Introduction to Programming Using MIT App Inventor
Freshman Exploratory Curriculum

Lesson 1 – Cowbell App
Lesson 2 – Zoltar Knows All

Prepared by: Larry Lambert
September, 2014

Lesson Plan

CTE Area: Programming and Web Development

Unit of Instruction: Introduction to Programming Using MIT App Inventor

Topic of Instruction: Freshman Exploratory – Computer Programming

Overview

Goal

The goal of this lesson is capture the interest of freshman students by exposing them to the creation of Android smart phone applications during their exploratory experiences in Programming and Web Development.

Student Learning Objectives

At the conclusion of these lessons, students will be able to:

- Utilize the Chrome web browser to access MIT's App Inventor.2.
- Implement a simple android app using MIT's App Inventor 2.
- Construct event driven programs using application controls and device sensors.
- Implement the following programming concepts:
  - Add a label and set the label text.
  - rename a component.
  - create a variable.
  - create a list.
  - select a random item from a list and display it.

State Frameworks/Professional Standards

2.C Programming Concepts

2.C.01 Implement concepts fundamental to programming.
- 2.C.01.01 Describe what a computer program is and how it runs.
- 2.C.01.02 Demonstrate the use of a debugger.
- 2.C.01.03 Utilize Integrated Development Environments.

2.C.03 Construct a program.
- 2.C.03.01 Create a menu driven application.
- 2.C.03.02 Create an interactive application.

Connections Across Curriculum

None in this lesson

Materials and Time

Material required by instructor:

- An internet connected computer loaded with Microsoft Windows.
- Basic software requirements; App Inventor running via the Chrome Browser)
- LCD Projector

Material required by students:

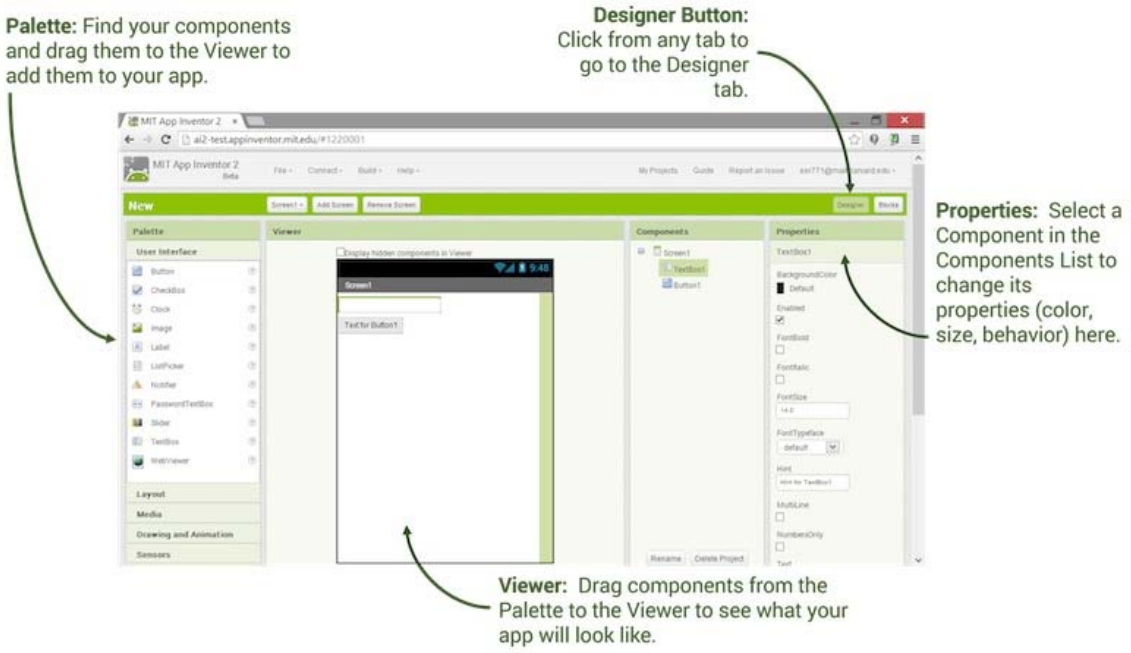
- An internet connected computer loaded with Microsoft Windows.
- Basic software requirements; App Inventor running via the Chrome Browser)

Instructional and activity time required to complete this lesson is approximately 5 hours

