Mobile Computational Thinking in App Inventor 2

Lyn Turbak

Wellesley College Computer Science Dept.
(on sabbatical at MIT CSAIL)

Rhode Island College CSTA-RI Talk
April 10, 2014

Computational Thinking Through Mobile Computing NSF Grant Team

Franklyn Turbak Eni Mustafaraj
Wellesley College

Ralph Morelli
Trinity College

Dave Wolber
U. of San Francisco

Larry Baldwin
BIRC

Hal Abelson Shay Pokress Josh Sheldon
MIT

Fred Martin Mark Sherman Karen Roehr
University of Massachusetts Lowell

Acknowledgment: This work was supported by the National Science Foundation under Grants 1225680, 1225719, 1225745, 1225976, and 1226216.

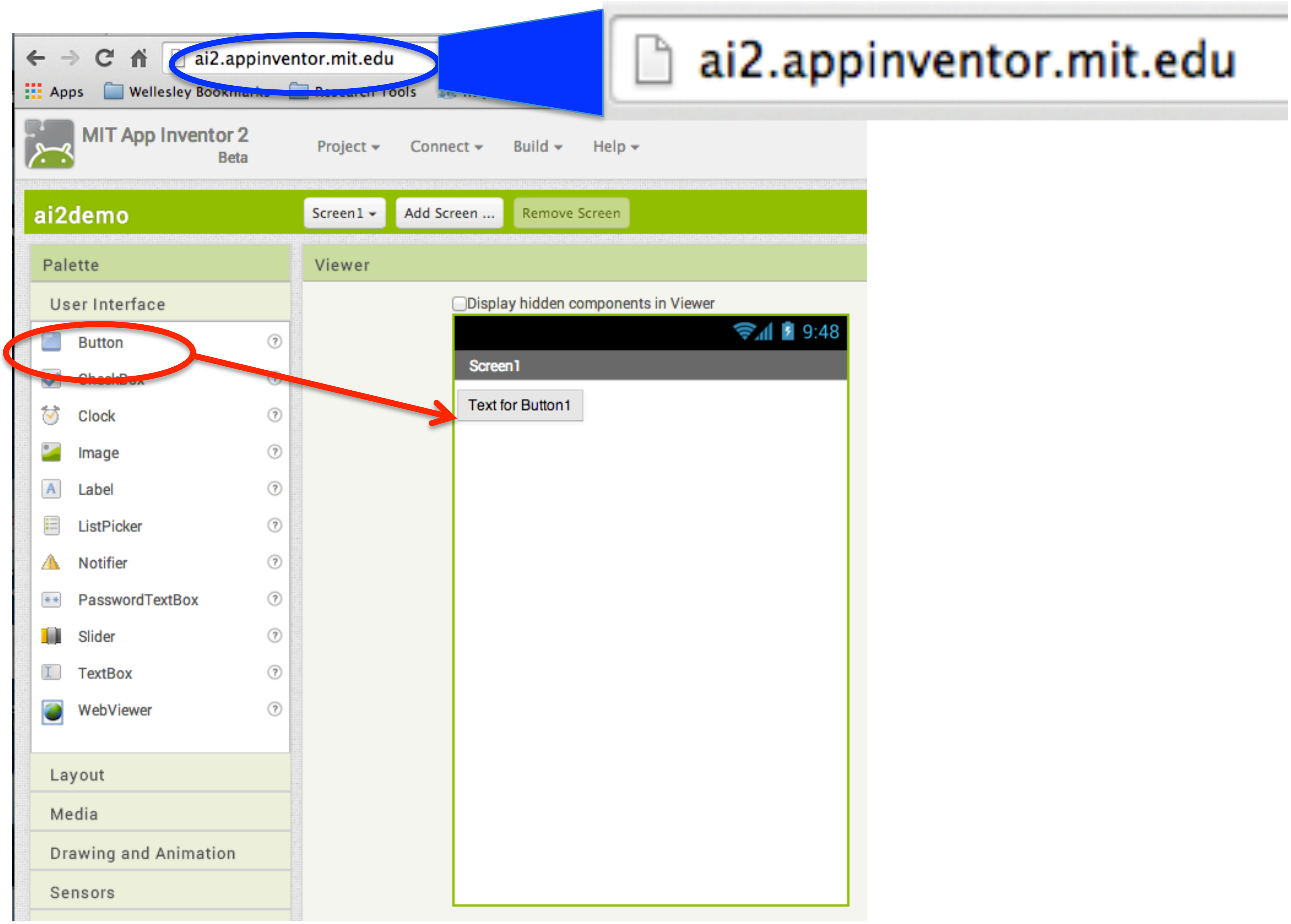
Talk Overview

- App Inventor 2 Demo
- App Inventor App Examples
- Situated Computing & Mobile Computational Thinking
- App Inventor 2 & Mobile Computational Thinking
- Looking Forward

Talk Overview

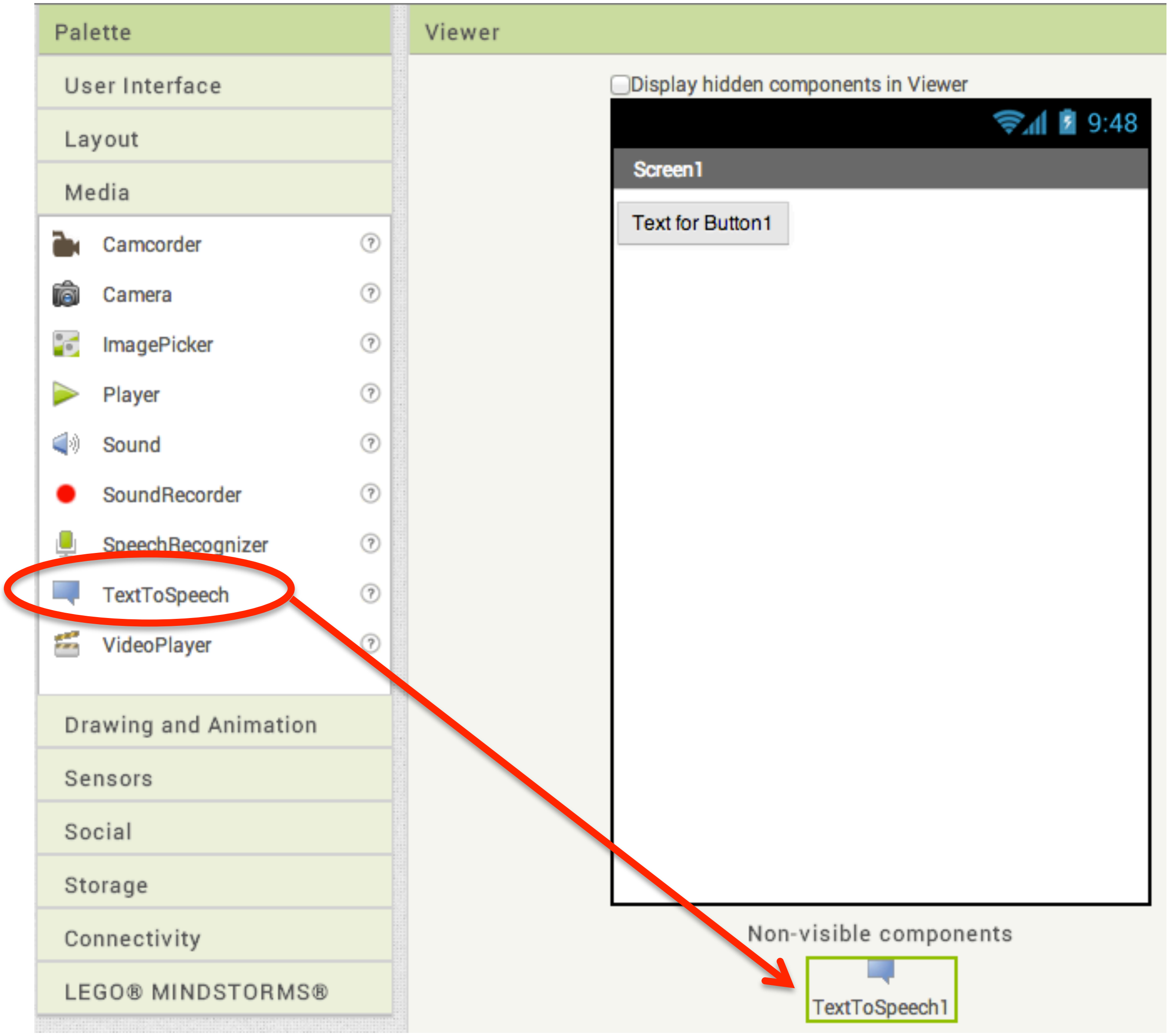
- App Inventor 2 Demo
- App Inventor App Examples
- Situated Computing & Mobile Computational Thinking
- App Inventor 2 & Mobile Computational Thinking
- Looking Forward

