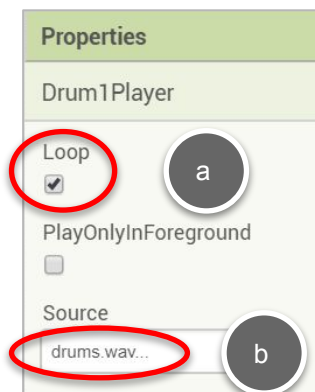
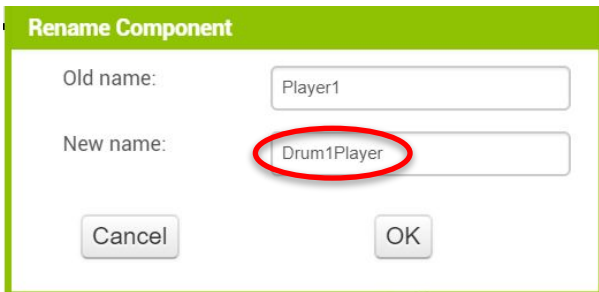
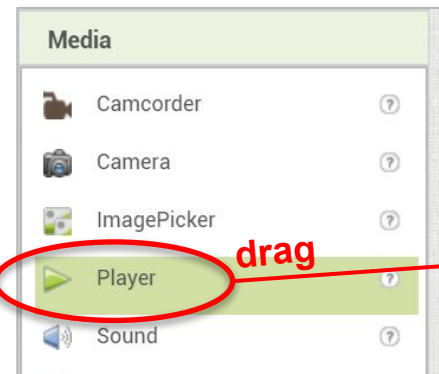
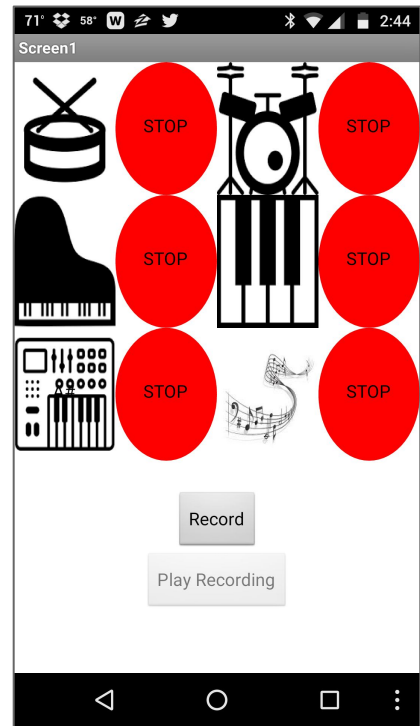


MUSIC MAKER: PART 2

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



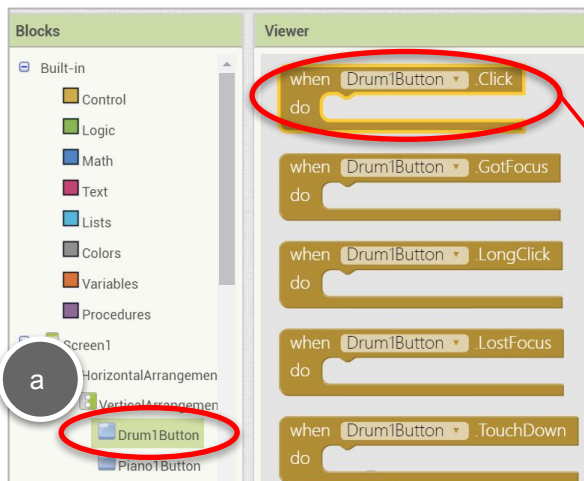
CODING THE BLOCKS

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

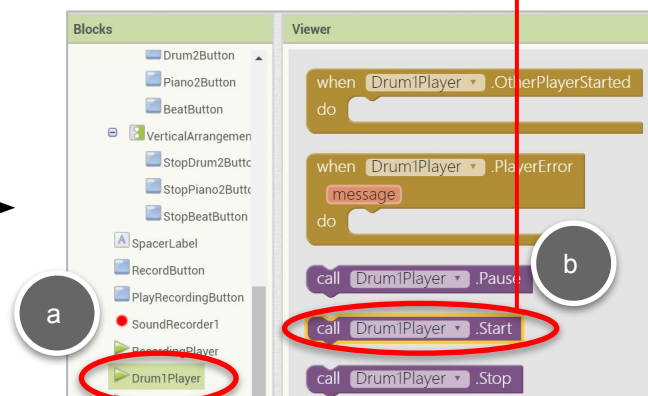


6 Code your first Button. Using the drums example, there could be a button named **Drum1Button**.

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.

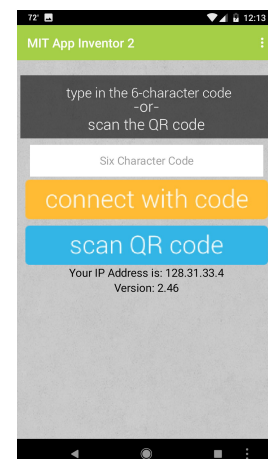
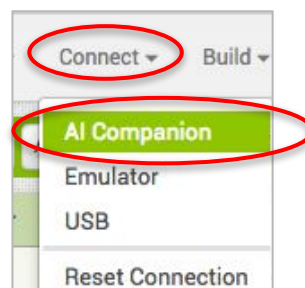
The image contains three screenshots of the MIT App Inventor 2 web interface, illustrating the steps to stop a drum player.

