



**Technovation**  
**CANADA**



Technovation 2018 Hack Day

# Hack Day Trainer: Ahmed

## Career pathway through Engineering

- ➔ B. Eng. in Software Engineering  
Lakehead University
- ➔ Project Management Office  
IBM (Automation, Legal Deliverables...)
- ➔ M. Eng. in Technology Innovation Management  
Carleton University
- ➔ Cybersecurity Research and Development  
Global Cybersecurity Resource.

**4th year at Technovation!**

# Welcome to Technovation

- ➡ Technovation will teach the skills you need to emerge as tech entrepreneurs and leaders.
- ➡ Working in teams of 2-5 you identify a problem in your community, and build a technology business to solve it.
- ➡ Technovation takes you through 4 stages of launching a mobile app startup, inspired by the principles of design thinking:
  - Ideation** - Identify a problem in the community
  - Technology** - Develop a mobile app solution
  - Entrepreneurship** - Build a business plan to launch the app
  - Pitch** - Take the business to market

# What we are covering today

1. What you will need for today
2. Problem solving exercises
3. Application development cycle
4. Setting up your phone
  - Installing AI Companion
5. Tips for mobile development
6. Creating your first app “Talk to Me” (App 1)
  - Tutorial walkthrough
  - Downloading application
  - Save and distribute
7. -----Lunch-----
8. Slide show (App 2)
9. Colored Dots (App 3)
10. Continue Learning
11. Technovation: Next Steps
12. Math App Challenge

# Getting Started

## What you need:

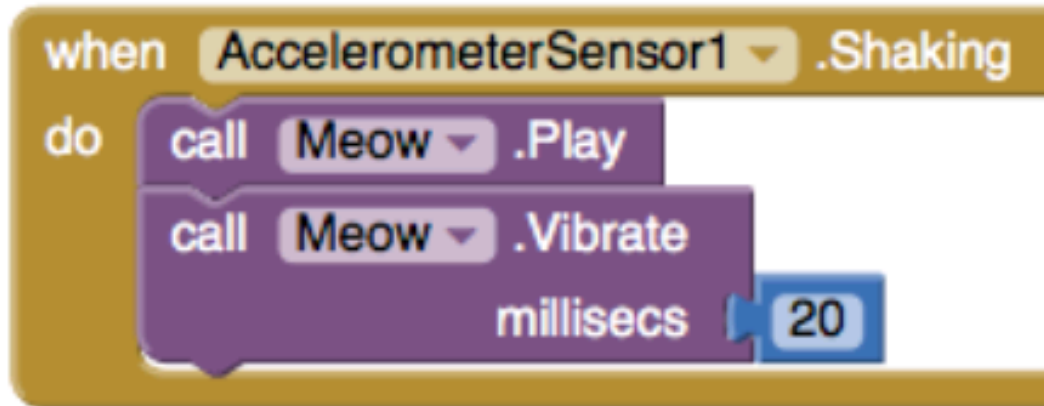
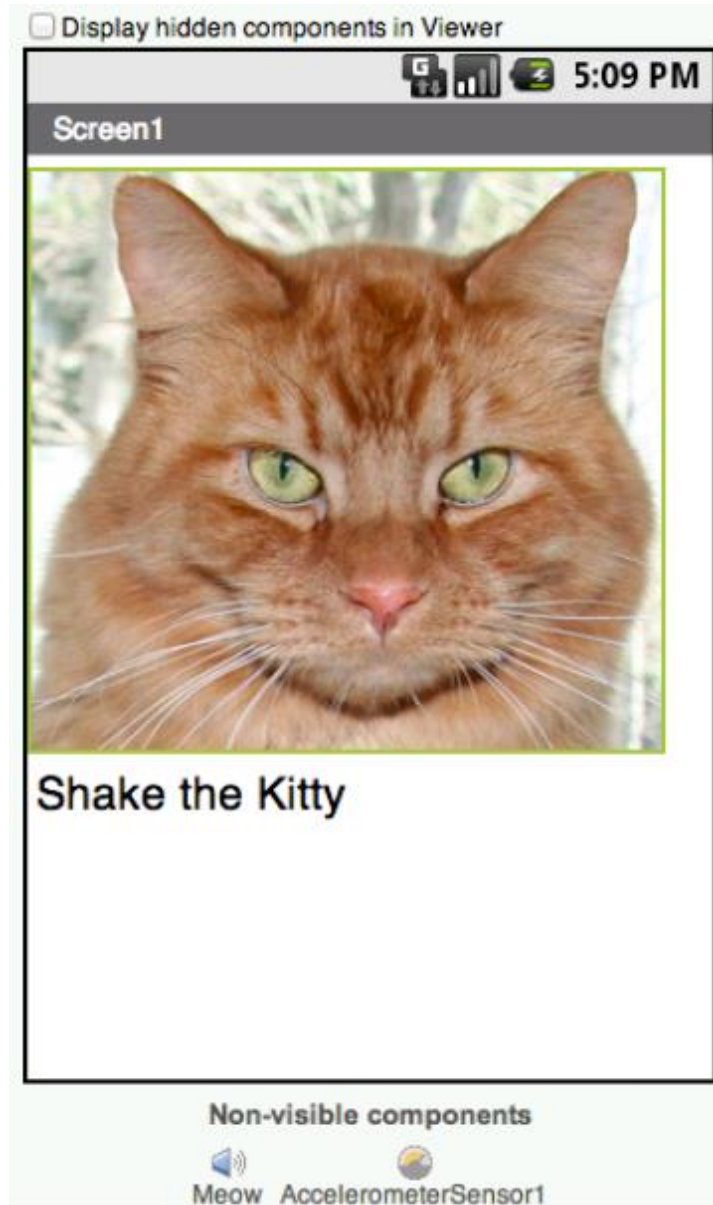
- Internet access
- Laptop (Windows or Mac)
- Gmail account
- Computer with Firefox 3.6/Chrome 4.0/ Safari 5.0 web browser
- Android phone or tablet with OS 2.3 or higher

# Why Learn App Inventor?

- ➡ Create your own mobile applications
- ➡ Use it for school projects
- ➡ Solve real world problems
- ➡ Create a prototype for your business

# Problem solving exercises

# What Does This App do?

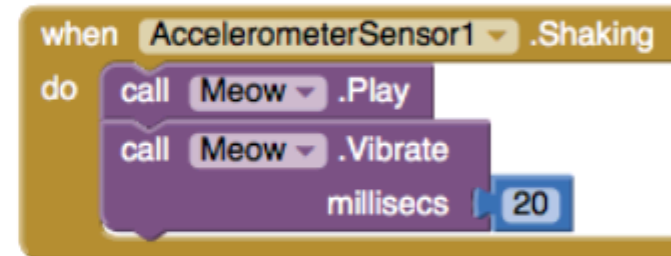
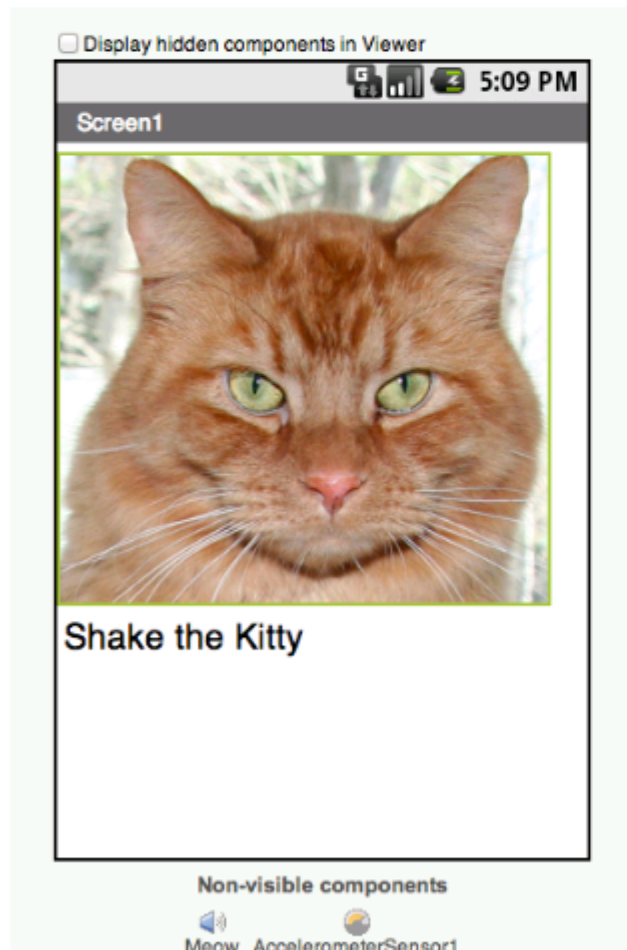






# Shaking Phone

*Make something happen when you shake your phone.*



The **AccelerometerSensor.Shaking** event will detect when the phone is shaking and then the Meow sound will play and the phone will vibrate for 20 milliseconds.

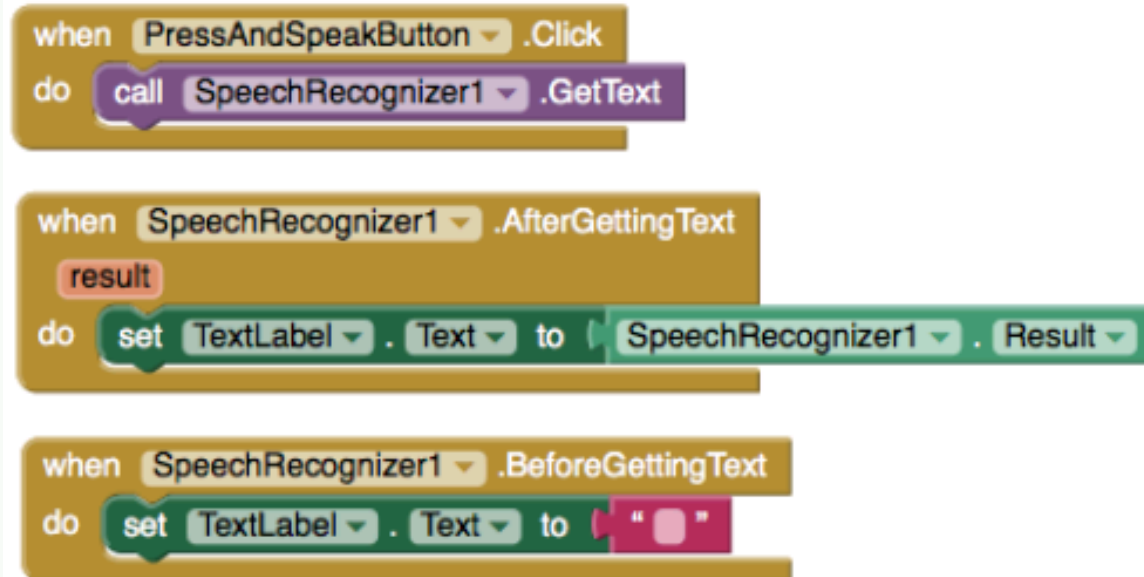


# What Does This App do?



Non-visible components

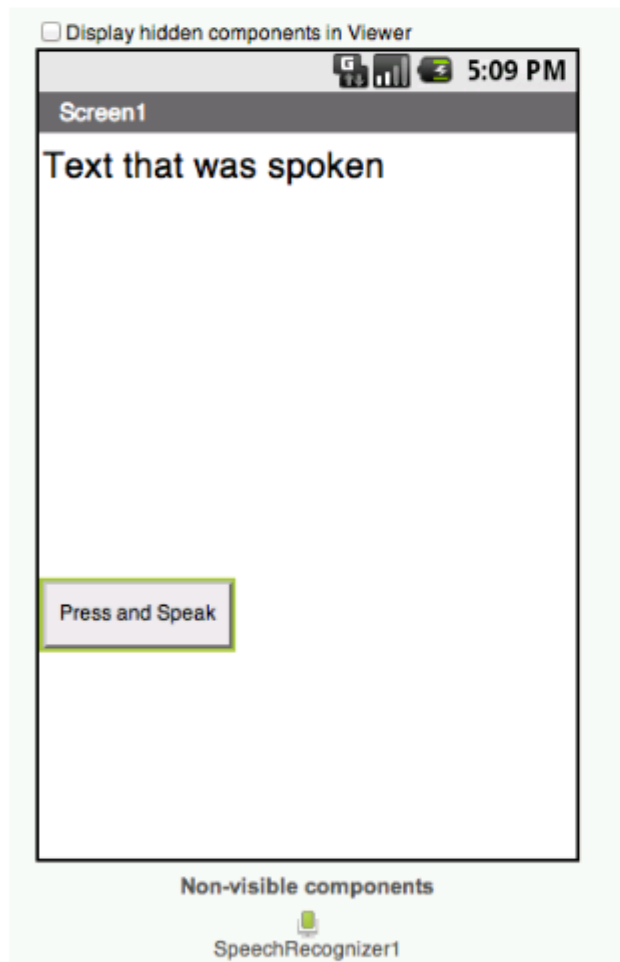
SpeechRecognizer1





# Speech Recognition

*Display the text of what is being said on the phone screen.*



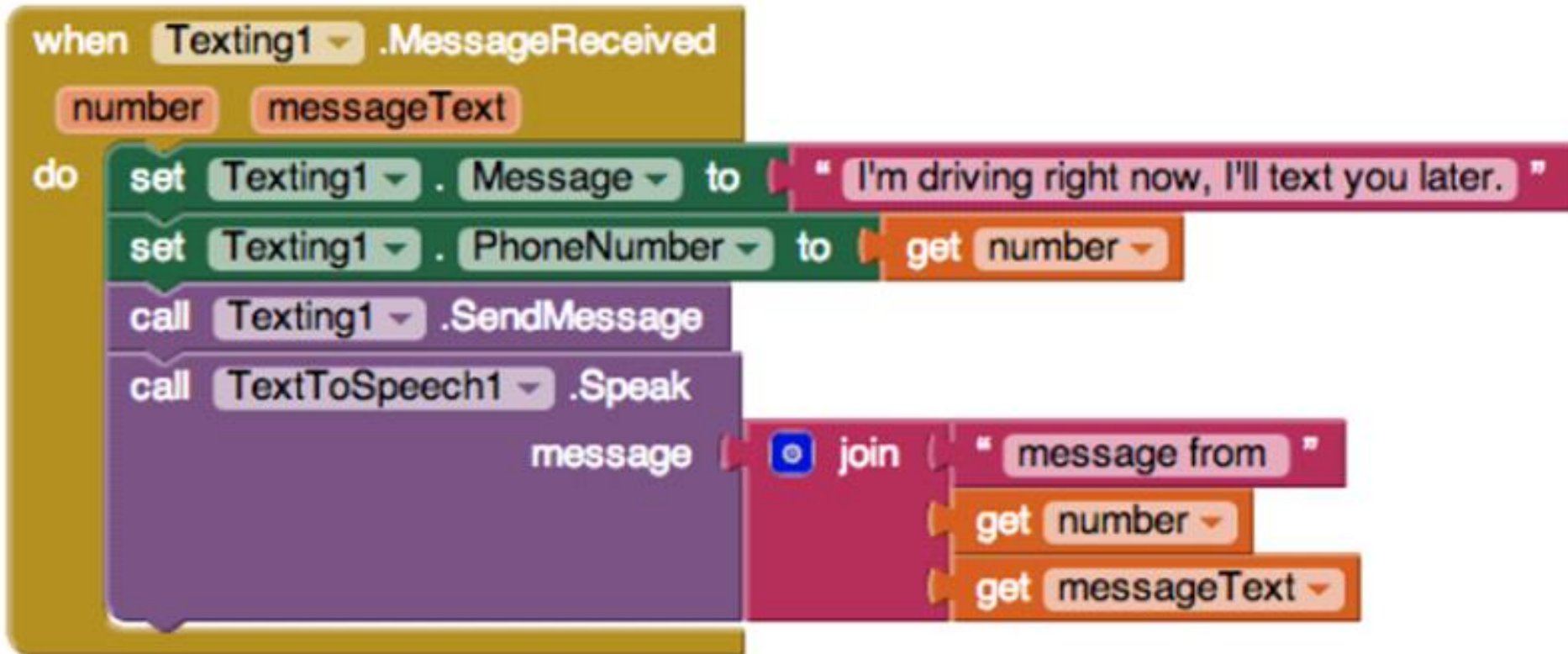
When the **PressAndSpeakButton** is clicked the **SpeechRecognizer** event is called and is ready for you to speak.