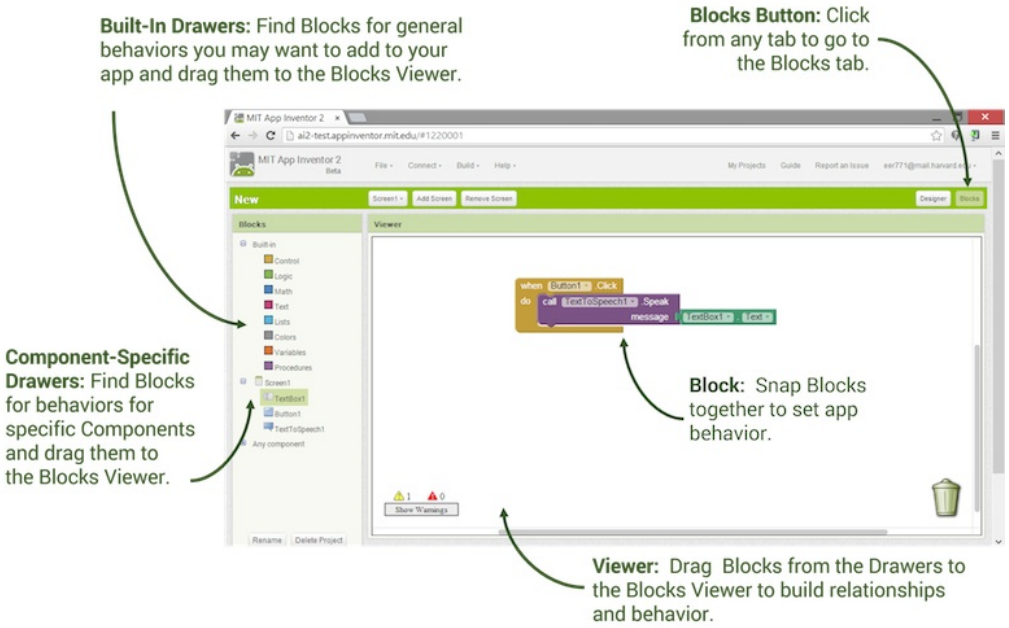
Instruction

Introduction – App Inventor Overview

The **App Inventor Designer** shown below is used to design the user interface, including both the on-screen and off-screen components included in the interface.



The **App Inventor Block Editor** shown below is used to program the application by including the code blocks necessary to cause the desired behavior by the app.

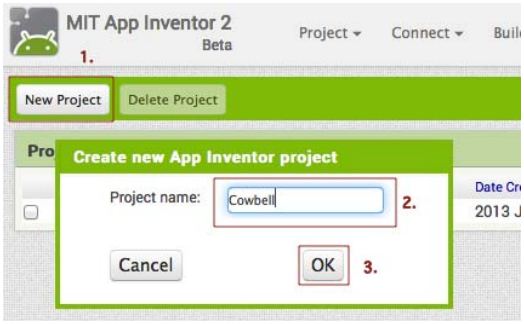


Lesson One – Cowbell App

Adapted from the website: http://ice-web.cc.gatech.edu/dl/?q=node/875

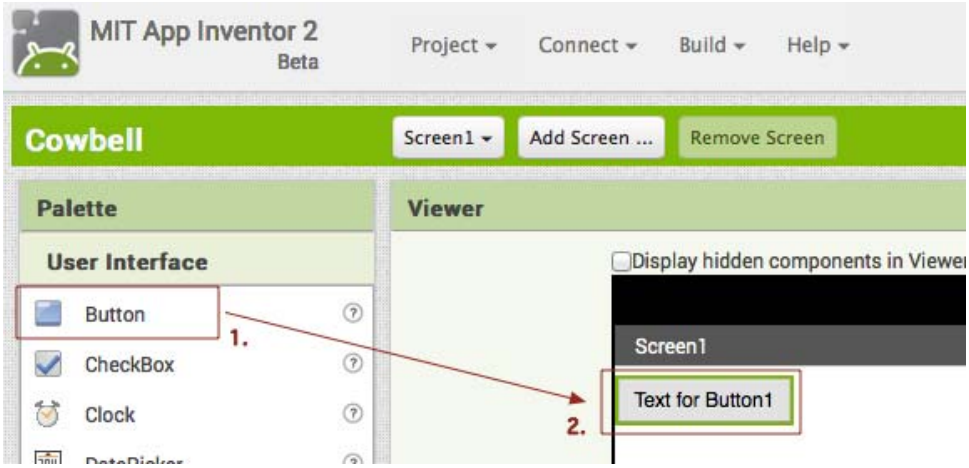
Create a project

1. Create a new project by clicking on the *New Project* button (1) and type in the name of the project: "Cowbell" (2). Then click on *OK* (3). This will create a new project named "Cowbell". Projects in App Inventor 2 are automatically saved to the cloud.