AI2 Demo: Add Button Component in Designer

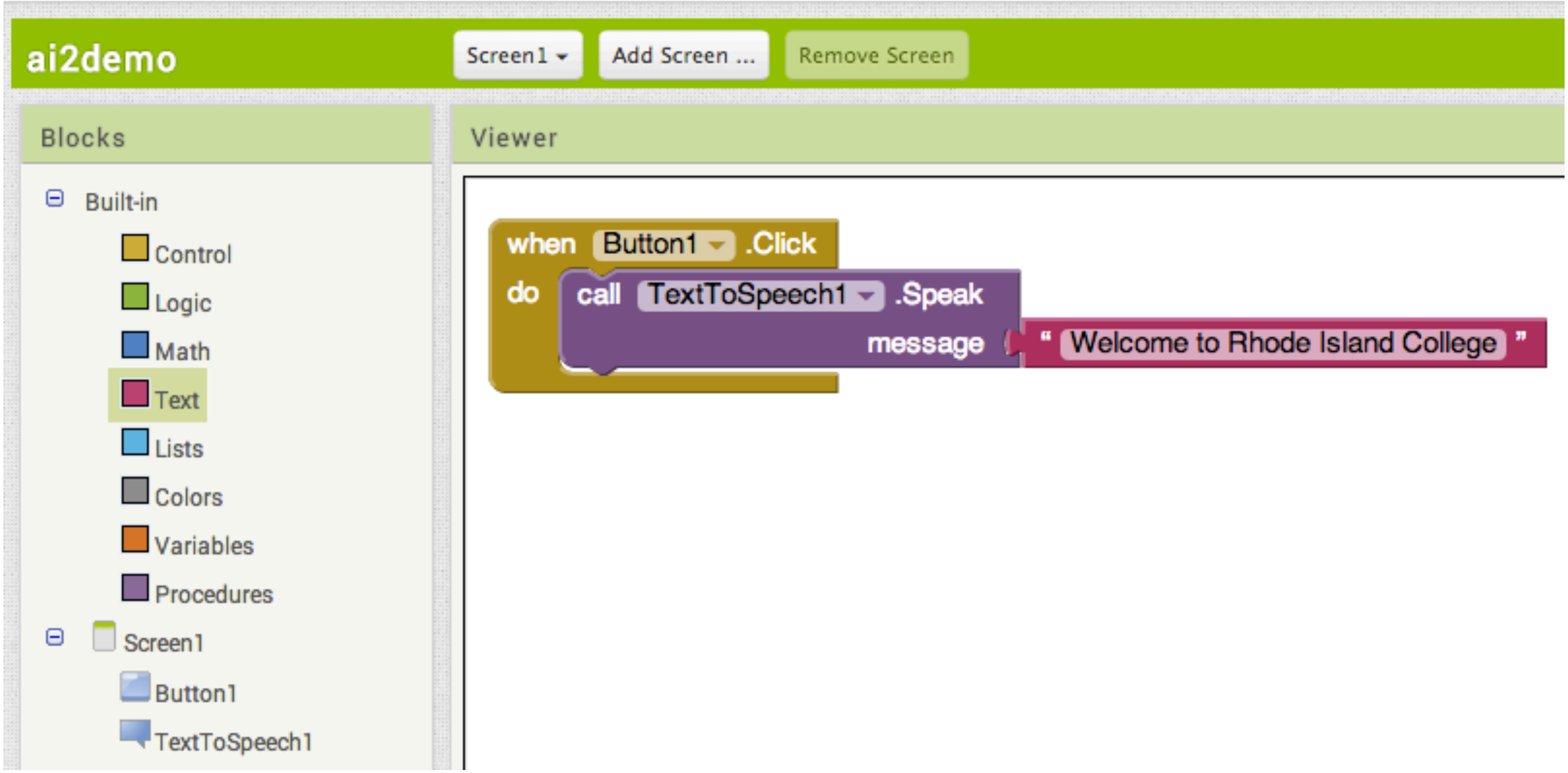


□ Display hidden components in Viewer

AI2 Demo: Add TextToSpeech Component in Designer



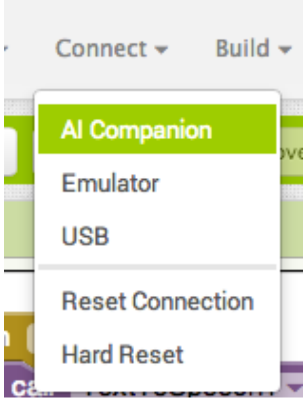
AI2 Demo: Specify Button Behavior in Blocks Editor

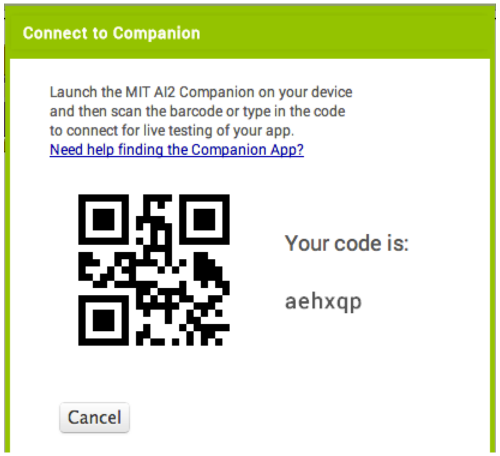


```
when Button1 .Click
do  call TextToSpeech1 .Speak
         message " Welcome to Rhode Island College "
```

AI2 Demo: Connect to Android Device (Live Development)

App Inventor




Connect to Companion

Launch the MIT AI2 Companion on your device and then scan the barcode or type in the code to connect for live testing of your app.

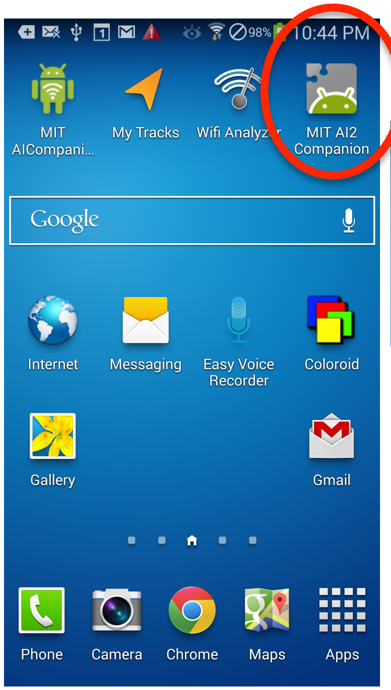
Need help finding the Companion App?

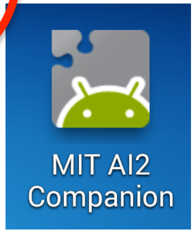
Your code is:

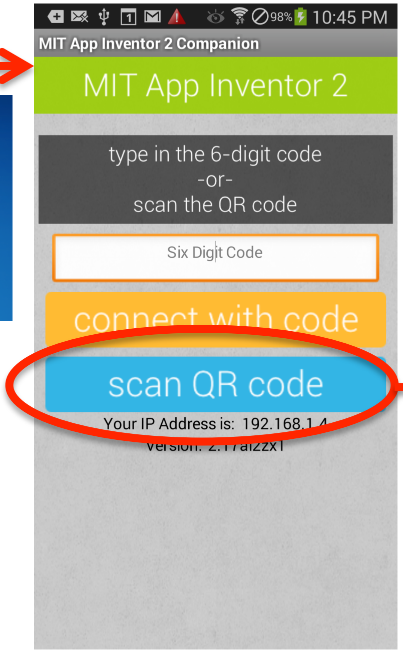
aehxqp

Cancel

Android Device







MIT App Inventor 2 Companion

MIT App Inventor 2

type in the 6-digit code
-or-
scan the QR code

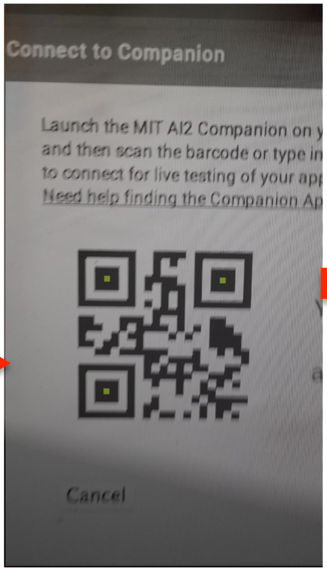
Six Digit Code

connect with code

scan QR code

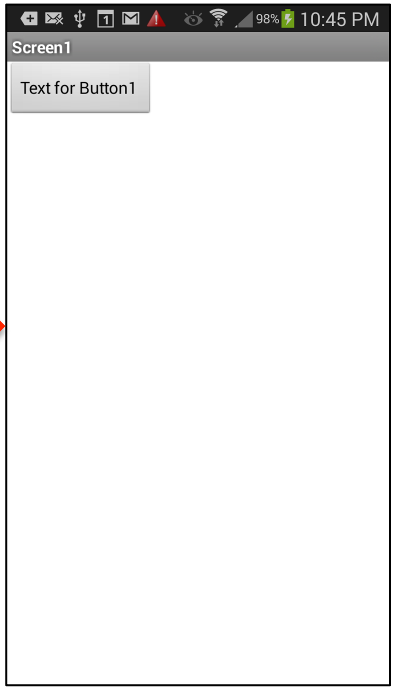
Your IP Address is: 192.168.1.4

Version: 2.17ai2zx1

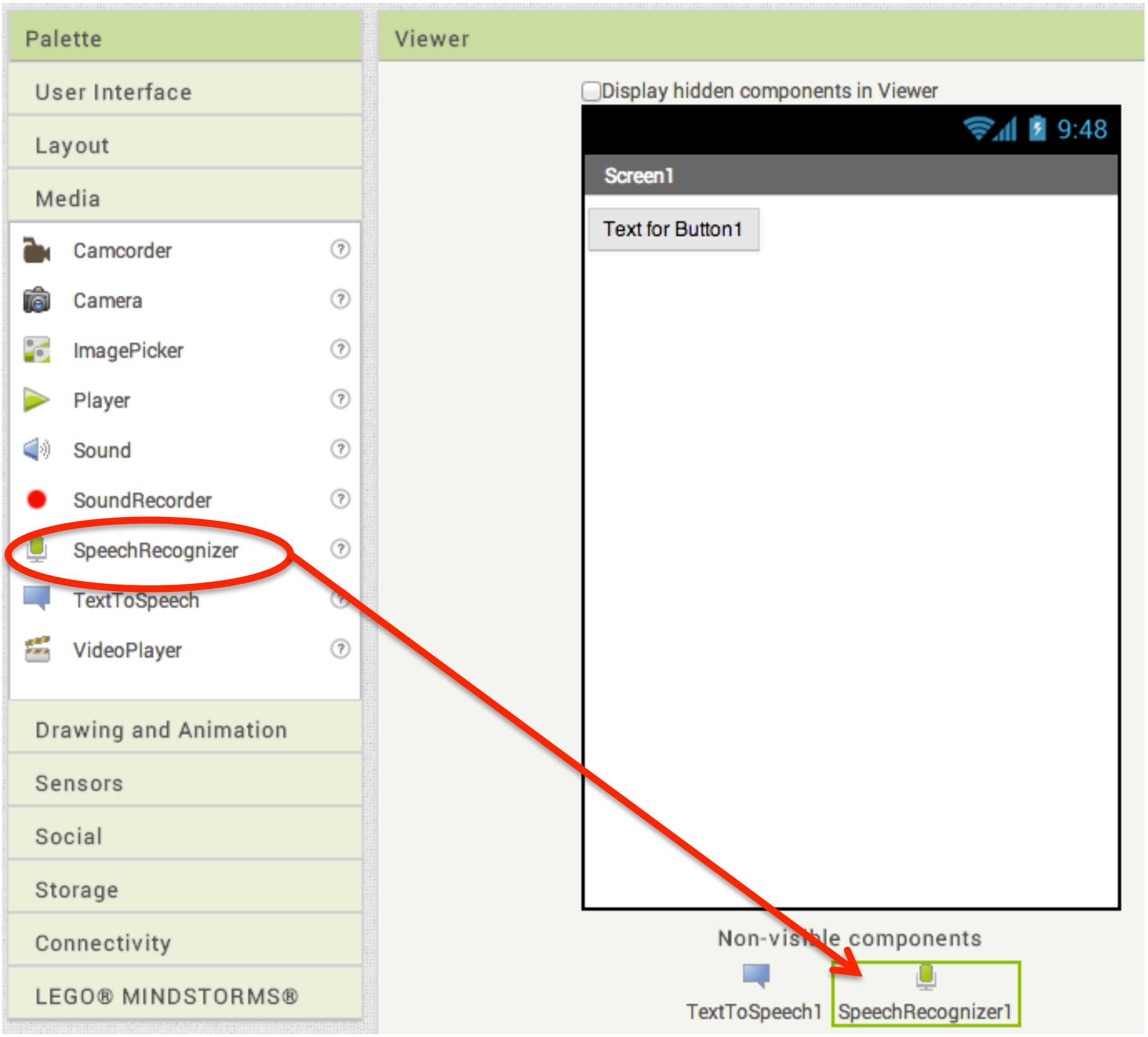

Connect to Companion

Launch the MIT AI2 Companion on y
and then scan the barcode or type in
to connect for live testing of your app
Need help finding the Companion Ap

