

Hello It's Me

Table of Contents

1. Synopsis	3
2. Learning Objectives	3
3. Mapping with the Computational Thinking Framework	4
4. Mapping with the CSTA Standards	6
5. Learning Prerequisites	7
6. Lesson Plan (3 x 45 minutes)	8
7. Assessment	11
8. Screen Design and Code	14
Appendix 1 Hello It's Me Teacher's Guide Lesson 1	16
Appendix 2 Hello It's Me Teacher's Guide Lesson 2	22
Appendix 3 Hello It's Me Teacher's Guide Lesson 3	24

1. Synopsis

This unit introduces students to App Inventor. In the first lesson, students will be introduced to App Inventor and learn to create mobile apps. Students will learn the main parts of the App Inventor interface - the “Designer” with its components, and the Blocks Editor with its “Drawers” - by changing some components’ properties and code using the TwoButtonGame app. In the second and third lessons, students will build a simple “app” (mobile application), HelloItsMe, with the teacher. Students will learn how to run and test their apps on Android devices using the MIT AI2 Companion app.

2. Learning Objectives

After completing this unit, students will be able to:

1. Log in and out; and create, save, and retrieve projects in the MIT App Inventor programming environment.
2. Demonstrate an understanding of the Designer and what its different sections represent.
3. Code an app using the App Inventor Blocks Editor.
4. Test and debug using the MIT AI2 Companion app (or emulator).
5. Identify and use the correct components and blocks for an audio and visual app.
6. Develop computational identity by creating a mobile app.



3. Mapping with the Computational Thinking Framework

The following tables show the alignment of this unit with the intended learning outcomes of the computational thinking framework. The entries indicate the expected relevance of this unit to each outcome:

- ✓✓✓ : High relevance
- ✓✓ : Some relevance
- ✓ : Low relevance

Computational Thinking Concepts

Unit 1: Hello It's Me		
1. Sequences	✓✓	Students learn that command blocks fit into event blocks.
2. Events	✓✓✓	Button click events are used in both apps of this unit.
3. Repetition		
4. Conditionals		
5. Parallelism		
6. Naming	✓	Left and right buttons are named properly in the TwoButtonGame app.
7. Operators		
8. Manipulation of data and elementary data structures		

Computational Thinking Practices

Unit 1: Hello It's Me		
1. Reusing and remixing		
2. Being incremental and iterative		
3. Abstracting and modularizing		
4. Testing and debugging	✓✓✓	Students need to test out the changes to the app and fix any mistakes.
5. Algorithmic thinking		

Computational Thinking Perspectives

Unit 1: Hello It's Me		
1. Expressing	✓✓✓	Students create an app with their image and voice to express themselves.
2. Connecting	✓✓	Students connect with their friends and family by sharing this app about themselves.
3. Questioning	✓✓	Students learn a new technology with MIT App Inventor.
4. Computational identity	✓✓	Students have the opportunity to build their first mobile app.
5. Digital empowerment	✓✓	Students feel empowered by building a mobile app about themselves.

4. Mapping with the CSTA Standards

This table shows our alignment with the intended learning outcomes of the CSTA CS Standards.

The entries in the tables indicate the expected relevance of the unit to each outcome:

2-CS-02	Design projects that combine hardware and software components to collect and exchange data. [C] CS: Hardware & Software [P] Creating (5.1)	Coding app combines software and hardware, even involving sound.
2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. [C] AP: Modularity [P] Computational Problems (3.2)	Students gather pictures and sounds, design, then code the app, a good example of decomposition.
2-AP-18	Systematically test and refine programs using a range of test cases. [C] AP: Program Development [P] Testing (6.1)	Students test using the MIT AI2 Companion.

5. Learning Prerequisites

None.



6. Lesson Plan (3 x 45 minutes)

This unit consists of three 45 minutes lessons. Expanded teacher guides are part of the appendix.

