



Jiangxi University of Science and Technology
School of information engineering

Task book

on

Mobile Application Design

Ata Jahangir Moshayedi



DR. Ata Jahangir Moshayedi
Jiangxi University of Science and Technology
School of information engineering

EMAIL: ajm@jxust.edu.cn

Autumn _2020

mswordcoverpages.com

Example 1:

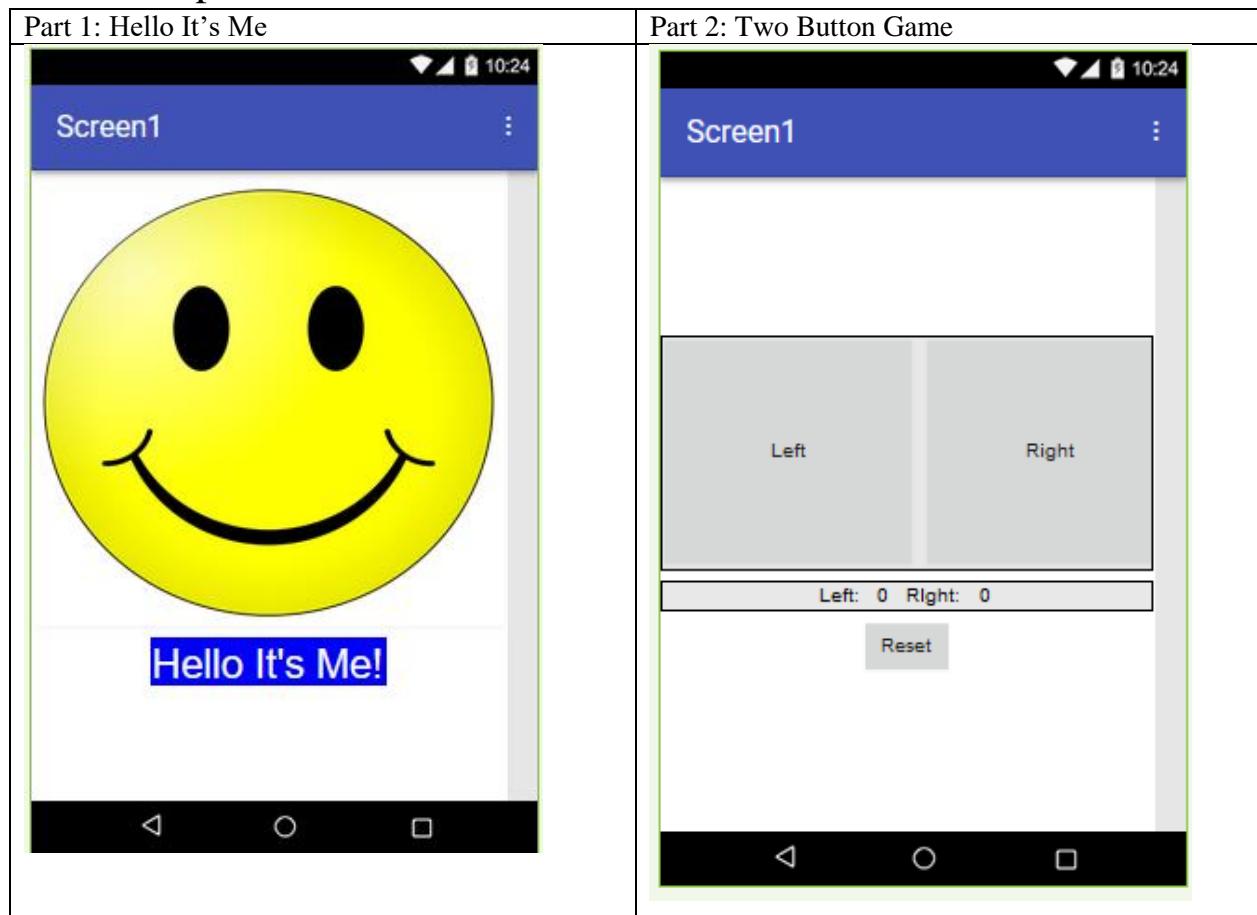
Task 1_1:Hello It's Me(App that speaks your voice)

Task 1_2:Two Button Game Student Guide

Objectives

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.

Final APP picture



Unit 1: Hello It's Me

HELLO, IT'S ME!

START HERE



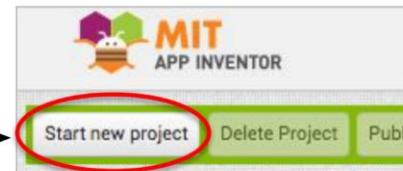
In this activity, you will create a project with App Inventor and use your own photo and voice to tell people who you are.



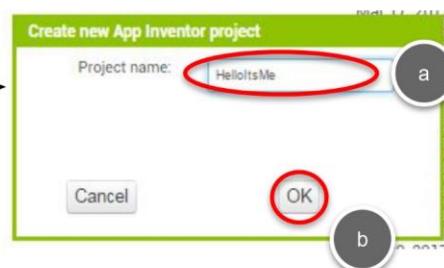
- 1 Go to the MIT App Inventor website (<http://ai2.appinventor.mit.edu>).
- 2 Log in with your Gmail address and password, or use the one supplied by your teacher.
- 3 Read the announcements, then click "Continue." - - - - -



- 4 Click the **Start new project** button. - - - - -



- 5 Name your project "HelloItsMe", then click **OK**. - - - - -



Copyright MIT 2019, Review Copy, Not for Distribution

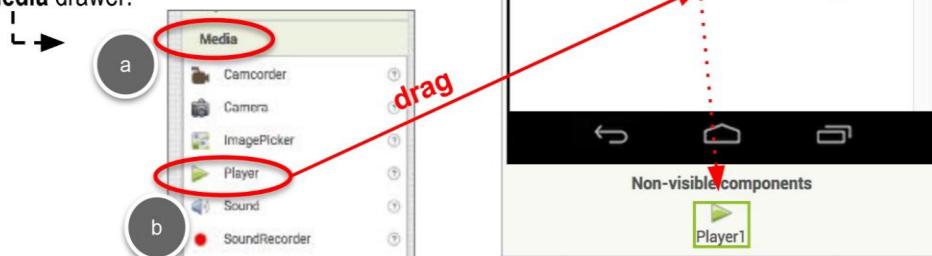
1

LET'S CONTINUE

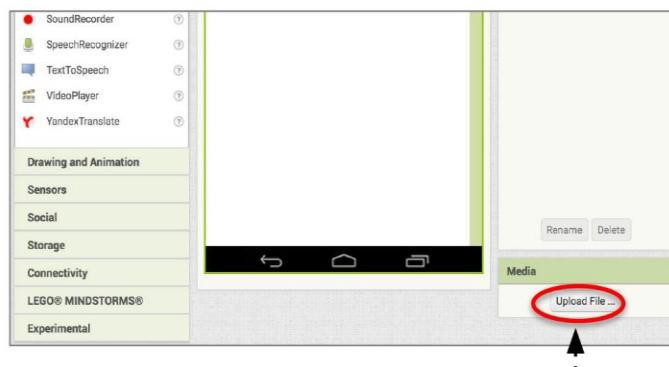
- 6** Add a **Button** and a **Label** component to your app by dragging them from the **User Interface** Drawer in the Palette to the Viewer.



- 7** Add a **Player** component, a non-visible component, to your app by dragging from the **Media** drawer.

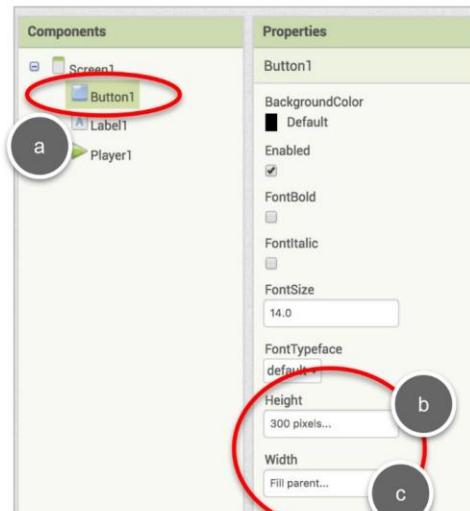


- 8** You should have an image of yourself and a recording of your voice, saying hello and something about you. Upload your picture and your voice recording as media for the app.

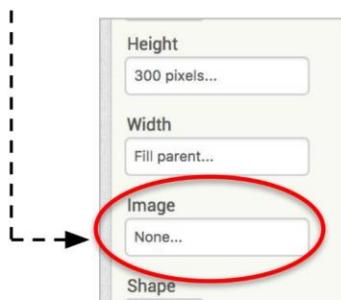


LET'S CONTINUE!

- 9 Click on **Button1** in the Components window and change the button properties as follows:
Height: 300 pixels
Width: Fill parent

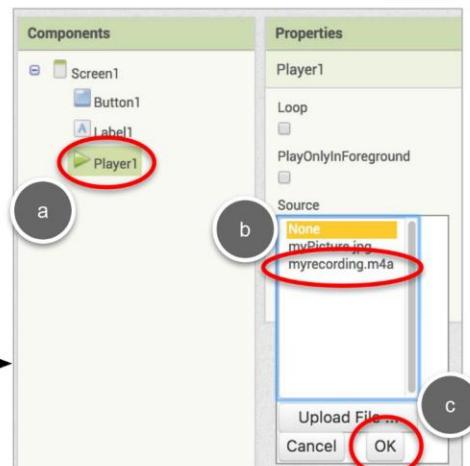


- 10 Find **Image** in the Properties window and click on the word "None".



