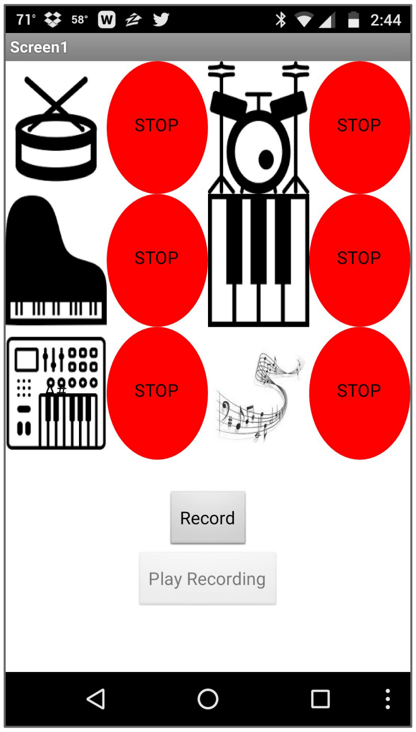
MUSIC MAKER: PART 2

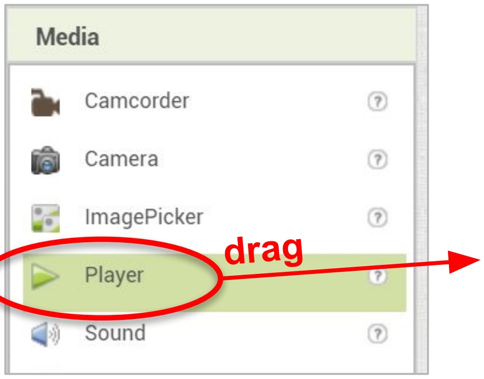
In Part 2, you will add code blocks to play and stop the instruments with the press of a button!