Lesson 1

Time	Activity
5 min	Introduction to App Inventor Explain that App Inventor is a block-based programming environment that allows us to make mobile apps for tablets and smartphones. Show the introductory video: https://youtu.be/sNjGAiOrX-o
15 min	Demonstration of App Inventor Programming Environment <ol style="list-style-type: none">1. Demonstrate the login and logout of the App Inventor service (https://ai2.appinventor.mit.edu) or through http://appinventor.mit.edu and clicking on Create Apps.2. Explain to students the Designer and the function of the Palette, Viewer and Properties windows.3. Explain to students the Blocks Editor and the difference between Built-In Blocks and component Blocks.
10 min	Using MIT AI2 Companion for Testing <ol style="list-style-type: none">1. Ask students to import the TwoButtonGame.aia project.2. Explain to students the auto-saving function, and the two main file formats (aia, apk) of App Inventor.3. Ask students to test their app using the MIT AI2 Companion app. Remind students to test and debug their apps regularly and frequently.4. Demonstrate how to use MIT AI2 Companion for immediate testing and debugging.
15 min	Modifying the TwoButtonGame app <ol style="list-style-type: none">1. Ask students to modify the app following Two Button Game Student Guide. Students may add their own modifications to the app also.2. Finally, remind students to log out from App Inventor at the end of each lesson.

Lesson 2

Time	Activity
10 min	Review and Introduction to Lesson <ol style="list-style-type: none">1. Review the parts of the App Inventor development environment. Ask students to identify the Designer, the Blocks Editor, and some of the panels and menus.2. Explain to students that they will make their own app in this lesson, using information about themselves in the app. Demonstrate the sample HelloItsMe app.
20 min	Preparation <p>Direct students to take photos of themselves and record their voices in preparation for making the HelloItsMe app. Make sure these files are saved. (Teachers can provide the image and sound files if time is a concern).</p>
15 min	Coding Activity <ol style="list-style-type: none">1. Work together on HelloItsMe app. Teacher models and students follow along. Students may also follow the Hello It's Me Student Guide or follow the video guide.2. Students test their app using MIT AI2 Companion.

Lesson 3

Time	Activity
15 min	Review and Introduction to Lesson <ol style="list-style-type: none">1. Review the parts of the App Inventor development environment again for the students.2. Ask students to answer the multiple choice questions found in the Assessment section of guide.
15 min	App Enhancements <ol style="list-style-type: none">1. Ask students to brainstorm enhancements to their app. Ideas could include changing the user interface (more colors and fonts), or adding more buttons for other people.2. Students implement their ideas into their apps.
15 min	Sharing <ol style="list-style-type: none">1. Instruct students in creating an apk file that can be installed on the tablet permanently.2. Students share their apps, either in small groups or with the entire class.

7. Assessment

Multiple-choice questions

Multiple-choice questions assess key concepts of the activity as shown below:

Questions on App Inventor Interface

1. In which section of App Inventor can you add components to your app's screen?

- A. Designer
- B. Properties
- C. Blocks Editor
- D. MIT AI2 Companion app

(Answer: A)

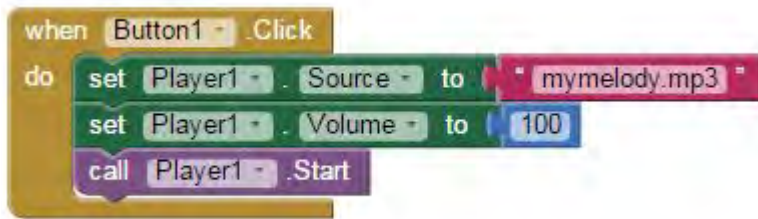
2. A button is an example of a _____.

- A. property
- B. component
- C. picture
- D. block

(Answer: B)

Question on Coding with App Inventor

3. What would happen when you press Button1 according to the following blocks?



- A. The sound file “mymelody.mp3” starts to play.
- B. The sound file “mymelody.mp3” starts to play and the volume becomes louder.
- C. The sound file “mymelody.mp3” stops and the volume becomes louder.
- D. The sound file “mymelody.mp3” stops and the volume becomes softer.