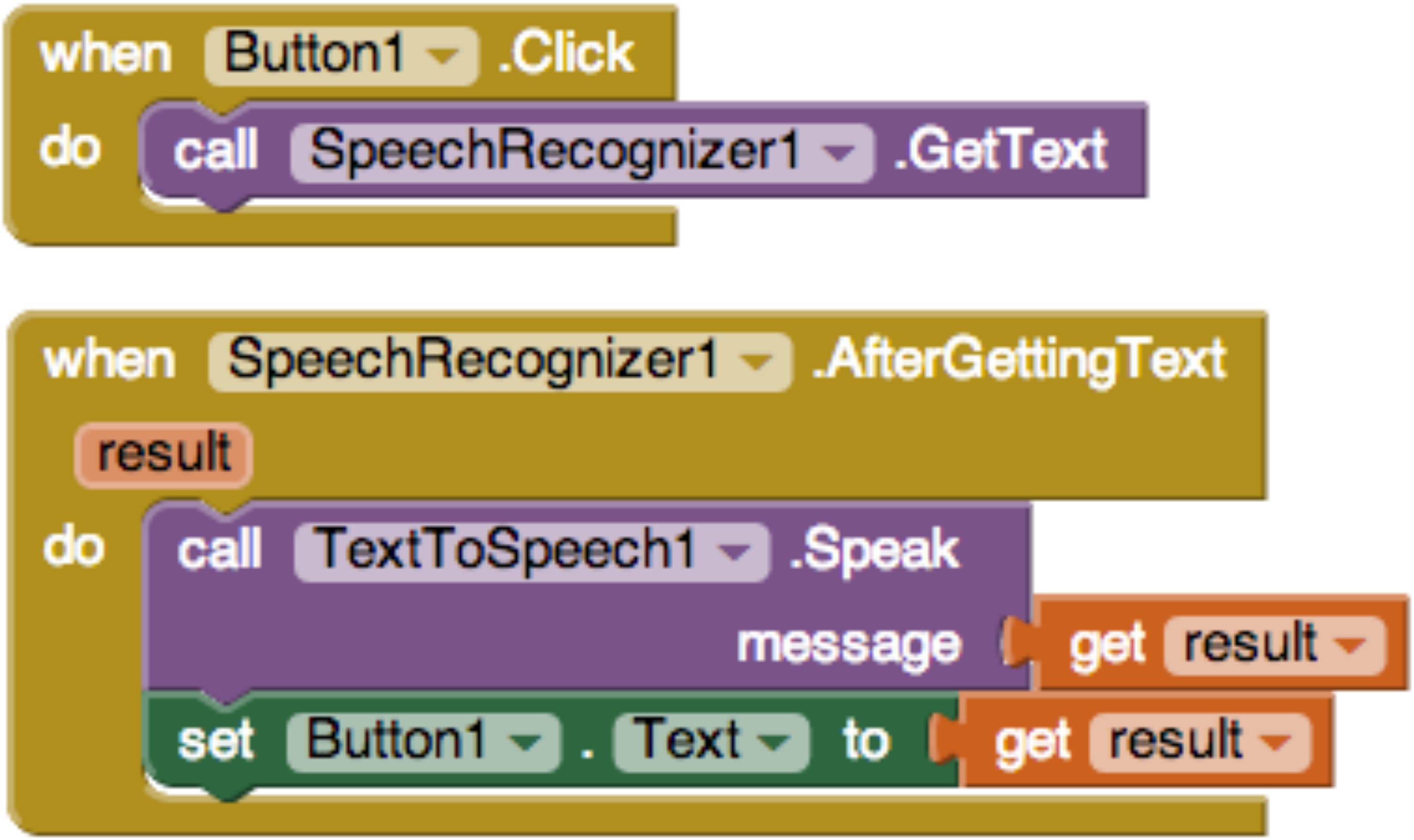
Cancel



AI2 Demo: Add SpeechRecognizer Component in Designer

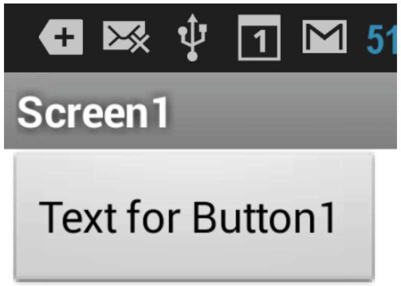


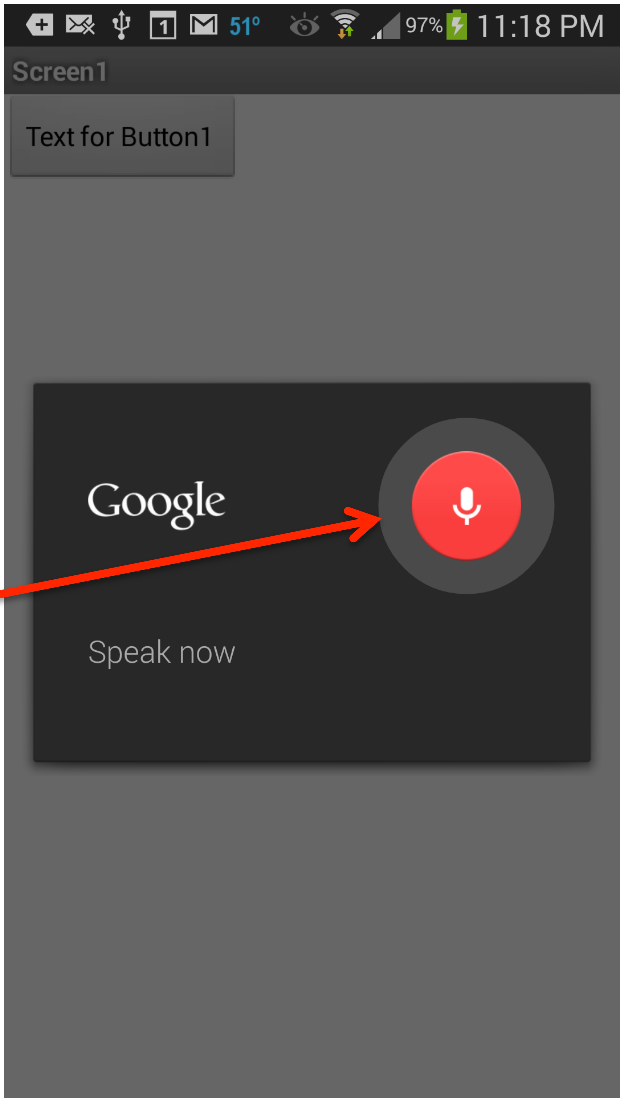
AI2 Demo: Blocks Using SpeechRecognizer



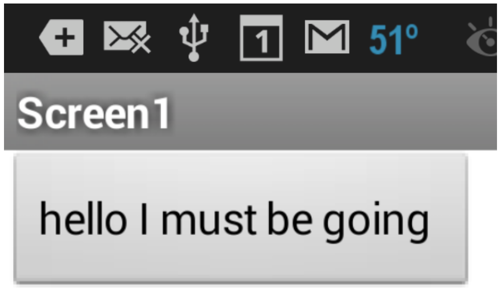
```
when Button1 .Click
do  call SpeechRecognizer1 .GetText

when SpeechRecognizer1 .AfterGettingText
  result
do  call TextToSpeech1 .Speak
                      message  get result
    set Button1 . Text to  get result
```

AI2 Demo: Live Development Test of SpeechRecognizer App





"hello I must be going"



"hello I must be going"

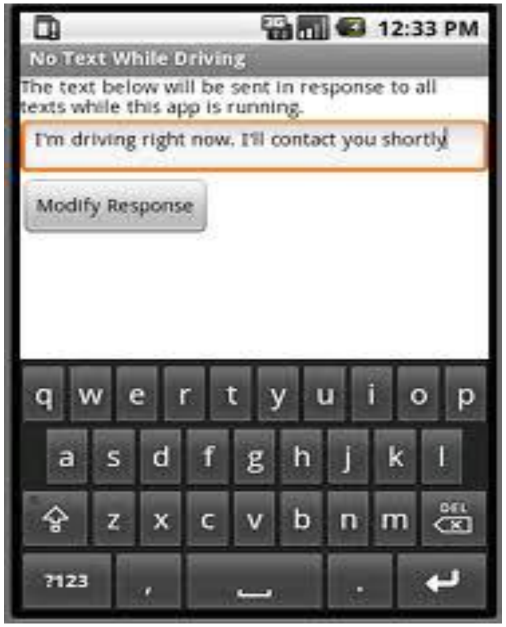
Talk Overview

- App Inventor 2 Demo
- App Inventor App Examples
- Situated Computing & Mobile Computational Thinking
- App Inventor 2 & Mobile Computational Thinking
- Looking Forward

USF CS107: Computing, Mobile Apps, and the Web

No Texting While Driving App

Daniel Finnegan.English Major

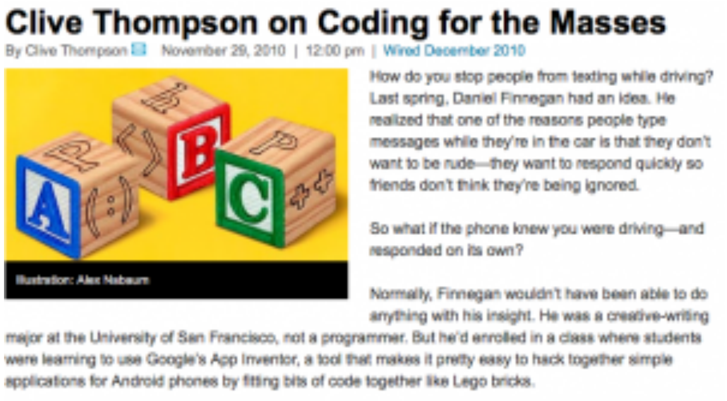


12:33 PM

No Text While Driving

The text below will be sent in response to all texts while this app is running.