The **BeforeGettingText** event will be triggered before speech has been received and recognized. Then the **Label** will display no text on the screen.

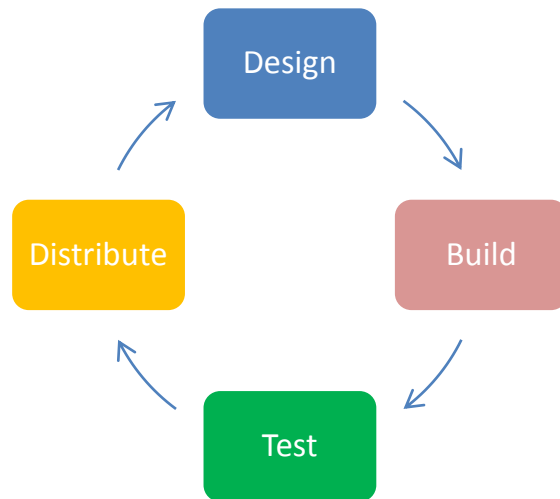
The **AfterGettingText** event will be triggered once speech has been received and recognized. Then the **Label** will display the text on the screen.

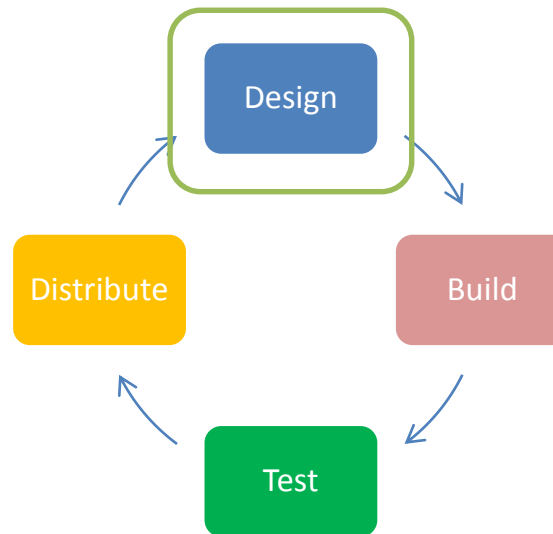
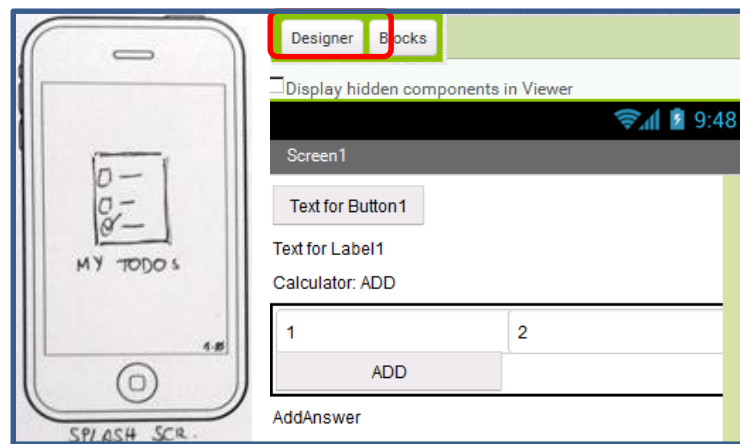


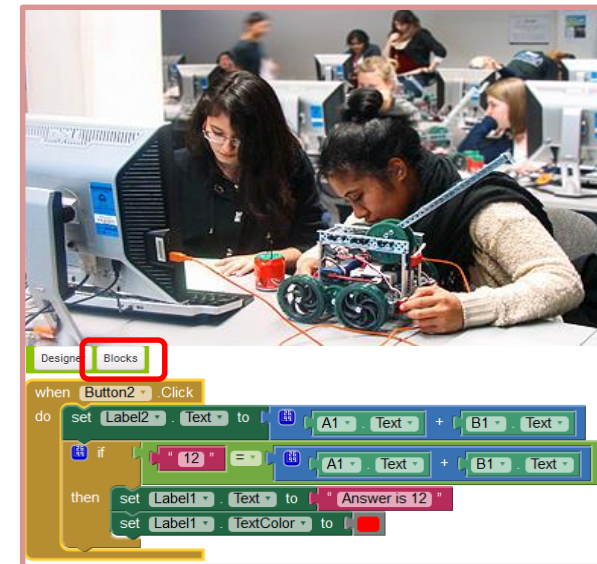
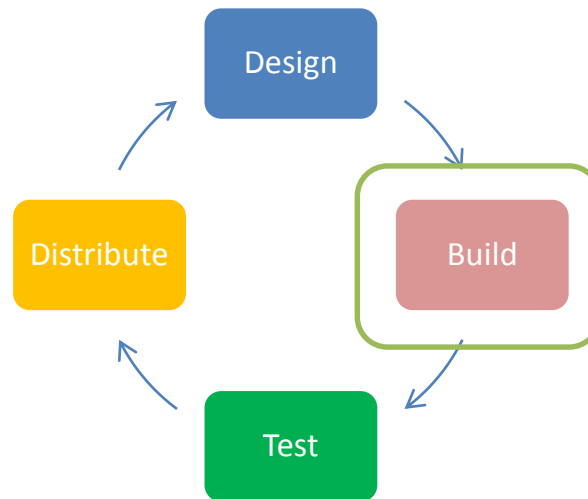
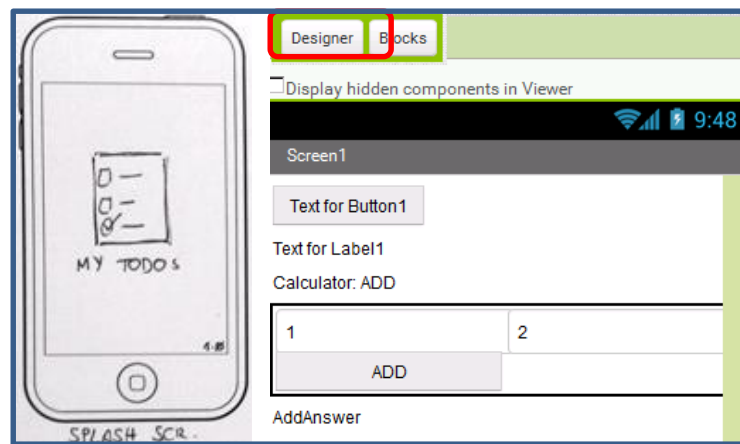
# What Does this App do?



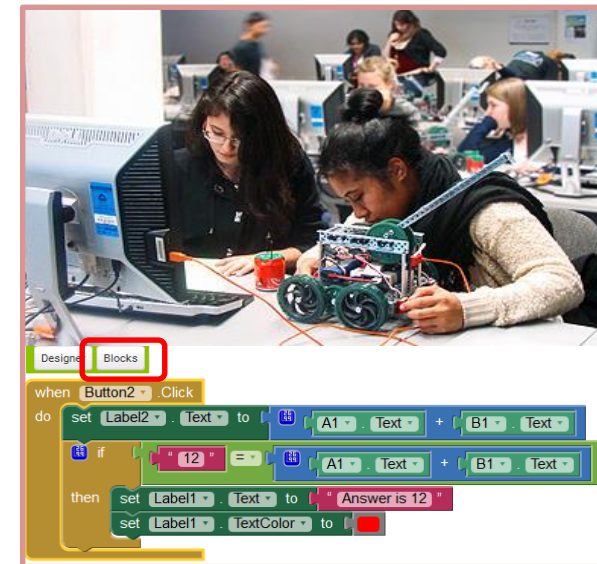
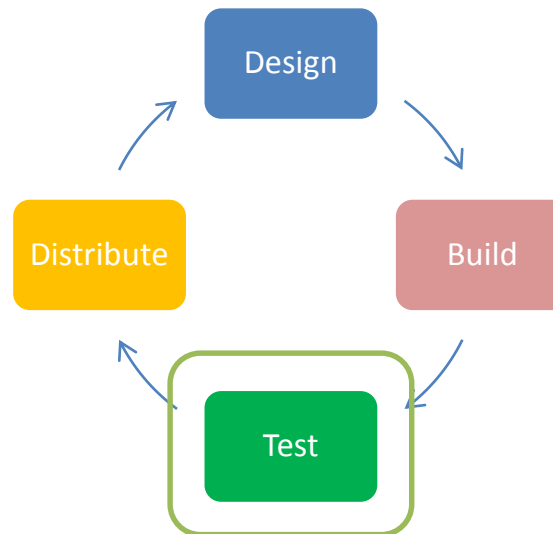
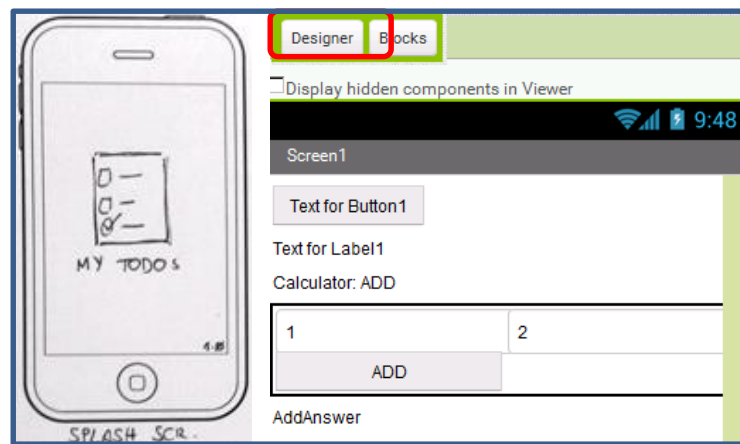
# Process for creating apps

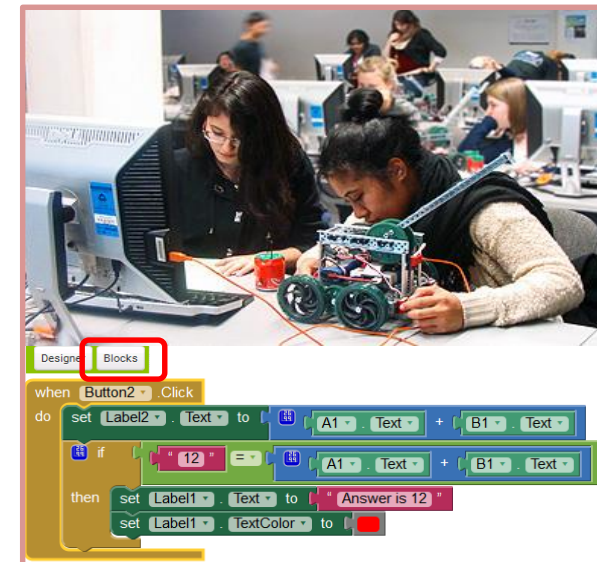
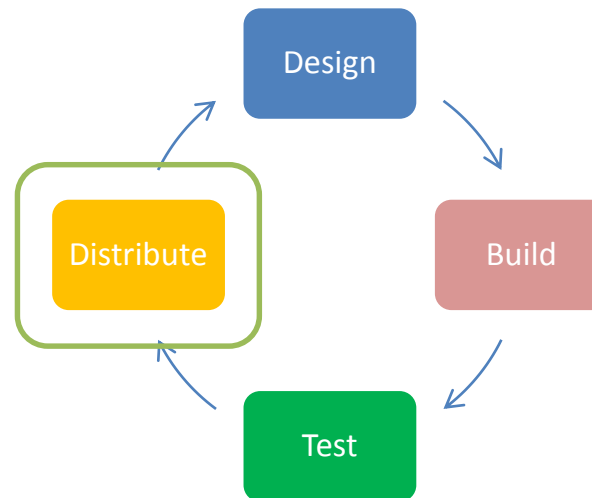
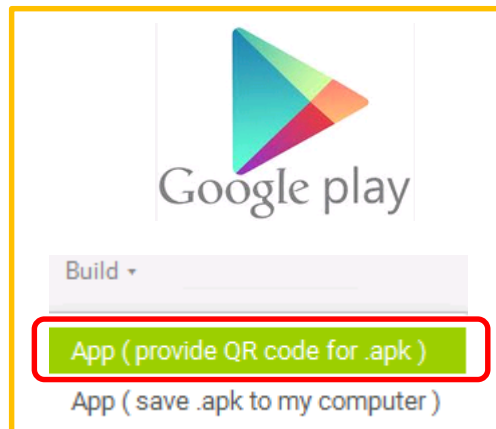
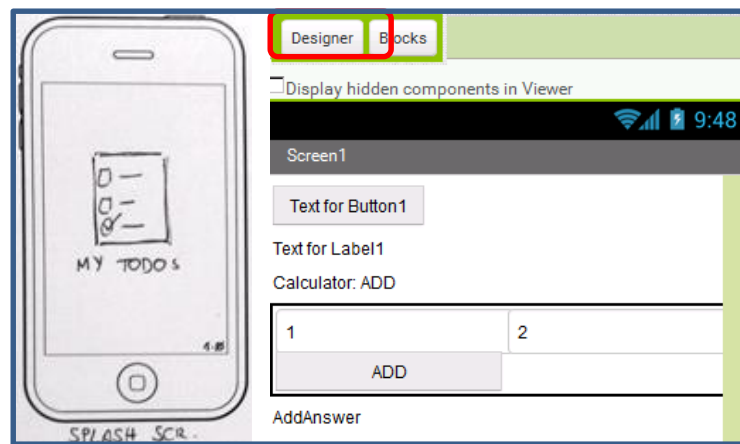












# Attention to Detail is Important

➡ “NameOne” is not the same as “name1”

# Setting Up Your Phones

# Getting Started

## What you need:

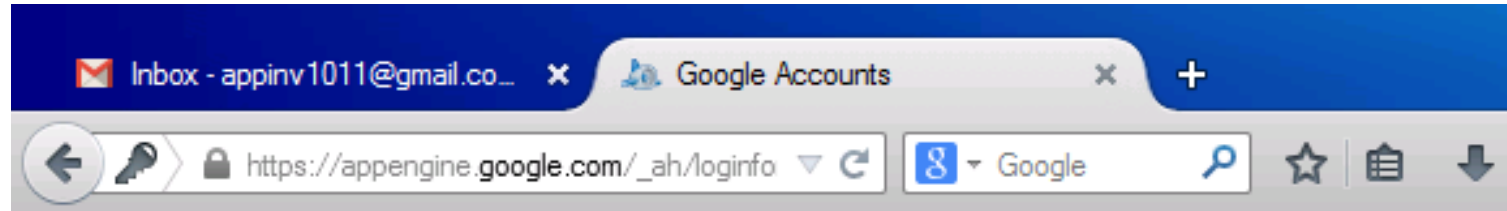
- Internet access
- Laptop (Windows or Mac)
- Gmail account
- Computer with Firefox 3.6/Chrome 4.0/ Safari 5.0 web browser
- Android phone or tablet with OS 2.3 or higher

# Log in to Gmail

- ➡ Go to Google and click on Sign In
- ➡ If you have an account sign in now
- ➡ Otherwise, click on New Account and create one

Go to App Inventor

**ai2.appinventor.mit.edu**



## Google accounts

**The application MIT AppInventor Version 2 is requesting permission to access your Google Account.**

**Please select an account that you would like to use.**

📧 appinv1011@gmail.com

Google is not affiliated with the contents of **MIT AppInventor Version 2** or its owners. If you sign in, Google will share your email address with **MIT AppInventor Version 2** but not your password or any other personal information.