- 11 Click on your photo that you uploaded earlier, step 8, then click **OK** to use your image.

- 12 Choose **Player1** from the Components list, and set its **Source** property to the voice recording file that you uploaded earlier, then click **OK** to use your sound file.



Unit 1: Hello It's Me

BLOCKS EDITOR

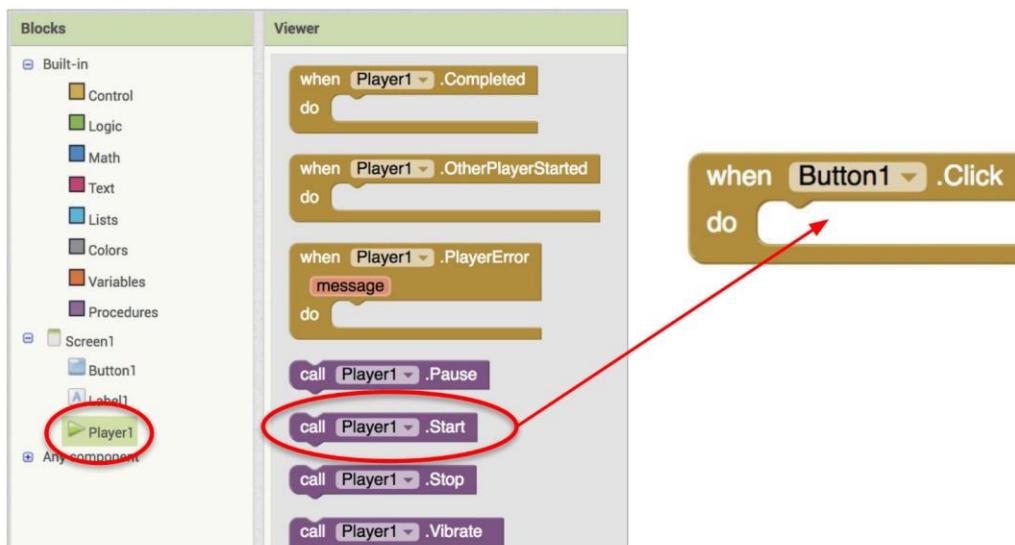
- 13 Click the **Blocks** button and go to the → Blocks Editor.



- 14 Click on **Button1** in the Blocks Component Drawer, then drag out the **Button1.Click** block.



- 15 Then click on **Player1** in the Blocks window on the left, drag out the **Player1.Start** block and snap it into the **Button1.Click** block.



TESTING!

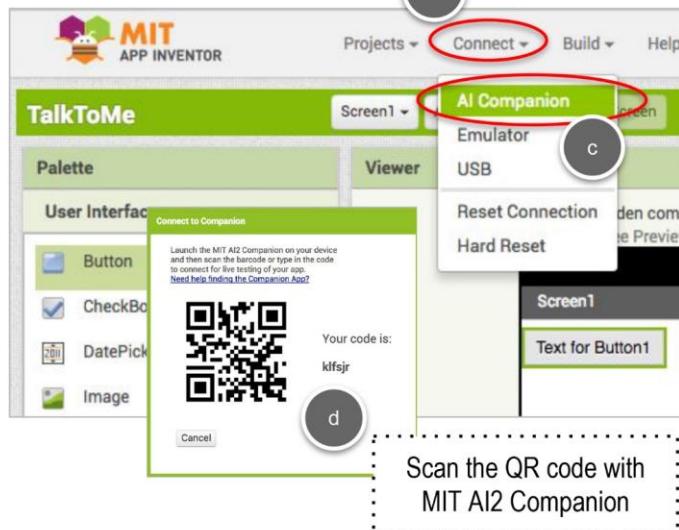
16

Now test the app on your tablet!

- a Start MIT AI2 Companion on your tablet



b



c

d

Scan the QR code with
MIT AI2 Companion

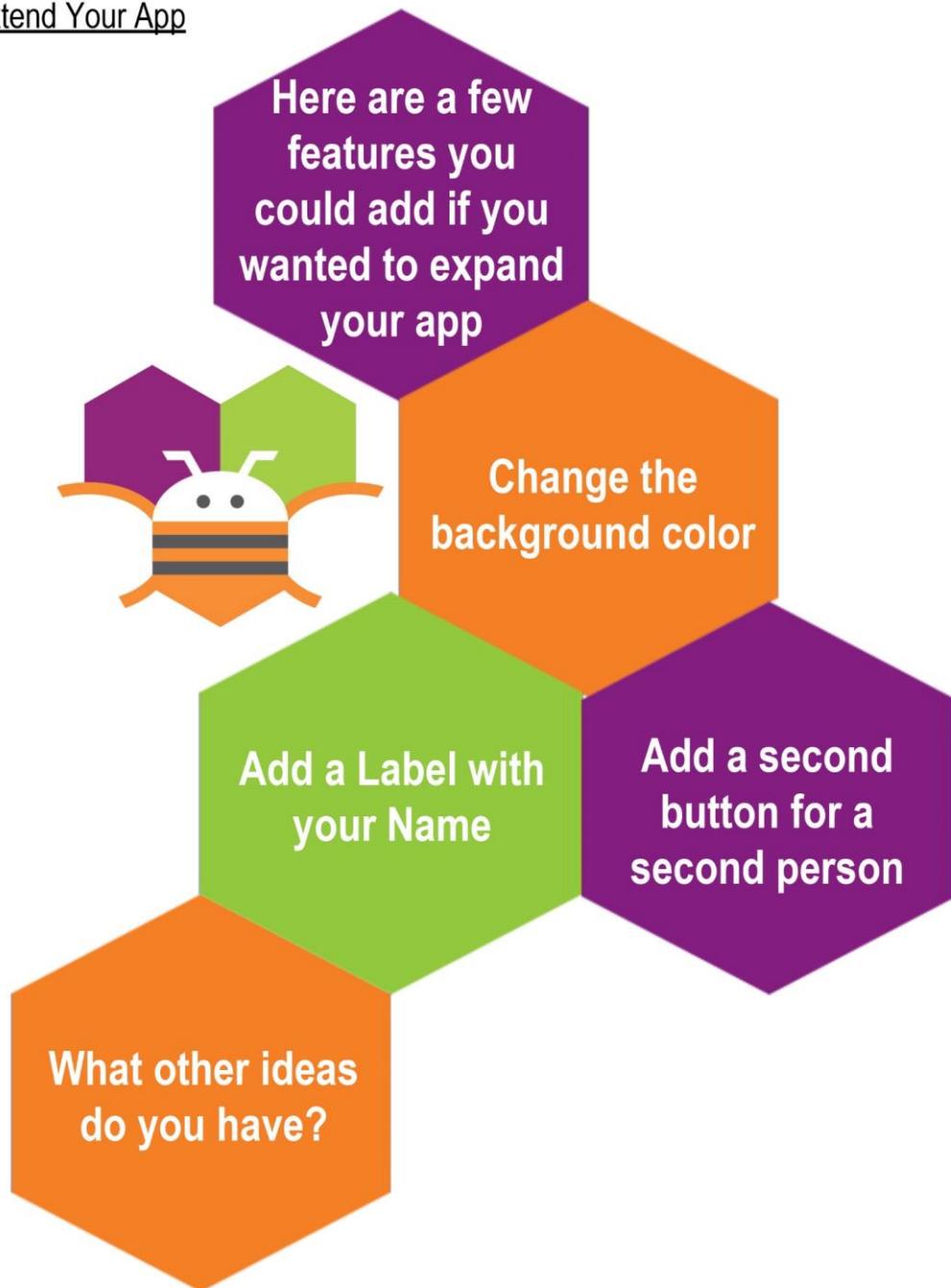
17

Test your app. If you press the button with your picture, can you hear your voice?



Congratulations,
you've made your
first app!

Extend Your App



Copyright MIT 2019, Review Copy, Not for Distribution

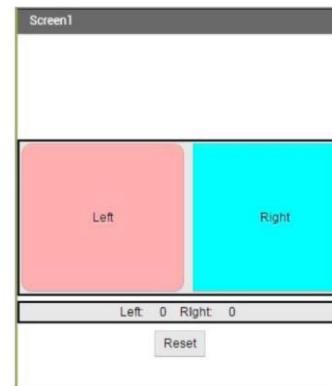
Unit 1: Two Button Game

TWO BUTTON GAME

START HERE



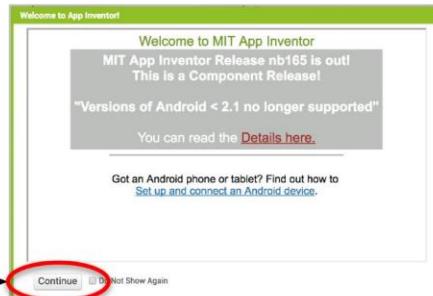
In this activity you will update some features of this simple button clicking game.