I'm driving right now. I'll contact you shortly



Clive Thompson on Coding for the Masses

By Clive Thompson | November 29, 2010 | 12:00 pm | Wired December 2010

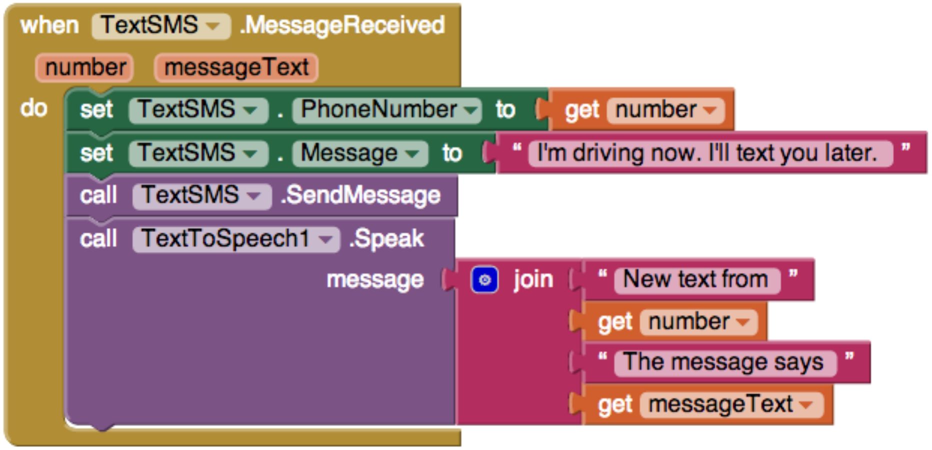
Illustration: Alex Nabaum

How do you stop people from texting while driving? Last spring, Daniel Finnegan had an idea. He realized that one of the reasons people type messages while they're in the car is that they don't want to be rude—they want to respond quickly so friends don't think they're being ignored.

So what if the phone knew you were driving—and responded on its own?

Normally, Finnegan wouldn't have been able to do anything with his insight. He was a creative-writing major at the University of San Francisco, not a programmer. But he'd enrolled in a class where students were learning to use Google's App Inventor, a tool that makes it pretty easy to hack together simple applications for Android phones by fitting bits of code together like Lego bricks.

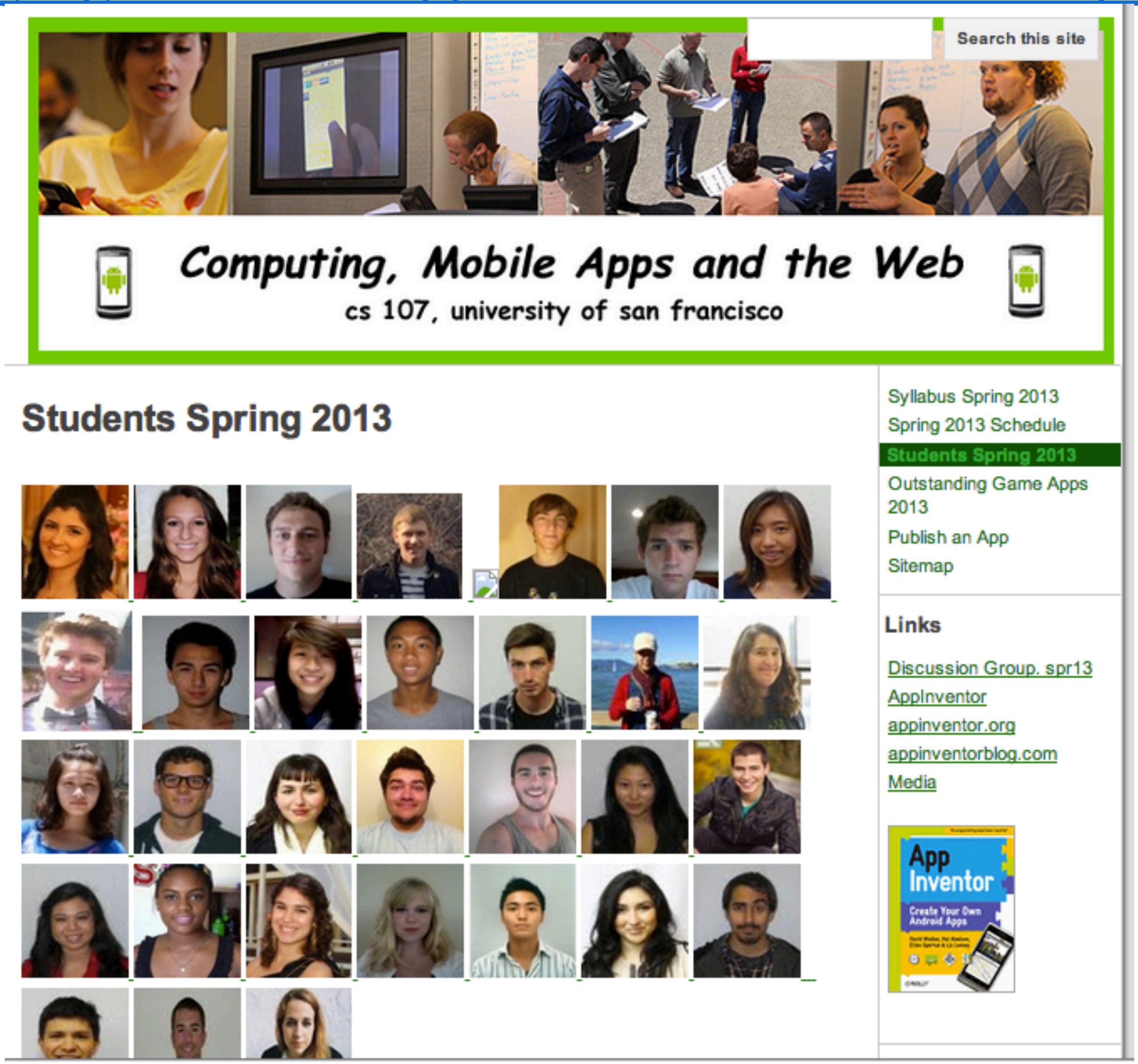
Daniel's code, translated into App Inventor 2



```
when TextSMS .MessageReceived
  number  messageText
do  set TextSMS . PhoneNumber to  get number
    set TextSMS . Message to  " I'm driving now. I'll text you later. "
    call TextSMS .SendMessage
    call TextToSpeech1 .Speak
                   message  join  " New text from "
                                  get number
                                  " The message says "
                                  get messageText
```

