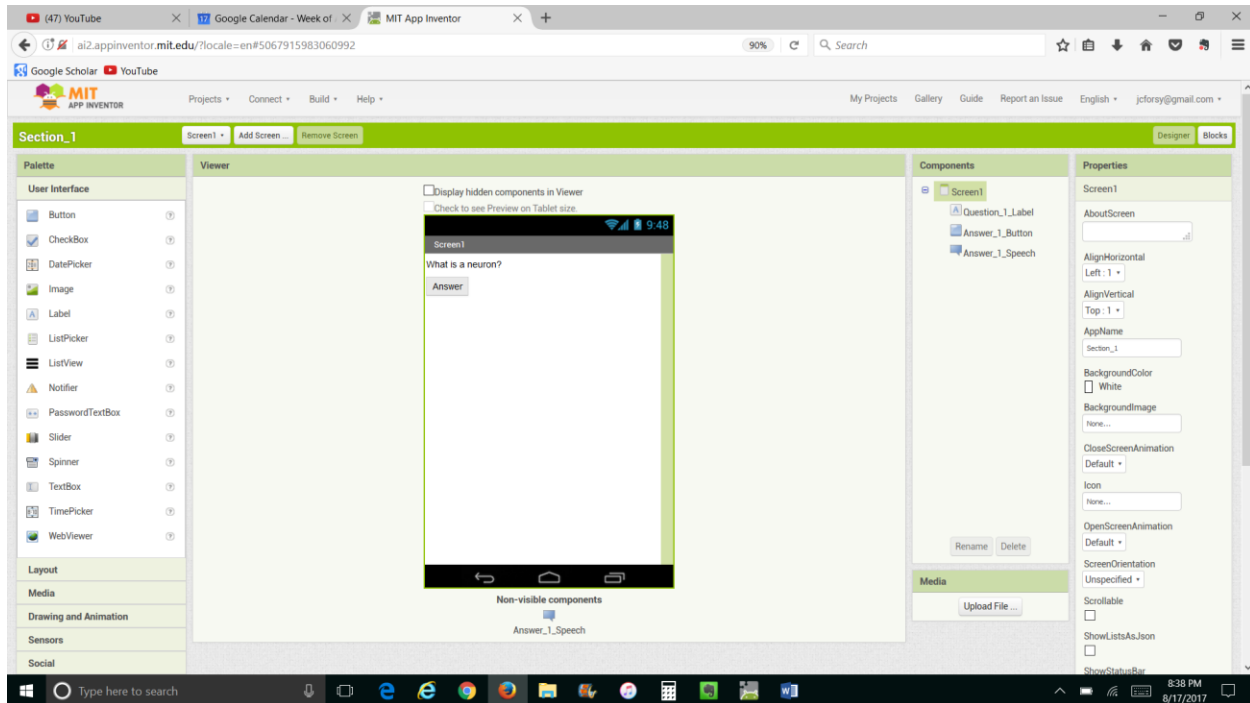


FIGURE 1. This image shows how the Designer workspace should appear. Note that in the Components panel, each of the components has been renamed

BLOCKS WORKSPACE

- From the Blocks panel, open the blocks for the Button and select the **when ____ .Click do** block, which inserts this block into the program – *NOTE: this block is referred to as an “Event Handler,” which provides instructions for what to do when a button click (i.e., the event) occurs*

Key Concepts

Event Handler – an event handler is triggered when an event such as a button press occurs and contains instructions for what to do in response to or as a result of the event

- Open the blocks for the Text-to-Speech and select the **call ____ .Speak** block and insert the **call ____ .Speak** block into the **when ____ .Click do** block – *NOTE: this block is referred to as a “Procedure Call” and directs the program to carry out a procedure, which here has the Text-to-Speech say the text given in a text string attached to its **message** slot*

Key Concepts

Procedure Call – contains instructions to perform a task

- From the Blocks panel, open the Text blocks and select the text string block, which is the first block in the list and has a space in which text may be inserted, and attach it to the **message** slot of the **call ____ .Speak** block – *NOTE: this block is a “text string,” which may contain words, numbers or special characters that will be presented in a Label or will serve as the input to some function such as Text-to-Speech*