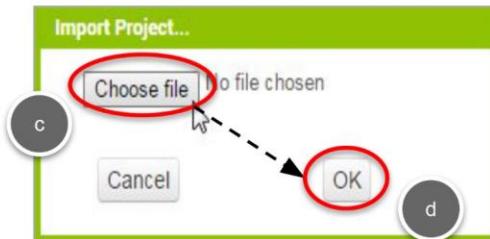
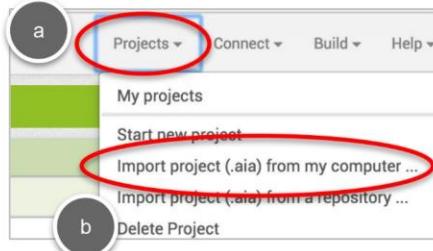
- 1 Go to the MIT App Inventor website (<http://ai2.appinventor.mit.edu>).

- 2 Log in with your Gmail address and password, or use the one supplied by your teacher.

- 3 Read the announcements, then click "Continue." - - - - -



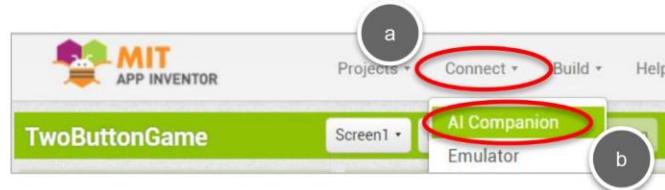
- 4 Import the TwoButtonGame.aia file provided by your teacher.



Unit 1: Two Button Game

LET'S RUN THE APP!

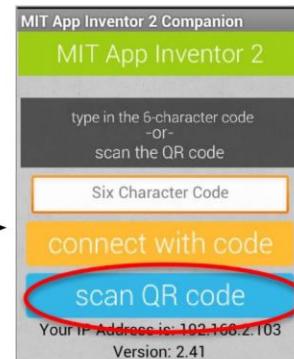
- 5 Under the **Connect** menu, choose **AI Companion**.



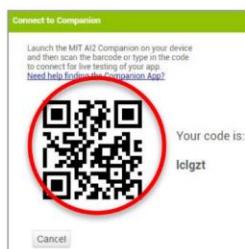
- 6 On your tablet or smartphone, run the **MIT AI2 Companion** app.



- 7 Click the blue **scan QR Code** button.



- 8 Scan the QR code on the computer screen with your device.



- 9 Make sure you are in the Designer window by clicking on the **Designer** button at the top right of the screen.

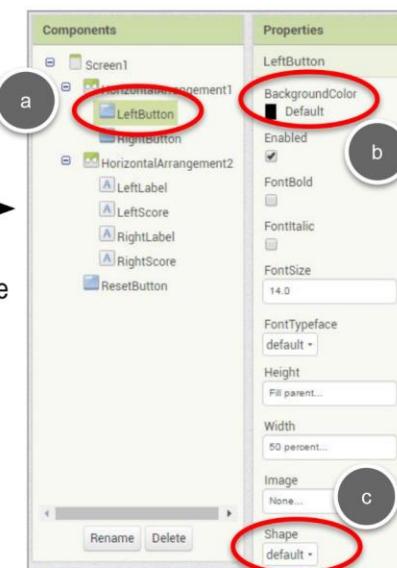


- 10 Click on **LeftButton** in the Components panel and change the properties as follows:

BackgroundColor: any color you like

Shape: rounded

- 11 Change the **RightButton** in the same way.



LET'S CONTINUE

12

Make any other changes that you want.

For example, how could you make the **ResetButton** larger?

13

Look at the app on your smartphone or tablet. Notice how the app changes when you make a change in App Inventor.

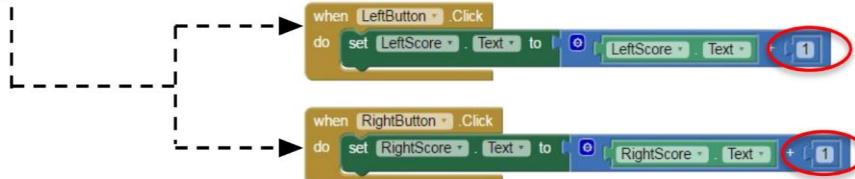
14

Click the **Blocks** button and go to the Blocks Editor.



15

In the **LeftButton.Click** and **RightButton.Click** blocks, change the number to any number other than 1.



16

Play the game again on your smartphone or tablet to see the changes!

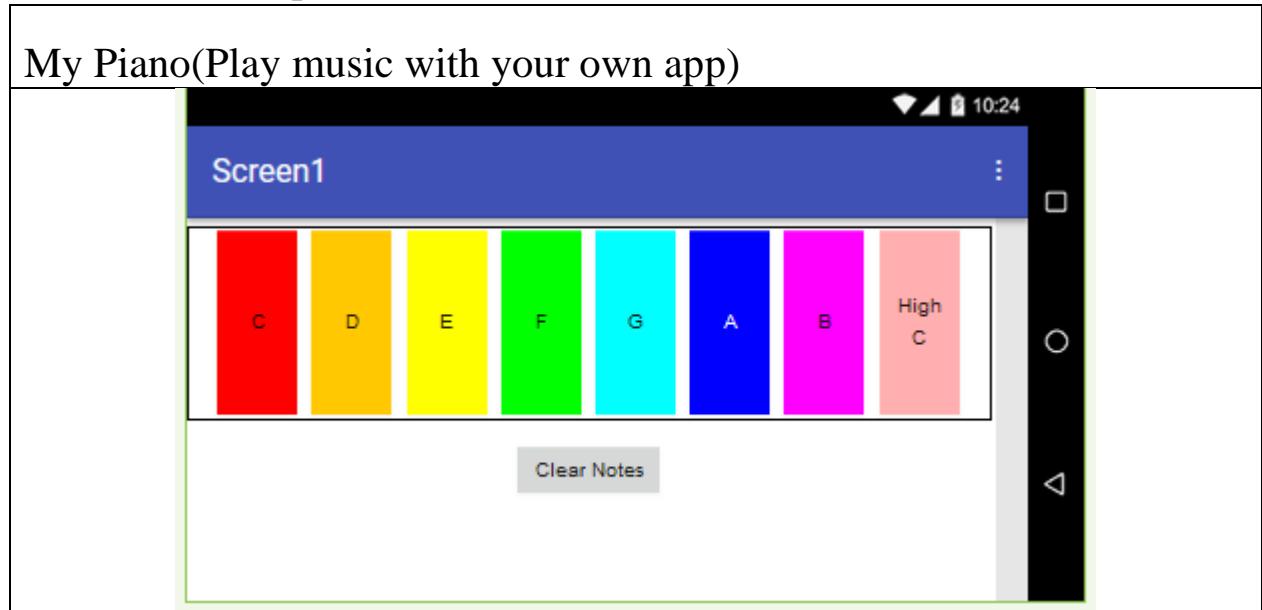
Example 2:

Task 1_1: My Piano(Play music with your own app)

Objectives

1. Code a piano app using the App Inventor Blocks Editor.
2. Apply the computational thinking practice of being incremental and iterative, developing and testing an app in stages.
3. Reuse code in an app.
4. Demonstrate abstraction by creating and implementing a procedure in App Inventor;
5. Test and debug an app using the MIT AI2 Companion.
6. Increase your positive perception of programming by creating a useful and fun app.

Final APP picture



Unit 2 MyPiano: Challenge

MY PIANO: CHALLENGE

ADD SHARP NOTE BUTTONS

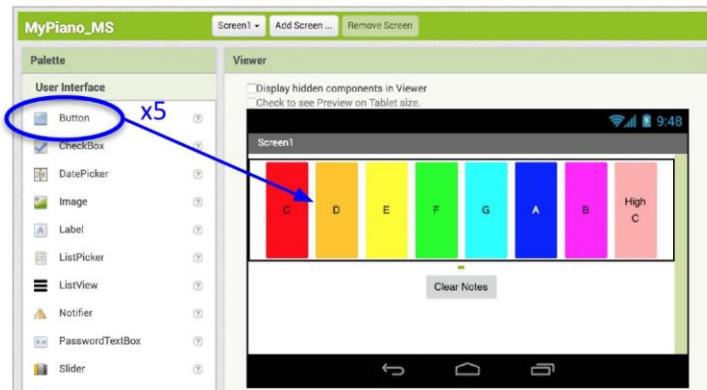


Thanks for trying some of the Challenges. Try one, two, or all of the suggestions, or come up with your own!

- 1 Switch to the Designer.



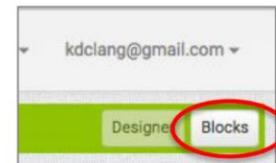
- 2 Add 5 more Buttons for the 5 Sharp Notes (C, D, F, G, and A). Remember to name them CSharpButton, DSharpButton, etc) so the sound file works properly.



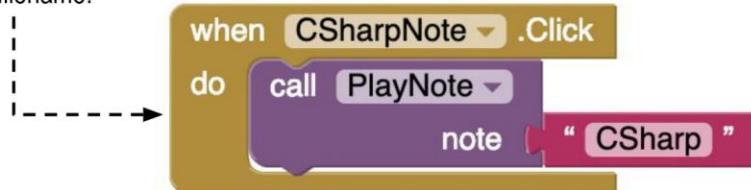
- 3 Since you added 5 new Buttons, you have a total of 13. If you want all the *Width's* to add up to 100%, what percentage should each **Button Width** be? You can round down to the nearest whole number.

ADD SHARP NOTES (continued)

4 Switch to the Blocks Editor. ----->



5 Add Button.Click event blocks for all your new Buttons. Remember to set the note parameter to match the Button name, since that matches the sound filename.