(Answer: B)

Question on App Inventor Testing and Debugging

4. In which section of App Inventor can you test your app?

- A. Designer
- B. Viewer
- C. Blocks Editor
- D. MIT AI2 Companion app

(Answer: D)

Learning attitudes

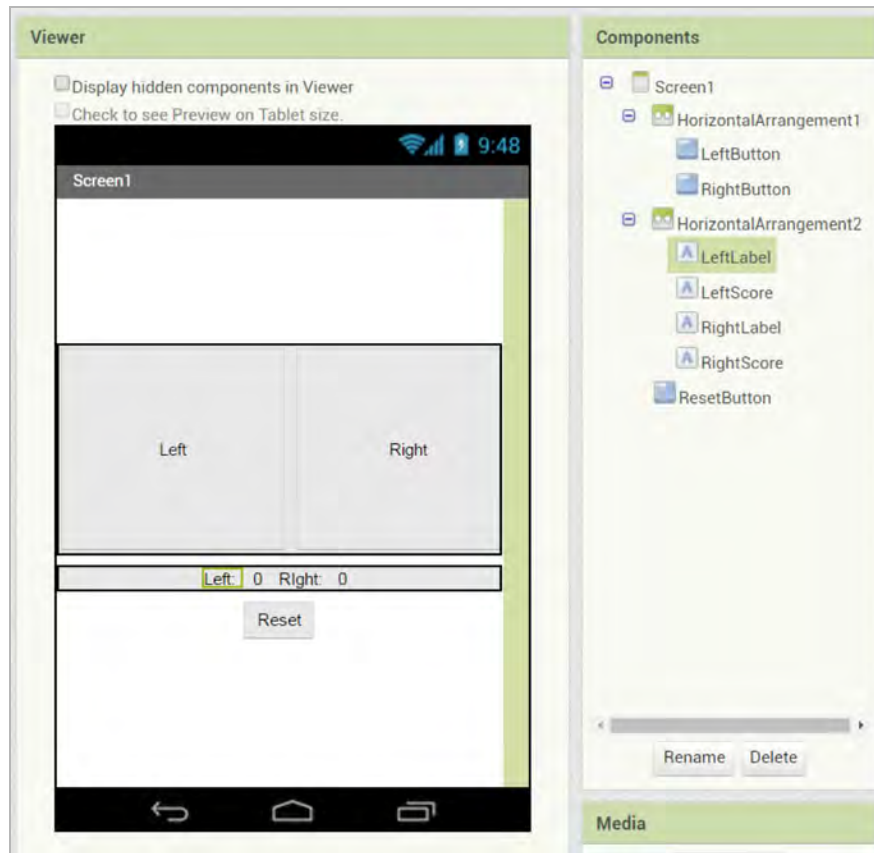
In order to evaluate students' attitude, perception, and understanding towards coding, students are required to finish a 5-point scale survey below by putting a “✓” in the appropriate box.

After completion of this unit, I think...	Disagree	Somewhat disagree	Neutral	Somewhat agree	Agree
Learning how to make apps makes me want to learn more about coding.					
I feel more connected to the technology around me when I make apps.					
I am excited to share this app with friends and family.					