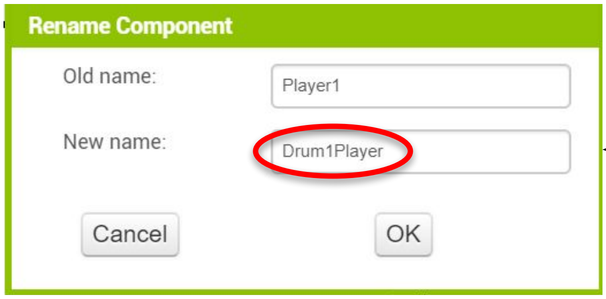
START HERE

1 Open your MusicMaker project in (http://ai2.appinventor.mit.edu).

2 Add a **Player** component to your app by dragging it from the **Media** Drawer in the Designer Palette to the Viewer.



3 Rename it appropriately for your chosen instrument. In this example, the app plays drums.

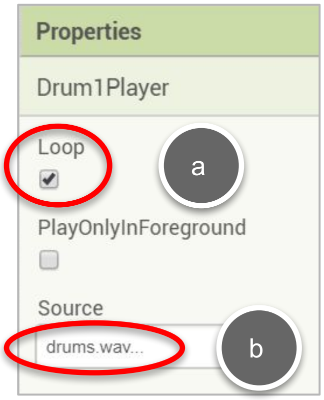

Rename Component

Old name: Player1

New name: Drum1Player

Cancel OK

4 Check the *Loop* option to play the sound repeatedly and set its *Source* to the appropriate sound file..


Properties

Drum1Player

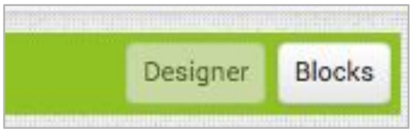
Loop a
☑

PlayOnlyInForeground
☐

Source
drums.wav... b

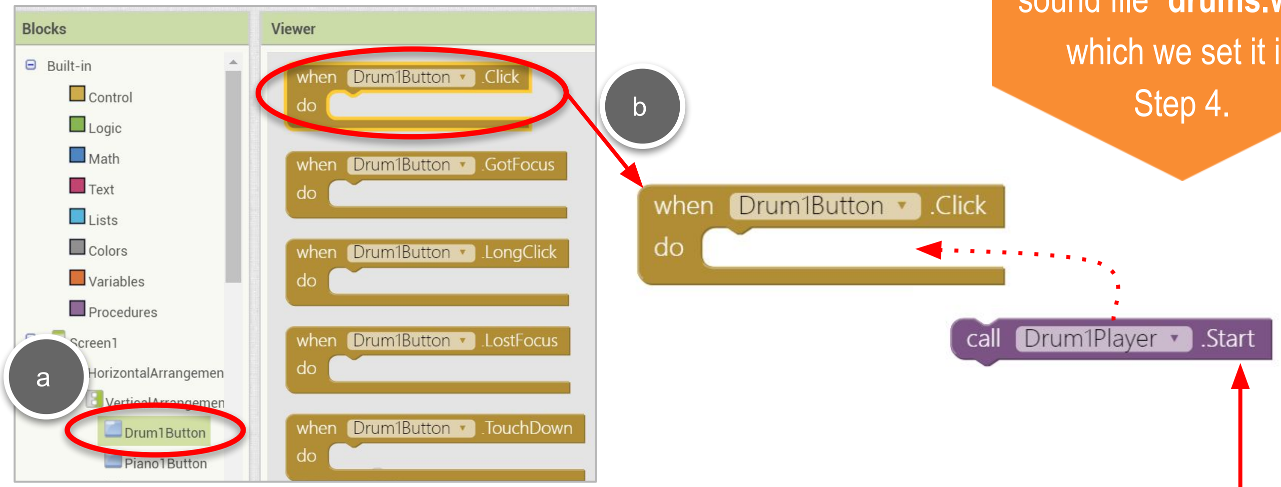
CODING THE BLOCKS

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

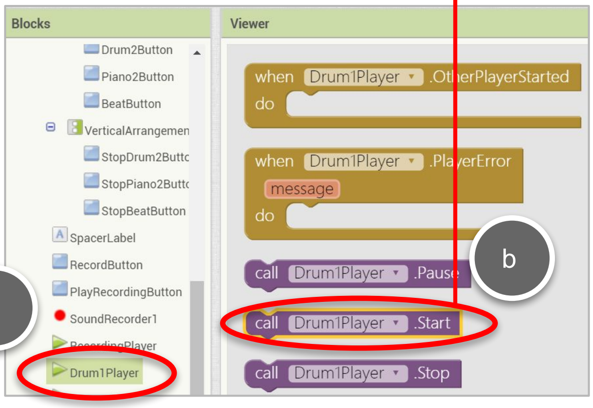


6 Code the your first Button. Using the drums example, there could be a button named **Drum1Button**.

This will set **Drum1Player** to play the sound file "**drums.wav**", which we set it in Step 4.



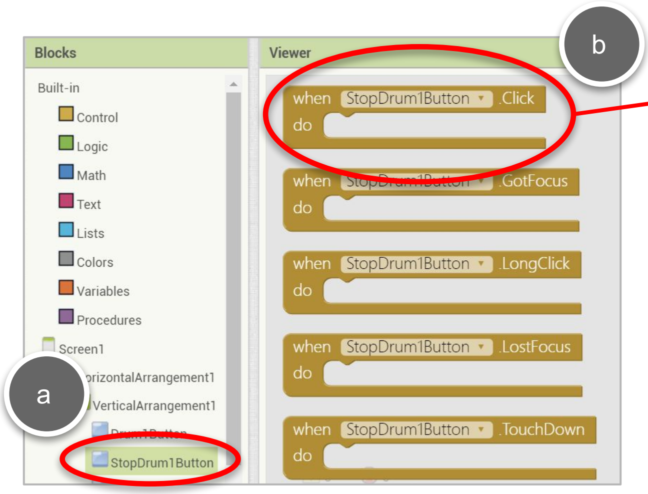
7 Click on the matching **Player** component you added, drag out a **.Start** block, and snap it into the **Button.Click** event block.