**Allow**

**No thanks**

[Sign in to another account](#)

☒ Remember this approval for the next 30 days



## Welcome to MIT App Inventor 2

Welcome to the nb146j Release!

Read the [Release Notes](#) for more information.

**This release uses Companion version 2.35**

---

Got an Android phone or tablet? Find out how to  
[Set up and connect an Android device.](#)

Don't have an Android device? Find out how to  
[Set up and run the Android emulator.](#)

Continue

☐

Do Not Show Again

### Description of MIT App Inventor

From this Site you can access MIT App Inventor, which lets you develop applications for Android devices using a web browser and either a connected phone or emulator. You can also use the Site to store your work and keep track of your projects. App Inventor was originally developed by Google. The Site also includes documentation and educational content, and this is being licensed to you under the Creative Commons Attribution 3.0 Unported license ([CC BY 3.0](#)).

### Account Required for Use of MIT App Inventor

In order to log in to MIT App Inventor, you need to use a Google account. Your use of that account is subject to Google's Terms of Service for



I accept the terms of service!

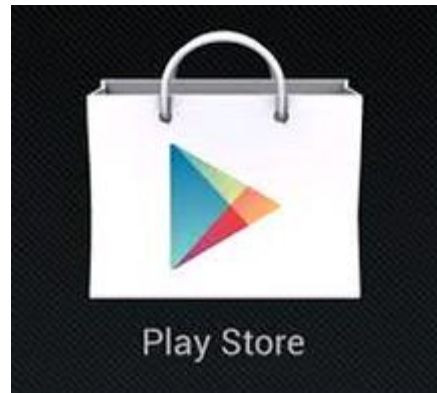
# AI Companion on your phone or tablet

- ➔ Can see changes made in real-time
- ➔ Operate and test your app with your actual phone
- ➔ GPS, Camera, and accelerometer might work
- ➔ Application is only temporarily running on the phone. Application is not stored in the phone

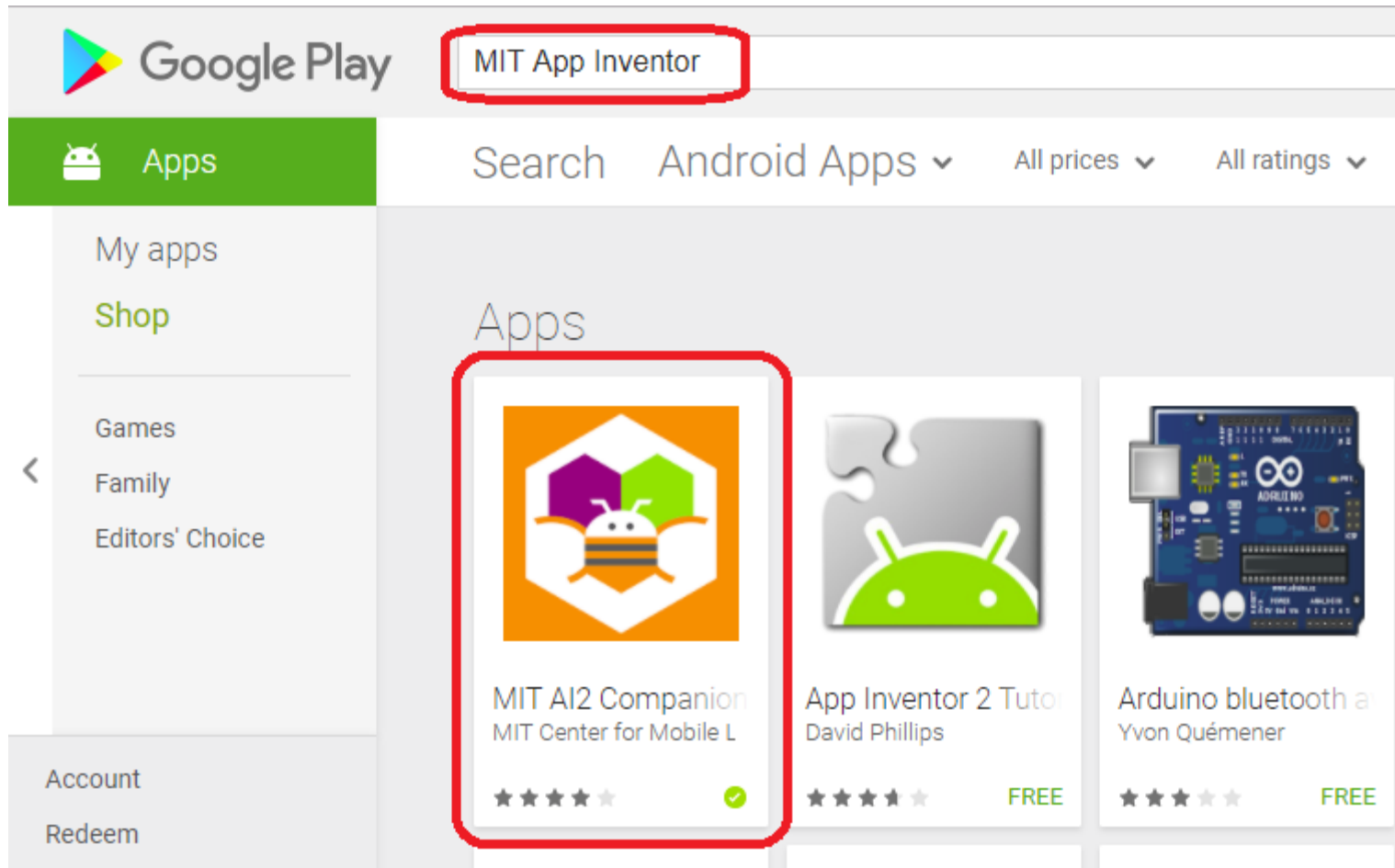


# Downloading AI Companion

➡ Go to the Play Store in your phone



# Search for “MIT app inventor”



# Creating your first app

## “Talk to Me”

# Let's get started

The screenshot shows the MIT App Inventor 2 web interface. The browser tabs at the top include 'Inbox - appinv1011@gmail.co...' and 'MIT App Inventor 2'. The address bar shows 'ai2.appinventor.mit.edu'. The page header features the MIT App Inventor 2 logo and the text 'Beta', along with navigation links for 'Projects', 'Connect', 'Build', and 'Help'. A green bar contains two buttons: 'Start new project' (highlighted with a red rectangle) and 'Delete Project'. Below this, a 'My Projects' section is visible on the left. A modal dialog box titled 'Create new App Inventor project' is open, containing a 'Project name:' label and a text input field with 'TalkToMe' (highlighted with a red rectangle). At the bottom of the dialog are 'Cancel' and 'OK' buttons.

Inbox - appinv1011@gmail.co... MIT App Inventor 2

ai2.appinventor.mit.edu

MIT App Inventor 2 Beta Projects Connect Build Help

Start new project Delete Project

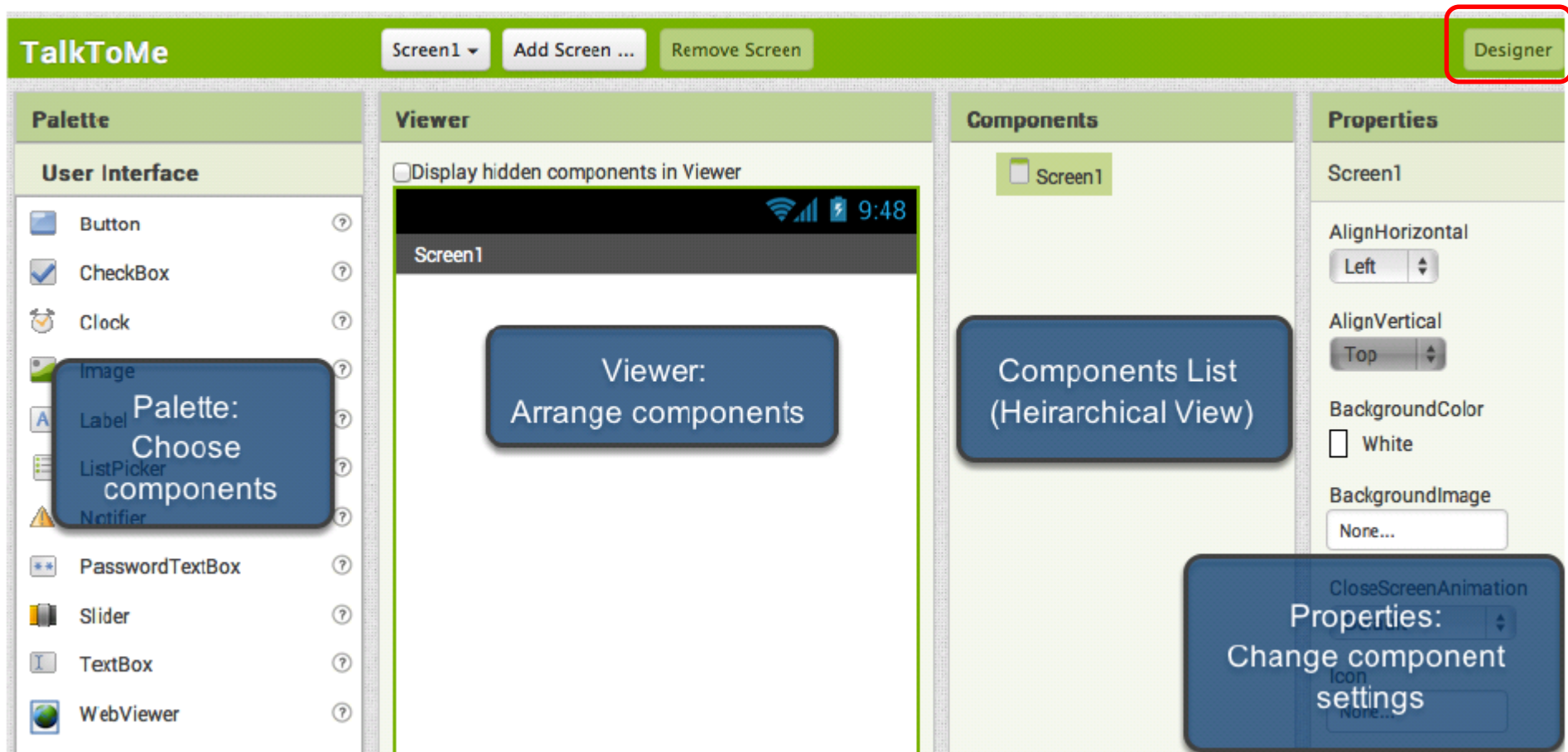
My Projects

Create new App Inventor project

Project name: TalkToMe

Cancel OK

# Getting Started





**Palette**

1 **User Interface**

- Button
- CheckBox
- Clock
- Image
- Label
- ListPicker
- Notifier
- PasswordTextBox

**Viewer**

☐ Display hidden components in Viewer

Screen1

Text for Button1

3

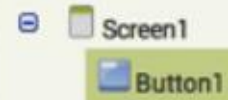
1. Click and hold on "Button"  
2. Drag over to the Viewer and drop.  
3. A Button appears on the Viewer.

## Viewer

☐ Display hidden components in Viewer



## Components



## Properties

Button1

BackgroundColor  
☐ Default

Enabled  
☒

FontBold  
☐

FontItalic  
☐

FontSize  
14.0

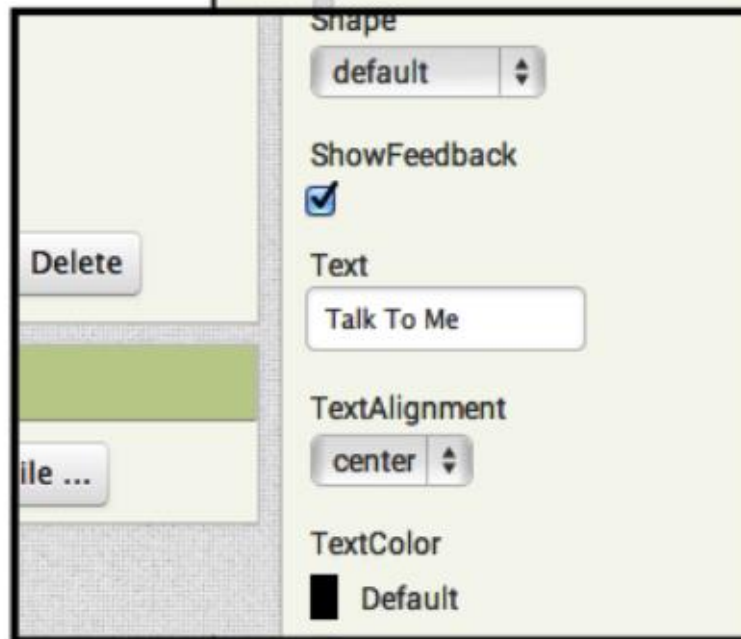
FontTypeface  
default

Image  
None...

Shape  
default

ShowFeedback  
☒

Text  
Talk T



Rename Delete

## Media

Palette

User Interface

Layout

Media

Camcorder ?

Camera ?

ImagePicker ?

Player ?

Sound ?

SoundRecorder ?

SpeechRecognizer ?

TextToSpeech ?

VideoPlayer ?