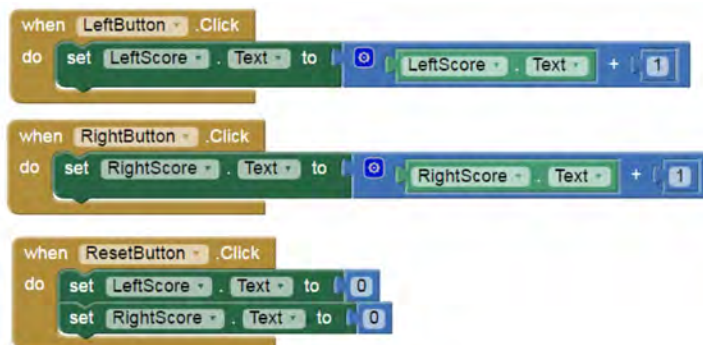
8. Screen Design and Code

TwoButton Game

Designer

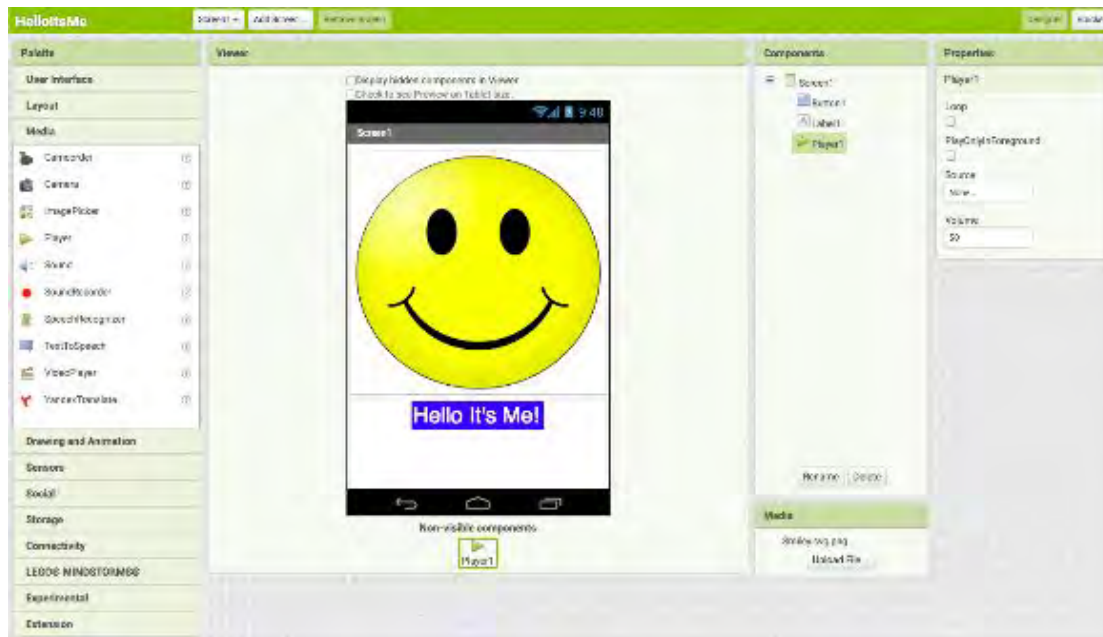


Blocks Editor



Hello It's Me

Designer *(this is just a sample project ... students will use their own pictures and sound recordings).*



Blocks Editor



Appendix 1

Hello It's Me Teacher's Guide

Lesson 1

Learning Objectives

At the end of this lesson, students should be able to:

1. Use the App Inventor environment to open, update and test an app.
2. Identify the Designer and Blocks Editor parts of the App Inventor environment.
3. Change code blocks and component properties to affect changes in an app's looks and behavior.
4. Test and debug a project.

Lesson Outline

Introduction to App Inventor (5 minutes)

This is a general introduction to MIT App Inventor 2, a blocks-based programming environment that allows users to create their own mobile apps

1. Explain that App Inventor is a block-based programming environment that allows us to make mobile apps for tablets and smartphones. Show the introductory video: <https://youtu.be/sNjGAiOrX-o>



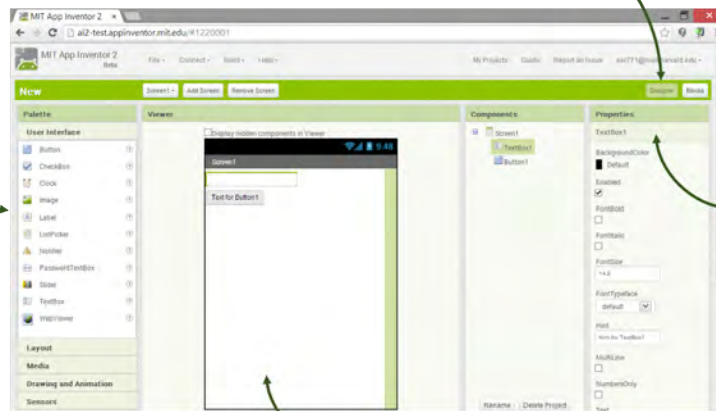
Demonstration of App Inventor Programming Environment (15 minutes)