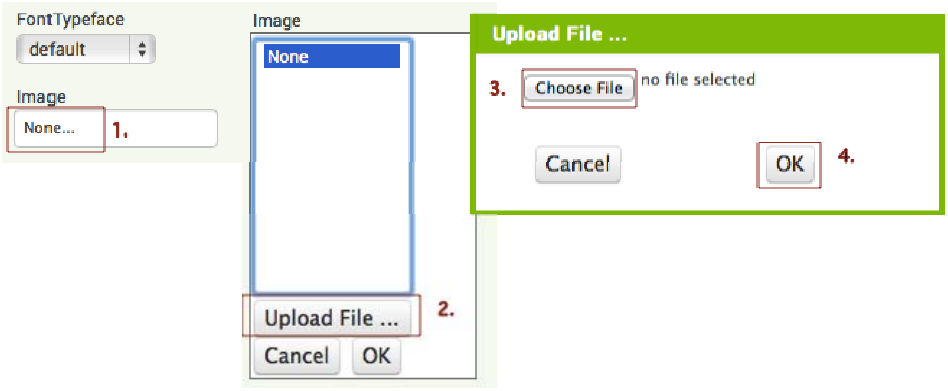
Create component - Button

From the *User Interface* palette (leftmost-column) drag a **Button** (1) to the viewer (column to the right that looks like a phone screen and says "Screen1") (2). A **Button** is something a user can push. The viewer shows the current user interface for the app.

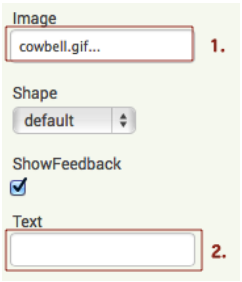


Set properties - Button

1. The properties for the button are shown in the right-most column. Set the image for the button to "cowbell.gif" by clicking on the *None...* (1) under Image, then click on *Upload File ...* (2), then click on *Choose File* (3) which will bring up a file picker. Select the "cowbell.gif" file that you just downloaded and click on*OK* (4). The button should then show the cowbell picture. If it doesn't show the picture check that you uploaded "cowbell.gif" and not "cowbell.wav". If you made a mistake just also upload "cowbell.gif" and set the image to that file.

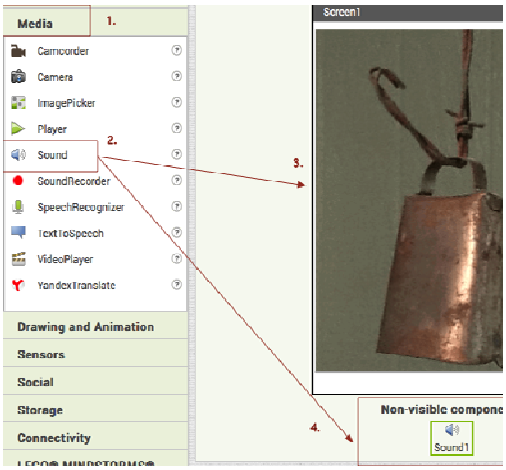


2. Check that the image property is set to "cowbell.gif" (1). Remove the default text for the button by selecting it and hitting the *delete* key on the keyboard (2).



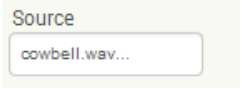
Create Component - Sound

Click on the *Media* category (1) in the palete (leftmost column) to view the media components and drag out a **Sound** (2) to the viewer (3). It will show up as a non-visible component under the display of the phone screen (4).



Set Properties - Sound

In the properties column set the source to "cowbell.wav" (upload the file), just like you did with "cowbell.gif".



