

# Music Maker

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>1. Synopsis</b>	<b>3</b>
<b>2. Learning Objectives</b>	<b>3</b>
<b>3. Mapping with the Computational Thinking Framework</b>	<b>4</b>
<b>4. Mapping with the CSTA Standards</b>	<b>7</b>
<b>5. Learning Prerequisites</b>	<b>8</b>
<b>6. Lesson Plan ( 45 minutes x 3)</b>	<b>9</b>
<b>7. Assessment</b>	<b>12</b>
<b>8. Screen Design and Code</b>	<b>15</b>
<b>Appendix 1</b>	
<b>Music Maker Teacher's Guide</b>	
<b>Lesson 1</b>	<b>18</b>
<b>Appendix 2</b>	
<b>Music Maker Teacher's Guide</b>	
<b>Lesson 2</b>	<b>20</b>
<b>Appendix 3</b>	
<b>Music Maker Teacher's Guide</b>	
<b>Lesson 3</b>	<b>22</b>

# 1. Synopsis

In this unit, students make musical apps as a follow-up to the MyPiano app made in the previous unit. Students are encouraged to design their own musical apps using sounds effects and instrument recordings, with provided images and matching music/sound recordings. They can also add their own images and sounds as an added challenge. This unit allows students to reinforce previously learned Computational Thinking concepts, such as events, parallelism, and repetition. Students will reuse and remix code from the MyPiano app. Students will also practice being incremental and iterative, as well as testing and debugging.

## 2. Learning Objectives

After completing this unit, students will be able to:

1. Utilize Layout components to organize the user interface of an app.
2. Use events, parallelism, and naming in their app.
3. Reuse and remix code.
4. Be iterative and incremental in developing their app.
5. Test and debug to make a working app.
6. Provide feedback and act on suggestions for improvement.

### 3. Mapping with the Computational Thinking Framework

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

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

#### Computational Thinking Concepts

Unit 3 Music Maker		
1. Sequences	✓✓	Sequencing of blocks is important in order for the app to function correctly.
2. Events	✓✓✓	Button click events are used to start and stop sound.
3. Repetition	✓✓	Students are introduced to the Loop property of the Player component.
4. Conditionals		
5. Parallelism	✓✓✓	Music sound files play simultaneously
6. Naming	✓✓✓	It is important that students name their components to avoid confusion.
7. Operators		
8. Manipulation of data and elementary data structure		

## Computational Thinking Practices

Unit 3 Music Maker		
1. Reusing and remixing	✓✓✓	Students will build this app by reusing code from the previous Piano app.
2. Being incremental and iterative	✓✓✓	Students build this app in incremental steps after confirming at every step that their app works.
3. Abstracting and modularizing		
4. Testing and debugging	✓✓✓	Students need to test changes to the app and fix any mistakes.
5. Algorithmic thinking	✓	Some students who do the challenge tasks need to follow a basic algorithm to play/stop/record their music.

## Computational Thinking Perspectives

Unit 3 Music Maker		
1. Expressing	✓✓✓	Students create an app with their own choice of sounds or music to express themselves.
2. Questioning	✓✓✓	Students learn new concepts like playing sounds simultaneously within MIT App Inventor.
3. Connecting	✓✓✓	Students are connecting with their personal interests in terms of sound or music.
4. Computational identity	✓✓	By creating the app, students can see themselves as creators of technology.
5. Digital empowerment	✓✓✓	Students feel empowered by creating an app they can share with friends and family.

## 4. Mapping with the CSTA Standards

This table shows the alignment of this unit with the intended learning outcomes to the CSTA CS Standards. The entries in the tables indicate the expected relevance of the unit to each outcome:

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 design the layout and implement the code for their apps.
2-AP-17	Incorporate existing code, media, and libraries into original programs, and give attribution. [C] AP: Program Development [P] Abstraction (4.2), Creating (5.2), Communicating (7.3)	Students use images and sound files provided in a template.
2-AP-18	Systematically test and refine programs using a range of test cases. [C] AP: Program Development [P] Testing (6.1)	Students test their apps as they develop.

## 5. Learning Prerequisites

Students should have completed two App Inventor units, “Hello It’s Me” and “My Piano App,” and should be familiar with button events and playing sounds, prior to starting this unit.



