# TECHNOLOGY CAMP

## DAY 2 : MOBILE APP DEVELOPMENT

# Building Mash A Mole App

## Session 4

**YELLOW CIRCLE INC**
PO Box 2383
Elk Grove, CA 95759-2383

**Teacher Lesson Plan**

## Session Name:

*Building Mash A Mole App*

## Summary:

*This course teaches the basics of mobile app development using the App Inventor development tool.*

## Time Allotment:

*65 minutes*

## Learning Objectives:

- *Utilize the App Inventor Designer to create a mobile app user interface*
- *Utilize the App Inventor Blocks Editor to program the behaviors for a mobile app*
- *Test the application using either an Android phone*
- *Modify the created app to change the user interface and/or programmed behavior*

## Supplies:

- *Scrap paper / notepad to take notes*
- *Android tablet*
- *Laptop / computer with Internet access*

## Learning Activities:

- **(5 minutes) - Session overview**

  During this lesson, we will continue working with the App Inventor development tool. We will build a mobile app together and then you will use what you have learned to modify and customize the app.

  - *Review App Inventor Tool*
    - *Designer & Blocks Editor*

- *Coding Concepts - Review Development Blocks*
  - *Event Handlers*
  - *Commands*
  - *Expressions ("getter and setter" blocks)*
- *Coding Concepts - New*
  - *Declaring and working with Variables*
  - *Creating and calling Procedures*
  - *Sprite placement and layers*
  - *Timer use*
- *Building in App Inventor*
  - *Mash A Mole Project*

- **(2 minutes) App Inventor Designer & Blocks Editor**
  - *Designer*
    - *Used for designing how the app will look to the user*
    - *The app "user interface" - allow the user to interact with the app*
    - *This is also where you add non-visible elements and user input features*
    - *Media files*
    - *Sensors - (e.g. AccelerometerSensor for shake inputs)*
    - *Clock - i.e. Timer*
  - *Blocks Editor*
    - *Used for designing the app behavior*
    - *The app "programming"*

- **(8 minutes) Coding Concepts - Review Development Blocks**
  - *Event Handlers*
    - *The brown blocks are "event handlers" that are used to determine how the phone responds to user input (e.g. opening the app, click,*

*shake)*
    - *In App Inventor, "event handler" blocks begin with the word "when"*
- *Commands*
    - *The purple blocks are "command" blocks that fit within"event handlers". The "command" blocks indicate the action that happens in response to the user input.*
    - *Command blocks can be used to create "procedures" - a action or group of actions that can be called by name to create a program action*
- *Expressions*
    - *The green blocks are expressions that either get or set the current value of a property (aka getter or setter blocks)*
    - *"getter" blocks establish (contain) the value of a component (e.g. FontBold)*
    - *"setter" blocks change the value of a property*
    - *They are formatted as "set" something "to" some value (e.g. set Button1.Text to "Hello")*
- *Variables*
    - *The orange blocks are used to work with "variables"*
    - *Item that stores a value that is used or acted upon by program conditions or information*
- *Lists*
    - *The light blue blocks are "list" blocks. Lists can be used directly or as part of variables*
- *Math*
    - *The dark blue blocks are used to perform math calculations that act in conjunction with other blocks*
    - *When fit together, the blocks can be read to describe the user and phone*

- **(60 minutes) - Group Activity : Build Mash A Mole**

**** Switch from slide show to Instructor screen share ****

Start New Project Build Mash A Mole

- *Login to App Inventor*
  - *ai2.appinventor.mit.edu*
- *Select "Start new project" (1)*
- *Enter Project name (2) and click OK (3)*


- **(20 minutes) - Group Activity : Build Mash A Mole App User Interface**
  - *Build the UI in Designer*

Create the following components by dragging them from the Palette into the Viewer:

| *Component Type* | *Palette Group* | *What you'll name it* | *Purpose of Component* |
|---|---|---|---|

| | | | |
|---|---|---|---|
| ***Canvas*** | *Drawing and Animation* | *MoleYard* | *"Field" where holes are located* |
| ***ImageSprite***<br>*Select and drop 5 on MoleYard* | *Drawing and Animation* | *Hole1 - Hole5* | *Holes where mole "pops up"* |
| ***ImageSprite*** | *Drawing and Animation* | *Mole* | *The mole* |
| ***HorizontalArrangement*** | *Layout* | *HorizontalArrangement1* | *To display the score* |
| ***Label*** | *User Interface* | *ScoreTextLabel* | *To hold "Score: "* |
| ***Label*** | *User Interface* | *ScoreValueLabel* | *Holds t score (# of times the mole was hit)* |
| ***Clock*** | *User Interface* | *MoleClock* | *mole movement control* |
| ***Sound*** | *Media* | *Buzz* | *Enables vibrate on mole touch* |