Program and test

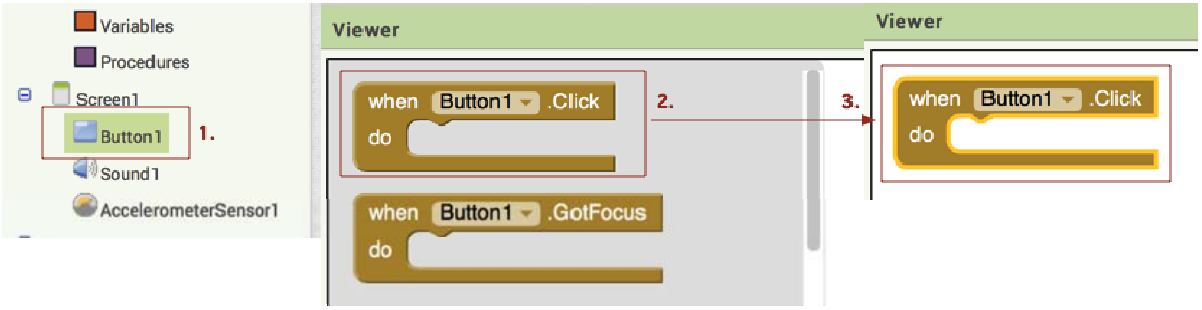
In this step we will program the app in the blocks editor to play the cowbell.wav sound file when the button is pushed and test the app. Doing something when an event occurs, like when a button is clicked, is called **handling an event**.

Open the Blocks Editor

Open the blocks editor by clicking on the *Blocks* button in the upper right corner of the Designer.

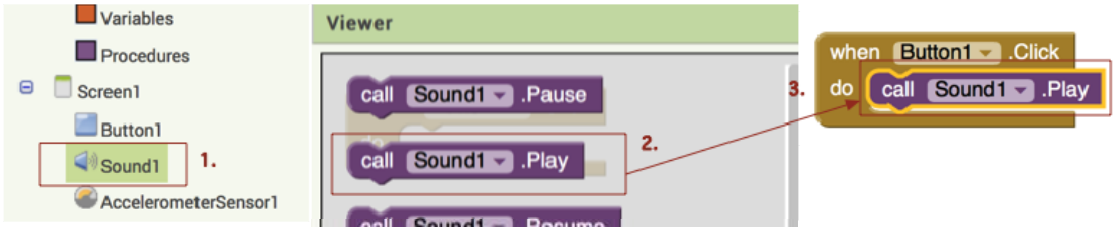
Handle Event - button click

Click on *Button1* (1) under *Screen1* in the left Blocks panel to see the blocks for the button you created. Drag out a **when Button1.Click** block (2) and drop it in the viewer area on the right (3). This block will run each time Button1 is clicked.



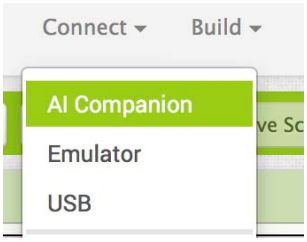
Specify Action - play sound

Click on *Sound1* (1) in the left Blocks panel and drag out a **call Sound1.Play** block (2) and put it inside the **do** area of the **when Button1.Click** block (3) as shown below. This will play Sound1, the cowbell.wav sound, each time you click on the button.



Test with a WiFi Connected Device

1. Click *Connect* > *AI Companion* on the upper task bar. A QR code will appear in the browser.

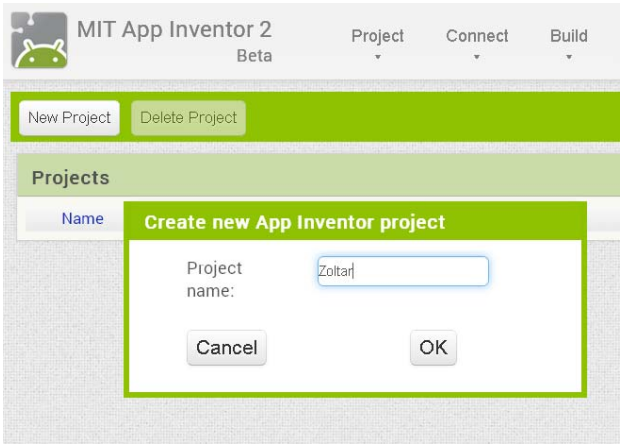


2. Open the MIT AI2 Companion app on the phone and scan the QR code or type in the code. In a few seconds, the app will appear on the device.