MAKE THE BUTTONS LOOK LIKE A PIANO

- 1 Switch to the Designer.



- 2 Changing the key color is really easy! Change all the regular buttons to a white *BackgroundColor*, and all the Sharp buttons to a black *BackgroundColor*. You will have to change the *TextColor* for the black buttons to white so they appear on the black background.



- 3 To make the white buttons show up, change **HorizontalArrangement1**'s *BackgroundColor* to a light grey (or some other color).

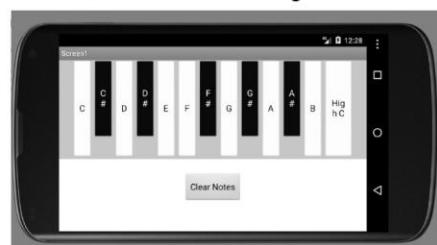


- 4 And change the *AlignVertical* property to "Top: 1" for **HorizontalArrangement1**.

- 5 You could make the Sharp Note **Buttons** not quite as tall as the regular notes. 40% is a good option, but you can try different values to see what you like.



Should look something like this!

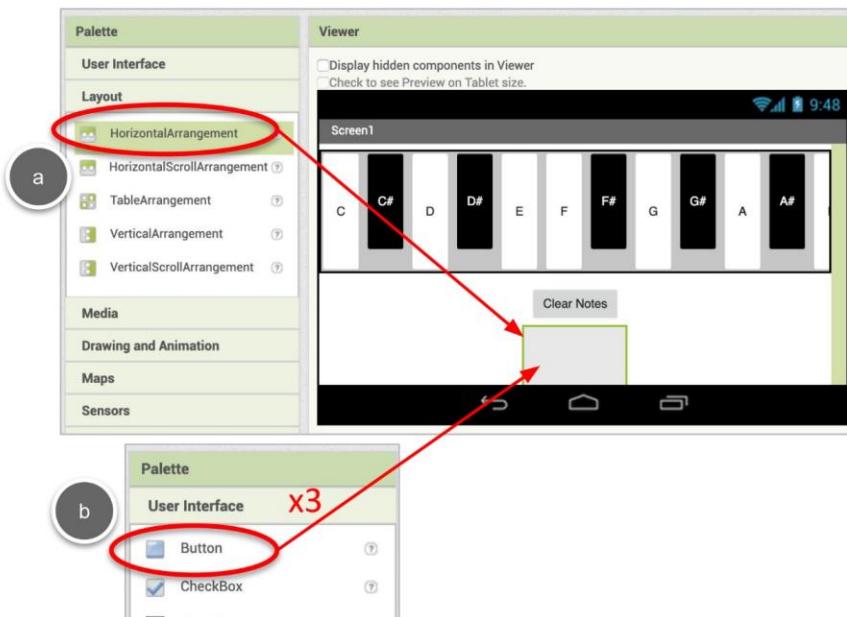


RECORD YOUR MUSIC

- 1 Switch to the Designer.

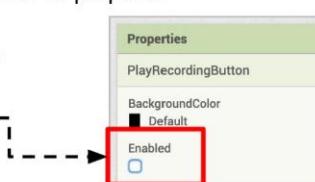


- 2 Add a **HorizontalArrangement** to the Viewer, and drop 3 **Buttons** into the **HorizontalArrangement**.

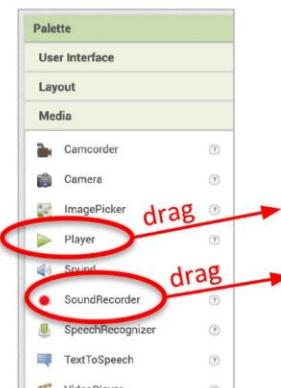


- 3 Name them **RecordButton**, **StopRecordingButton**, and **PlayRecordingButton**, in that order and change the *Text* property for each to its purpose.

- 4 Uncheck the *Enabled* box for the **StopRecordingButton** and the **PlayRecordingButton**.



- 5 Drag in a **SoundRecorder** component and another **Player** component from the Media drawer. Rename the Player component **RecordingPlayer**.

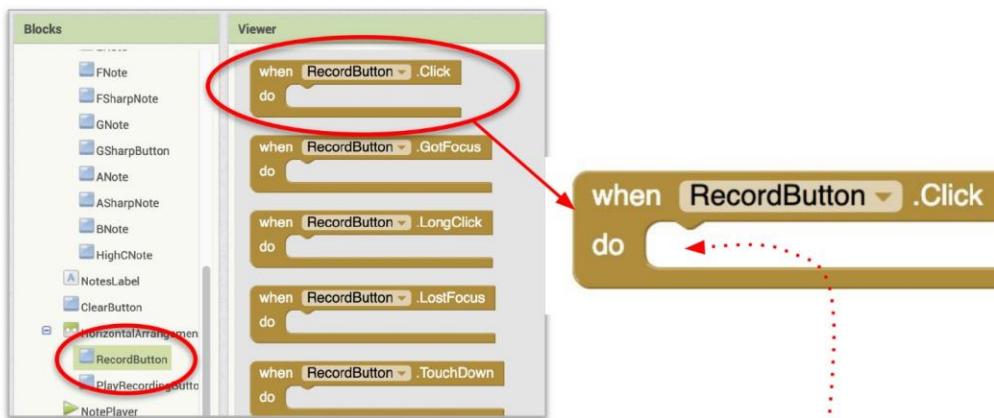


RECORD YOUR MUSIC (continued)

6 Switch to the Blocks Editor.



7 Drag out a RecordButton.Click event block.



8 When the user clicks this **Button**, you want to start the **SoundRecorder**.

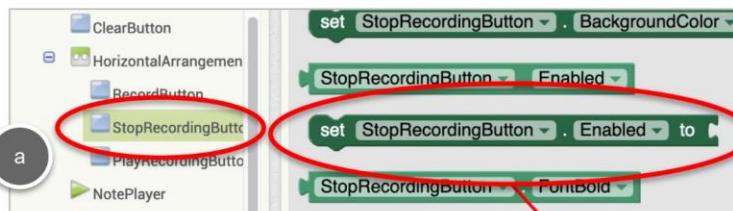


RECORD YOUR MUSIC (continued)

```
when RecordButton .Click
do call SoundRecorder1 .Start
```

9

You also want to enable the **StopRecordingButton** so they can stop the recording when they wish.



a

b

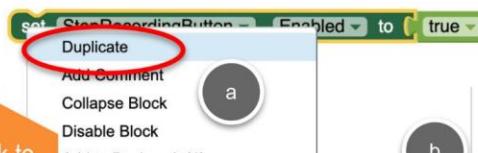
```
set StopRecordingButton .Enabled to [true v]
```



d

10

Since the app is recording, disable the **RecordButton** by Duplicating the **set StopRecordingButton.Enabled** block and changing **StopRecordingButton** to **RecordButton** in the dropdown.



Right click to the get popup menu

11

And change **true** to **false** in the dropdown.



RECORD YOUR MUSIC (continued)

12

Duplicate the **set RecordButton.Enabled to false** block and change it for **PlayRecordingButton**. You want to make sure the user doesn't try to play back a recording while the app is recording.

```
set [PlayRecordingButton v].Enabled to [false]
```

13

Drag out a **StopRecordingButton.Click** event block.



14

When the user stops recording, you want to:

- o Stop the **SoundRecorder1**.
- o Disable **StopRecordingButton**.
- o Enable **RecordButton**.

Use the Duplicate feature to duplicate and change what you need from the **RecordButton.Click** event.

```
call [SoundRecorder1 v].Stop
```

b



c

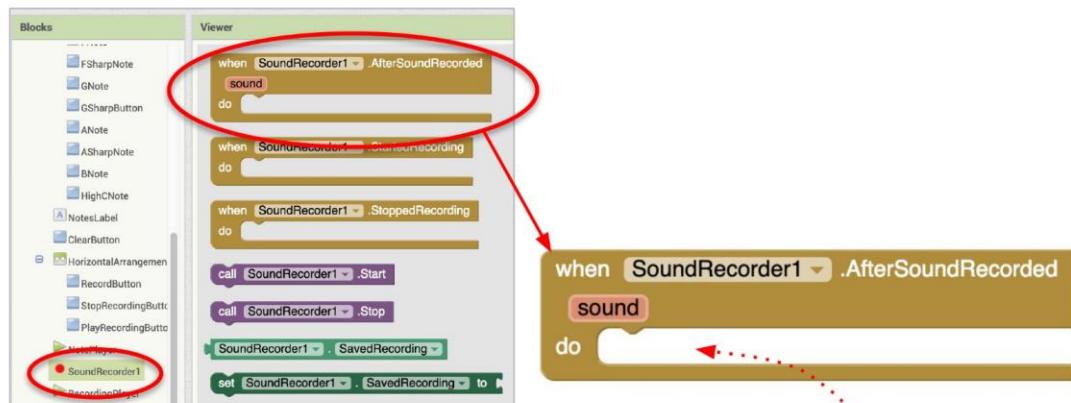
```
set [StopRecordingButton v].Enabled to [false]
```