1. Demonstrate how to login and logout of the App Inventor website by playing the video (<http://goo.gl/pNAxkr>).
 - a. Open browser window and go to the MIT App Inventor 2 website (<http://appinventor.mit.edu>).
 - b. Click on the “Create apps!” button in the top right of the window to go to the MIT App Inventor 2 development website. You can also go directly to <http://ai2.appinventor.mit.edu>.
 - c. Follow the prompts to login (or sign up) with their Google Account username and password to enter the website.
2. Identify the following four main parts of the **Designer**:
 - a. **Palette**, where the user can choose from the different components. Open up some of the Palette Drawers, like “Layout”, “Media”, “Social” to show students there are many features that can be added to an app.
 - b. **Viewer**, where the user can drag components to add them to the app. Note that adding a component will add it vertically, relative to other components already in the Viewer.
 - c. **Components**, where the user can see all components currently in the app. Users can rename their components in the list.
 - d. **Properties**, where the user can change specific properties of a given component by first selecting the component in the Viewer or the Component List, and then changing the property. Characteristics such as colour and font size can be changed here.



Palette: Find your components and drag them to the Viewer to add them to your app.

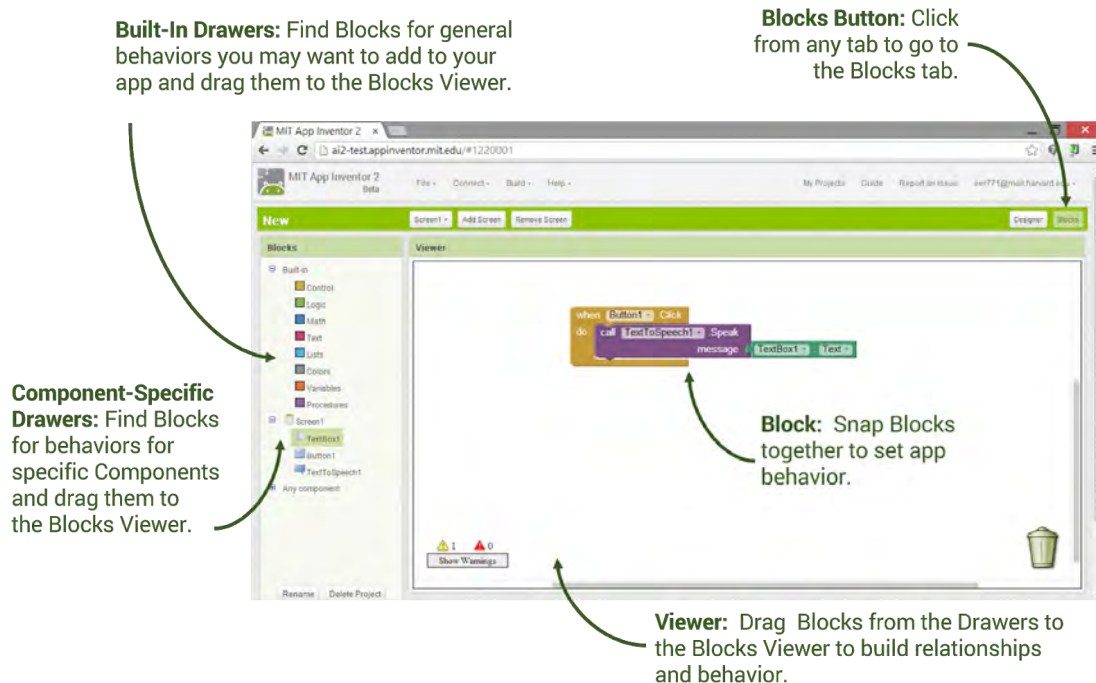
Designer Button: Click from any tab to go to the Designer tab.