Lesson Two – Zoltar Knows All

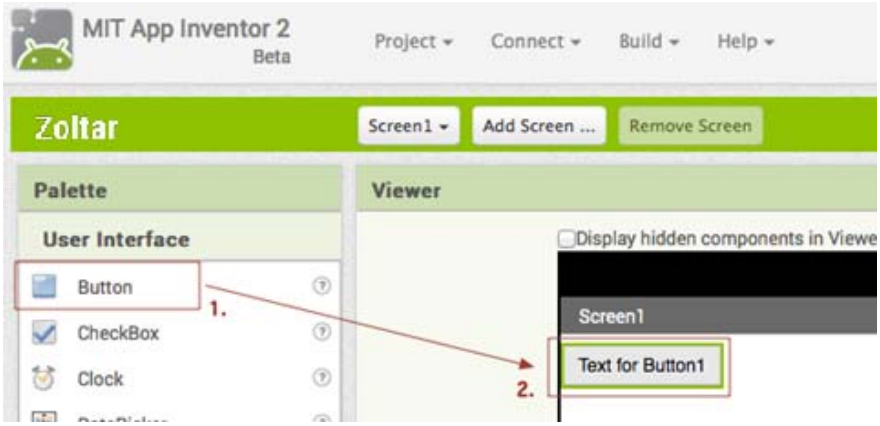
Create a project

2. Create a new project by clicking on the *New Project* button (1) and type in the name of the project: "Zoltat" (2). Then click on *OK* (3). This will create a new project named "Zoltar". Projects in App Inventor 2 are automatically saved to the cloud.



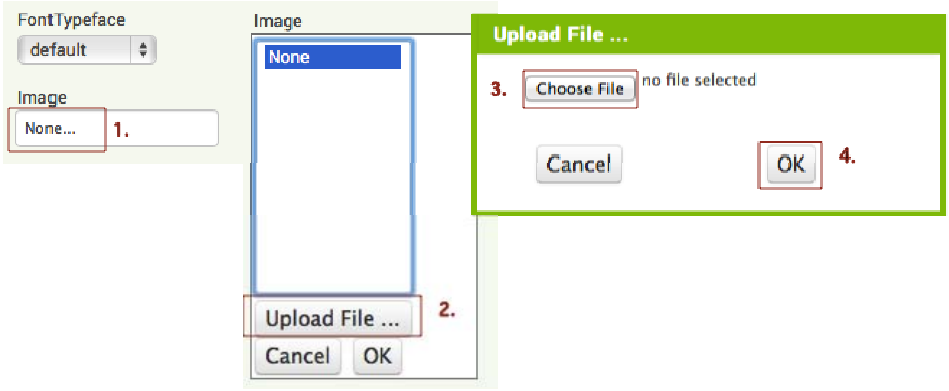
Create component - Button

From the *User Interface* palette (leftmost-column) drag a **Button** (1) to the viewer (column to the right that looks like a phone screen and says "Screen1") (2). A **Button** is something a user can push. The viewer shows the current user interface for the app.

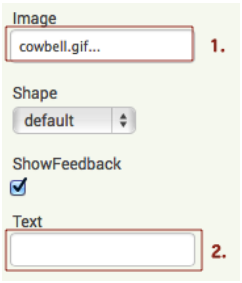


Set properties - Button

3. The properties for the button are shown in the right-most column. Set the image for the button to "zoltar.png" by clicking on the *None...* (1) under Image, then click on *Upload File ...* (2), then click on *Choose File* (3) which will bring up a file picker. Select the "zoltar.png" file that you just downloaded and click on *OK* (4). The button should then show the zoltar picture. If it doesn't show the picture check that you uploaded "zoltar.png". If you made a mistake just upload "zoltar.png" and set the image to that file.

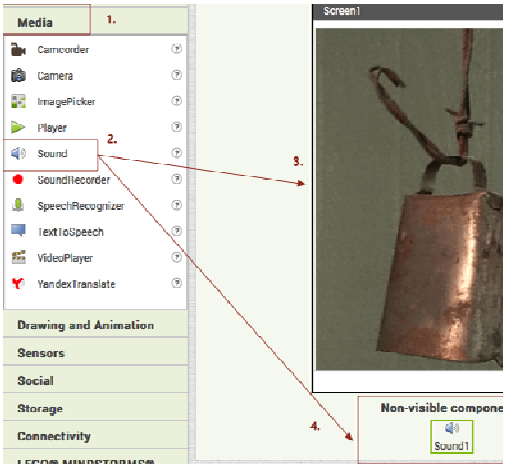


4. Check that the image property is set to "zoltar.png" (1). Remove the default text for the button by selecting it and hitting the *delete* key on the keyboard (2).