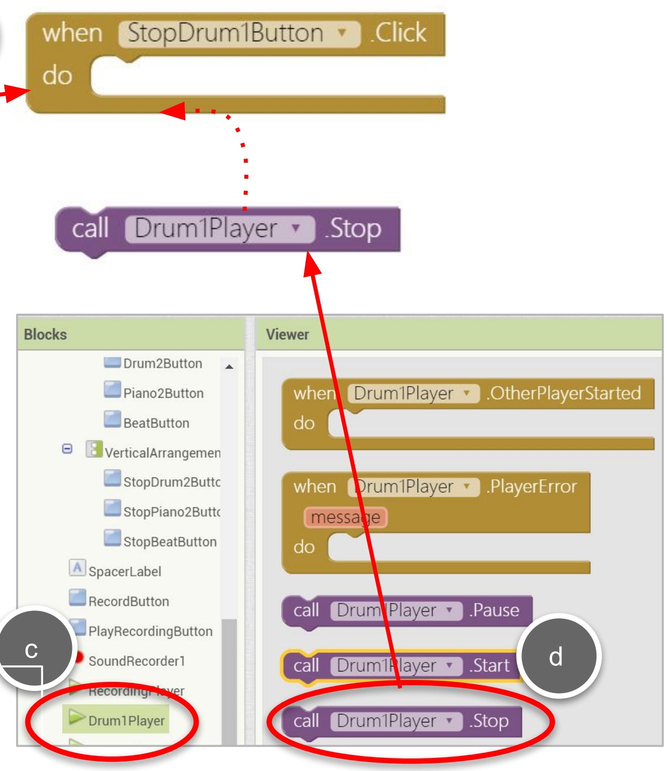


STOP THE DRUMS!

8 The **Button** to stop the instrument sound uses the same idea. Drag out a **Button.Click** event block for the matching Stop Button for this instrument. Again, we'll use the Drums as an example.

9 Drag out a matching **Player.Stop** block and snap it into the **StopButton.Click** event. This will make the **Player** stop playing.

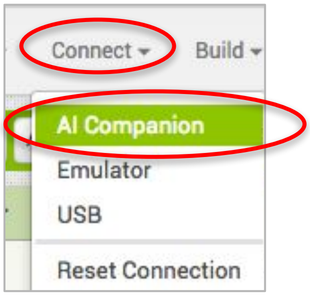

Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Colors
Variables
Procedures
Screen1
HorizontalArrangement1
VerticalArrangement1
Drum1Button
StopDrum1Button
a
b
when StopDrum1Button .Click do
when StopDrum1Button .GotFocus do
when StopDrum1Button .LongClick do
when StopDrum1Button .LostFocus do
when StopDrum1Button .TouchDown do

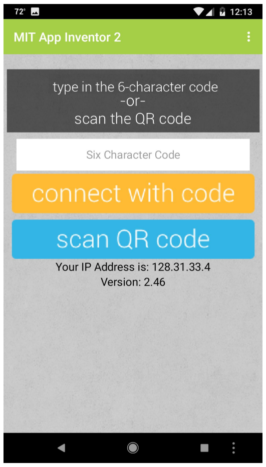


TESTING!