Drawing and Animation

Sensors

Social

Storage

Connectivity

LEGO® MINDSTORMS®

Viewer

☐ Display hidden components in Viewer

Screen1

Talk To Me

Drop here.  
Component will  
automatically  
show up in  
Non-visible  
components area  
below

Non-visible components

TextToSpeech1

Components

Screen1

Button1

TextToSpeech1

Rename

Delete

Media

Upload File ...

My Projects

Guide

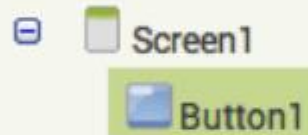
Report an Issue

appinventorskilz@gmail.com ▾

Designer

Blocks

## Components



## Properties

Button1

BackgroundColor

☐ Default

Enabled



FontBold



FontItalic



## Blocks

## Built-in

- Control
- Logic
- Math
- Text
- Lists
- Colors
- Variables
- Procedures



## Screen1

- Button1
- TextToSpeech1



## Any component

## Viewer

**Built-in Blocks** are always available. They handle things like math, text, logic, and control.

**Component Blocks** correspond to the components you've chosen for your app.



0



0

Show Warnings

**Workspace** where you assemble the blocks into a program.

**Trash** for deleting unneeded blocks.



# TalkToMe

Screen1 ▾

Add Screen ...

Remove Screen

## Blocks

### Built-in

Control

Logic

Math

Text

Lists

Colors

Variables

Procedures

### Screen1

Button1

Textuspeech1

### Any component

## Viewer

when Button1 ▾ .Click

do

when Button1 ▾ .GotFocus

do

when Button1 ▾ .LongClick

do

when Button1 ▾ .LostFocus

do

Button1 ▾ BackgroundColor

3

when Button1 ▾ .Click

do



# TalkToMe

Screen1 ▾

Add Screen ...

Remove Screen

## Blocks

☰ Built-in

Control

Logic

Math

Text

Lists

Colors

Variables

Procedures

☰ Screen1

Button1

TextToSpeech1

☰ Any component

## Viewer

when TextToSpeech1 ▾ .AfterSpeaking

result

do

when TextToSpeech1 ▾ .BeforeSpeaking

do

call TextToSpeech1 ▾ .Speak  
message

TextToSpeech1 ▾ . Country ▾

set TextToSpeech1 ▾ . Country ▾ to

TextToSpeech1 ▾ . Language ▾

when Button1 ▾ .Click

do

call TextToSpeech1 ▾ .Speak  
message

# TalkToMe

Screen1 ▾

Add Screen ...

Remove Screen

## Blocks

### Built-in

Control

Logic

Math

Text

Lists

Colors

Variables

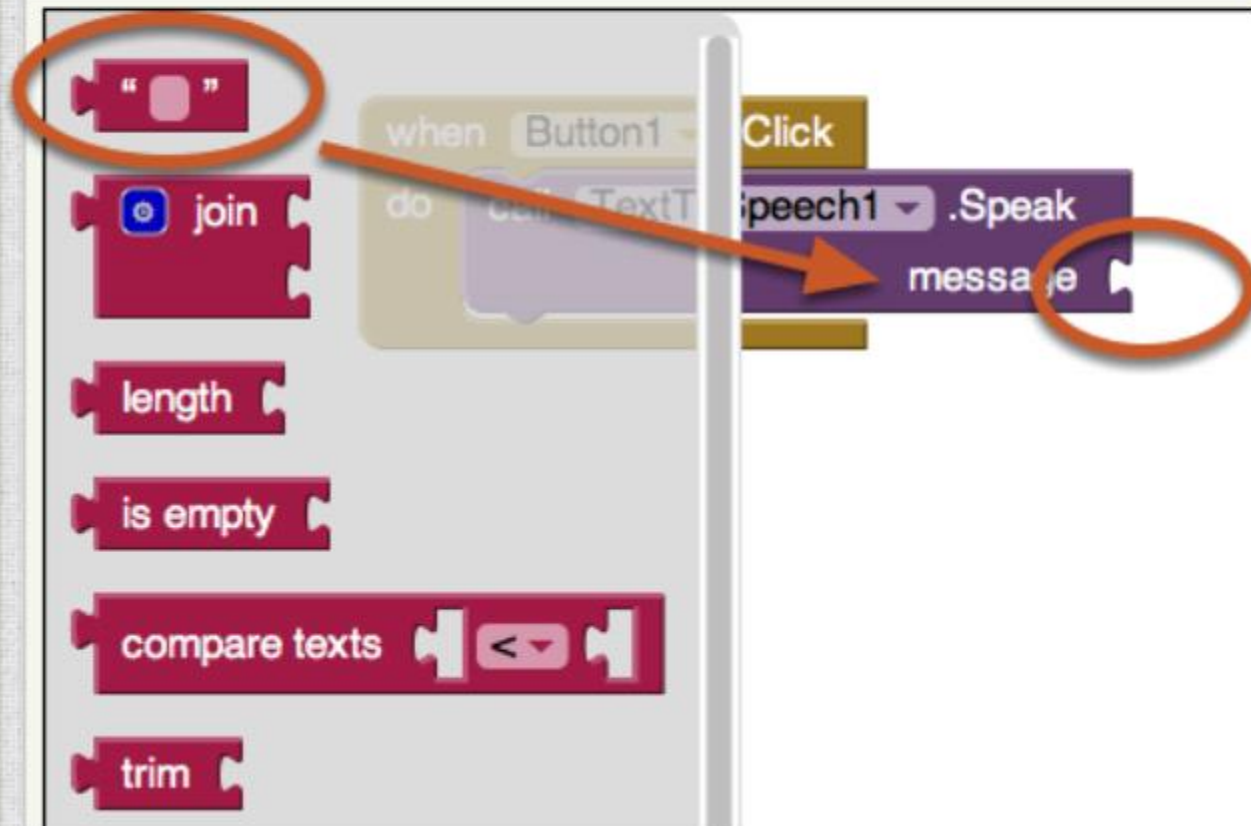
Procedures

### Screen1

Button1

TextToSpeech1

## Viewer





```
when Button1 .Click  
do call TextToSpeech1 .Speak  
message " Congratulations! You've made your first app. "
```



MIT App Inventor 2  
Beta

Project ▾

Connect ▾

Build ▾

TalkToMe

Screen1 ▾

AI Companion

Emulator

USB

Reset Connection

Palette

User Interface



Button



Checkmark

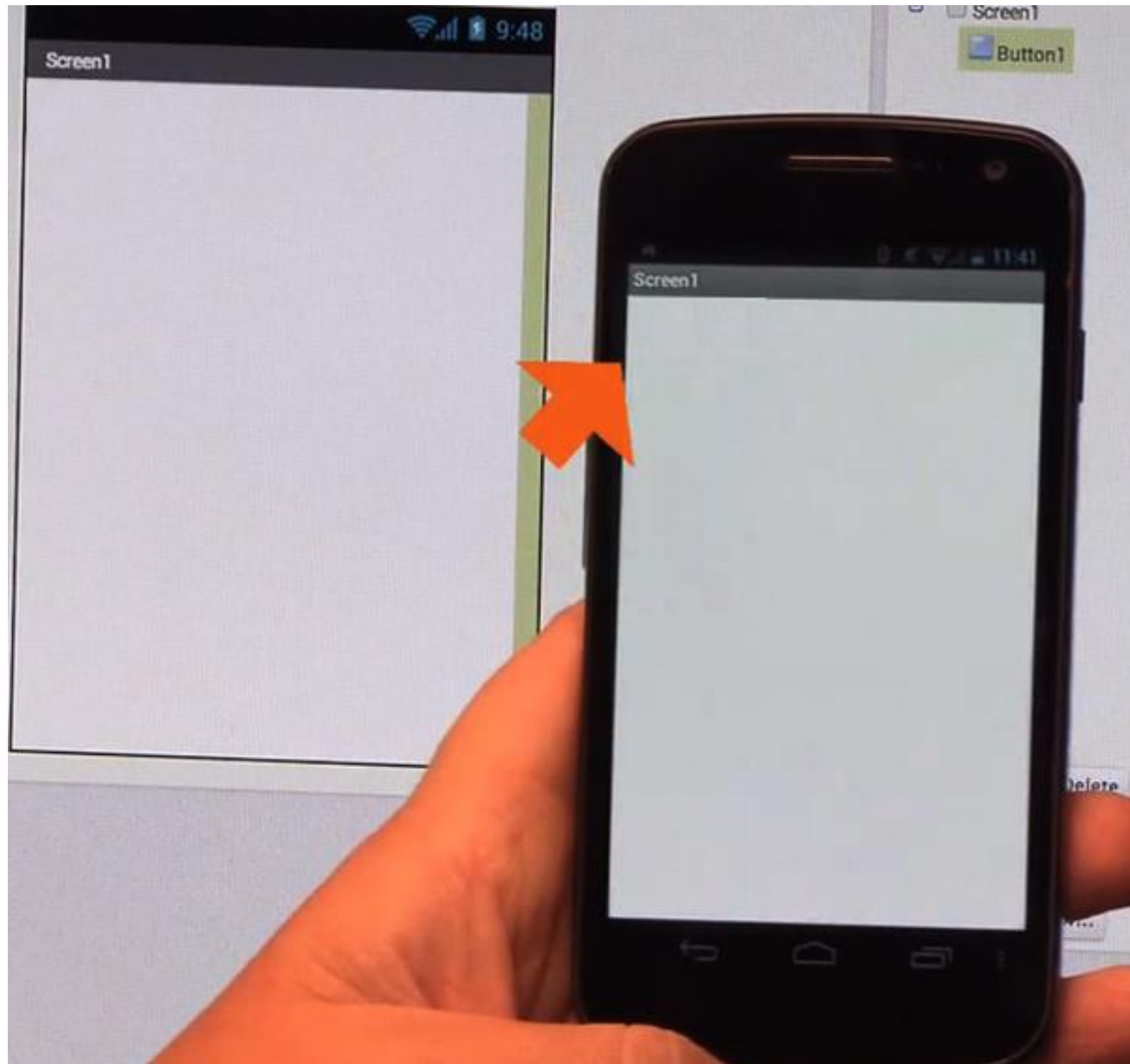


Viewer

☐ Display hidden

Screen1





Inbox (1) - appinv1011@gmail... x MIT App Inventor 2 x Connect y

ai2.appinventor.mit.edu/#5306835864125440

MIT App Inventor 2 Beta

Projects v Connect v Build v Help v

## TalkToMe

### Blocks

- Built-in
  - Control
  - Logic
  - Math
  - Text**
  - Lists
  - Colors
  - Variables
  - Procedures
- Screen1
  - TextBox1

My projects

---

Start new project

Import project (.aia) from my computer ...

Import project (.aia) from a repository ...

Delete Project

---

**Save project**

Save project as ...

Checkpoint

---

Export selected project (.aia) to my computer

Export all projects

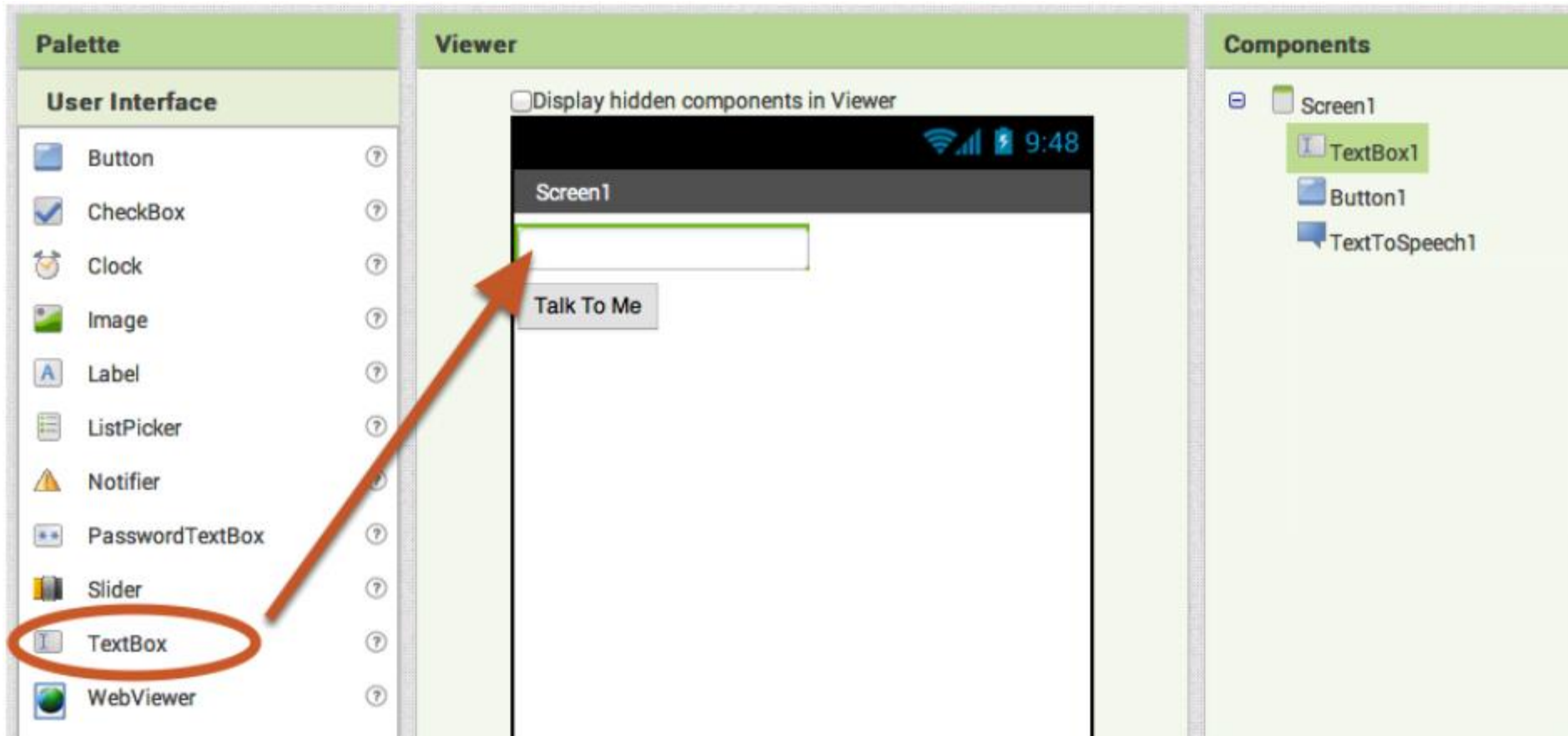
---

Import keystore

Export keystore

Delete keystore

# Step 2: Inputting what to say



## Blocks

### Built-in

- Control
- Logic
- Math
- Text
- Lists
- Colors
- Variables
- Procedures

### Screen1

- TextBox1
- Button1
- TextToSpeech1
- AccelerometerSensor1

### Any component

## Viewer

set TextBox1 . Height to

TextBox1 . Hint

set TextBox1 . Hint to

TextBox1 . MultiLine

set TextBox1 . MultiLine to

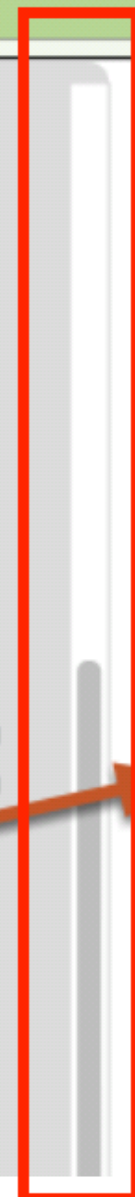
TextBox1 . NumbersOnly

set TextBox1 . NumbersOnly to

TextBox1 . Text

set TextBox1 . Text to

TextBox1 . TextColor







" Congratulations! You've made you



Blocks

Built-in

Control

Logic

Math

Text

Lists

Colors

Variables

Procedures

Screen1

HorizontalArrangement1

Viewer

initialize global name to

get

set to

initialize local name to

in

do

call TextToSpeech1

.Speak

message

TextToTalk

.Text

initialize local name to

in

# TalkToMe

Screen1 ▾

Add Screen ...