- Top Left Screenshot:** Shows the 'Blocks' palette on the left. A red circle labeled 'a' highlights the 'StopDrum1Button' block under the 'Control' category. A red circle labeled 'b' highlights a 'when StopDrum1Button.Click' event block in the 'Viewer' pane.
- Top Right Screenshot:** A close-up of the event block from the previous screenshot. A red arrow points from the 'do' slot to a 'call Drum1Player.Stop' block.
- Bottom Screenshot:** Shows the 'Blocks' palette and 'Viewer' pane. A red circle labeled 'c' highlights the 'Drum1Player' block in the 'Blocks' palette. A red circle labeled 'd' highlights the 'call Drum1Player.Stop' block in the 'Viewer' pane. A red arrow points from the 'Drum1Player' block in the palette to the 'call Drum1Player.Stop' block in the viewer.

TESTING!

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

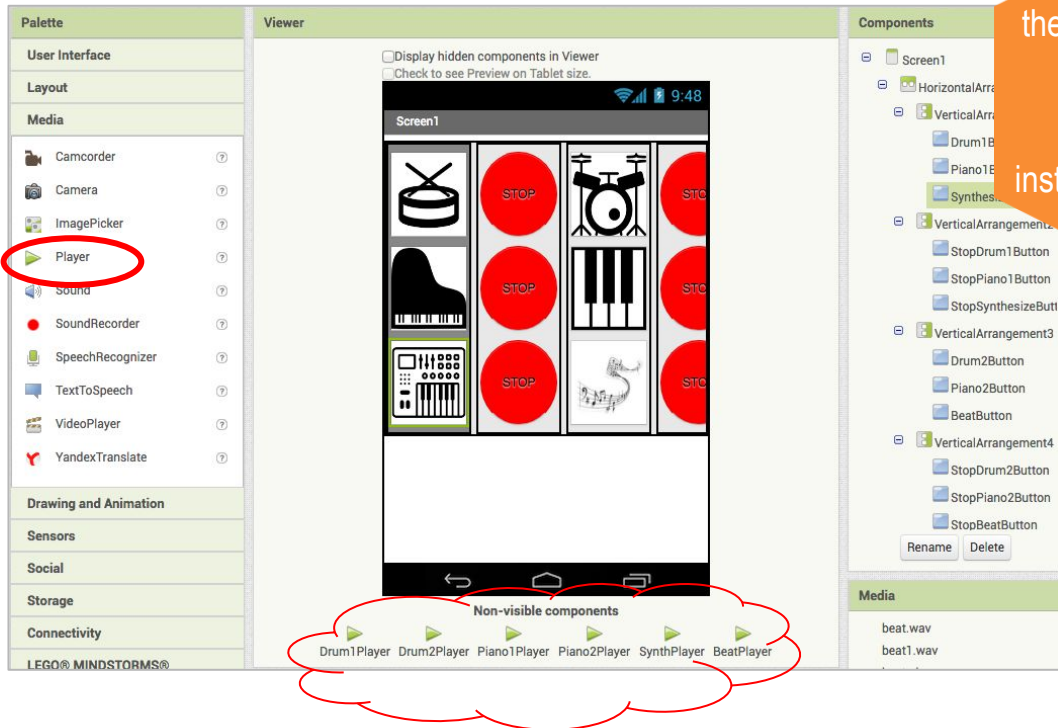
Start MIT AI2 Companion on your tablet



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

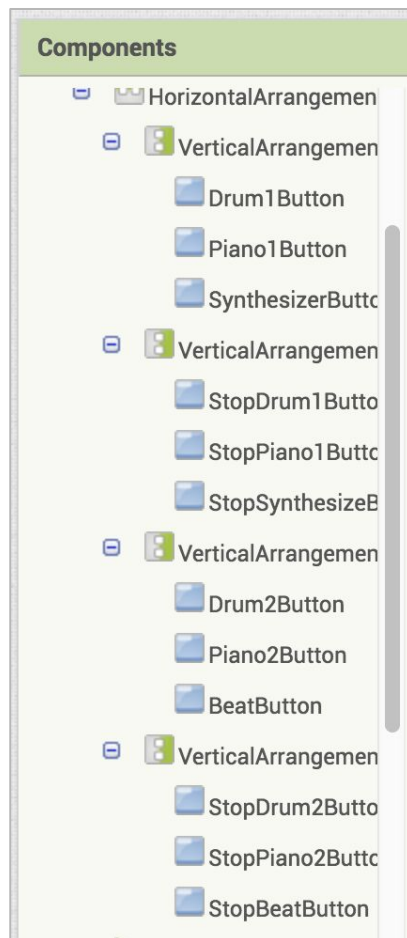


COMPUTATIONAL THINKING CONCEPTS

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

Music Maker

1. Naming:



2. Events