USF CS107 Spring 2013 Portfolios

https://sites.google.com/site/appinventorcourse/students-spring-2013



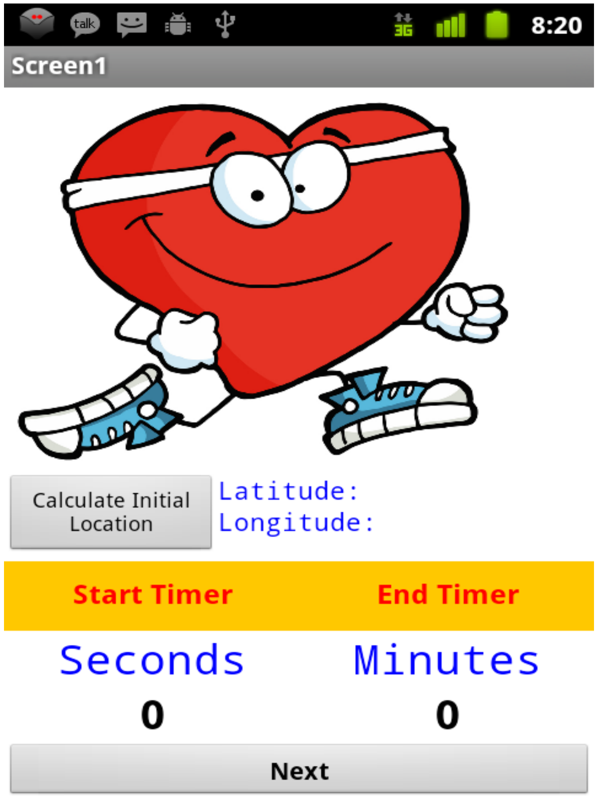
Computing, Mobile Apps and the Web

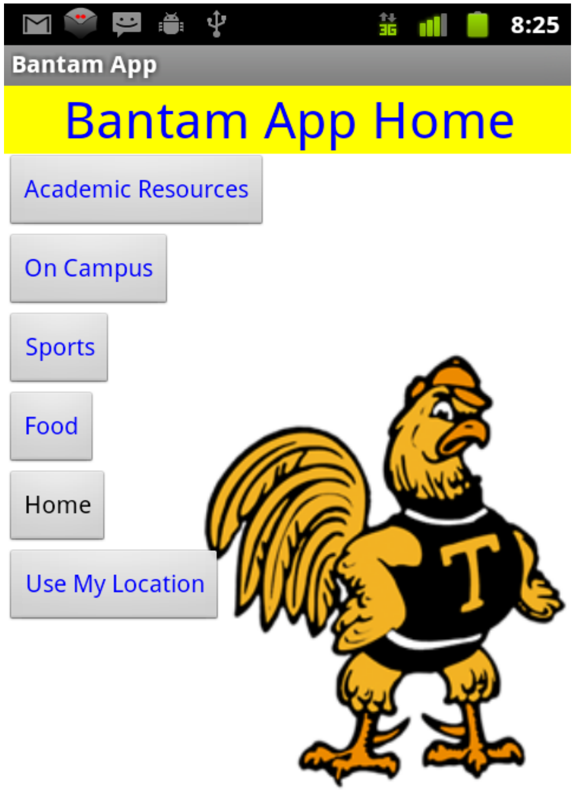
cs 107, university of san francisco

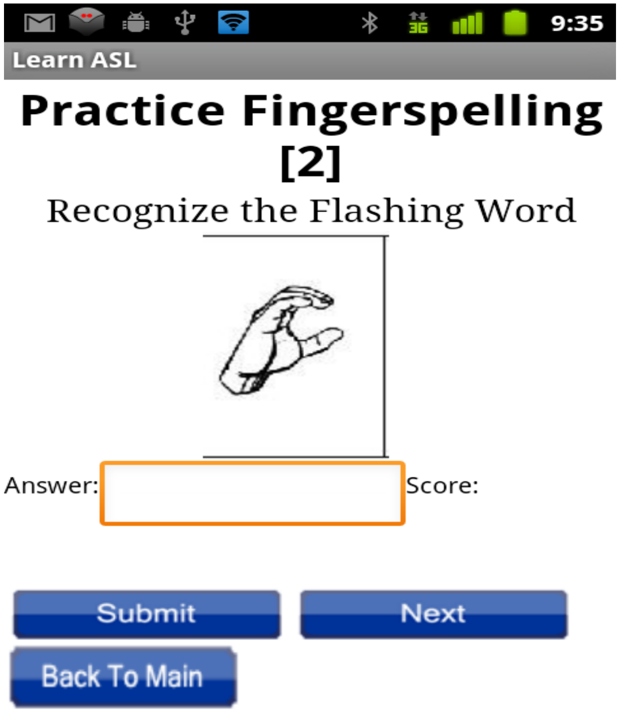
Students Spring 2013

Trinity CPSC 110 Computing with Mobile Phones

http://turing.cs.trincoll.edu/~ram/cpsc110/portfolios.html

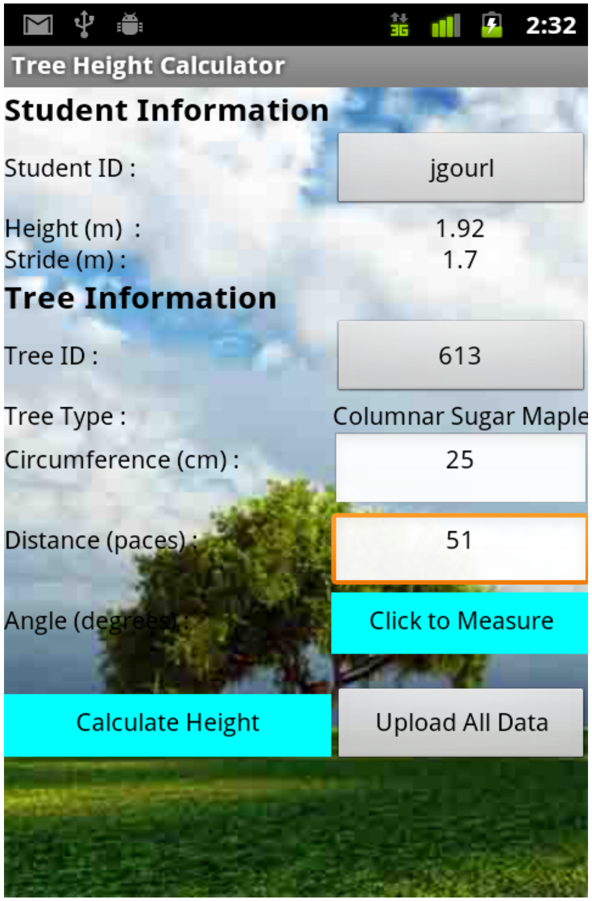






Trinity College: Tree Height Calculator

http://notes.hfoss.org/index.php/TreeCalc



Tree Height Calculator

Student Information

Student ID : jgourl

Height (m) : 1.92
Stride (m) : 1.7

Tree Information

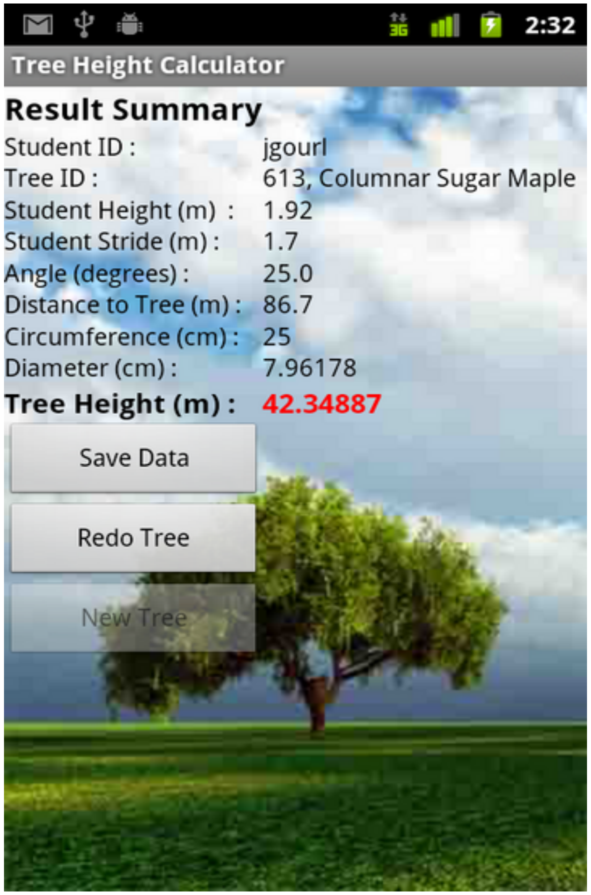
Tree ID : 613

Tree Type : Columnar Sugar Maple

Circumference (cm) : 25

Distance (paces) : 51

Angle (degrees) :



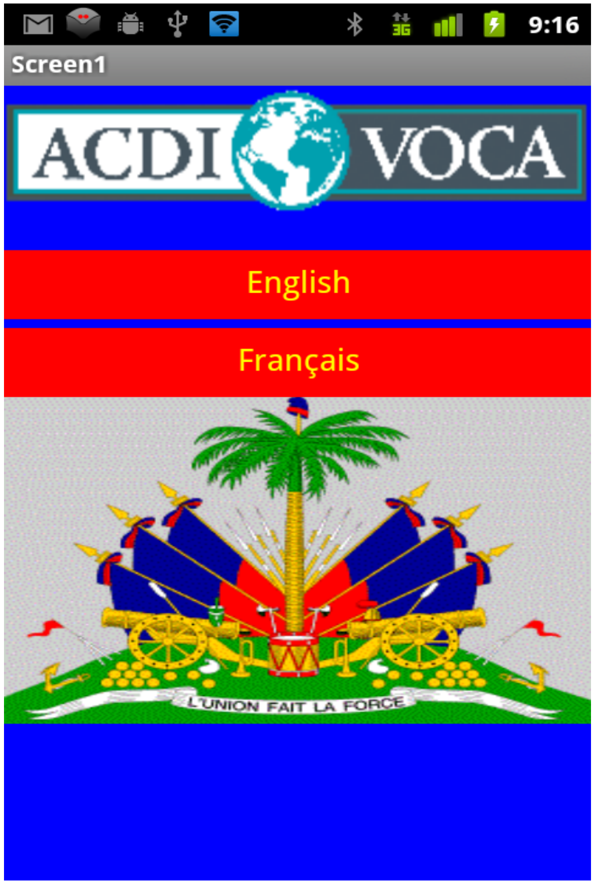
Tree Height Calculator

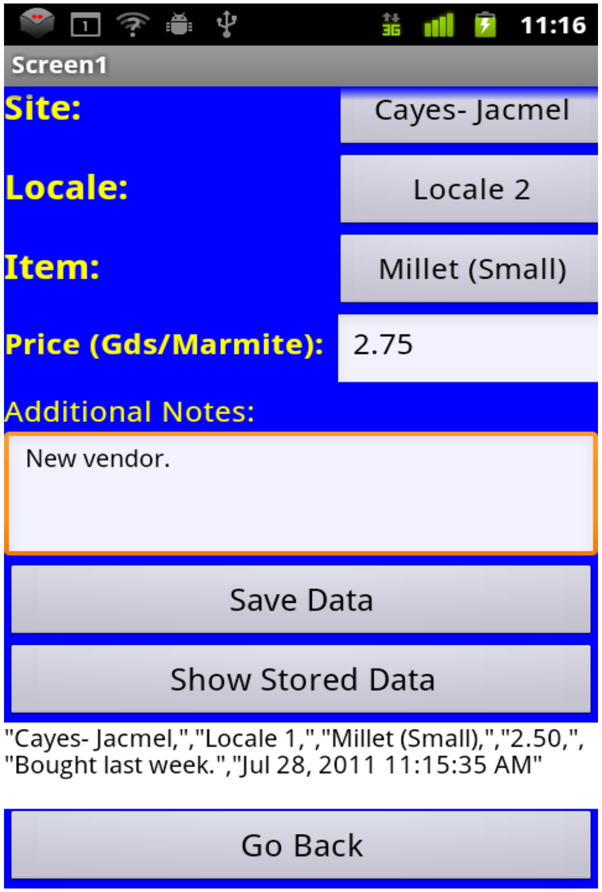
Result Summary

Student ID :	jgourl
Tree ID :	613, Columnar Sugar Maple
Student Height (m) :	1.92
Student Stride (m) :	1.7
Angle (degrees) :	25.0
Distance to Tree (m) :	86.7
Circumference (cm) :	25
Diameter (cm) :	7.96178
Tree Height (m) :	**42.34887**

Trinity College: Commodity Tracker App for Haiti

http://notes.hfoss.org/index.php/Haiti_Commodity_Collector





Site: Cayes- Jacmel

Locale: Locale 2

Item: Millet (Small)

Price (Gds/Marmite): 2.75

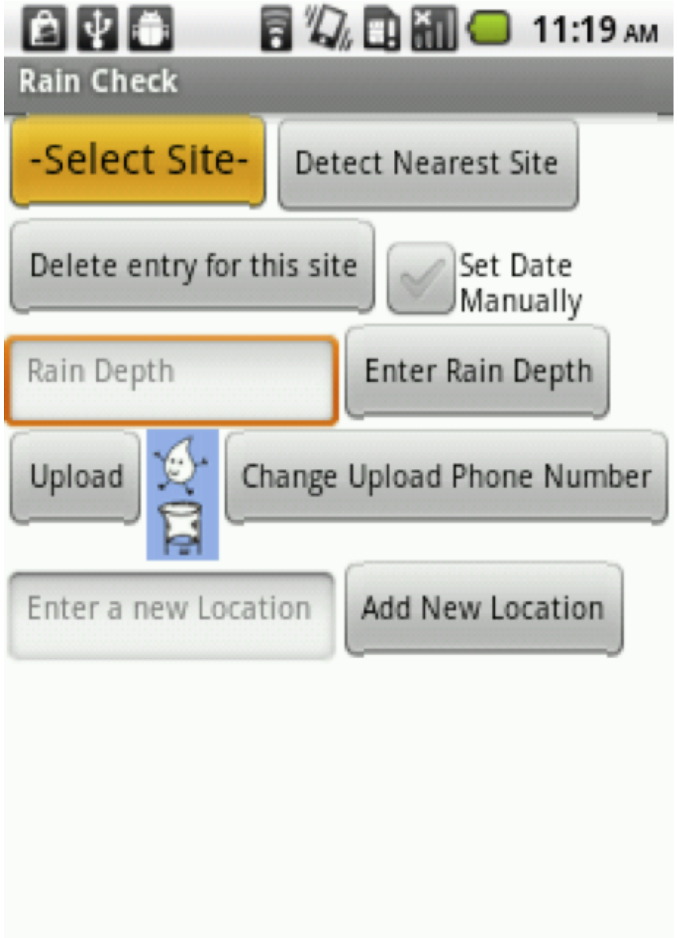
Additional Notes:

New vendor.