## 6. Lesson Plan ( 45 minutes x 3)

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

### Lesson 1

Time	Activity
10 min	<b>Introduction to Music Maker Project</b> Ask students: <ol style="list-style-type: none"><li>1. What did you like about the MyPiano app?</li><li>2. How might the MyPiano app be made better, more interesting or fun?</li><li>3. Have you tried the GarageBand app before?</li></ol> Explain to students that they will make their own Music Maker, with various instruments and beat sound files for users to play. Demonstrate an example MusicMaker app.
15 min	<b>App Design</b> Students design their own app on paper: <ol style="list-style-type: none"><li>1. Give students access to media files available.</li><li>2. Give students the Music Maker Design Worksheet.</li><li>3. Have students draw the layout of their apps. Encourage them to think about the placement of buttons for the ease of starting and stopping/pausing the sound plays.</li><li>4. Remind students that their apps should include:<ol style="list-style-type: none"><li>a. Minimum of four sounds</li><li>b. The ability to start and stop any sound</li></ol></li></ol>
20 min	<b>Adding of Components in Designer</b> Based on the layout drawn on the worksheet, ask students to add the necessary components for their apps, following the Student Guide.

## Lesson 2

Time	Activity
10 min	<b>Demonstration of Player Components</b> <ol style="list-style-type: none"><li>1. Ask students: “Is there any difference between the sounds that play in the MyPiano app and this Music Maker app?” (The answer is that in this app, multiple sounds can play at once, so they will need multiple Player components).</li><li>2. Demonstrate how to add Player components.<ol style="list-style-type: none"><li>a. Explain the Loop property and compare to loops in code (from Scratch).</li></ol></li><li>3. Demonstrate the Start, Stop, and Pause blocks for the Player component.</li></ol>
30 min	<b>Coding Activity</b> <ol style="list-style-type: none"><li>1. Students code their apps following Student Guide: Lesson 2.</li><li>2. If some students finish early, they may try the Challenge, using Student Guide: Challenge.</li></ol>
5 min	<b>Wrapup</b> <p>Check in with students to see what progress they have made. There is a third lesson in which some students can finish up and others can add enhancements.</p>

## Lesson 3

Time	Activity
5 min	<b>Introduction to Lesson 3</b> <ol style="list-style-type: none"><li>1. Review what has been accomplished so far.</li><li>2. Explain that students can work on completing the app if they haven't already, or they can use the lesson to add more notes or new features. Students may also try the Challenge to create a toggle button to record music.</li></ol>
20 min	<b>Add New Features</b> <p>Based on suggestions at the end of the Student Guide: Part 3, students can add more notes. Note files are included in the template for the sharp notes.</p>
10 min	<b>Sharing</b> <p>Ask students to share with a classmate. They should demonstrate their app, then let the other person try it. Each person should complete a Feedback worksheet for the other person, providing two positive statements and a constructive suggestion for improvement.</p>
10 min	<b>Wrapup</b> <ol style="list-style-type: none"><li>1. Ask students to reflect on creating their own apps. Ask if it felt different from following steps to make a set app.</li><li>2. Ask students to complete the multiple choice and Learning Attitudes Survey.</li></ol>

## 7. Assessment

### Multiple-choice questions

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

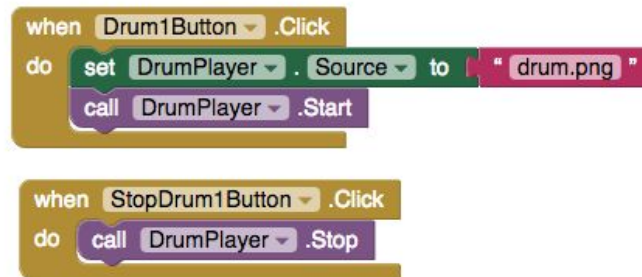
1. What would happen if you press **Piano1Button** and then three seconds later press **Drum1Button** using the following blocks?



- A. Piano1Player would start playing a sound, after three seconds Drum1Player would start playing and Piano1Player would stop playing.
- B. Piano1Player would play its sound continuously, and Drum1Player would never play.
- C. Piano1Player would start playing a sound, after three seconds Drum1Player would start playing and Piano1Player would not stop playing.
- D. Piano1Player would start playing a sound, after three seconds (when Drum1Player is pressed) the program would stop working.