Remove Screen

## Blocks

### Built-in

- Control
- Logic
- Math
- Text
- Lists
- Colors
- Variables
- Procedures

## Viewer

initialize global name to

get

set to

initialize local name to

in

initialize global textToSpeak to

set to

global textToSpeak

Blocks

Built-in

Control

Logic

Math

Text

Lists

Colors

Variables

Viewer

initialize global name to

get ▾

set ▾ to

initialize local name to

in

initialize global textToSpeak to " "

when Button1 ▾ .Click

do

set global textToSpeak ▾ to

TextBox1 ▾ . Text ▾

call TextToSpeech1 ▾ .Speak

message

get ▾

global textToSpeak

Inbox (1) - appinv1011@gmail...

MIT App Inventor 2

Connect your Phone or Tablet ...

+

ai2.appinventor.mit.edu/#5306835864125440

MIT App Inventor 2

Beta

Projects ▾ Connect ▾ Build ▾ Help ▾

TalkToMe

Screen1 ▾ Add Screen ... Remove Screen

Blocks

Built-in

Control

Logic

Math

Text

Lists

Colors

Variables

Procedures

Screen1

TextBox1

Button1

TextToSpeech1

Any component

Viewer

initialize global textToSpeak to " "

when Button1 .Click

do

set global textToSpeak to TextBox1 . Text

call TextToSpeech1 .Speak

message get global textToSpeak

" Congratulations! You've made your first app. "

Inbox (1) - appinv1011@gmail... x MIT App Inventor 2 x Connect y

ai2.appinventor.mit.edu/#5306835864125440

MIT App Inventor 2  
Beta

Projects v Connect v Build v Help v

## TalkToMe

### Blocks

- Built-in
  - Control
  - Logic
  - Math
  - Text**
  - Lists
  - Colors
  - Variables
  - Procedures
- Screen1
  - TextBox1

My projects

---

Start new project

Import project (.aia) from my computer ...

Import project (.aia) from a repository ...

Delete Project

---

**Save project**

Save project as ...

Checkpoint

---

Export selected project (.aia) to my computer

Export all projects

---

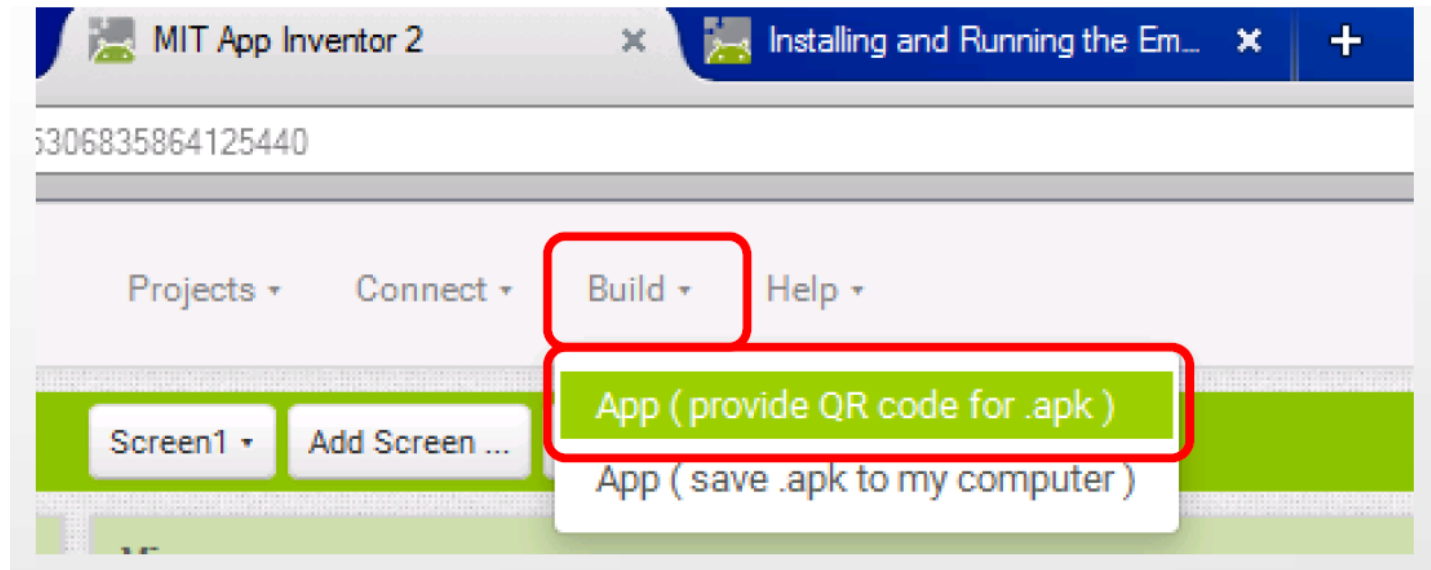
Import keystore

Export keystore

Delete keystore

# Sharing your app

# Sending App to Friends and Family



# Install blocked

For security, your device is set to block installation of applications not obtained from Google Play

Cancel

Settings





## Security

Device administration

Device administrators

View or disable device administrators

Unknown sources

Allow installation of apps from sources other than the Play Store



Verify apps

Block or warn before installing apps that may cause harm



## Unknown sources

Your device and personal data are more vulnerable to attack by applications from unknown sources. You agree that you are solely responsible for any damage to your device or loss of data that may result from using these applications



Allow initial installation only

Cancel

OK



TalkToMe



internet



message



eMail



call



Apps

# Lunch

# Lunch Instructions

- ➔ You will be eating at the University Residence Cafeteria
- ➔ You are expected to get back to this room by 1:00pm
- ➔ We will go over in groups and come back in groups – STICK TOGETHER!

Have a good lunch

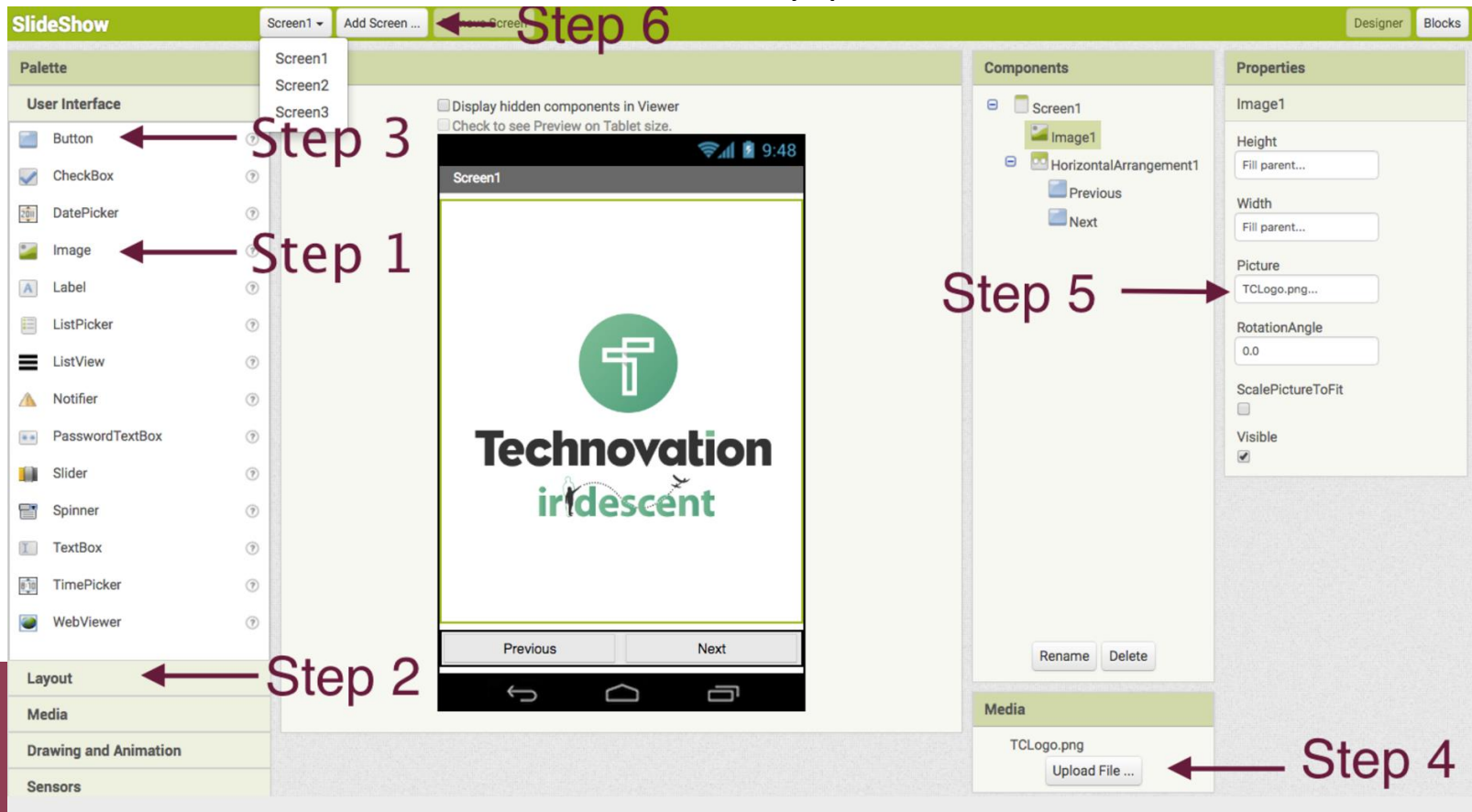
# Making a slideshow

# Designing your screen

- ➡ For this app you want to create a slideshow by allowing your user to be able to navigate through images
- ➡ When you design your screen, you will need to put in an image
- ➡ You also need 'previous' and 'next' buttons for your user to click. This part will be done in the designer



1. Add an image to your screen.
  - Make it fit your screen by changing the width and the height!
  - We choose to make our image “fill parent”. When you select “fill parent” you are telling the component to fill the space of whatever it is in.
2. Add a horizontal layout to your screen below the image.
3. Add two buttons into the horizontal layout.
  - Click on the buttons in the component menu
    - Edit the text to make the buttons say ‘previous’ and ‘next’.





4. Upload a picture you want to use in your slide show!
5. Select the picture you want to use on the first screen of your slide show.
  - Click on “Image1” in the “Components” menu and then “Picture” in the properties menu. You can then select a picture you uploaded.
6. Add two more screens.
  - For now you do not need to do anything with those screens, but after you code the blocks you will add buttons and images to them just like you did for this screen



# Building your code

- ➡ Once the layout is complete use the Blocks Editor to make the buttons do something.
- ➡ For each button, make an action for the event it is clicked.
- ➡ For the Next button you want it to call the next screen (if you are on 1, call 2)
- ➡ For the Previous button call the previous screen (if you are on 1, call 3)

## Click on “Blocks” Button

- Click on button you named ‘previous’ to see all the event handlers you can use. Grab the ‘when button.click’ event handler and drag it to your workspace.
- Click on ‘control’ and find the ‘open another screen screenName’ block.
- Get an empty text box and click it into the “open another screen screenName” block. Type the name of the last screen that will be in your slide show. For us, it was Screen3.
- Click on your next button and grab the “when button.click” event handler again
- Get another “open another screen screenName” block from control.
- Get an empty text block and type in the next screen that will appear in your slide show. (For us, that was Screen2.)

The screenshot shows the SlideShow application interface. The top bar has a green background with the text "SlideShow" and buttons for "Screen1", "Add Screen ...", "Remove Screen", "Designer", and "Blocks". The "Blocks" panel on the left lists categories: Built-in, Control, Logic, Math, Text, Lists, Colors, Variables, Procedures, Screen1, Image1, HorizontalArrangement1, Previous, Next, and Any component. The "Viewer" panel on the right shows two event handler blocks: "when Previous .Click" and "when Next .Click", both with "do" blocks containing "open another screen screenName" and "Screen3" or "Screen2". Red arrows point from the text "Steps 8 & 11" to the "Control" category, "Steps 9 & 12" to the "Text" category, and "Steps 7 & 10" to the "Previous" and "Next" buttons. A blue arrow points from the "Blocks" button in the top bar to the "Blocks" panel.