Key Concepts

Text String – a series of letters, numbers and/or special characters

- In the text string block, enter the text, “A specialized cell that can transmit electrical currents”

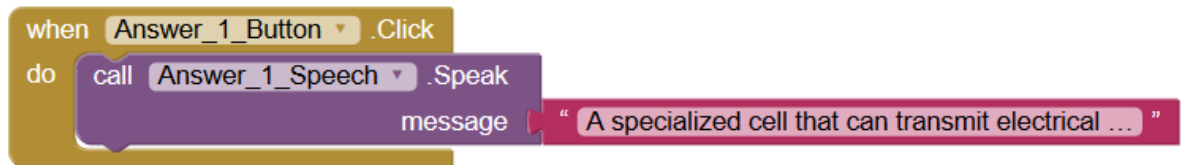


FIGURE 2. The Blocks workspace should consist of a single **when ____ .Click do** block that appears as shown

ADJUST PROPERTIES OF TEXT-TO-SPEECH

Text-to-speech is also known as “Speech Synthesis,” which refers to the capability to create a human-like voice with a computer. App Inventor allows you to adjust the properties of the computer-generated voice.

- From the Properties panel for the Text-to-Speech component, select the Language pulldown menu and change the accent to either German (de), Spanish (es), French (fr) or Italian (it) – *NOTE: additional components that will be covered in a later section are required to translate the text to another language*

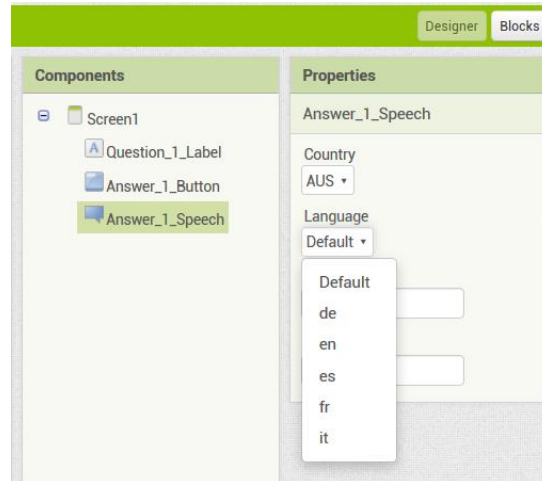


FIGURE 3. The Language setting for the Text-to-Speech properties allow the accent of the spoken voice to be changed

- To increase or decrease the frequency of the spoken voice, change the setting for Pitch
– *NOTE: a value larger than “2” will produce no effect*
- To make the Text-to-Speech talk faster or slower, adjust the setting for the SpeechRate -
– *NOTE: a value larger than “2” will produce no effect*

QUESTION-ANSWER APP EXERCISE

Create a series of questions and answers based on an assigned topic and practice answering – be sure to share the app with others and take turns answering each other’s questions. Be aware that each answer will require a separate Text-to-Speech.

FINALIZING AN APP

There are a few additional steps needed to create an app that will look professional and can be loaded on your phone or distributed through the Google Play Store or other sites.

- From the Properties panel for Screen 1, change the AppName to “QA App” or some other name that would tell a user about the app – *NOTE: this is the name that will appear for the app on the phone or device*
- Download a relevant image (e.g., a question mark) and save it to the computer – *NOTE: if you plan to share your app with others, you should only use copyright free images, which is a setting in Advanced Settings with Google Image Search*

- For the Icon, which is the image that will appear for the app on a phone or device, select Upload File – *NOTE: this will open a separate window to identify the file to be uploaded*
- Locate the image to be used as an icon and upload the image file