Add Media Files

- *Click Upload File button in the Media Pane to open Upload File Window (1)*
- *Click Choose File button and select image and sound files from ProjectFiles Folder (2)*
- *Click OK to upload media (3) - Repeat steps 2 and 3 to upload all media files*


Set the Component Properties

- *Make the following changes to the components' properties:*

| ***Component*** | ***Action*** |
|---|---|
| *Screen1* | *Set:* ***AboutScreen*** *to "Mash A Mole"* ***Icon to*** *"mole.png"* ***Title to*** *"Mash A Mole"* |
| *MoleYard* | *Set* ***BackgroundColor*** *to Green. Set* ***Width*** *to 320 pixels. Set* ***Height*** *to 320 pixels.* |
| *Hole1* | *Set* ***X*** *to 20 and* ***Y*** *to 60 (upper left).* |
| *Hole2* | *Set* ***X*** *to 130 and* ***Y*** *to 60 (upper center).* |
| *Hole3* | *Set* ***X*** *to 240 and* ***Y*** *to 60 (upper right)* |

| | |
|---|---|
| *Hole4* | *Set **X** to 75 and **Y** to 140 (lower left).* |
| *Hole5* | *Set **X** to 185 and **Y** to 140 (lower right).* |
| *Mole* | *Set **Picture** to "BigMole.png". Set **Z** to 2 so the mole appears in front of the other **ImageSprite** s, which have the default **Z** value of 1.* |
| *ScoreTextLabel* | *Set **Text** to "Score: ".* |
| *ScoreTextValue* | *Set **Text** to "0".* |

*About Sprites [Hole1-Hole5 and Mole]*

- *Explain Sprite setting for **X** is the horizontal position of the sprite [picture] on the screen*
- *Explain Sprite setting for **Y** is the vertical position of the sprite [picture] on the screen*
- *Explain that Sprite setting for **Z** is the front to back position of the sprite [picture] - positions move from low to high. Images with a **Z** value of 2 will be placed over those with a Z value of to at the same **X** and **Y** settings*

*Don't worry now about setting the **Picture** property for the holes; we'll set the property in the Blocks Editor.*