d

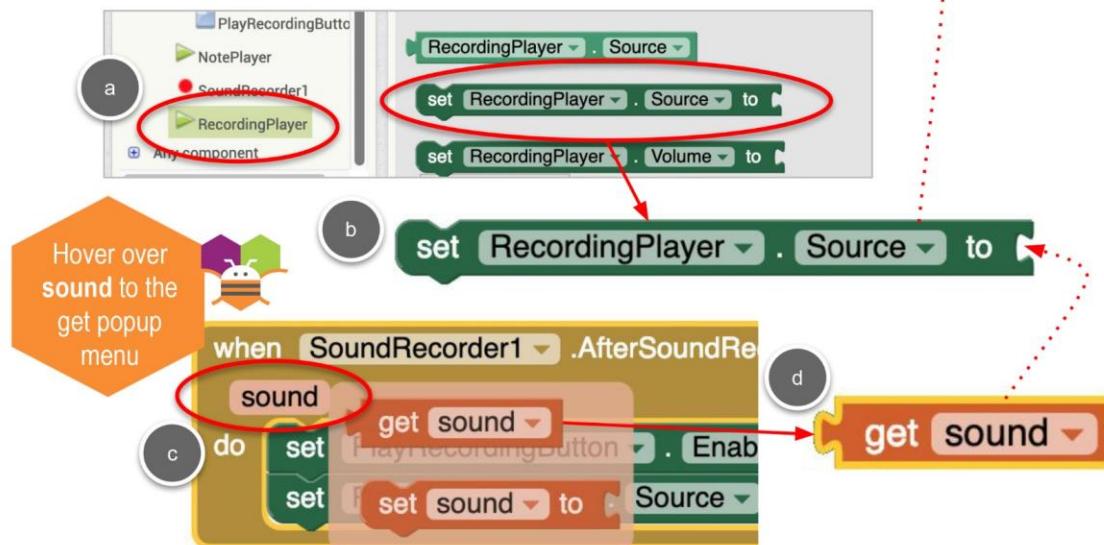
```
set [RecordButton v].Enabled to [true]
```

RECORD YOUR MUSIC (continued)

- 15 When the **SoundRecorder** finishes, it triggers an event, **SoundRecording.AfterSoundRecorded**. Drag out this block.



- 16 Set the Source for **RecordingPlayer** to the **sound** returned by the event.



RECORD YOUR MUSIC (continued)

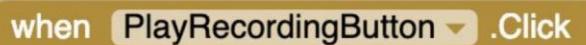
17

Now that you've set the Player's Source, enable the PlayRecordingButton so the user can play it back.



18

Add the **PlayRecordingButton.Click** event, and start the Player!



19

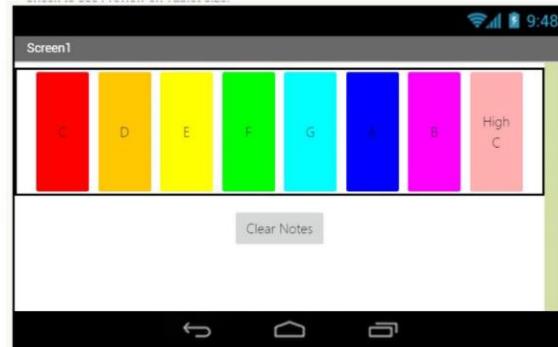
Try it out! Press the Record button, play some music, Stop the recording, and then play it back! How does it sound?

Unit 2 MyPiano: Part 1

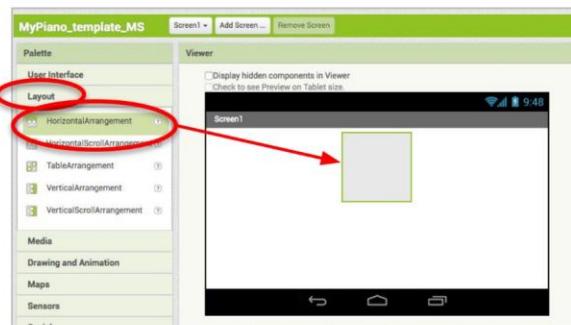
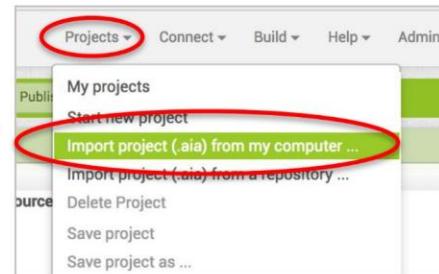
MY PIANO: PART 1

[START HERE](#)

In this lesson, you will create a project with App Inventor that will play notes like a piano!



- 1 Go to the MIT App Inventor website (<http://ai2.appinventor.mit.edu>) and sign in to your account.
- 2 Import the "MyPiano_Template.aia" project provided by your teacher.
- 3 Your Designer is empty except for media files. Note that all the sound files are included and appear in the Media panel.
- 4 Drag out a **HorizontalArrangement** from the Layout Drawer.

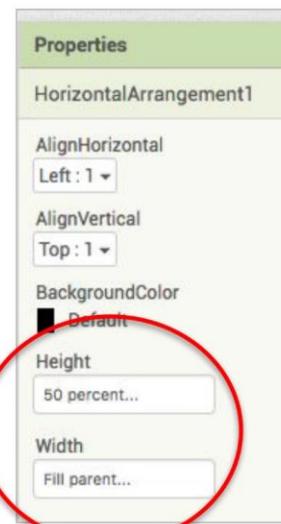


Copyright MIT 2019, Review Copy, Not for Distribution

LET'S CONTINUE

- 5** Change the *Height* of the **HorizontalArrangement** to **50%** and its *Width* to **Fill Parent**.

- 6** Drag *eight* Buttons into the **HorizontalArrangement**. They will appear side by side.

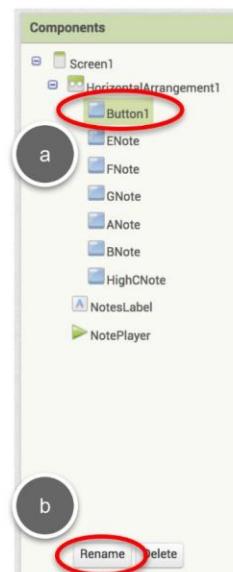


The correct order of piano keys is **C D E F G A B HighC**, so name each button according to its note.

- 7** Click on **Button1** in the Components window and change its name to "**CNote**".

- 8** Rename **Button2** "**DNote**".

- 9** Rename the remaining keys in this order **"C D E F G A B HighC"**.



You may not be able to see all the keys in the Viewer. We'll fix that next.

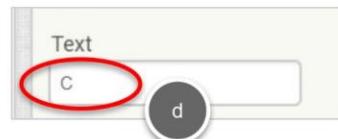


SETTING PROPERTIES

10

Change **CNote**'s Properties as follows:

- *Background Color: Red*
- *Height: Fill parent*
- *Width: 10 percent*
- *Text: C*



Components	Properties
Screen1	CNote
HorizontalArrangement1	BackgroundColor Red Enabled FontBold FontItalic FontSize 14.0 FontTypeface default Height Fill parent... Width 10 percent...
CNote	a
DNote	b
ENote	c
FNote	
GNote	
ANote	
BNote	
HighCNote	
NotesLabel	
NotePlayer	

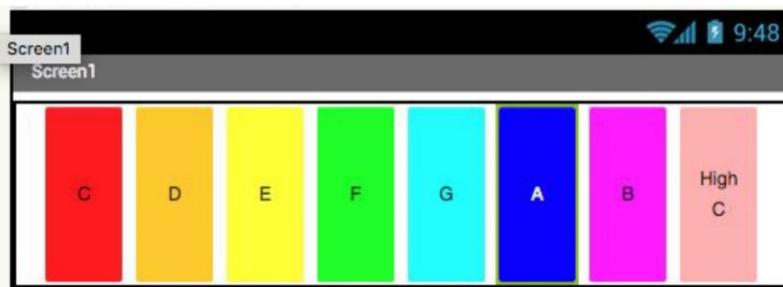
11

Change properties for **DNote**:

- *Background Color: Orange*
- *Height: Fill parent*
- *Width: 10 percent*
- *Text: D*

12

Change the remaining keys so your layout looks like the one below.

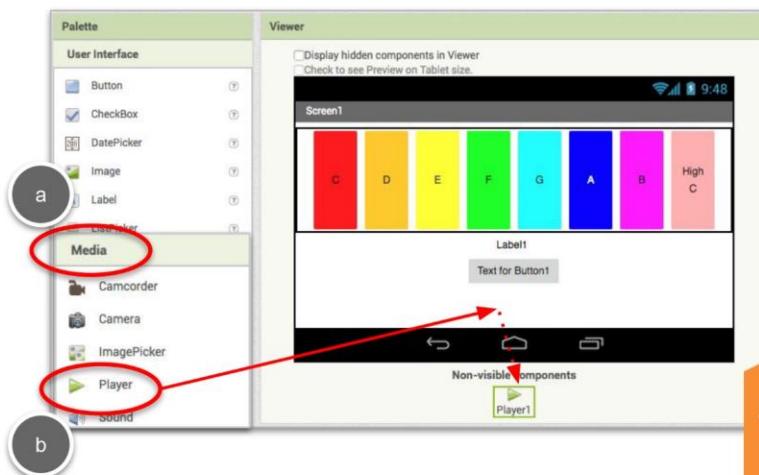


ADDING MORE COMPONENTS

13

Drag a **Label** and a **Button** and place them both below **HorizontalArrangement1**.

14

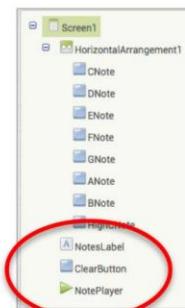
From the **Media** drawer, drag a **Player** component onto the Viewer.