(Answer: C)

2. A student writes an app to play sounds. Her code blocks for starting and stopping the Drum sounds are below. When testing, she presses the Drum1Button but instead of hearing the drum sound, an error appears. What is the problem?



- A. The DrumPlayer source file is not a sound file.
- B. The purple block should be DrumPlayer.Stop instead of DrumPlayer.Start in the Drum1Button.Click event.
- C. The DrumPlayer source file should be set in the Designer, not in blocks.
- D. She needs to reset the MIT AI2 Companion.

(Answer: A)

## Learning attitudes

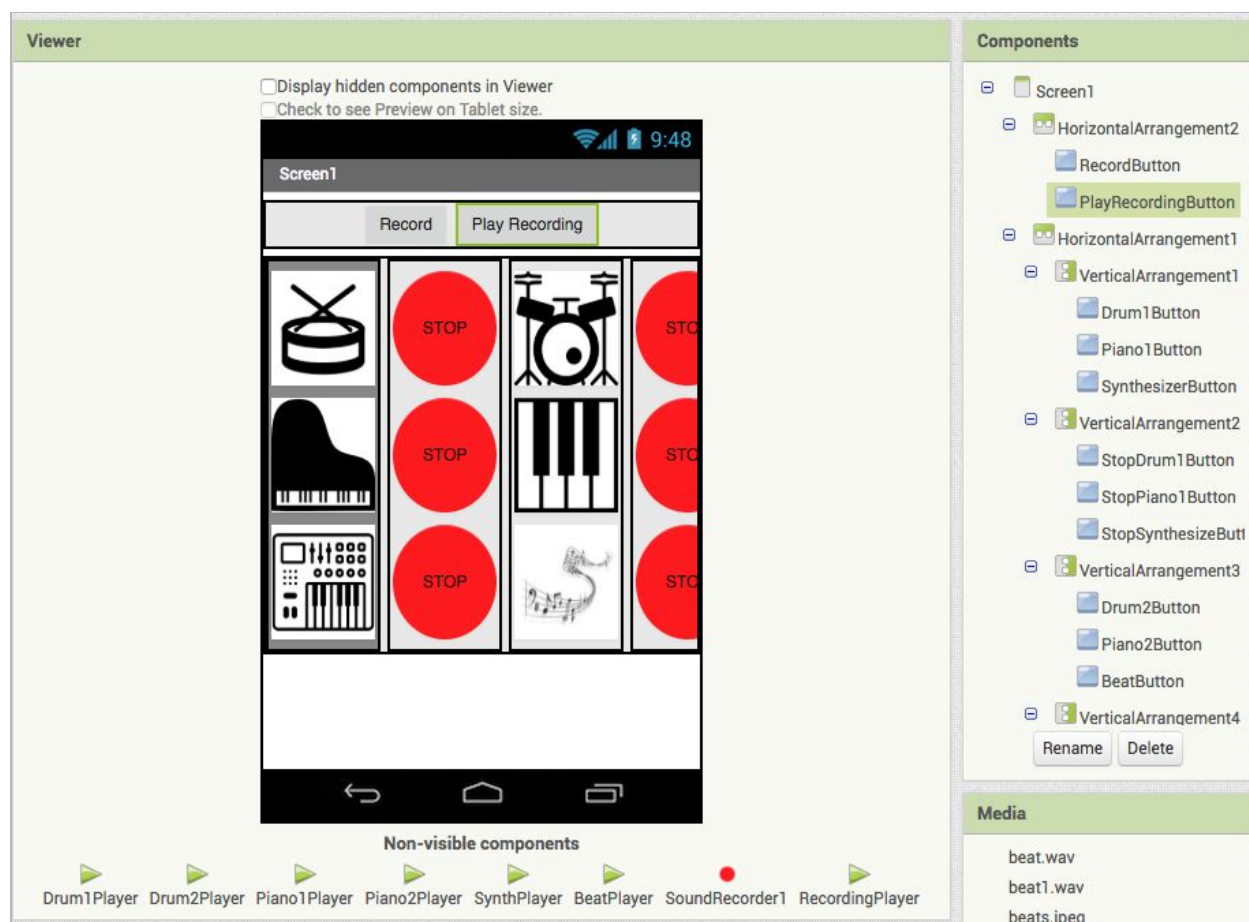
In order to evaluate students' attitude, perception, and understanding towards coding, students are required to finish a 5-point scale survey below by putting a “✓” in the appropriate box.