SlideShow

Screen1 Add Screen ... Remove Screen Designer Blocks

Blocks

Built-in

- Control
- Logic
- Math
- Text
- Lists
- Colors
- Variables
- Procedures

Screen1

- Image1
- HorizontalArrangement1
  - Previous
  - Next
- Any component

Viewer

Steps 8 & 11

Steps 9 & 12

Steps 7 & 10

when Previous .Click

do open another screen screenName " Screen3 "

when Next .Click

do open another screen screenName " Screen2 "

0 0

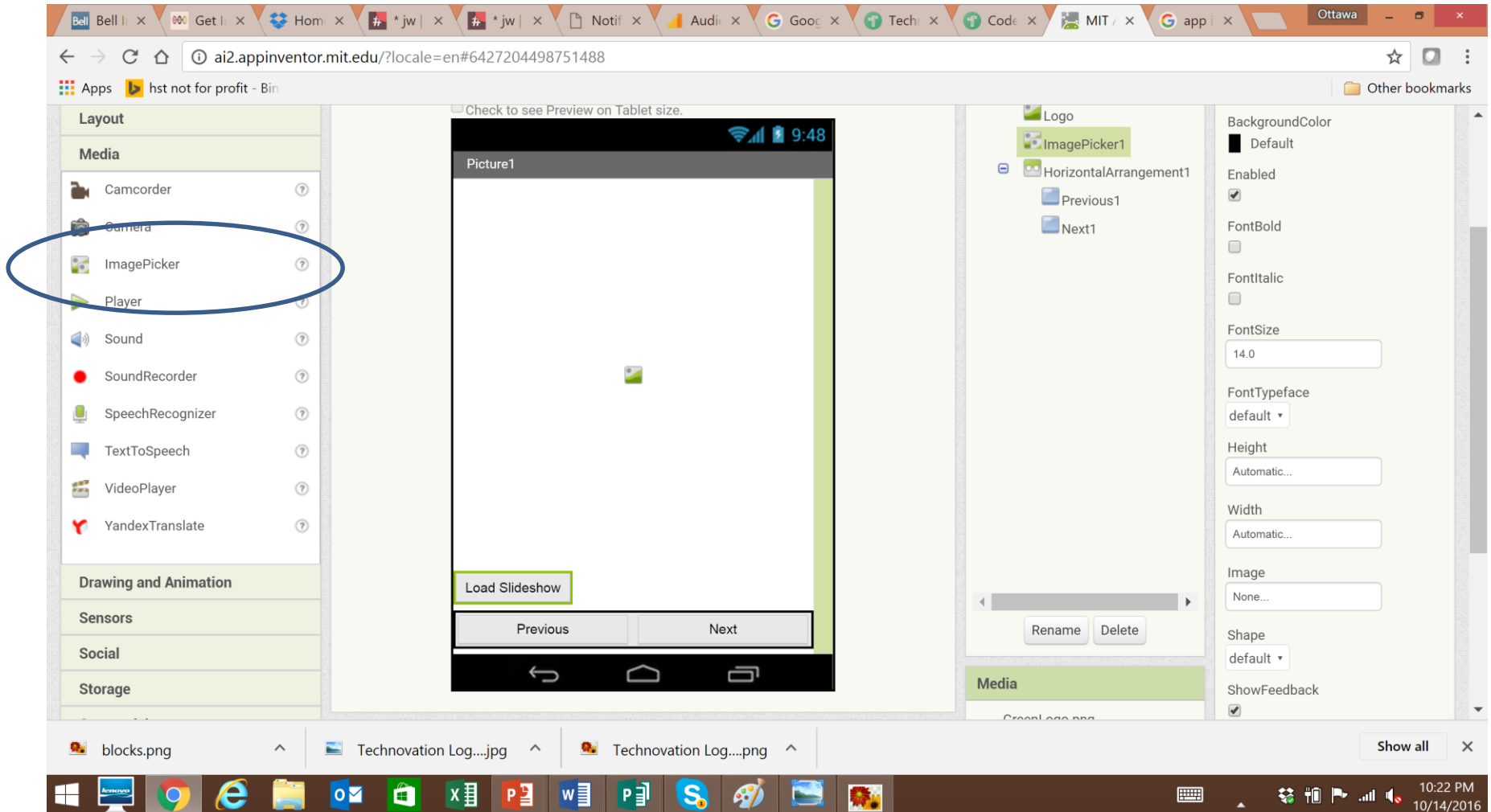
Show Warnings

Rename Delete

# Using photos on the phone

- ➔ Your first slide show used images from your computer.
  - You used the app inventor image upload to do that
- ➔ To make a slide show with images from your user's phone we will use the image picker.
- ➔ The image picker lets the user on the phone see all their photos and pick one.

# Using an image on your phone



# Setting the image

Browser tabs: Bell, Get, Home, \*jw, \*jw, Notif, Audi, Goog, Tech, Code, MIT, app, Ottawa

Address bar: ai2.appinventor.mit.edu/?locale=en#6427204498751488

Page title: Apps hst not for profit - Bin

Other bookmarks

Slideshow Screen1 Add Screen ... Remove Screen Designer Blocks

Blocks

- Built-in
  - Control
  - Logic
  - Math
  - Text
  - Lists
  - Colors
  - Variables
  - Procedures
- Screen1
  - Logo
  - ImagePicker1
- HorizontalArrangement1
  - Previous1
  - Next1
- Any component

Viewer

when Next1 .Click  
do open another screen screenName "Screen2"

when Previous1 .Click  
do open another screen screenName "Screen3"

when ImagePicker1 .AfterPicking  
do set Logo . Picture to ImagePicker1 . Selection

0 0  
Show Warnings

blocks.png Technovation Log....jpg Technovation Log....png Show all

Windows taskbar: Windows, Chrome, Edge, File Explorer, Outlook, Teams, Excel, PowerPoint, Word, Publisher, Skype, Paint, Photos, Weather, Clock, 10:26 PM 10/14/2016

# Types of Data

# Types of Data?

	WhatsApp	Angry Birds	Slideshow
About	Message and call your friends	Game: shoot birds at pigs	Cycle through images (you made this)
Types of data	<ul style="list-style-type: none"><li>• Your username</li><li>• Your friend's usernames</li><li>• The message you want to send</li><li>• What time it is</li><li>• Your location</li></ul>	<ul style="list-style-type: none"><li>• Your score</li><li>• levels you've completed</li></ul>	<ul style="list-style-type: none"><li>• Your favorite images</li></ul>

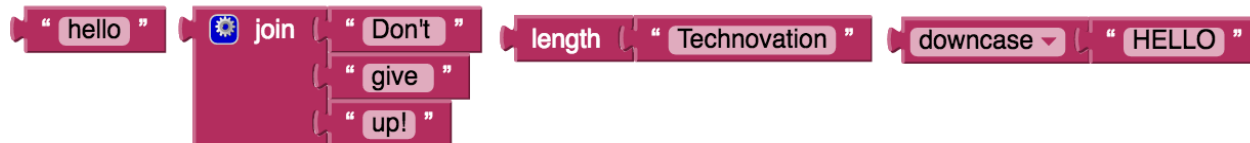


# Types of Data

## ➔ Numbers



## ➔ Strings



## ➔ Booleans



# Variables




**Variable: Data that can change in value**

- Your Age (string, number, or boolean?)
- Your Address (string, number, or boolean?)
- Student[yes or no] (string, number, or boolean?)



# Variables

➡ Variable: Data that can change in value

- Your Age (string, **number**, or boolean?) 
- Your Address (string, number, or boolean?)
- Student[yes or no] (string, number, or boolean?)




# Variables

## Variable: Data that can change in value

- Your Age (string, **number**, or boolean?) 
- Your Address (**string**, number, or boolean?) 
- Student[yes or no] (string, number, or boolean?)




# Variables

## ➔ Variable: Data that can change in value


- Your Age (string, **number**, or boolean?) 
- Your Address (**string**, number, or boolean?) 
- Student[yes or no] (string, number, or **boolean**?) 

# Variables

## ➔ Variable: Data that can change in value

- Your Age (string, **number**, or boolean?) 
- Your Address (**string**, number, or boolean?) 
- Student[yes or no] (string, number, or **boolean**?) 

## ➔ Use Global variables to share the value in your program

initialize global  to 

# App Inventor Colored Dots

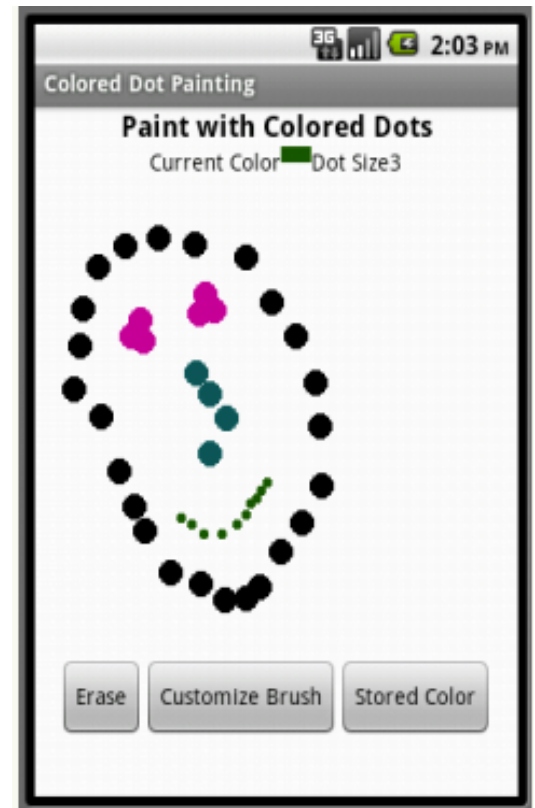
# AI: Colored Dots

## (Create multiple screens)

<http://appinventor.mit.edu/explore/ai2/colored-dots.html>

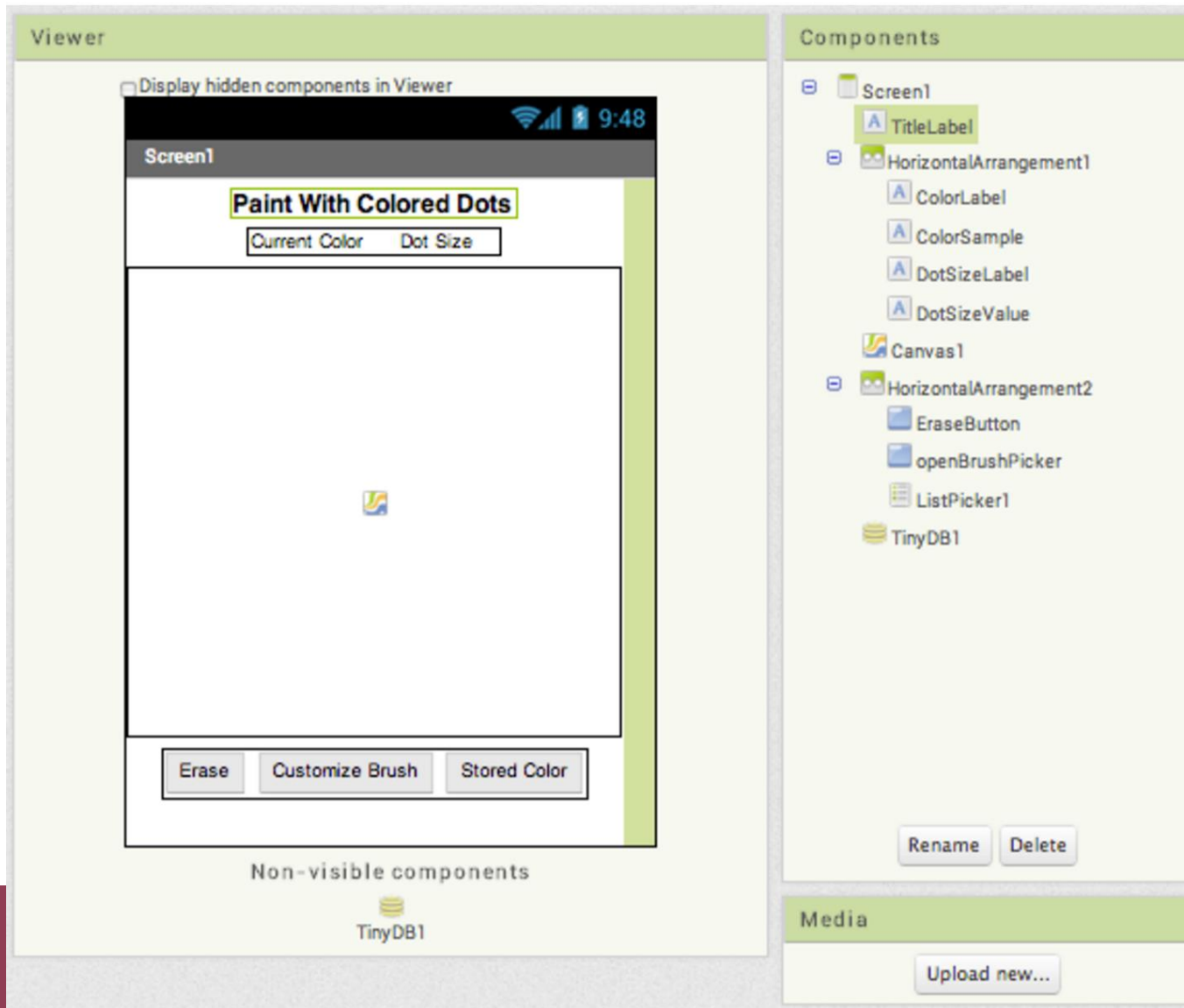
The [Colored Dot tutorial](#) teaches you how to create apps that have multiple screens. You'll learn how to:

- ➔ make an app with multiple screens
- ➔ pass values from one screen to another using TinyDB
- ➔ how to fill and use the ListPicker element



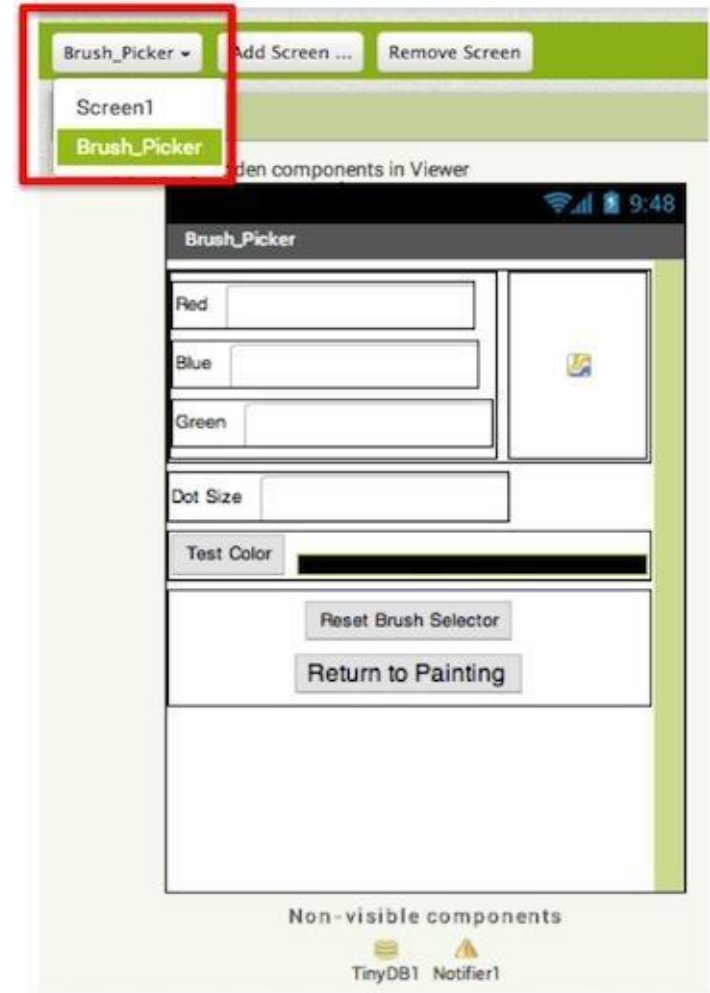


# Starting Screen

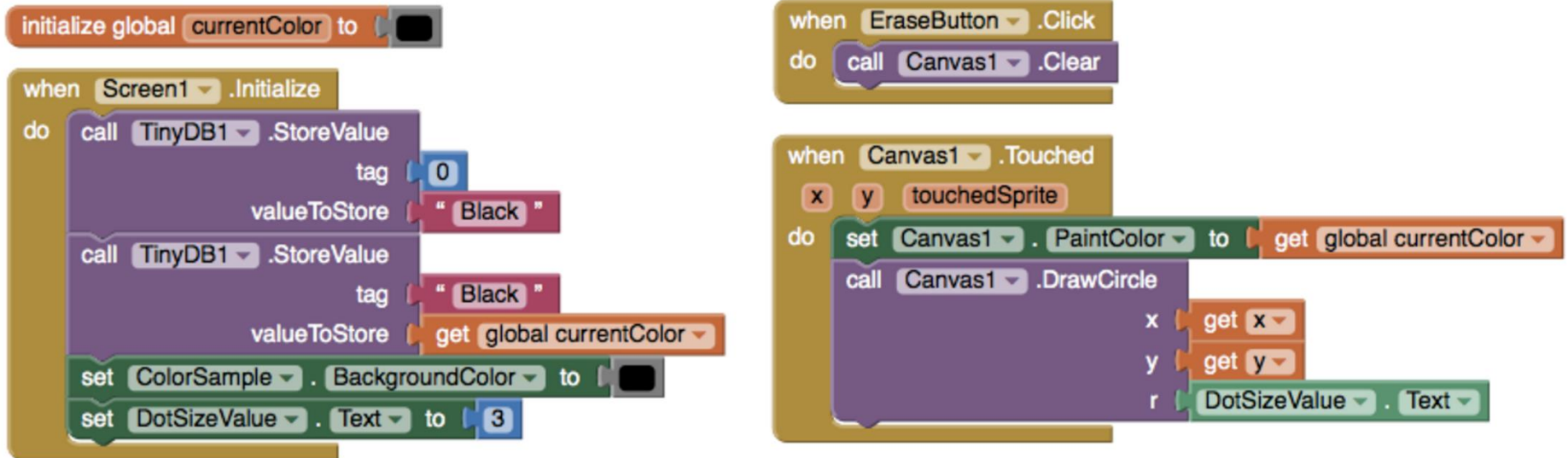


# Multiple Screens

- ➔ You can add screens in the designer and use the screen transitions in blocks editor to decide which screen to go to next
  - For Example: pushing the menu button go to the menu screen
- ➔ Screen 1 will always be the screen the app starts on – it's probably best to make it a welcome screen