 While you may use a Sound component for short sounds, the Player component is more reliable to play all sound files.

ADDING MORE COMPONENTS (continued)

15

Name the Button “**ClearButton**”,
the Label “**NotesLabel**”,
and the Player “**NotePlayer**”.



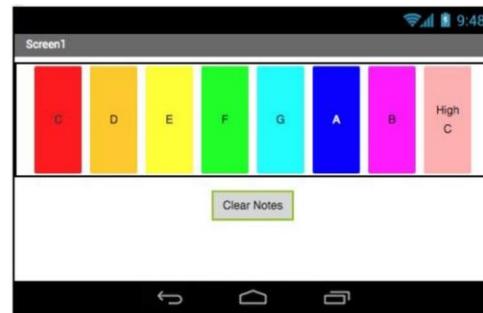
16

Set the *Text* property for **NotesLabel** to **blank** (erase all the text).

17

Set the *Text* property for
ClearButton to “**Clear Notes**”.

Your layout should look like this:



You'll code
the blocks
in Part 2!

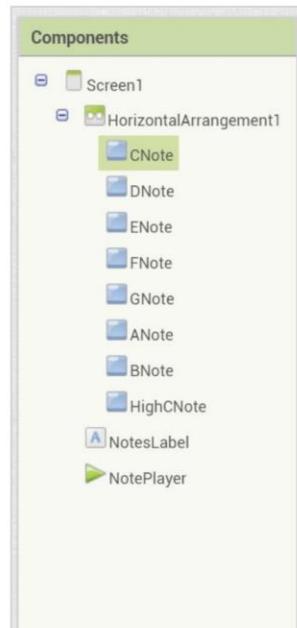


COMPUTATIONAL THINKING CONCEPTS

The following are the Computational Thinking Concepts learned in Part 1.

My Piano

1. Naming:



Unit 2 MyPiano: Part 2

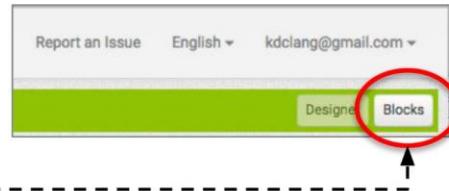
MY PIANO: PART 2

START HERE



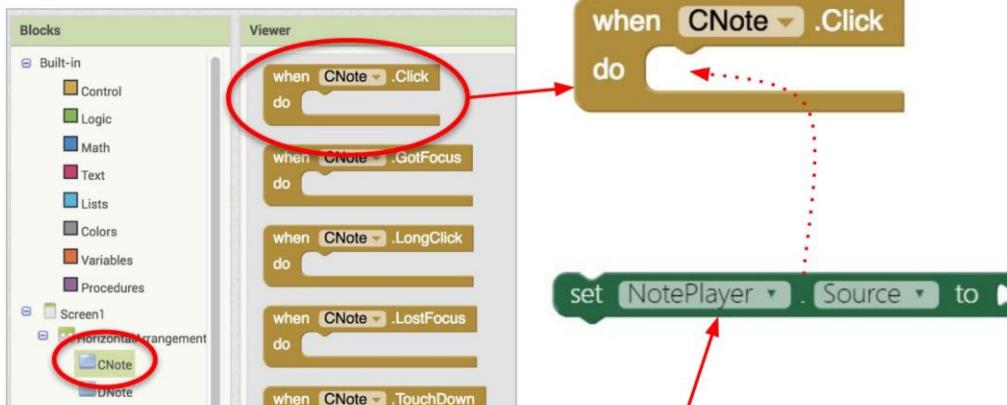
In this section, you will create a project with App Inventor that will play notes like a piano!

- 1 Go to the MIT App Inventor website (<http://ai2.appinventor.mit.edu>) and open your MyPiano project.



- 2 Click the **Blocks** button and go to Blocks Editor.

- 3 Drag out a **CNote.Click** event block.



- 4 Drag out a **set NotePlayer.Source** block and snap it into the **CNote.Click** event block.



Copyright MIT 2019, Review Copy, Not for Distribution

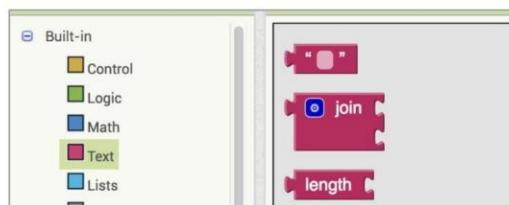
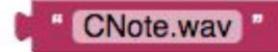
SET THE SOUND FILE

5

Drag out a **Text** block, modify its content to **"CNote.wav"** and snap it to **set NotePlayer.Source**. This will set **NotePlayer**'s sound file source to one of the pre-uploaded sound files: **CNote.wav**.



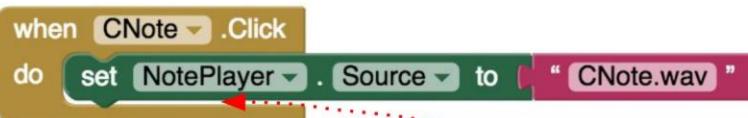
```
when CNote.Click
do set NotePlayer.Source to [CNote.wav v]
```

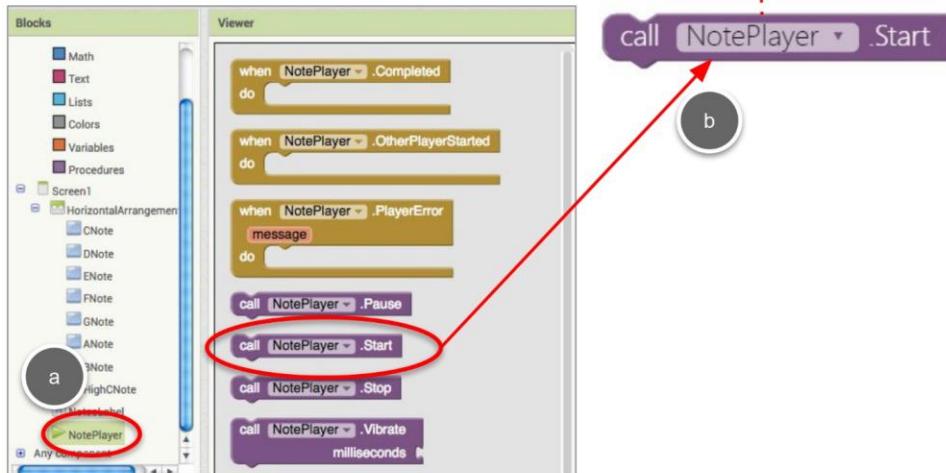
```
[CNote.wav v]
```

6

Drag out a **NotePlayer.Start** block. It will play the sound file you specified in the previous step.



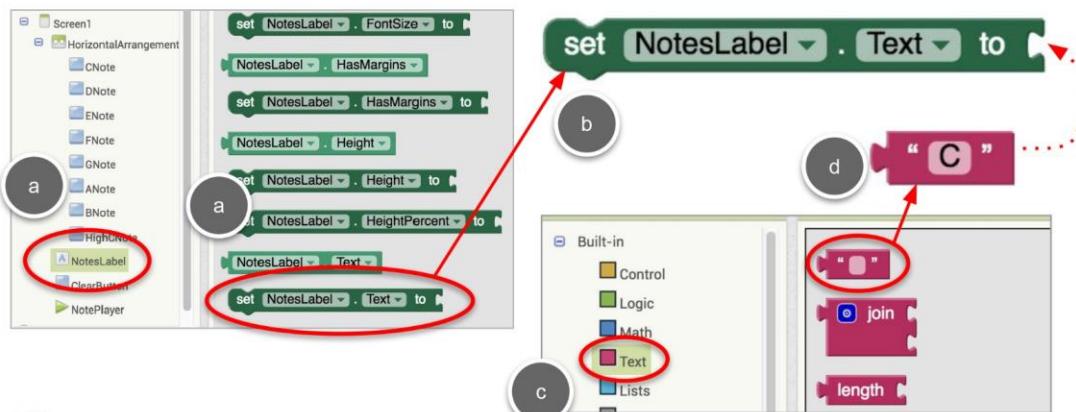
```
when CNote.Click
do set NotePlayer.Source to [CNote.wav v]
call NotePlayer.Start
```



DISPLAY THE NOTE

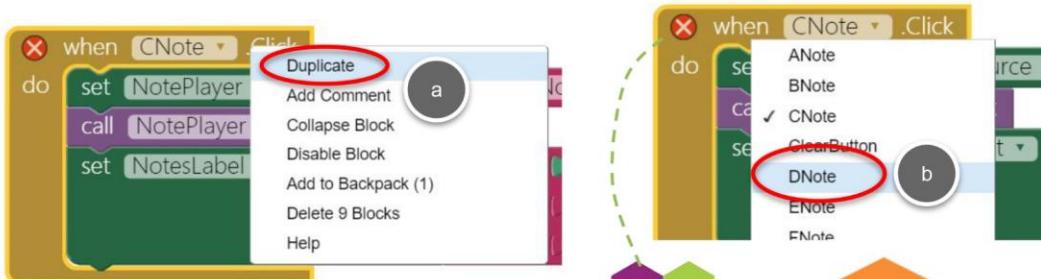
7

Drag out a **set NotesLabel.Text** block and a **Text** block with "C". This will display the current note being played as it plays. Snap it in below **NotePlayer.Start**.