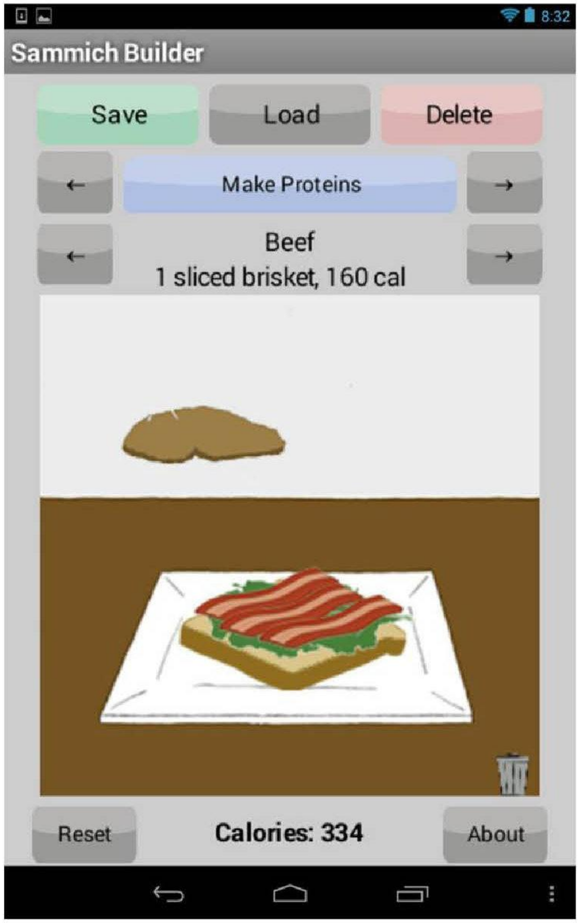
"Cayes- Jacmel,","Locale 1,","Millet (Small),","2.50,",
"Bought last week.","Jul 28, 2011 11:15:35 AM"

Trinity College: Rainfall Tracker App for Haiti

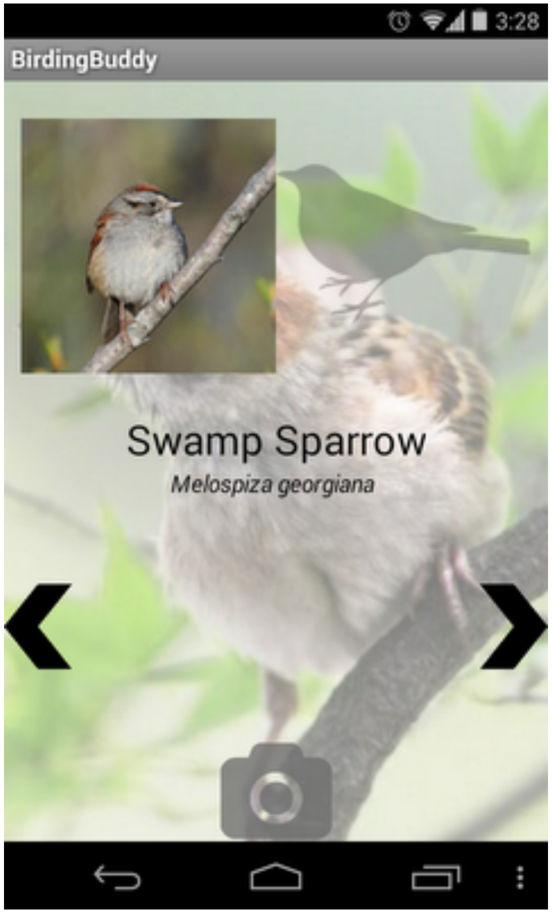
http://notes.hfoss.org/index.php/Rain_Check