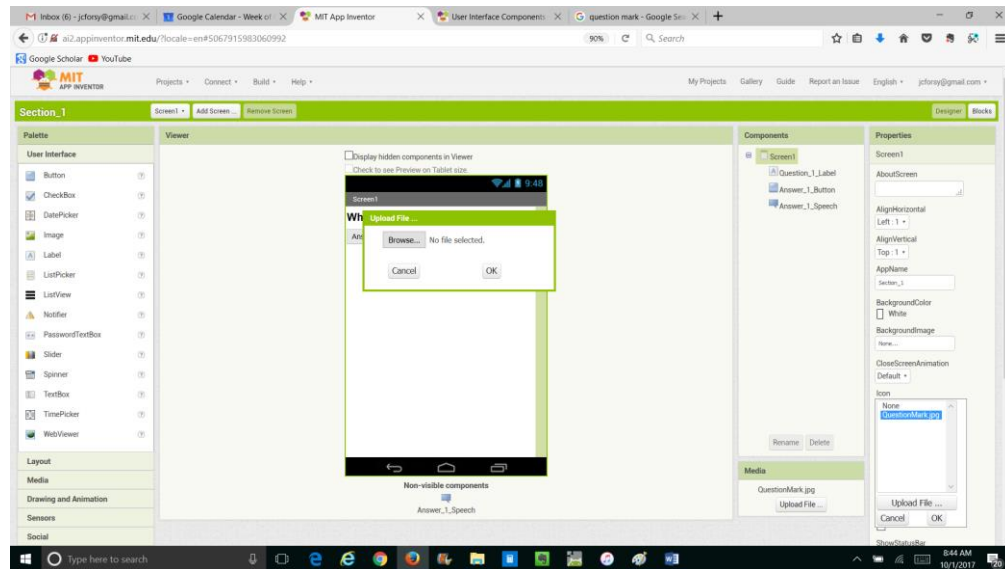


FIGURE 4. The icon that will appear on the phone or device for the app can be set by uploading an image file to serve as the icon

- For apps that will be best viewed in either Portrait or Landscape orientation, set the ScreenOrientation
- Change the Title to the name of the app – *NOTE: the title will appear at the top of the screen when running the app*

PACKAGING THE APP

There are two ways to package an app so it can be loaded on a phone or device:

1. Directly load the app onto the phone or device
 - From the Build menu, select “App (provide QR code for .apk) – *NOTE: this will open a barcode that when scanned with the phone or device, will load the app*
 - Using the MIT AI2 Companion app on the phone or device, scan the QR code

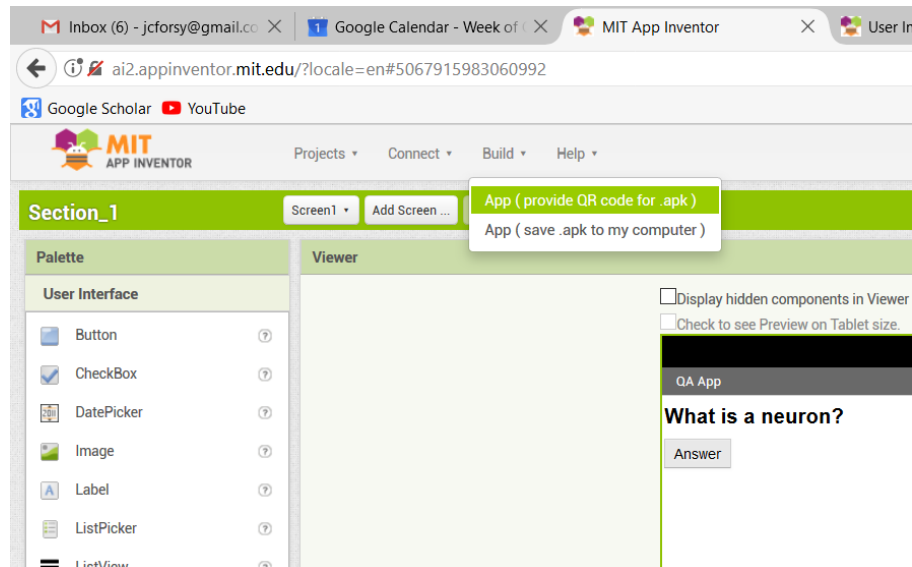


FIGURE 5. By selecting “App (provide QR code for .apk,” a barcode is generated that when scanned using the MIT AI2 Companion app, will load the app on the phone or device

2. Distribute the app through email

- From the Build menu, select “App (save .apk to my computer) – *NOTE: this will save a .apk file to the Downloads folder of your computer*
- Send an email, with the .apk file as an attachment to the email
- Once the email is received, open the attachment with the .apk file – *NOTE: opening the attachment will install the app on the phone or device*

Be aware that for either of these solutions to work with a given Android phone or device, it is necessary within the security settings of the phone or device to allow apps to be installed from unknown sources. This should be a toggle that either allows or disallows.