10 Now test starting and stopping that instrument with your two Buttons!

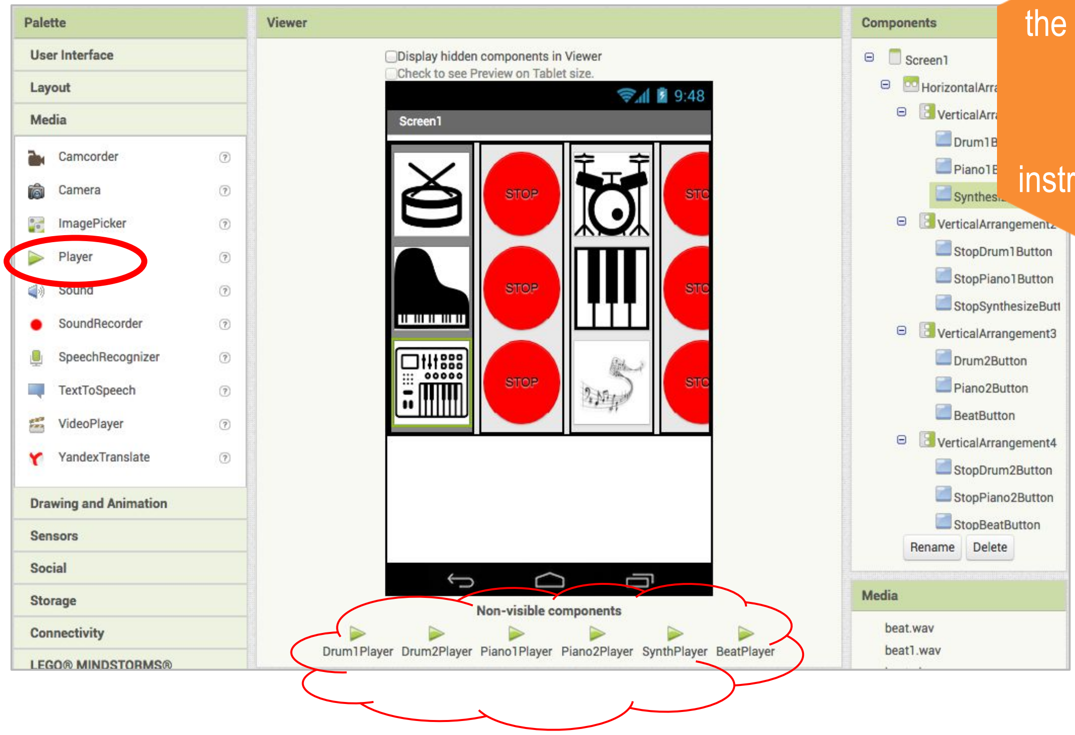
Start MIT AI2 Companion on your tablet





NOW DO THE REST!

11 Now that you have one Button set working, add code for your other instruments. Make sure you have a Button to start the instrument, and one to stop it. Also, make sure you add a new Player component for each instrument. Remember to name them appropriately!



Because the instruments will play simultaneously, *in parallel*, each instrument needs its own Player.

12 After adding all your Buttons, test your app again. You should be able to play multiple instruments at the same time. Check that all the start and stop buttons work!

If you want a challenge, try recording the music as it plays, and then play it back!

Choose Ways to Extend Your App

Here are a few features you could add if you want to expand your app

Try the Challenge: record and play back your music

Add a Pause button for each instrument

Instead of 2 buttons, make one button toggle play/stop.

Add more instruments

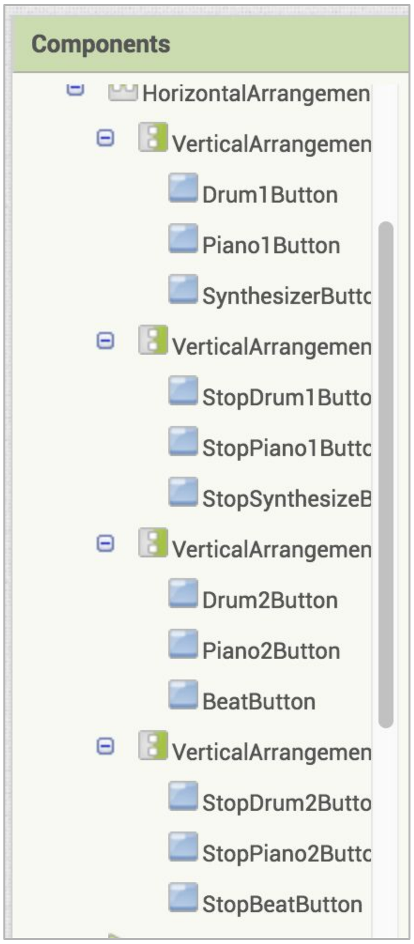
What other ideas do you have?

COMPUTATIONAL THINKING CONCEPTS

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

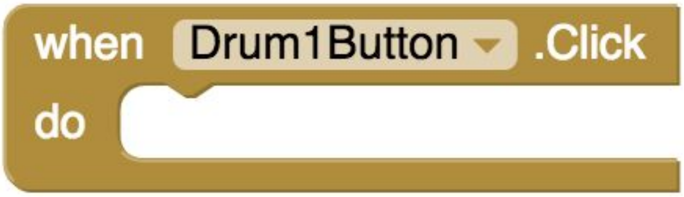
Music Maker

1. Naming:



Components
HorizontalArrangemen
VerticalArrangemen
Drum1Button
Piano1Button
SynthesizerButtc
VerticalArrangemen
StopDrum1Butto
StopPiano1Buttc
StopSynthesizeB
VerticalArrangemen
Drum2Button
Piano2Button
BeatButton
VerticalArrangemen
StopDrum2Butto
StopPiano2Buttc
StopBeatButton

2. Events



```
when Drum1Button .Click
do
```