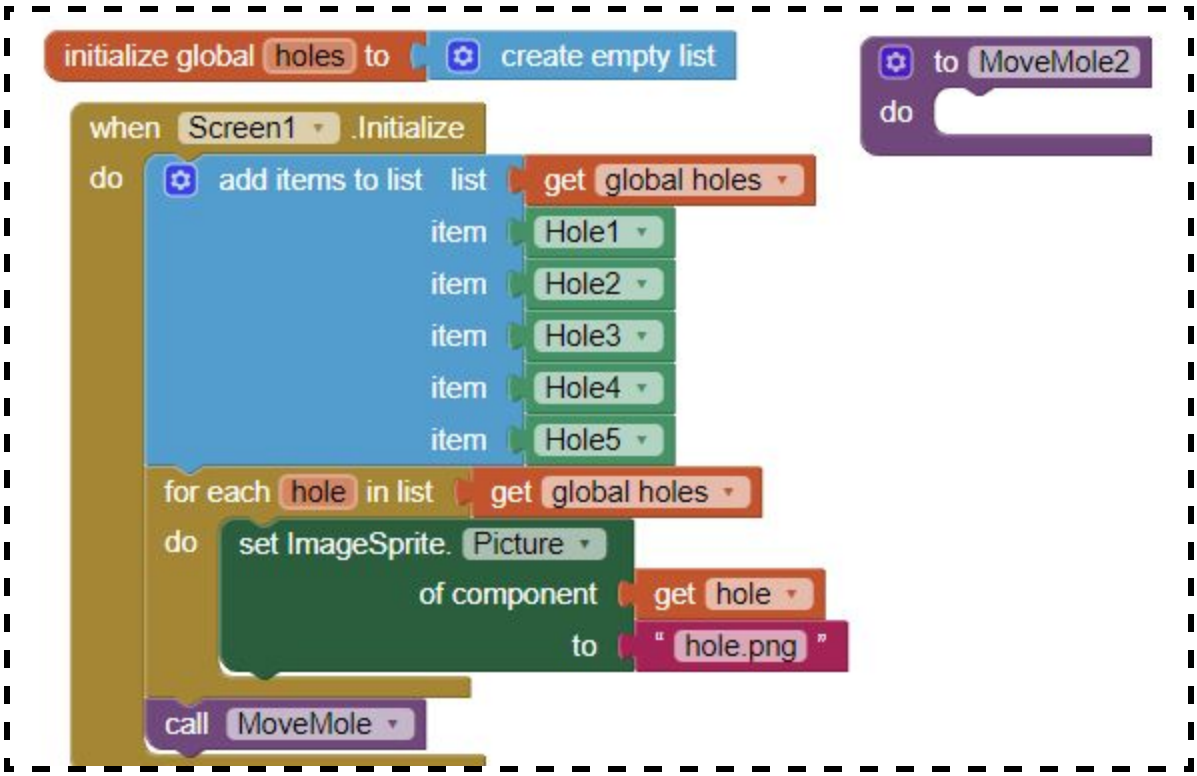
Build the App Behavior in Blocks Editor

- *Create global variable "holes"*
  - *For now, we will give it a "dummy" initial value of an empty list; we'll set the real initial value in the Screen1.Initialize event handler, which gets executed each time the app loads the screen.*
  - *Note: components cannot be referred to in variable initialize blocks, which are run before the app has started.*

| *Block Type* | *Drawer* | *Purpose* |
|---|---|---|
| *initialize global [name] to* | *Variables* | *Create global variable “holes”* |
| *create empty list* | *Lists* | *Holes where mole “pops up”* |

App Startup

- *The first event to occur in any program is Screen1.Initialize, so we will put start-up code in that handler.*
  - *We will add the hole components to the list holes*
  - *Set each hole's **Picture** property to "hole.png"*
  - *Call MoveMole. procedure*
  - *Note: Since we have not yet written MoveMole, we will create an empty procedure with that name, which we will fill in later.*

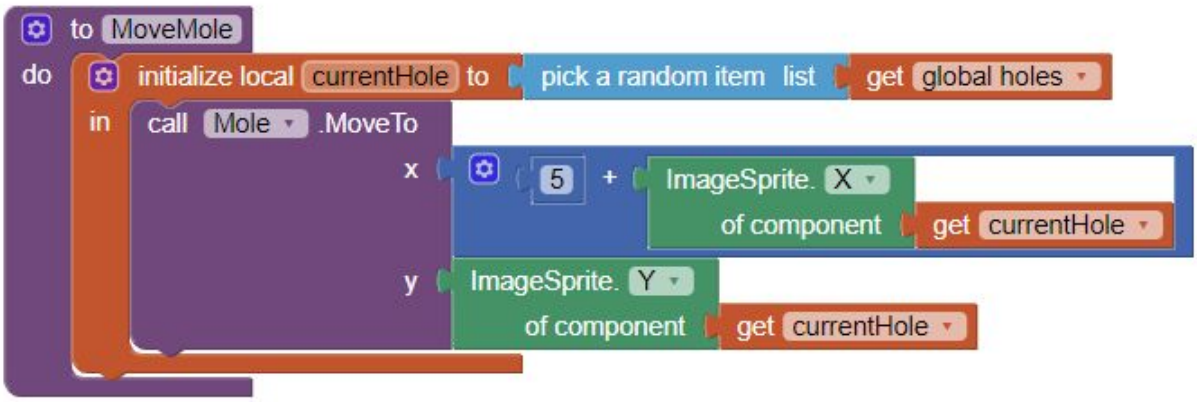



| *Block type* | *Drawer* | *Purpose* |
|---|---|---|
| *Screen1.Initialize* | *Screen1* | *Specify what should happen when the app starts.* |
| *add items to list* | *Lists* | *Add the following values to...* |
| *get global holes* | *Variables* | *...the list of holes:* |
| *Hole1* | *Hole1* | *-the upper left hole* |
| *Hole2* | *Hole2* | *-the upper center hole* |