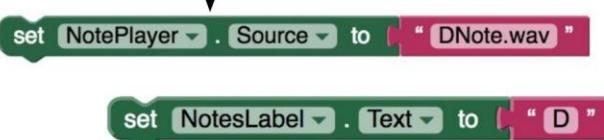
8

Right-click the **CNote.Click** event and Duplicate it. Another set of blocks will appear. Click the drop-down menu and select "**DNote**" to change it to that Button's Click event.



9

In the **DNote.Click** event, change all references to "C" to "D".



Note: don't worry about the X when you duplicate the event, this X indicates that there are two identical events in your app and will disappear when you change to the other event.

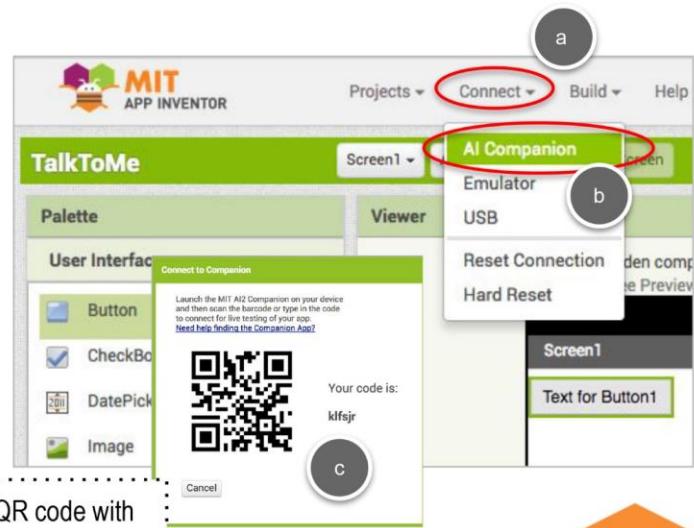
TESTING!

- 10 Stop here and test just these two notes!

Start MIT AI2 Companion
on your tablet



- 11 Connect to your tablet and try pressing the C and D buttons. Do they play different notes?



Scan the QR code with
MIT AI2 Companion

In the next lesson,
you will use a new block,
a Procedure, to make the
rest of your buttons play
notes too!



COMPUTATIONAL THINKING CONCEPTS

The following are the Computational Thinking Concepts learned in Part 2.

My Piano

1. Events:

```
when CNote .Click
do [ ]
```

2.

Sequences

```
when CNote .Click
do set NotePlayer . Source to "CNote.wav"
call NotePlayer .Start
set NotesLabel . Text to "C"
```

MD_2020

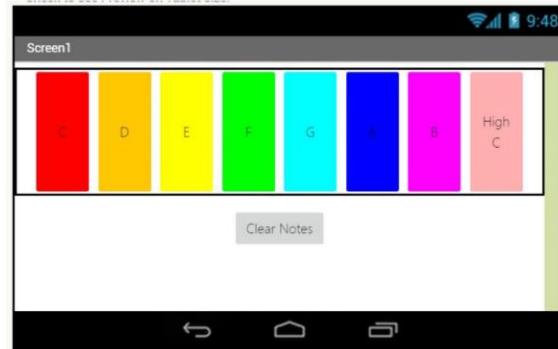
Unit 2 MyPiano: Part 3

MY PIANO: PART 3

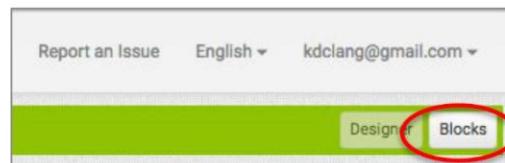
START HERE



In this activity, you will learn how to use a procedure to manage the blocks of all the piano keys.

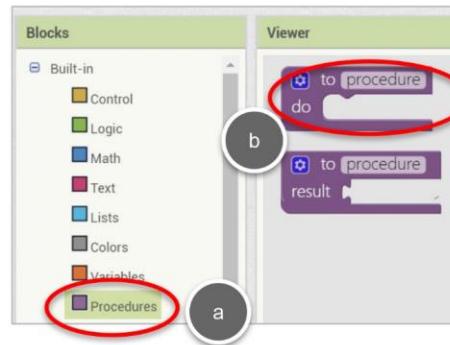


- 1 Go to the MIT App Inventor website (<http://ai2.appinventor.mit.edu>) and click the **Blocks** button to go to the Blocks Editor.

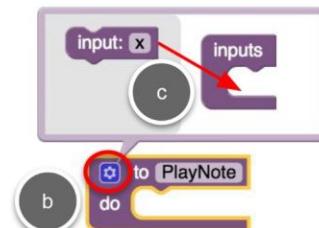
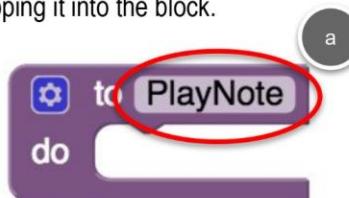


Because the code blocks for **CNote.Click** and **DNote.Click** are so similar, we are going to make a procedure to play the notes.

- 2 Click on **to procedure** in the **Procedures** Drawer, then drag out a **to procedure** block.



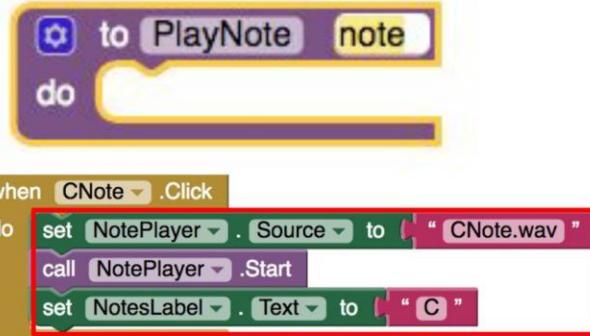
- 3 Change the name to **PlayNote**. Add an input by clicking on the blue circle and snapping it into the block.



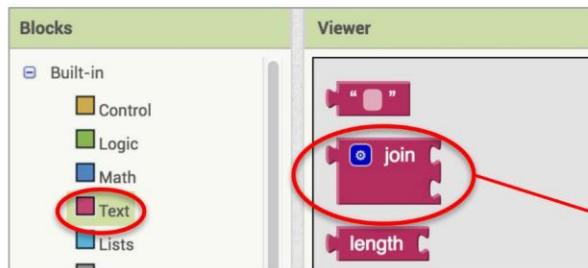
WRITE A PROCEDURE

4 Then, rename the input "note".

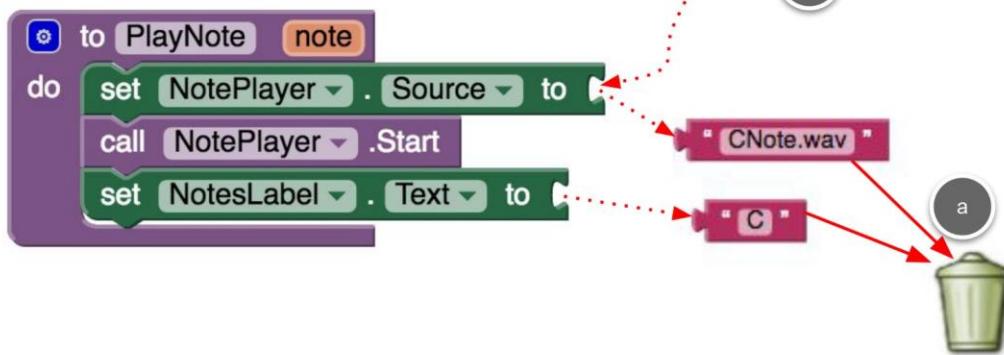
5 Drag the code inside the **CNote.Click** block to the new **PlayNote** procedure Block.



6 We want to be able to have different .wav filenames, based on the note, so drag out a **join** block from the **Text** drawer.



7 Delete the original text blocks and snap in the **join** block.



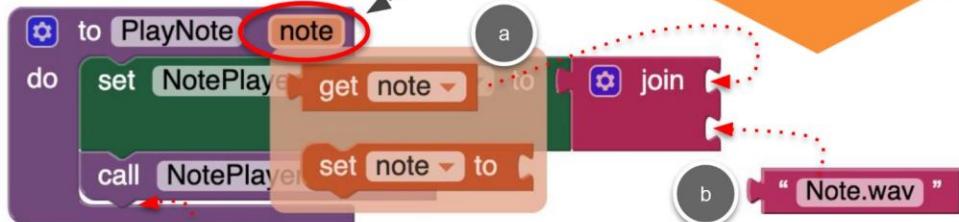
PLAYNOTE PROCEDURE

- 8 The only thing that changes is the note, so join note and "Note.wav".

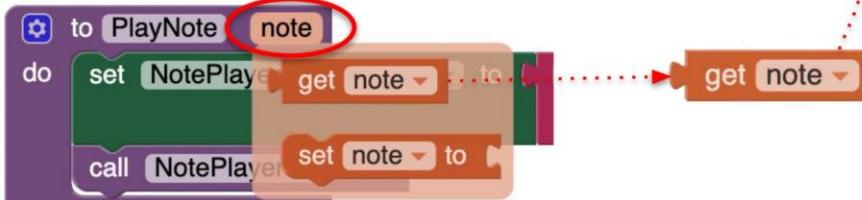
Hover over note for get note to pop-up.