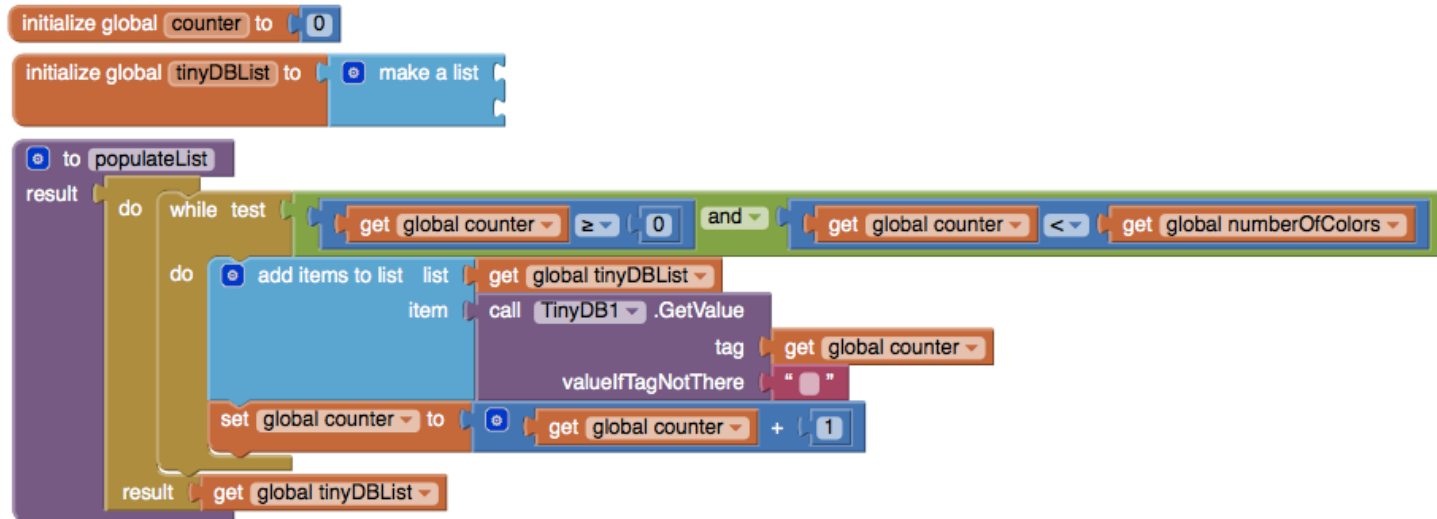
# Starting with one colour



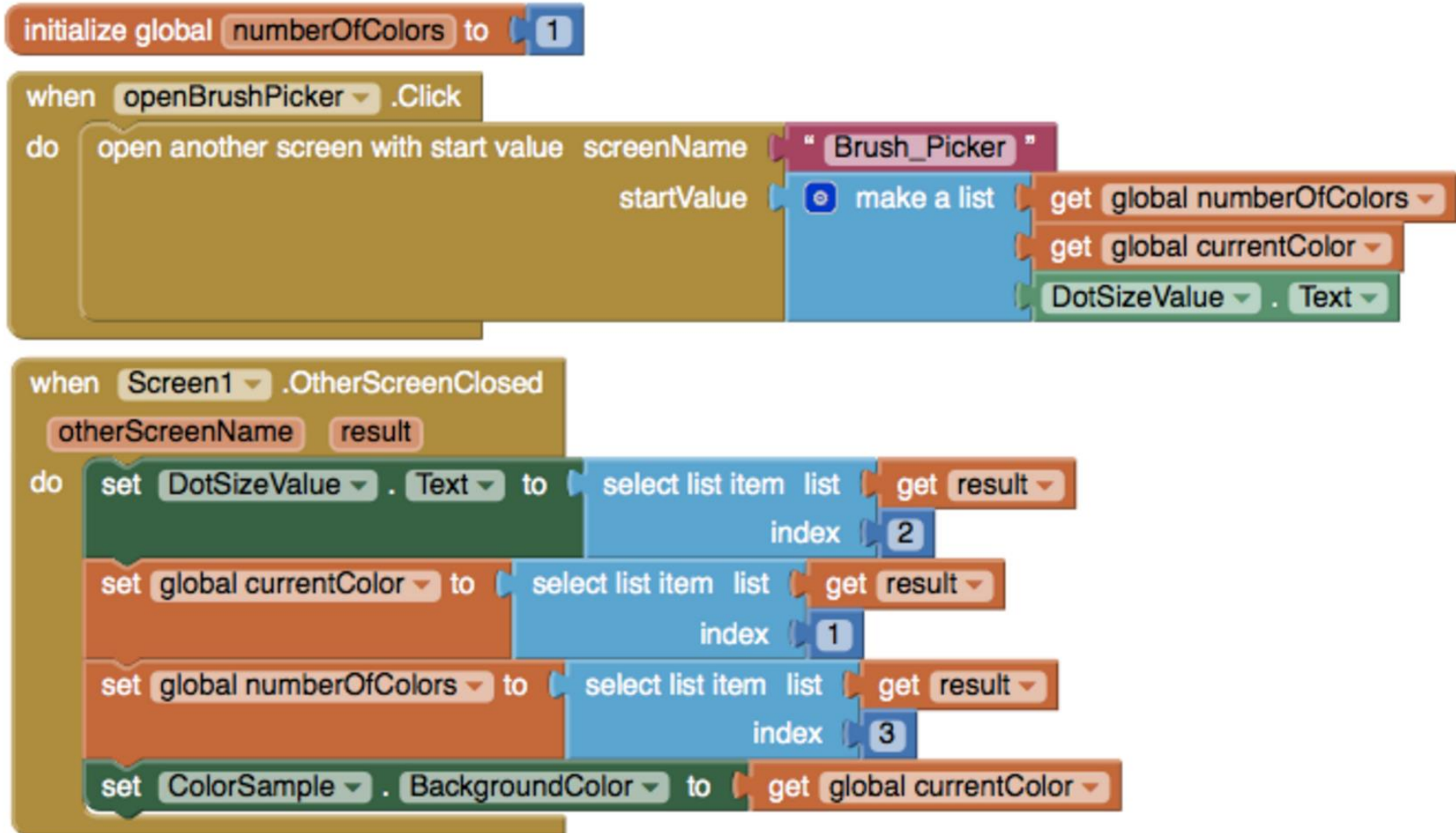
- ➡ On Screen 1 set up your starting colour
- ➡ Set up what happens when the screen is touched

# TinyDB

- ➔ Besides opening screens and returning values, the different screens in a multiple screen app can communicate through TinyDB. To do this, give every screen its individual TinyDB component.
- ➔ ColoredDots uses TinyDB to let you name the colors you create and save them for later use. The saving and naming will be done in Brush\_Picker

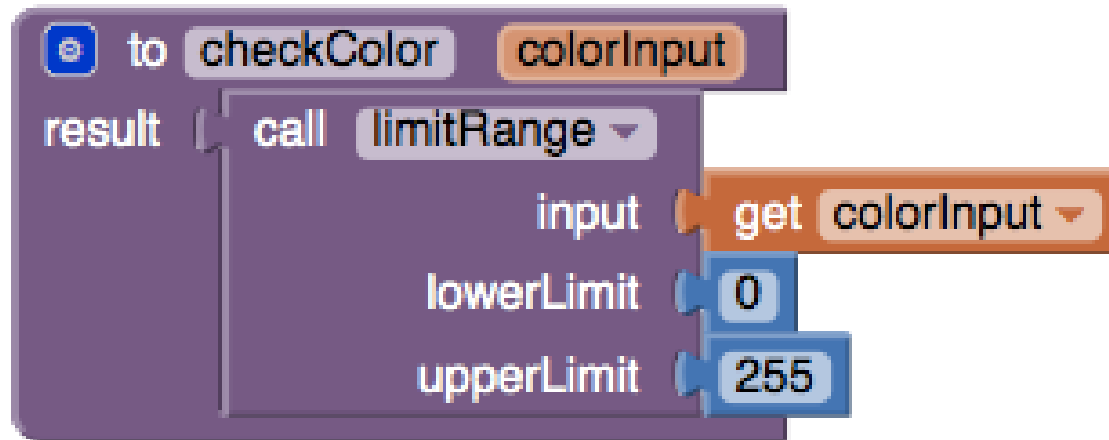


# Getting ready to add colours

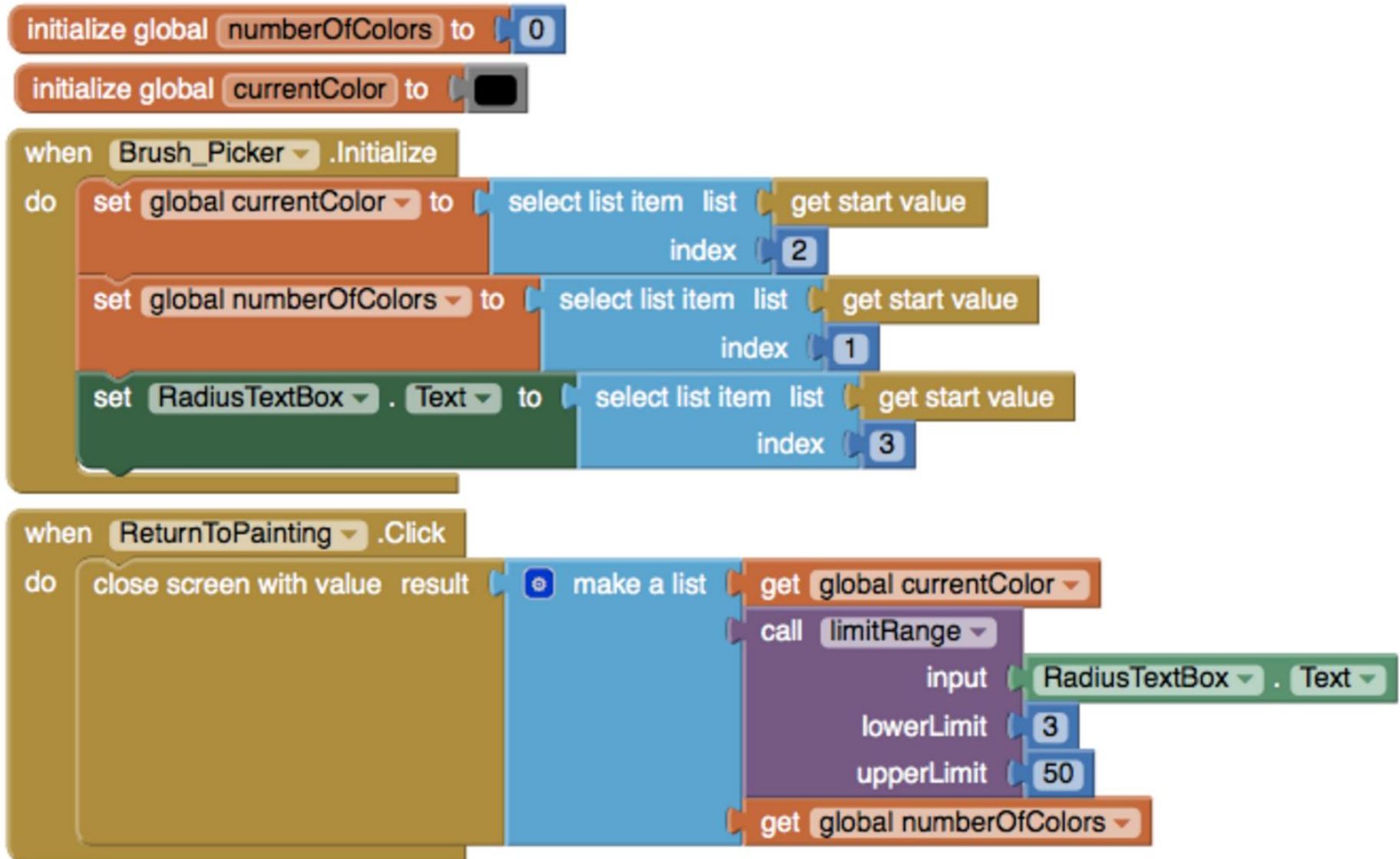


# Brush Picker

- ➔ The main job of Brush\_Picker is to create a color from the red-green-blue values entered in the text boxes and provide that color to Screen1.
- ➔ One thing Brush\_Picker needs to check is that it's using good values for colors and dot size. Each of the red, green, blue values should be a number between 0 and 255.