| | | |
|---|---|---|
| *Hole3* | *Hole3* | *-the upper right hole* |
| *Hole4* | *Hole4* | *-the lower left hole* |
| *Hole5* | *Hole5* | *-the lower right hole* |
| *for each hole in list* | *Control* | *Specify that we would like a variable named "hole"....* |
| *get global holes* | *Variables* | *...to take on each of the values in the list holes .* |
| *set ImageSprite.Picture of component ... to* | *Any ImageSprite* | *Set the Picture property of...* |
| *get global hole* | *Variables* | *...the ImageSprite referred to by the variable hole ...* |
| *" " (hole.png)* | *Text* | *...to the picture of the empty hole.* |
| *to procedure (MoveMole)* | *Procedures* | *Create an procedure, to be filled in later, for moving the mole.* |
| *call MoveMole* | *Procedures* | *Call MoveMole to make the first placement of the mole.* |

Game Play - Move Mole

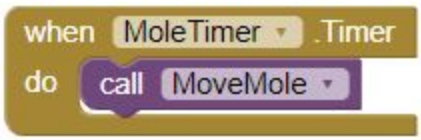
- *Build the MoveMole. procedure*
  - *Randomly select one of the holes [currentHole] from the global holes list*
  - *Move the mole to the position of the selected [currentHole] hole*
  - *Note: Type value or select dropdown as shown below:*




| *Block type* | *Drawer* | *Purpose* |
|---|---|---|
| *initialize local currentHole to*<br>*(there are two types of 'initialize local': take the one that fits the procedure block)* | *Variables* | *Save the...* |
| *pick a random item* | *Lists* | *...randomly selected...* |
| *get global holes* | *Variables* | *...hole.* |
| *call Mole.MoveTo* | *Mole* | *Move the mole to the...* |
| *ImageSprite.X* | *Any component > Any ImageSprite* | *..x-coordinate of...* |
| *get currentHole* | *Variables* | *...the chosen hole...* |
| *ImageSprite.Y* | *Any component > Any ImageSprite* | *...and the y-coordinate of...* |

| | | |
|---|---|---|
| *get currentHole* | *Variables* | *...the chosen hole.* |

- ***Set when the mole moves (i.e. when the MoveMole. procedure is called)***
  - *Select the MoleClock.Timer event handler (recall the TimerInterval value was set in the Desiginer MoleTimer Component Properties)*
  - *Insert the call MoveMole procedure to move the mole each time the TimerInterval is reached*



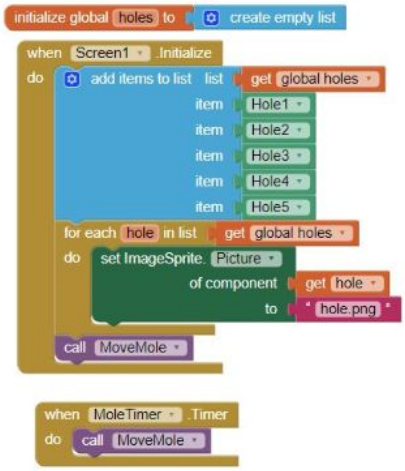

| ***Block type*** | ***Drawer*** | ***Purpose*** |
|---|---|---|
| *MoleClock.Timer* | *MoleClock* | *When the timer goes off...* |
| *call MoveMole* | *Procedures* | *...move the mole.* |

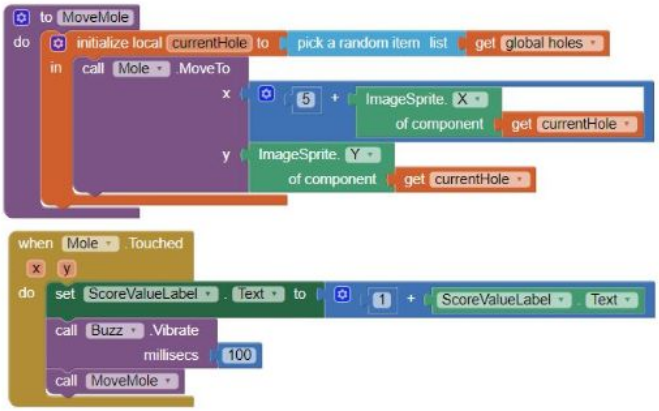
Game Play - Register Action & Update Score