Properties: Select a Component in the Components List to change its properties (color, size, behavior) here.

Viewer: Drag components from the Palette to the Viewer to see what your app will look like.

3. Click on the “Blocks” button in the upper right of the screen to switch to the **Blocks Editor**. Identify the two main parts of the Blocks Editor:
 - a. **Drawers**
 - i. built-in
 - ii. components
 - b. **Blocks Viewer**

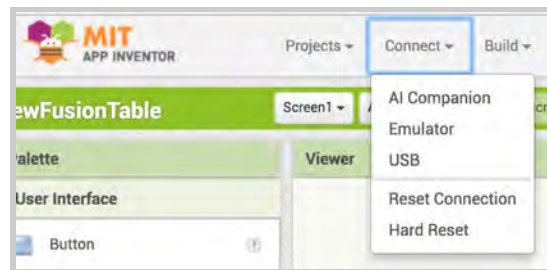


Using MIT AI2 Companion for Testing (10 minutes)

1. Ask students to import the TwoButtonGame.aia project. Demonstrate how to import a project into your project list.
 - 1.1. Click the “Project” button to pull down the menu, and select “Import project (.aia) from my computer” The “Import Project...” Dialogue Box will appear.
 - 1.2. Click the “Choose file” button to open the “Open” Dialogue Box.
 - 1.3. Go to the source folder, select the specific .aia file, and click the “open” button.
 - 1.4. Click the “OK” button in the “Import Project...” Dialogue Box.
2. Explain to students the auto-saving function, and the two main file formats (aia, apk) of App Inventor.
3. Ask students to test their app using the MIT AI2 Companion app. Remind students that it is a good habit and important step to test and debug their apps regularly and frequently.

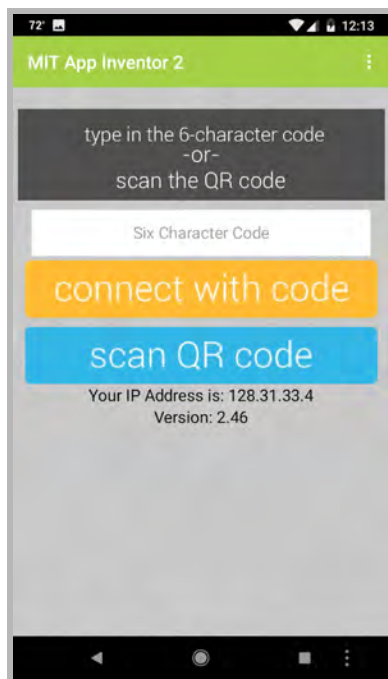
4. Demonstrate how to use MIT AI2 Companion for immediate testing and debugging.

4.1. Click on Connect menu and then select AI2 Companion.



4.2. MIT AI2 Companion should already be installed on student devices.

4.3. Open the MIT AI2 Companion on the device and click the blue button to scan the QR code, and scan the QR code that appears on the computer.



Modifying the TwoButtonGame app (15 minutes)

1. Ask students to modify the app following the Two Button Game Student Guide. Students may add their own modifications to the app also.
2. Finally, remind students to logout from App Inventor at the end of each lesson.

Appendix 2

Hello It's Me Teacher's Guide

Lesson 2

Learning Objectives

At the end of this lesson, students should be able to:

1. Create a HelloItsMe app that includes an image, sound, and event handling.
2. Test and debug their apps using the MIT AI2 Companion app and live testing.

Lesson Outline

Review and Introduction to Lesson (10 minutes)

1. Review the App Inventor environment to reacquaint students with the different parts of the interface by opening App Inventor on teacher's screen for students to see. Encourage class discussion with the following prompts:
 - (1) Where is the layout of components made and added to? (Designer)
 - (2) In which window do you write the blocks for your app? (Blocks Editor)
 - (3) What did you change in the TwoButtonGame app? (Shape and colours)
2. Explain to students that they will make their own app in this lesson, using information about themselves in the app. Demonstrate the sample app and tell students that they are going to create their first app by adding their photo and voice.