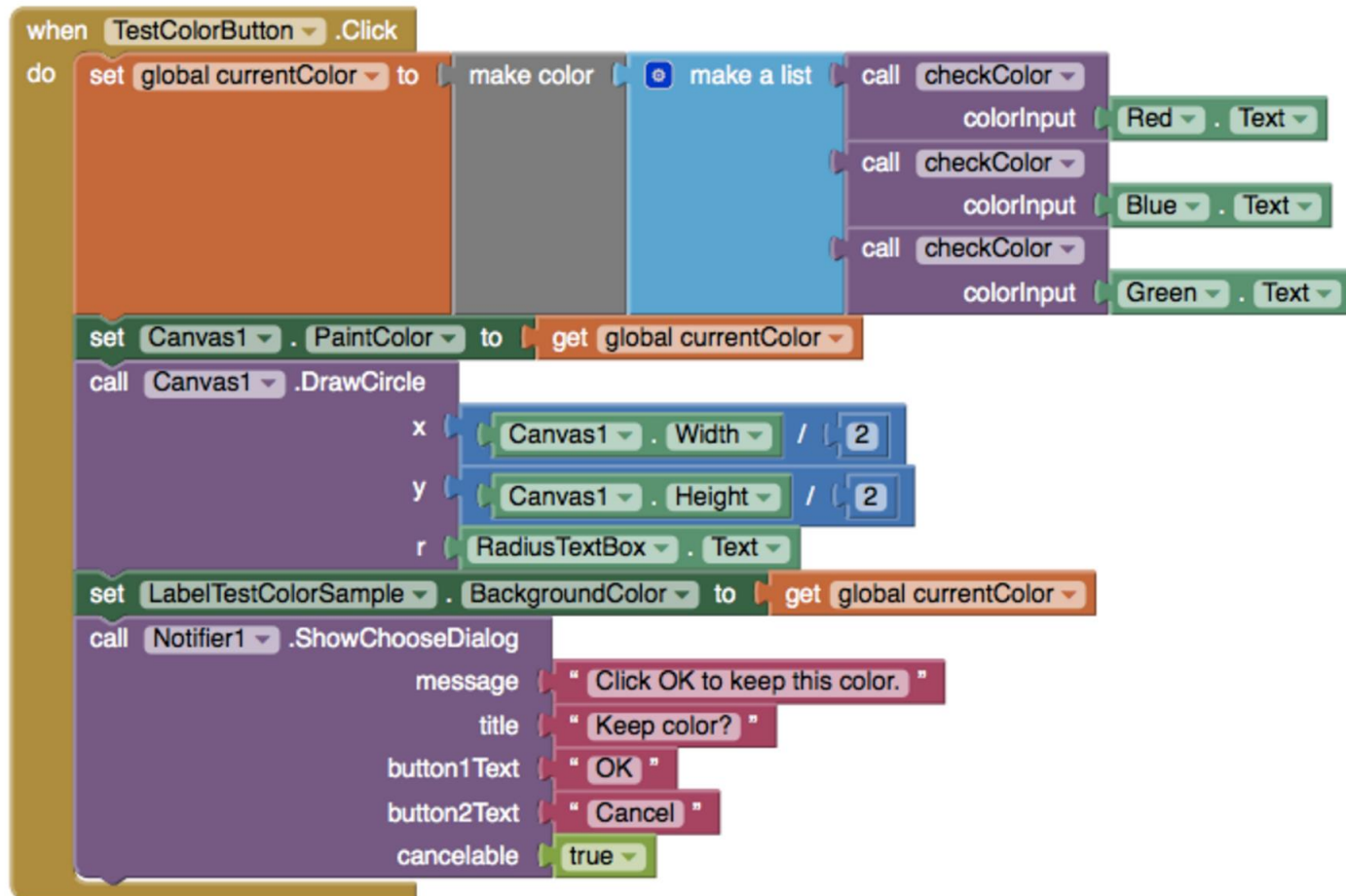


# Setting up the brush picker



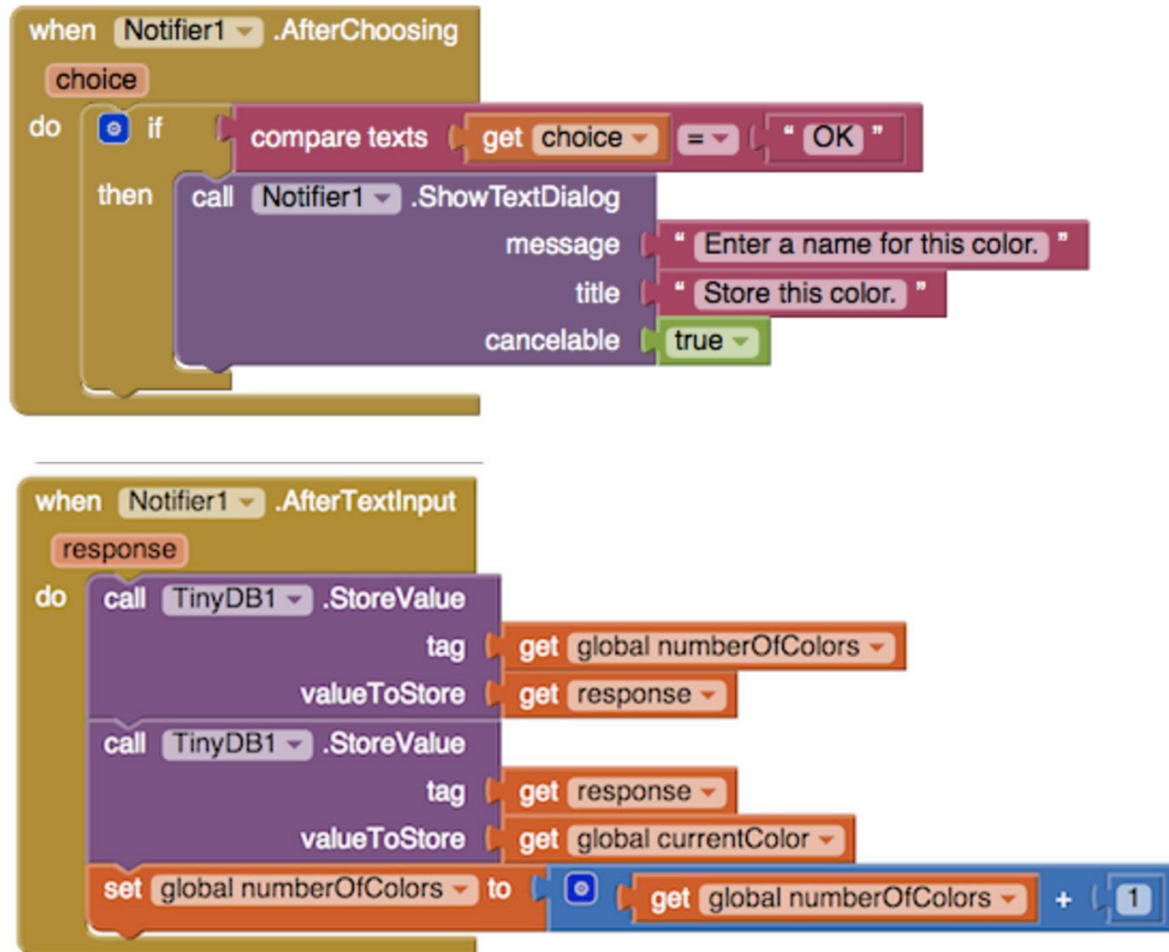


# Testing the colour

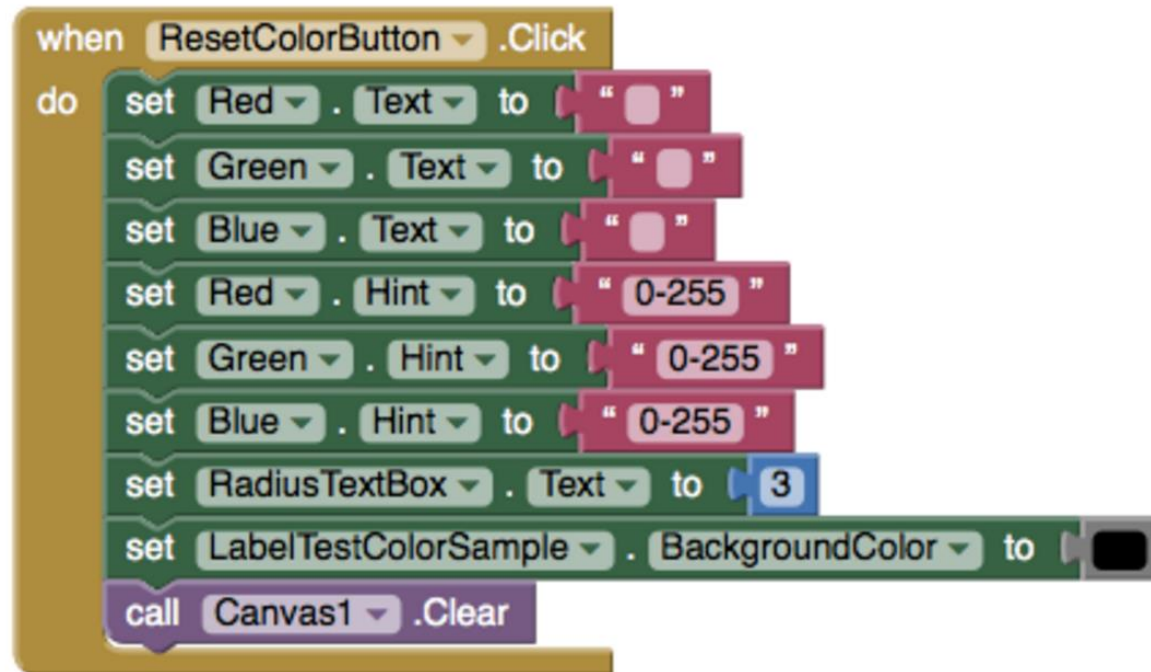




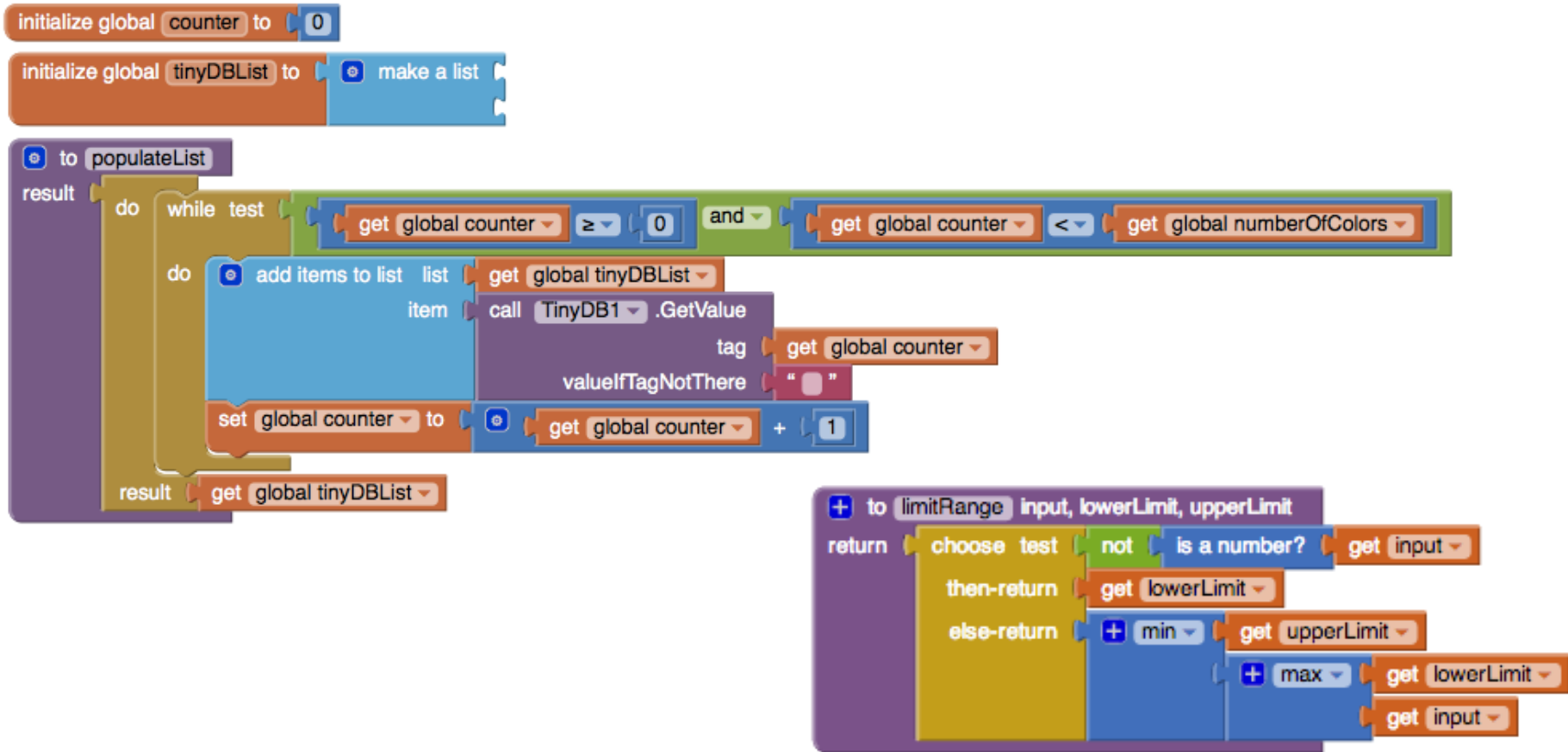
# Saving the colour



# Resetting colours



# Putting it all together




# Continuing your learning (Tutorials)

# appinventor.mit.edu/explore/ai2/tutorials

Technovation - Invitation to co... x MIT App Inventor 2 x Connect your Phone or Tablet ... x Tutorials for App Inventor 2 | E... x +

appinventor.mit.edu/explore/ai2/tutorials.html

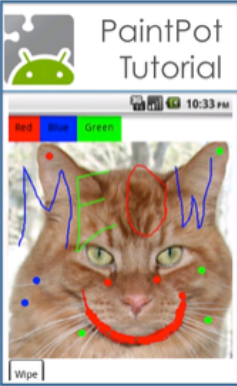
## Tutorials for App Inventor 2



### Beginner Tutorials

Video & Text

- Talk To Me  
Easy text-to-speech app
- Ball Bounce  
Starter for a game app
- Digital Doodle  
Finger drawing app

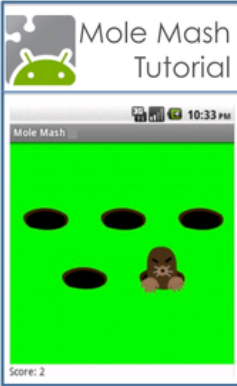


### PaintPot Tutorial

Red Blue Green

MEOW


Wipe



### Mole Mash Tutorial

Mole Mash

Score: 2



### appinventor.org

app building for everyone.

Tutorials • Online Book  
Course-in-a-box

## App Inventor

Create Your Own  
Android Apps

David Welber, Hal Abelson,  
Ellen Spertus & Liz Looney

O'REILLY

There are many more tutorials available below. Scroll down to browse the list, or check the appropriate boxes and click "Filter":

#### Filter by Tutorial Topic

<input type="checkbox"/> Sprites	<input type="checkbox"/> Drawing Canvas	<input type="checkbox"/> Multiple Screens	<input type="checkbox"/> Clock Timer
<input type="checkbox"/> Game	<input type="checkbox"/> SMS Texting	<input type="checkbox"/> Camera	<input type="checkbox"/> Video
<input type="checkbox"/> Activity Starter	<input type="checkbox"/> ListPicker	<input type="checkbox"/> Accelerometer	<input type="checkbox"/> File Sharing
<input type="checkbox"/> Data Storage	<input type="checkbox"/> External API	<input type="checkbox"/> Location Sensor	<input type="checkbox"/> GPS
<input type="checkbox"/> NFC (Near Field Comm.)			

[appinventor.mit.edu/explore/ai2/tutorials](http://appinventor.mit.edu/explore/ai2/tutorials)

# Have Fun!