- *Register Mole Mash (when mole is touched)*
  - *Select Mole.Touched event handler to direct actions when a mole is touched*
  - *Select the green "setter" block to set the score value*
  - *Use Math blocks to increment the ScoreValueLabel.Text by 1*
  - *Select a purple command block to call buzz.vibrate sound component*
  - *Use a Math block to set the vibrate duration to 100 milisecs*
  - *Select a purple command block to call MoveMole procedure to move the mole to the next hole*


| *Block type* | *Drawer* | *Purpose* |
|---|---|---|
| *Mole.Touched* | *Mole* | *When the mole is touched...* |
| *set ScoreValueLabel.Text to* | *ScoreValueLabel* | *...update the visible score to...* |
| *+* | *Math* | *...the result of adding...* |
| *1* | *Math* | *...1 [and]...* |
| *ScoreValueLabel.Text* | *ScoreValueLabel* | *...the previous score.* |
| *call Buzzer.Vibrate* | *Buzzer* | *Make the phone vibrate for...* |
| *100* | *Math* | *...100 milliseconds.* |
| *call MoveMole* | *Procedures* | *Move the mole to a new location.* |

Game Complete - Full Blocks Editor View



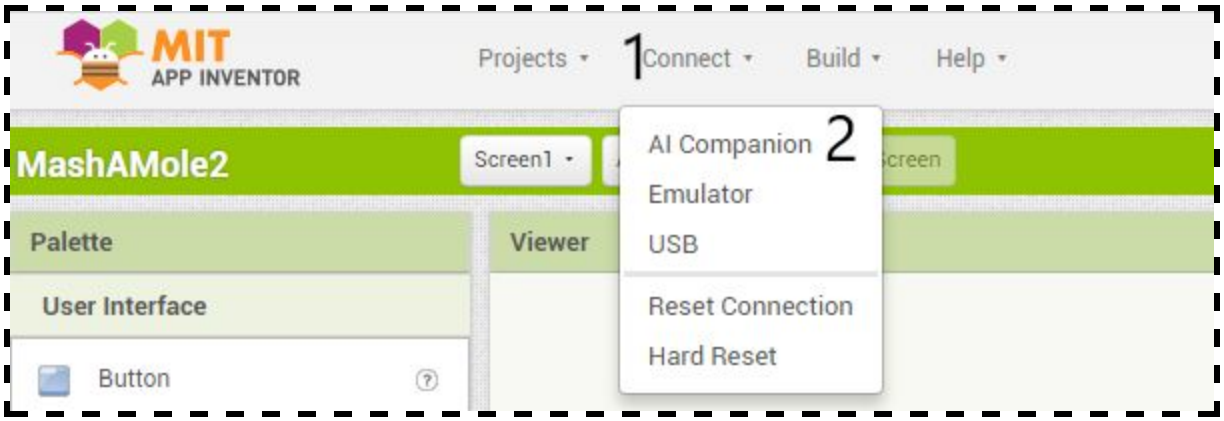

- **(5 minutes) - Group Activity : Test Mash A Mole App**

  Test using QR code

  - ***On your phone/tablet:*** *Open the MIT AI2 Companion App*


  - ***On your computer:*** *Select* ***Connect*** *(1) dropdown*

- *Select **AI Companion** (2) wait while system compiles and packages the code (progress bar may be displayed)*




- *When packaging is complete, QR code will be displayed.*

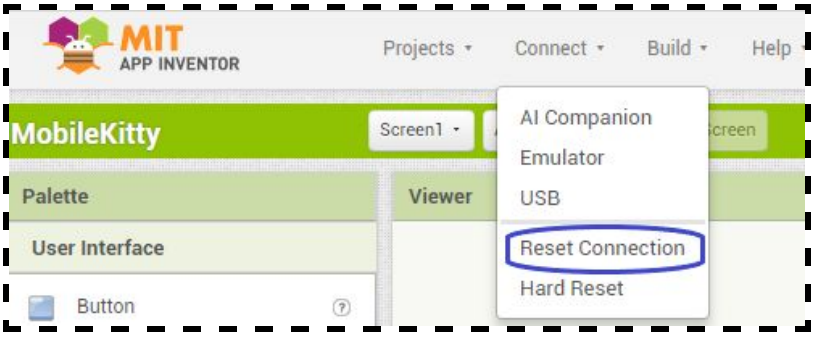

**On your phone/tablet:** Select scan QR code (1) and scan.

- *Note: If you do not want to use the QR scanner, you may select the connect with code button (orange) and enter the six character code (below "Your code is:" from the Connect to Companion window) in the app text box*