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

*Because students will design their own apps, the Designer and Blocks code below are just examples. The apps do not need to look just like the examples below.*

### Designer



## Blocks

```
when StopPiano1Button .Click
do
  set Piano1Player . Loop to false
  call Piano1Player .Stop
```

```
when SynthesizerButton .Click
do
  if not SynthPlayer . IsPlaying
  then
    set SynthPlayer . Loop to true
    call SynthPlayer .Start
```

```
when StopPiano2Button .Click
do
  set Piano2Player . Loop to false
  call Piano2Player .Stop
```

```
when Drum2Button .Click
do
  if not Drum2Player . IsPlaying
  then
    set Drum2Player . Loop to true
    call Drum2Player .Start
```

```
when StopBeatButton .Click
do
  set BeatPlayer . Loop to false
  call BeatPlayer .Stop
```

```
when Piano1Button .Click
do
  if not Piano1Player . IsPlaying
  then
    set Piano1Player . Loop to true
    call Piano1Player .Start
```

```
when Piano2Button .Click
do
  if not Piano2Player . IsPlaying
  then
    set Piano2Player . Loop to true
    call Piano2Player .Start
```

## Blocks for Challenge

```
when Drum1Button .Click
do
  if not Drum1Player . IsPlaying
  then
    set Drum1Player . Loop to true
    call Drum1Player .Start
```

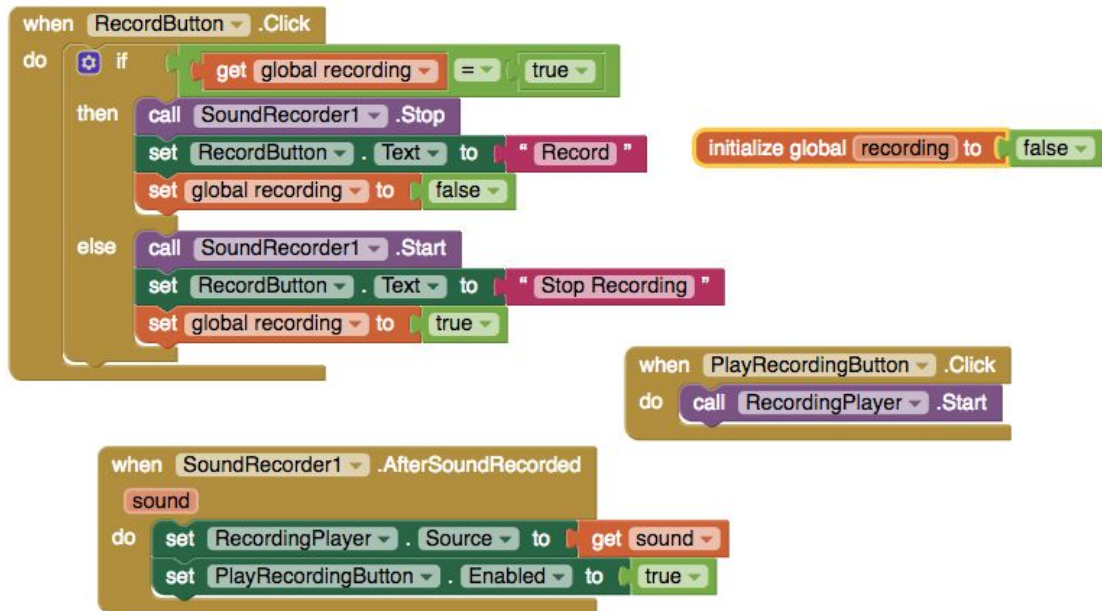
User tapped and released the button.

```
when StopSynthesizeButton .Click
do
  set SynthPlayer . Loop to false
  call SynthPlayer .Stop
```

```
when StopDrum1Button .Click
do
  set Drum1Player . Loop to false
  call Drum1Player .Stop
```

```
when BeatButton .Click
do
  if not BeatPlayer . IsPlaying
  then
    set BeatPlayer . Loop to true
    call BeatPlayer .Start
```





# **Appendix 1**

## **Music Maker Teacher's Guide**

### **Lesson 1**

#### **Learning Objectives**

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

1. Design a layout for an app using design planning skills.
2. Use the Layout components in App Inventor to control placement of components in their apps.
3. Apply the Computational Thinking Practices of reusing and remixing code and being iterative and incremental, by taking the time to plan and design to create their apps.