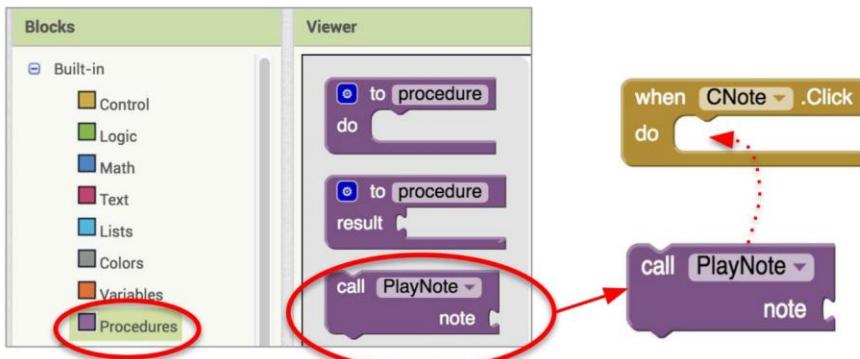
If note is "C", joining it with "Note.wav" makes "C"+"Note.wav", or "CNote.wav", the name of the sound file.



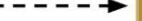
- 9 And set NotesLabel.Text to note.

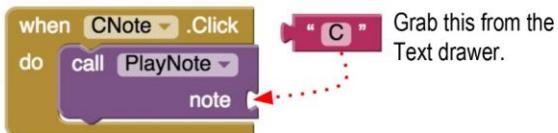


- 10 Drag out a call PlayNote block from the Procedures drawer and add to CNote.Click so that the PlayNote code runs when the C note is pressed.



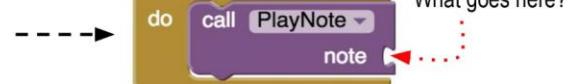
CALL THE PROCEDURE

- 11 Complete the puzzle piece and pass "C" as the note to **PlayNote**. 

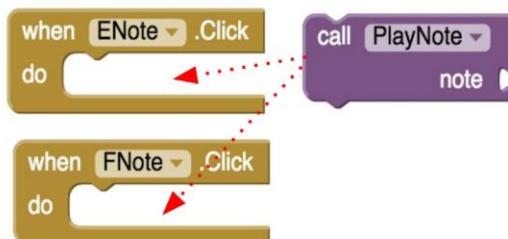


- 12 Do the same for **DNote.Click**.

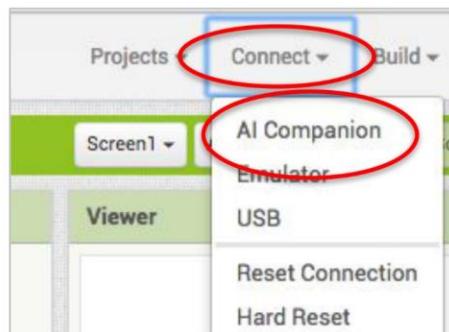
Delete the code blocks that were in **DNote.Click**, and replace them with a call to **PlayNote**.



- 13 Add **.Click** event blocks for all the other note buttons, and call **PlayNote** with the correct note for each button.



- 14 Test your app with the MIT AI2 Companion to make sure you can play all eight notes and see the correct notes displayed.

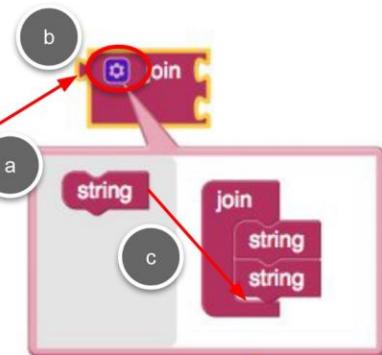
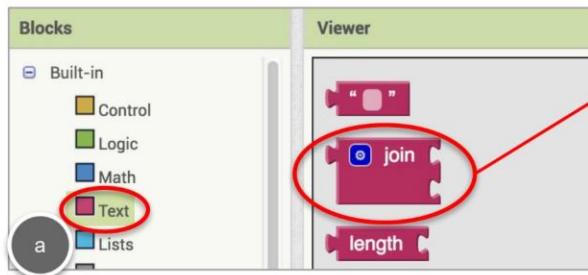


CHANGING THE APP

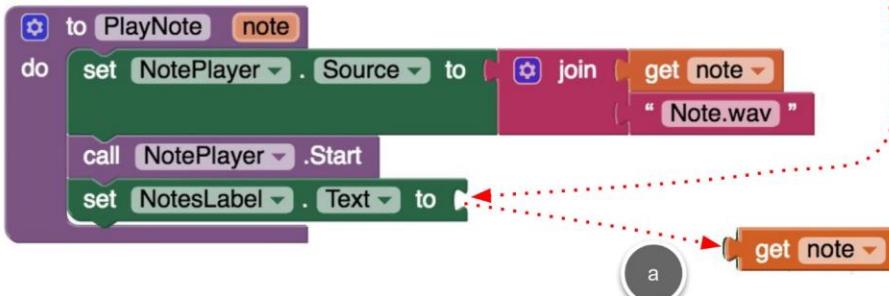
Let's make a change to our app.
Instead of just displaying the current note, let's display **all** the notes pressed in sequence , like "A C C C D E F" etc.



- 15 In the **PlayNote** procedure, let's update **set NotesLabel.text** so it uses a **join** block. Add a third string to the **join** block.

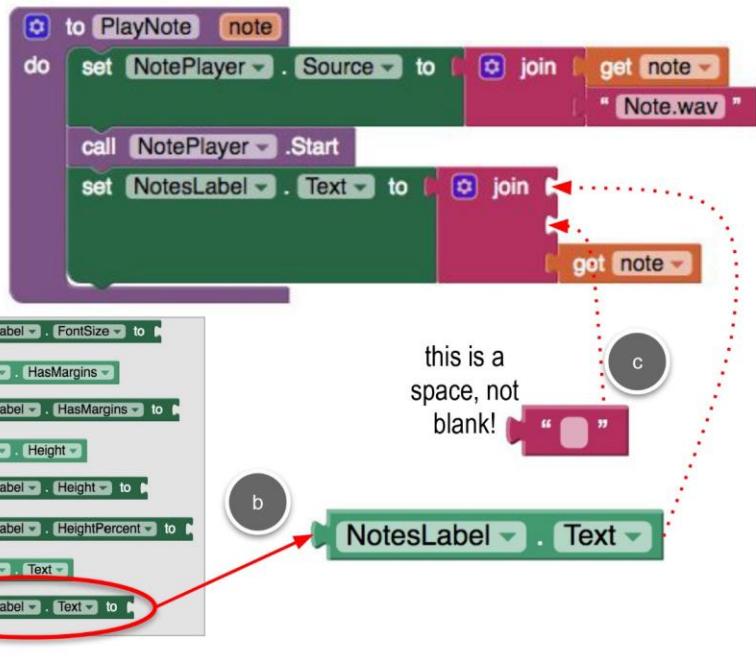


- 16 Move **get note** out and replace it with the **join** block.



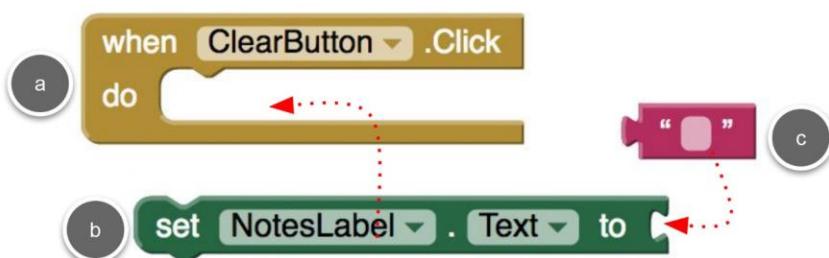
COMPLETE THE JOIN

- 17 Add **NotesLabel.Text** and a Text space block as the other two strings in the **join** block.

CLEAR BUTTON

Sometimes the string of notes can get too long, so let's code the Clear button to reset the string.

- 18 Drag out the **ClearButton.Click** block and clear the **NotesLabel**.



TESTING!

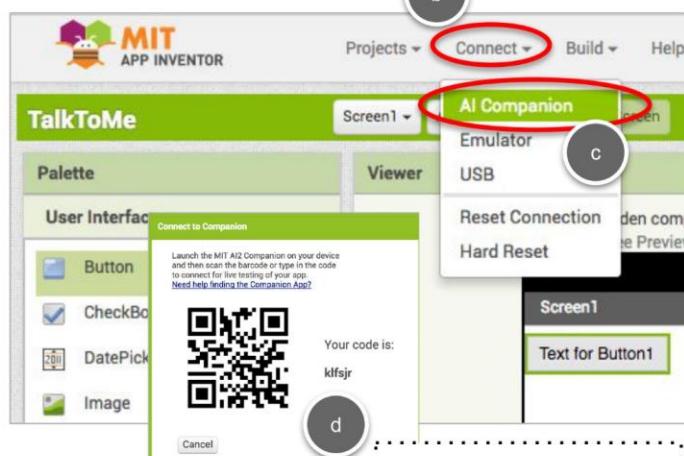
19

Now test your app on your tablet!

a

Start MIT AI2 Companion
on your tablet

b



d

Scan the QR code with
MIT AI2 Companion

20

Play with your piano. Try to play all the “keys”. You should hear the corresponding notes, and they should also appear in **NotesLabel**.