- ***On your phone/tablet:*** *The app you created will run. Test all functions and when done, use the back arrow to exit.*
- ***On your computer:*** *Click cancel to clear QR code window popup.*
- *Make any necessary changes and continue the process until you are satisfied with your results.*

**Note:** If re-packaging and re-testing, causes the the AI Companion dropdown to stop responding (grey out) complete the following steps to reset*:*

- ***On your phone/tablet:*** *Stop the app from the "3dot" menu in the upper right corner and select Stop and Exit from the pop-up window*
- ***On your computer:*** *From the Connect dropdown select Reset Connection*



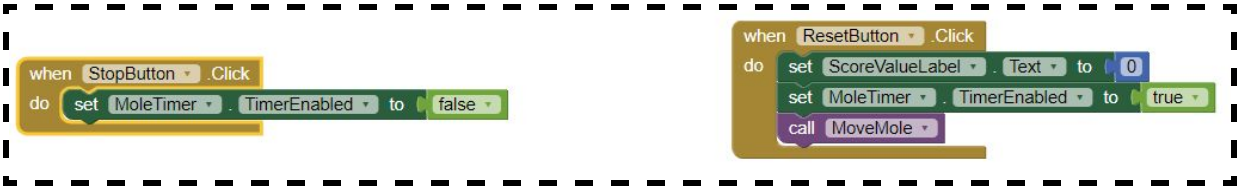
- **(30 minutes) - Student Activity : Modify & Test Mash A Mole App**

  This is student time to use App Inventor to make the Mash A Mole app their own.

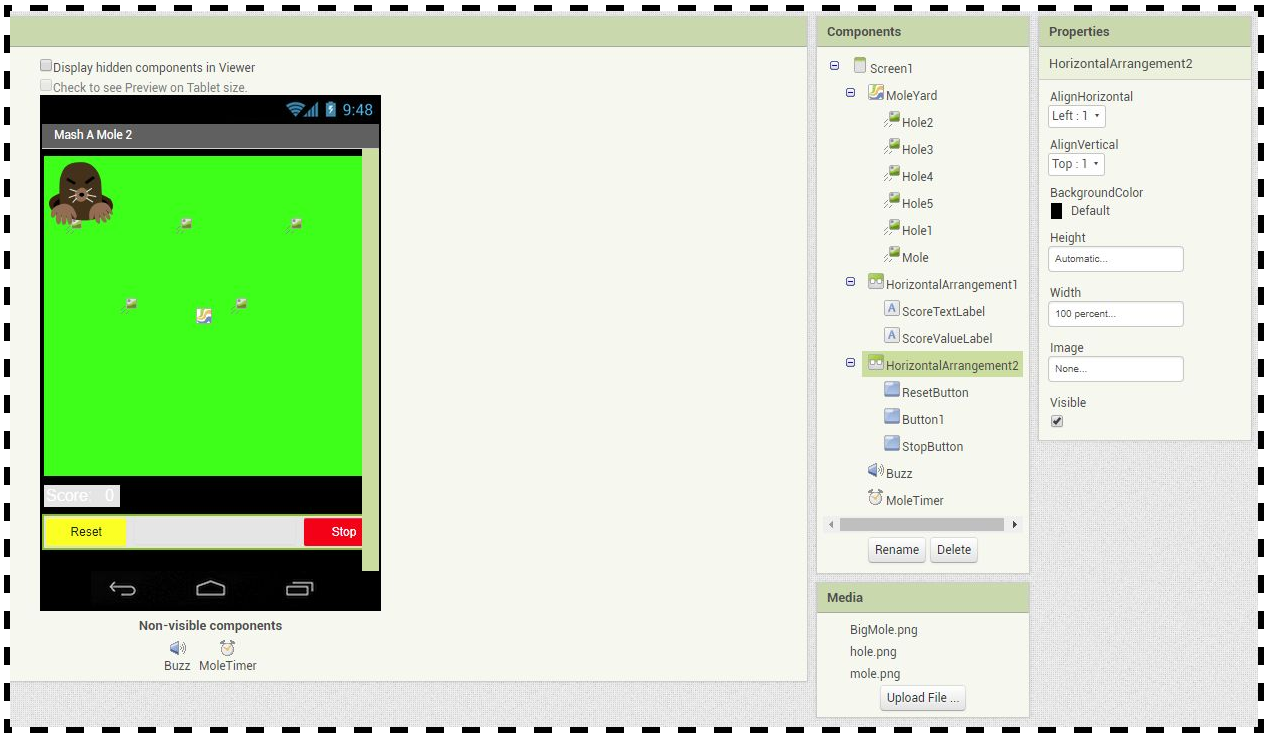
  - *Encourage students to change the appearance & behaviors of the app*
    - *Modify colors, text, images*
    - *Add buttons to enable additional features*
    - *Add Reset Button*
    - *Add Stop Button*
    - *Add sound to vibration*
    - *Add additional holes*
    - *Change Icon Image*
    - *Change App Name*
  - *Have students test their new apps for functionality & share with class if time allows*