UMass Lowell 91.108/70.108 Intro to App Design & Mobile Computing



Sammich Maker

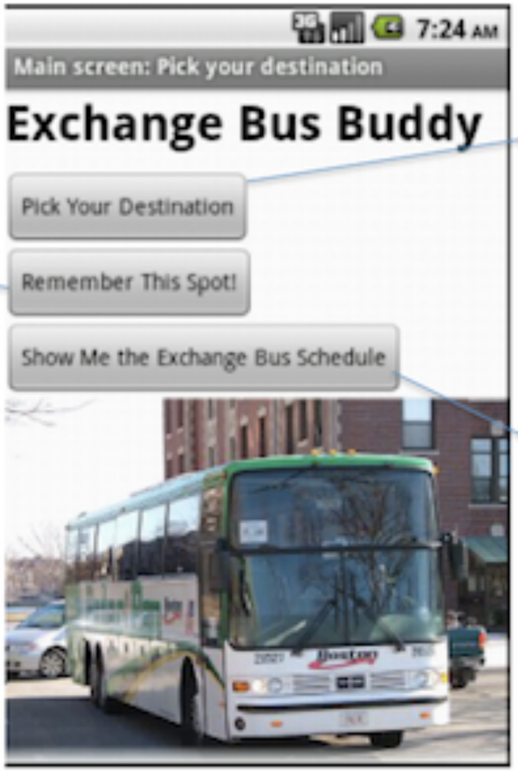


Birding Buddy

Wellesley CS117 Inventing Mobile Apps

galleries of location based-apps and web-service apps

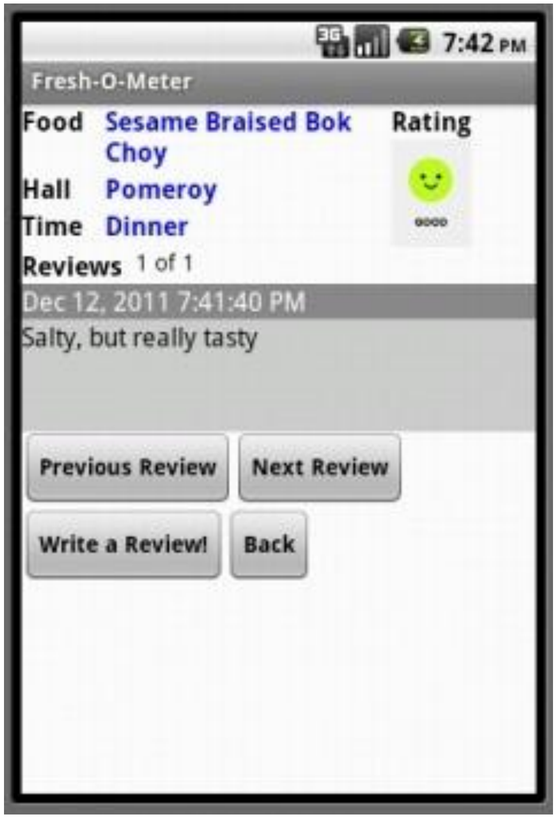
Exchange Bus Buddy



Main screen: Pick your destination

Exchange Bus Buddy

Wellesley Fresh-O-Meter



Fresh-O-Meter

Food Sesame Braised Bok Choy

Hall Pomeroy

Time Dinner

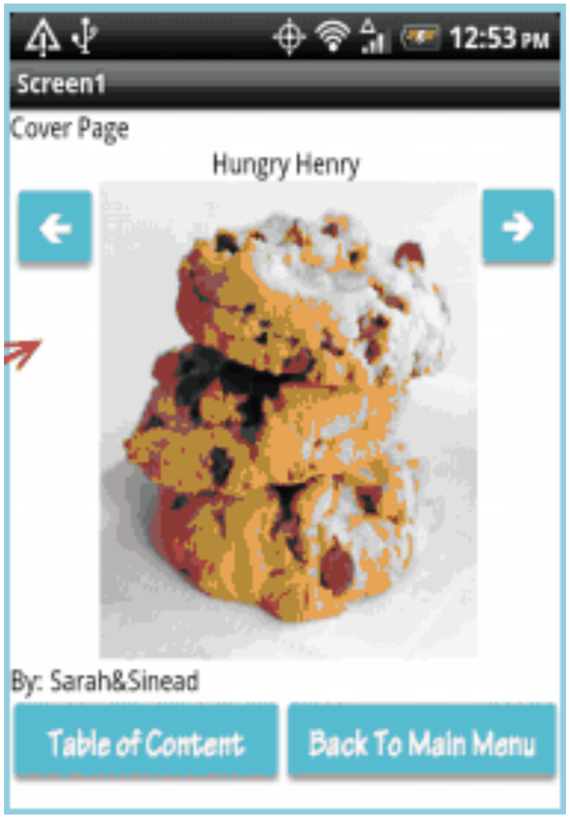
Rating

Reviews 1 of 1

Dec 12, 2011 7:41:40 PM

Salty, but really tasty

StoryBook



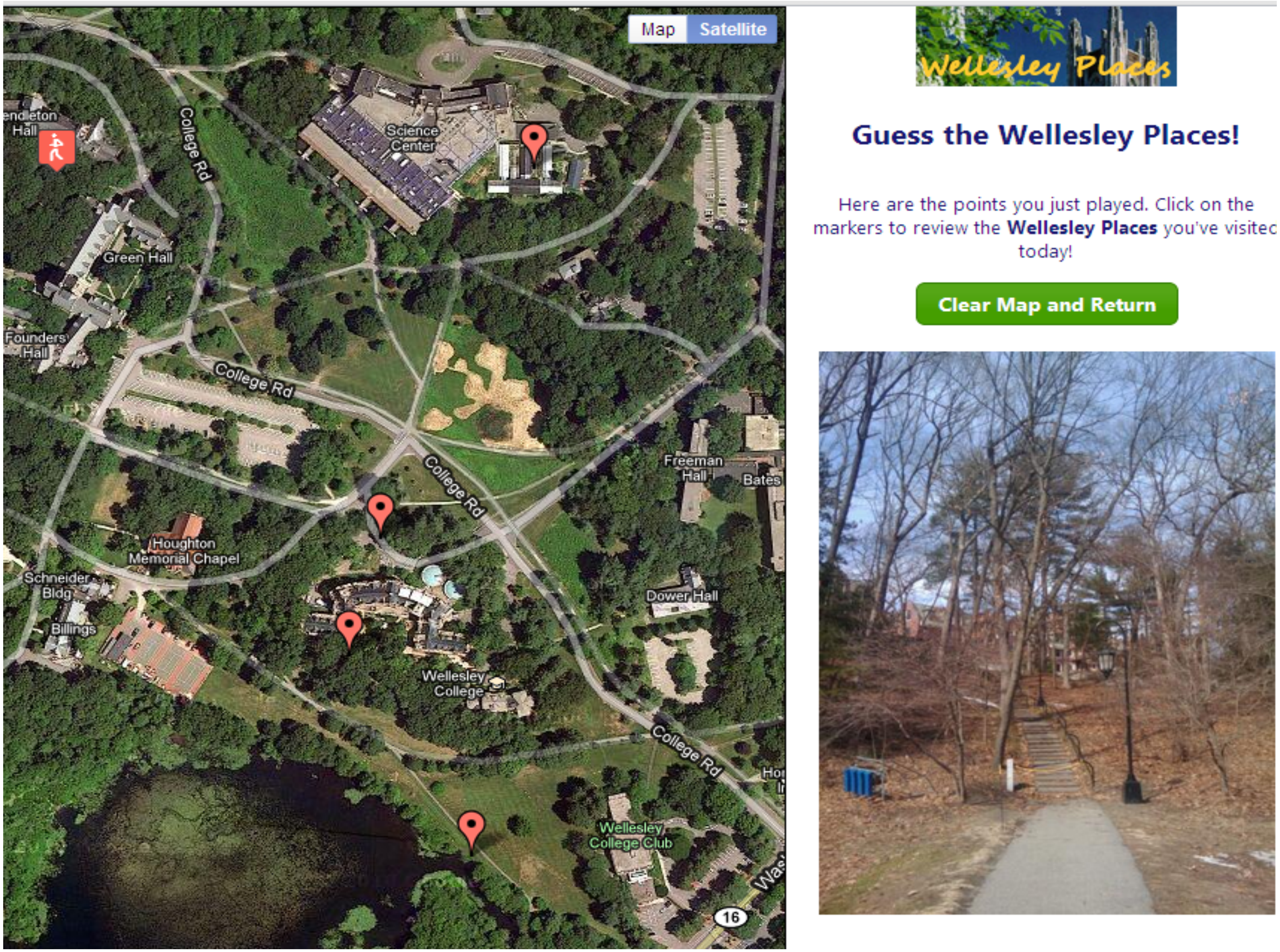
Screen1

Cover Page

Hungry Henry

By: Sarah&Sinead

Wellesley CS249 Web Mashups



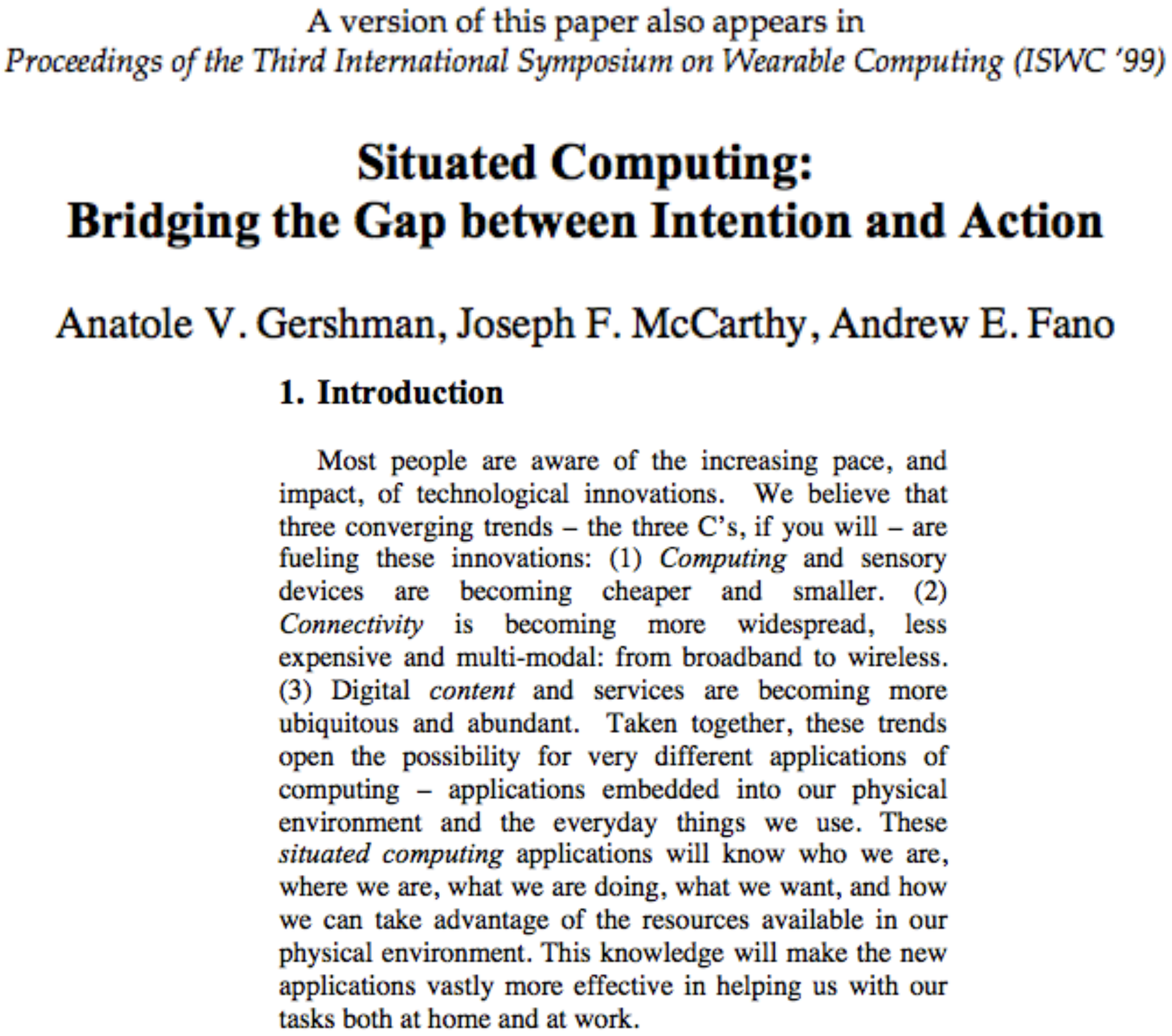
Guess the Wellesley Places!

Here are the points you just played. Click on the markers to review the **Wellesley Places** you've visitec today!

Talk Overview

- App Inventor 2 Demo
- App Inventor App Examples
- Situated Computing & Mobile Computational Thinking
- App Inventor 2 & Mobile Computational Thinking
- Looking Forward

Situated Computing and Convergence (1999)



A version of this paper also appears in
Proceedings of the Third International Symposium on Wearable Computing (ISWC '99)

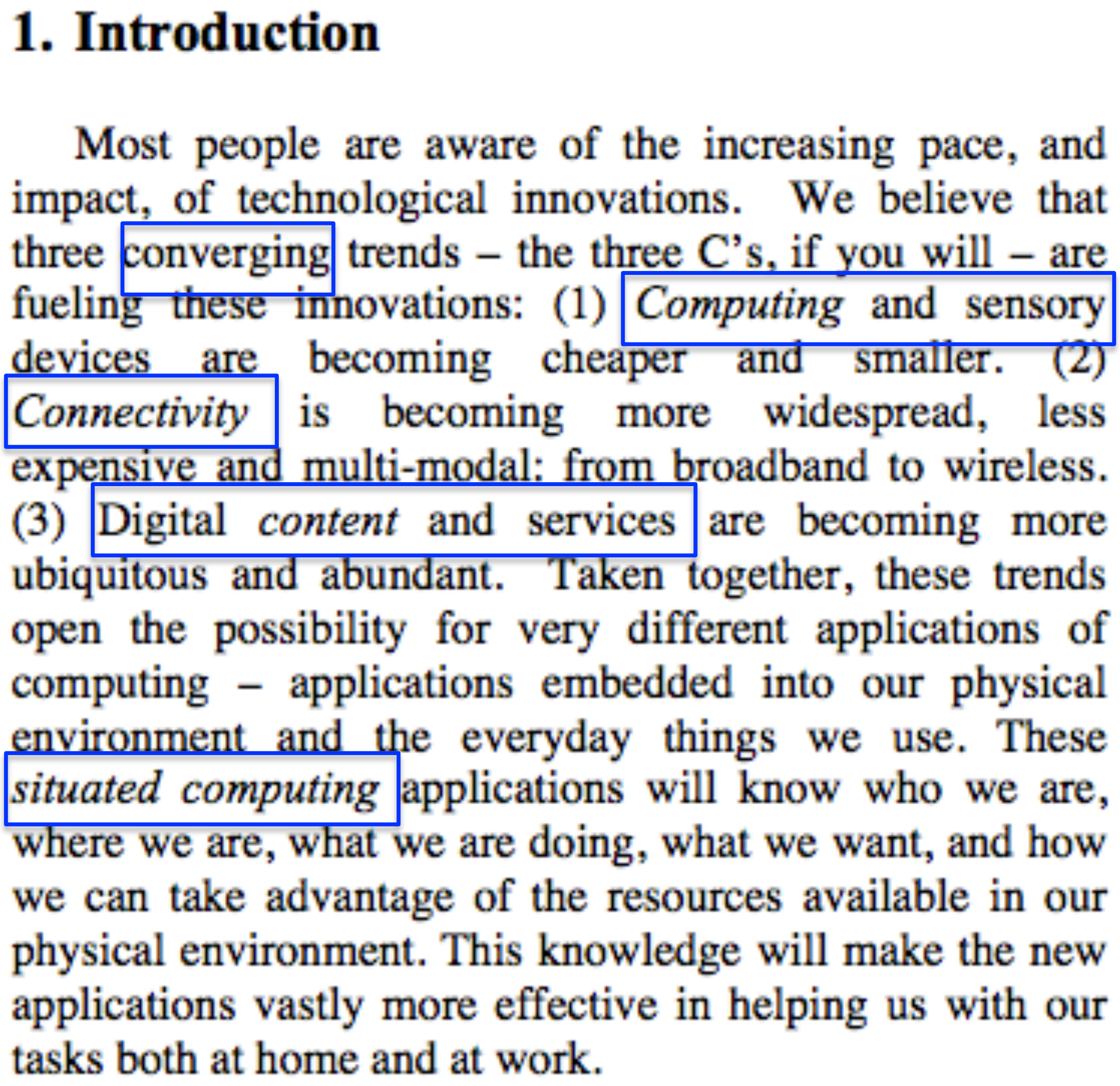
Situated Computing:
Bridging the Gap between Intention and Action

Anatole V. Gershman, Joseph F. McCarthy, Andrew E. Fano

1. Introduction

Most people are aware of the increasing pace, and impact, of technological innovations. We believe that three converging trends – the three C's, if you will – are fueling these innovations: (1) *Computing* and sensory devices are becoming cheaper and smaller. (2) *Connectivity* is becoming more widespread, less expensive and multi-modal: from broadband to wireless. (3) Digital *content* and services are becoming more ubiquitous and abundant. Taken together, these trends open the possibility for very different applications of computing – applications embedded into our physical environment and the everyday things we use. These *situated computing* applications will know who we are, where we are, what we are doing, what we want, and how we can take advantage of the resources available in our physical environment. This knowledge will make the new applications vastly more effective in helping us with our tasks both at home and at work.

Situated Computing and Convergence (1999)



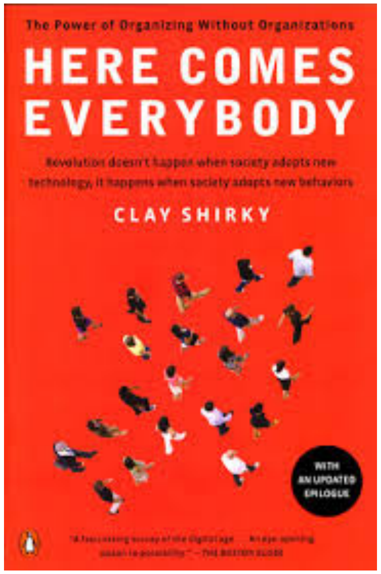
1. Introduction

Most people are aware of the increasing pace, and impact, of technological innovations. We believe that three converging trends – the three C's, if you will – are fueling these innovations: (1) *Computing* and sensory devices are becoming cheaper and smaller. (2) *Connectivity* is becoming more widespread, less expensive and multi-modal: from broadband to wireless. (3) Digital *content* and services are becoming more ubiquitous and abundant. Taken together, these trends open the possibility for very different applications of computing – applications embedded into our physical environment and the everyday things we use. These *situated computing* applications will know who we are, where we are, what we are doing, what we want, and how we can take advantage of the resources available in our physical environment. This knowledge will make the new applications vastly more effective in helping us with our tasks both at home and at work.