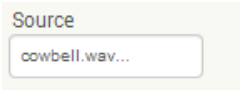


Create Component - Sound

Click on the *Media* category (1) in the palete (leftmost column) to view the media components and drag out a Sound (2) to the viewer (3). It will show up as a non-visible component under the display of the phone screen (4).



Set Properties - Sound
In the properties column set the source to "cowbell.wav" (upload the file), just like you did with "cowbell.gif".



Program and test

In this step we will program the app in the blocks editor to play the cowbell.wav sound file when the button is pushed and test the app. Doing something when an event occurs, like when a button is clicked, is called **handling an event**.

Open the Blocks Editor

Open the blocks editor by clicking on the *Blocks* button in the upper right corner of the Designer.

Handle Event - button click

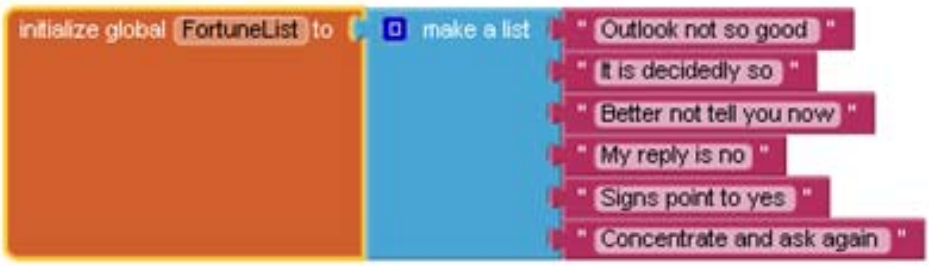
Click on *Button1* (1) under *Screen1* in the left Blocks panel to see the blocks for the button you created. Drag out a **when Button1.Click** block (2) and drop it in the viewer area on the right (3). This block will run each time Button1 is clicked.

Specify Action – select random item from list

Add Action – Call to TextToSpeech

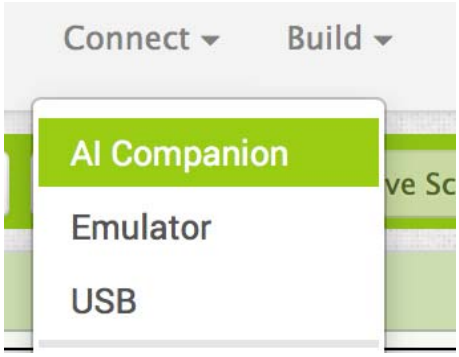
Specify Action – select random item from list

Click on *Sound1* (1) in the left Blocks panel and drag out a **call Sound1.Play** block (2) and put it inside the **do** area of the **when Button1.Click** block (3) as shown below. This will play Sound1, the cowbell.wav sound, each time you click on the button.



Test with a WiFi Connected Device

3. Click *Connect* > *AI Companion* on the upper task bar. A QR code will appear in the browser.



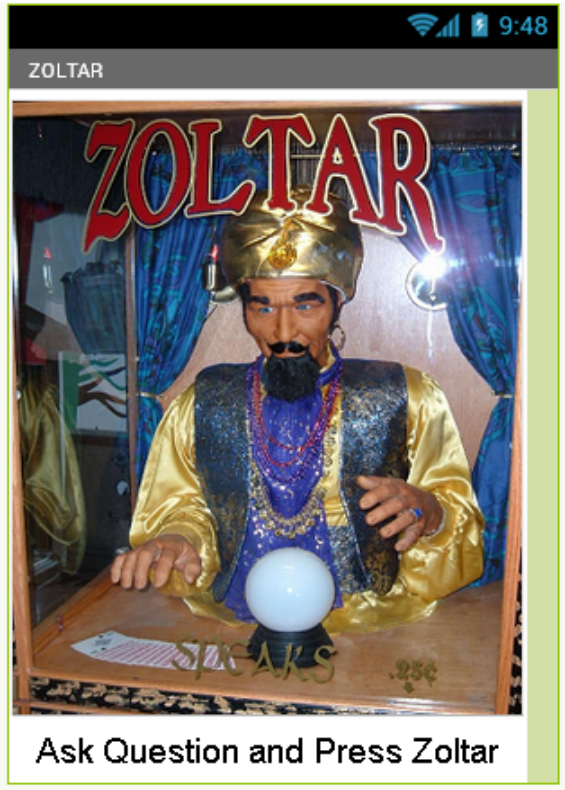
4. Open the MIT AI2 Companion app on the phone and scan the QR code or type in the code. In a few seconds, the app will appear on the device.