#### **Lesson Outline**

##### **Introduction of the project (5 minutes)**

Ask students:

1. What did you like about the MyPiano app?
2. How might the MyPiano app be made better, more interesting or fun?
3. Have you tried the GarageBand app before?

Explain to students that they will make their own Music Maker, where they will create an app using various instruments and beat sound files. This will build upon what on their work with the MyPiano app, and adds extended and varied sound files.

Demonstrate an example app, so students can see and hear the sounds can start and stop.

### App Design (15 minutes)

Students design their own apps on paper:

1. Give students access to media files. These can be in a Dropbox or Google Drive or some other online storage used by your school. Allow students to play the sounds so they know what they sound like. Headphones are advised.
2. Give students the *Music Maker Design Worksheet*.
3. Have students draw the layout of their apps. Encourage them to think about the placement of buttons for the ease of starting and stopping/pausing the sound plays.
4. Remind students that their apps should include:
  - a. Minimum of four sounds
  - b. The ability to start and stop any sound

### Adding of Components in Designer (20 minutes)

1. Have students upload the MusicMaker\_template.aia file.
  - a. Show them the media files that are included. These are the same media files they listened to above.
  - b. Demonstrate **HorizontalArrangements** and **VerticalArrangements** as a way to organize components in their app. Placement of components inside an Arrangement is based on the type of Arrangement. Components inside **VerticalArrangements** are placed vertically, below each other. Components inside **HorizontalArrangements** are placed horizontally, beside each other.
2. Based on the layout drawn on the worksheet, ask students to add the necessary components for their apps, following the Student Guide: Part 1.

# Appendix 2

## Music Maker Teacher's Guide

### Lesson 2

#### Learning Objectives

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

1. Reuse and remix code from a previous app.
2. Demonstrate understanding that multiple Player components can create parallel actions.
3. Demonstrate understanding of Events, which trigger actions in an app.
4. Follow a design to implement an app.

#### Lesson Outline

##### Demonstration of Player Components (10 minutes)

1. Ask students: “Is there any difference between the sounds that play in the MyPiano app and this Music Maker app?” (The answer is that in this app, multiple sounds can play at once, so they will need multiple Player components). The concept is *parallelism*, where multiple things can occur simultaneously in an app. While technically, things are not actually being executed at the same time, multiple sounds files can be launched so they play in parallel.
2. Demonstrate how to add Player components.
  - a. Explain the Loop property and compare to loops in code (from Scratch).
3. Demonstrate the Start, Stop, and Pause blocks for the Player component.

### **Coding Activity (25 minutes)**

1. Students code their apps following Student Guide: Lesson 2. Most of this lesson will be similar to the My Piano app, where the user presses a button to start a sound. The addition of a Stop or Pause feature is usually implemented by an additional button. If some students want to use a single button, refer them to the Challenge Student Guide, which explains the process for recording, and also shows a toggle button.
2. If some students finish early, they may try the Challenge, using Student Guide: Challenge. The Challenge adds a Record button so users can record their music and then play it back.

### **Sharing (10 minutes)**

Ask students to swap tablets with a partner, and try each other's apps. Provide feedback using the Feedback Worksheet. Stress to students the need for constructive and thoughtful feedback. They can comment on the user interface or the sounds chosen.

# **Appendix 3**

## **Music Maker 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**

##### **Introduction to Lesson 3 (5 minutes)**

1. Review what has been accomplished so far.
2. Explain that students can work on completing the app if they haven't already, or they can use the time to add more notes or new features. Students may also try the Challenge to create a toggle button to record music.

### **Add New Features (20 minutes)**

Students may continue to work on their apps if they have not yet finished.

Based on suggestions at the end of the Student Guide: Part 2, students can add new features.

### **Sharing (10 minutes)**

Ask students to share with a classmate. They should demonstrate their app, then let the other person try it. Each person should complete a Feedback worksheet for the other person, providing two positive statements and a constructive suggestion for improvement.

### **Wrapup (10 minutes)**

1. Ask students to reflect on creating their own apps. Ask if it felt different from following steps to make a set app.
2. Ask students to complete the multiple choice questions and Learning Attitudes Survey.