Preparation to make the App (20 minutes)

Direct students to take a photo of themselves and record their voice in preparation for making the HelloItsMe app. Make sure these files are saved (Teachers can provide the image and sound files if time is a concern).

1. On Macs, students can use the QuickTime app to record their voices. On Windows, students can use Voice Recorder.
2. On Macs, students can use Photo Booth to take their picture. On Windows, students can use the Camera icon to take a picture.

Another option is to take a picture from the Internet of a famous person, and find a recording of their voice (MLK and “I have a dream”) and have students use that media for the app. The idea with “Hello It’s Me” is to personalize the app.

Coding Activity (15 minutes)

Work together on the HelloItsMe app. Teacher models and students follow along using the Hello It’s Me Student Guide, or follow the [Youtube video](#).

1. Start a New project: “HelloItsMe”
2. Add a Button, a Label, and a Player to the viewer.
3. Upload the image and sound files just prepared (or are provided by teachers).
4. Change the properties of the Button and Label.
5. Add the **Button1.Click** event in Blocks Editor.
6. Call the Player to play the sound file.
7. Students test their app, build the apk file, and install it on the tablet to show to their classmates.



Appendix 3

Hello It's Me Teacher's Guide

Lesson 3

Learning Objectives

At the end of this lesson, students should be able to:

1. Add an enhancement to their app.
2. Test and debug their apps using the MIT AI2 Companion app and live testing.
3. Share their work with their peers.

Lesson Outline

Review and Introduction to Lesson (15 minutes)

1. Review again for students the parts of the App Inventor development environment.
2. Ask students to answer the multiple choice questions found in the Assessment section of this unit outline.

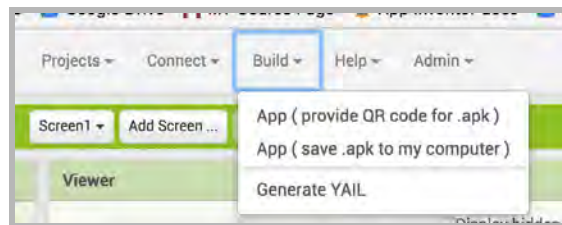


App Enhancements (15 minutes)

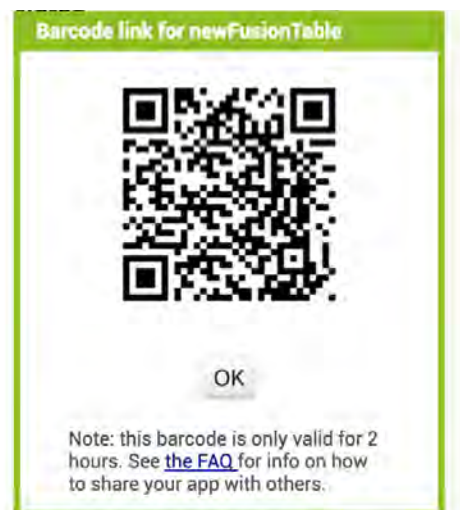
1. Ask students to brainstorm enhancements to their apps. Ideas could include changing the user interface (more colors and fonts), or adding more buttons for other people.
2. Students implement their ideas in their apps

Sharing (15 minutes)

1. Instruct students how to create an apk file that can be installed on the tablet permanently.
 - a. Under the Build menu, select “App (provide QR code for apk)”



- b. When QR code appears, scan it using the tablet. It is a good idea to have student devices preloaded with a QR Code Scanner app, but if not, MIT AI2 Companion can be used.



- c. When instructed, students will install their apps on the tablet. The apps will remain permanently on the devices (unless uninstalled manually).
2. Students share their apps. This could be with a group, where they pass around their tablets, or students could share with the entire class.