Extension

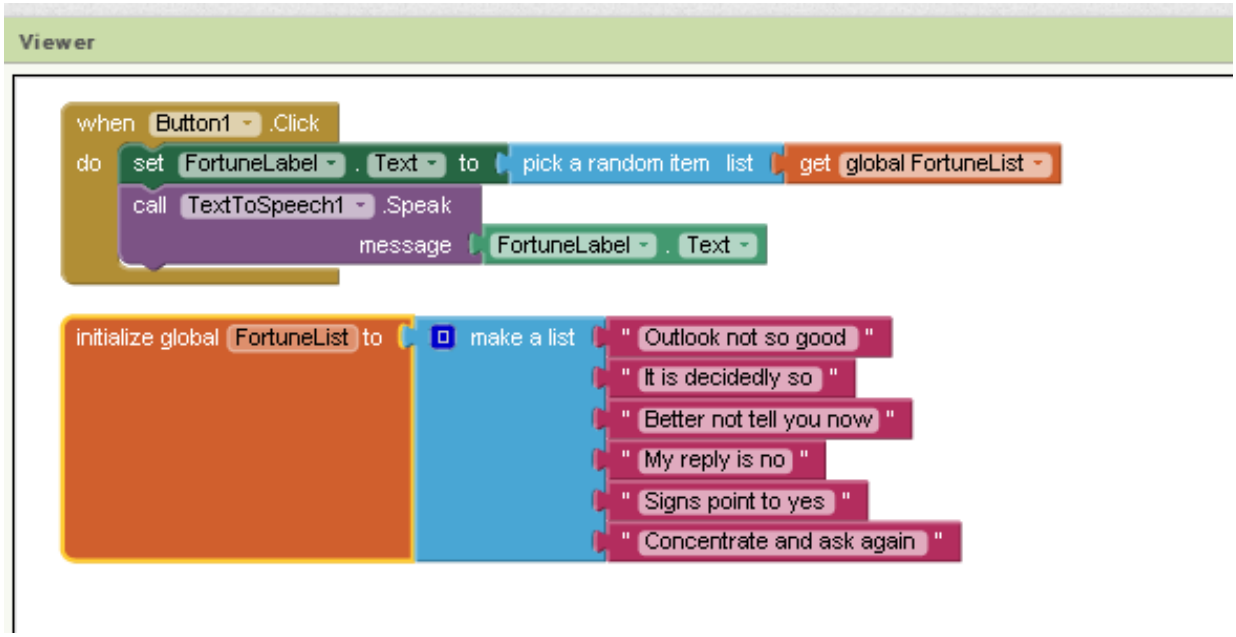
Any students who complete the project should add additional responses to the list that Zoltar uses to provide responses to the users questions.

The 20 answers inside a Magic 8 Ball are:

- It is certain
- It is decidedly so
- Without a doubt
- Yes definitely
- You may rely on it
- As I see it, yes
- Most likely
- Outlook good
- Yes
- My sources say no
- Outlook not so good
- Very doubtful
- Signs point to yes
- Reply hazy try again
- Ask again later
- Better not tell you now
- Cannot predict now
- Concentrate and ask again
- Don't count on it
- My reply is no



Below is a screen capture of the same page as viewed with Internet Explorer:



- add a label and set the label text.
- rename a component.
- create a variable.
- create a list.
- select a random item from a list and display it.

Assessment

Formative Assessment of this lesson is accomplished by the direct observation of students during the activity. At the conclusion of the activity, all students should have a completed, basic web page and be capable of opening it in the web browser.

Summative Assessment of this lesson will consist of several questions that will be included in a unit exam at a future date.