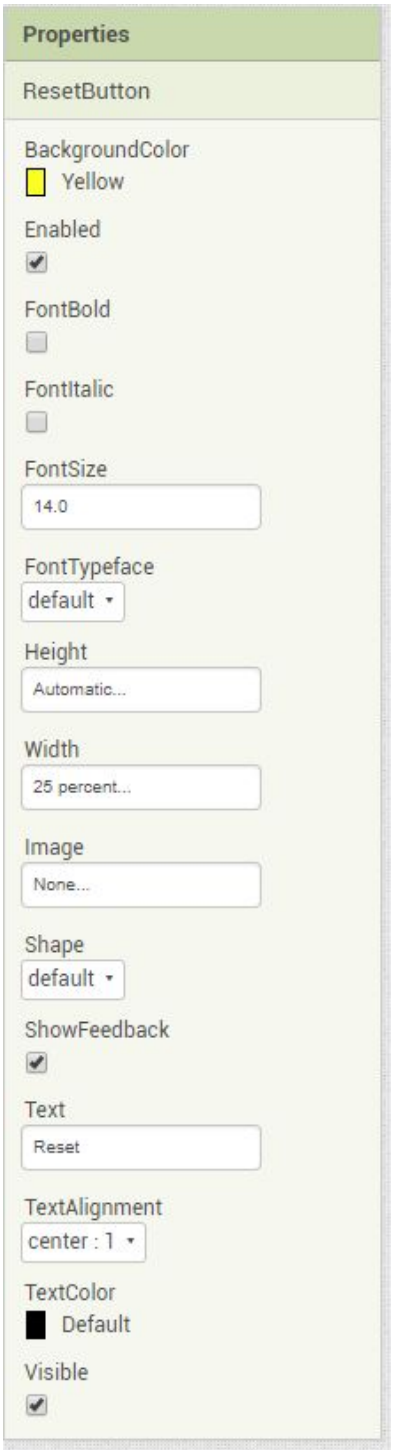
- **(10 minutes) - Design Options**

  Designer and Block Editor for Stop and Reset Buttons



# Building Mash A Mole App




Properties
ResetButton
BackgroundColor
Yellow
Enabled
FontBold
FontItalic
FontSize
14.0
FontTypeface
default
Height
Automatic...
Width
25 percent...
Image
None...
Shape
default
ShowFeedback
Text
Reset
TextAlignment
center : 1
TextColor
Default
Visible


Properties
Button1
BackgroundColor
None
Enabled
FontBold
FontItalic
FontSize
14.0
FontTypeface
default
Height
Automatic...
Width
50 percent...
Image
None...
Shape
default
ShowFeedback
Text
TextAlignment
center : 1
TextColor
Default
Visible


Properties
StopButton
BackgroundColor
Red
Enabled
FontBold
FontItalic
FontSize
14.0
FontTypeface
default
Height
Automatic...
Width
25 percent...
Image
None...
Shape
default
ShowFeedback
Text
Stop
TextAlignment
center : 1
TextColor
White
Visible

- **(1 minutes) - What's next?**

  *Inform students that they are finished with Mobile App Development day of the camp. Students can head back to the cafeteria to be picked up by their parents, and remind them to be back tomorrow for another fun day of learning. Day 3 is all about Cloud Computing.*

**This page is intentionally left blank**