Clay Shirky on Situated Software vs. Web School (2004)

Target small population

- NYU ITP *Teachers on the Run*
  vs. RateMyProfessors.com
- scaling issues unimportant
- simple hardwired data vs. scalable databases
- software for your mom

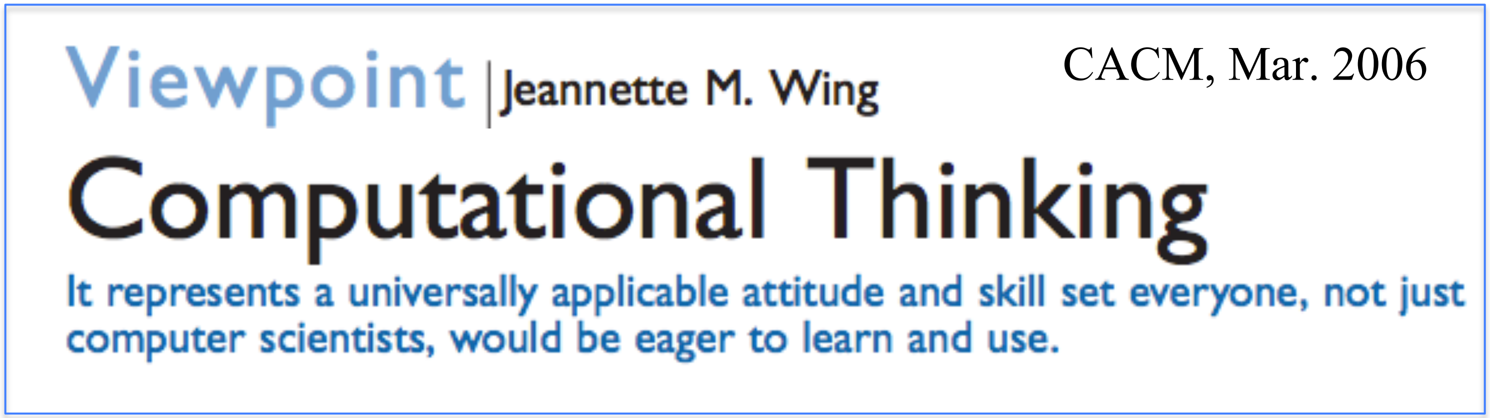

The Power of Organizing Without Organizations
HERE COMES EVERYBODY
Revolution doesn't happen when society adopts new technology, it happens when society adopts new behaviors
CLAY SHIRKY
WITH AN UPDATED EPILOGUE

Leverage small groups

- local knowledge
- trust of other users
- publicly shame deadbeats in group purchase apps

http://shirky.com/writings/herecomeseverybody/situated_software.html

Computational Thinking



Viewpoint | Jeannette M. Wing

CACM, Mar. 2006

Computational Thinking

It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.

CS PRINCIPLES

www.csprinciples.org

Mobile CSP

Mobile CSP

Computer Science Principles

Ralph Morelli's Mobile CSP in App Inventor resources: `mobile-csp.org`

Principles of Mobile Computational Thinking (MCT)

1. Leverages features that situate app in the world.

2. Requires event-oriented behavior.

3. Emphasizes useful programs embedded in a social context.

4. Takes advantage of larger informational ecosystem.

5. Involves design, engineering, and entrepreneurship.

Talk Overview

- App Inventor 2 Demo
- App Inventor App Examples
- Situated Computing & Mobile Computational Thinking
- App Inventor 2 & Mobile Computational Thinking
- Looking Forward

App Inventor & Mobile Computational Thinking

1. Visual blocks language, cloud-based environment, and live programming with connected device lower barriers to programming.

2. High-level abstractions for mobile device features facilitates creating situated apps

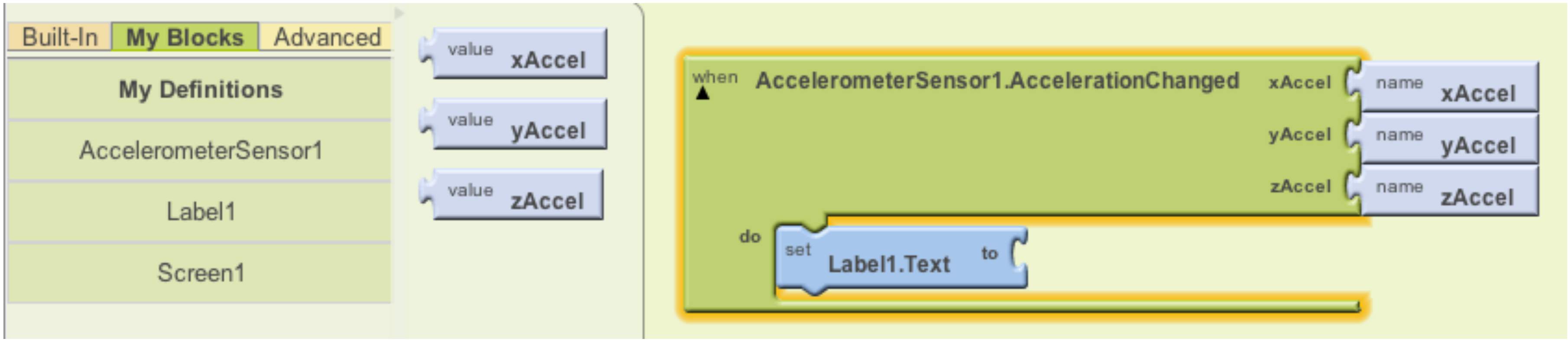
3. Simple approach to event handling makes it easy to specify app behavior.

4. Advantages of App Inventor 2 over App Inventor Classic:

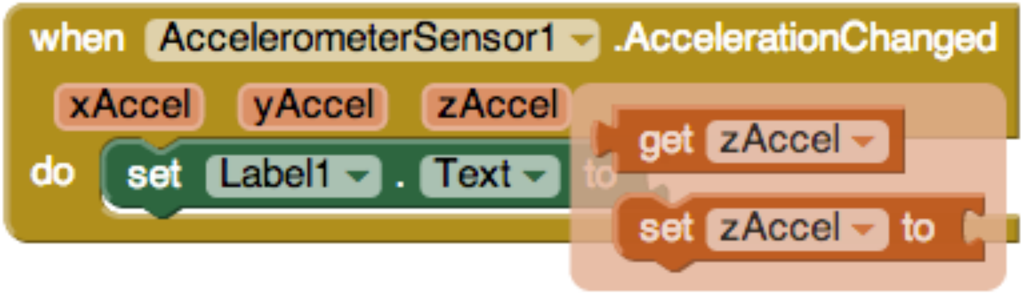
   - Browser-based blocks editor
   - Mutators
   - Improved naming features illustrating CS principles

Referencing an Event Parameter

AI Classic

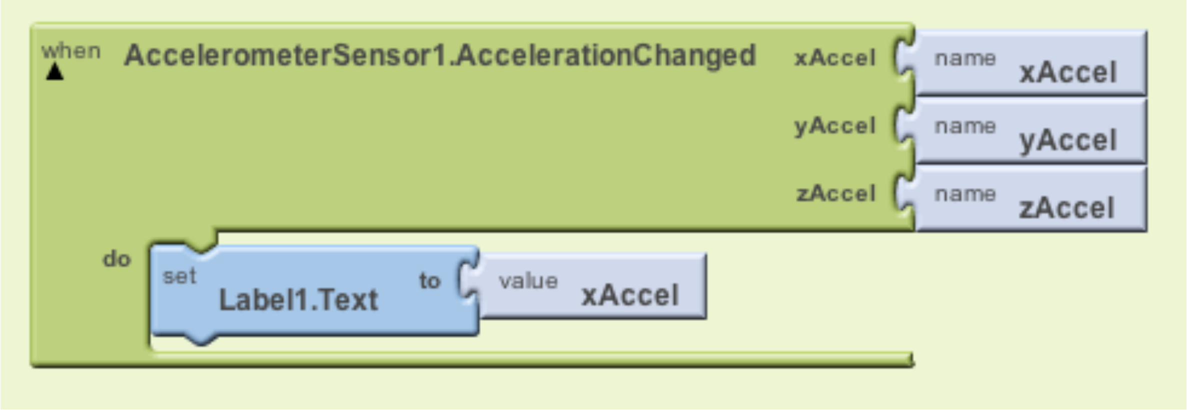


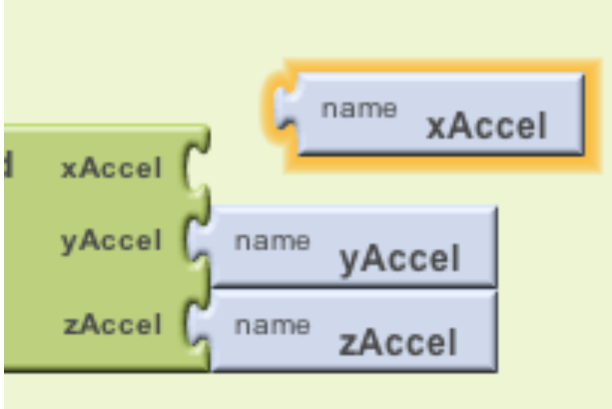
AI2



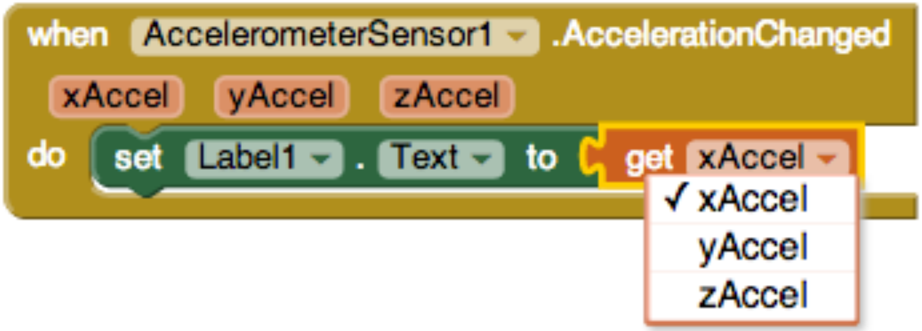
More Event Parameters

AI Classic

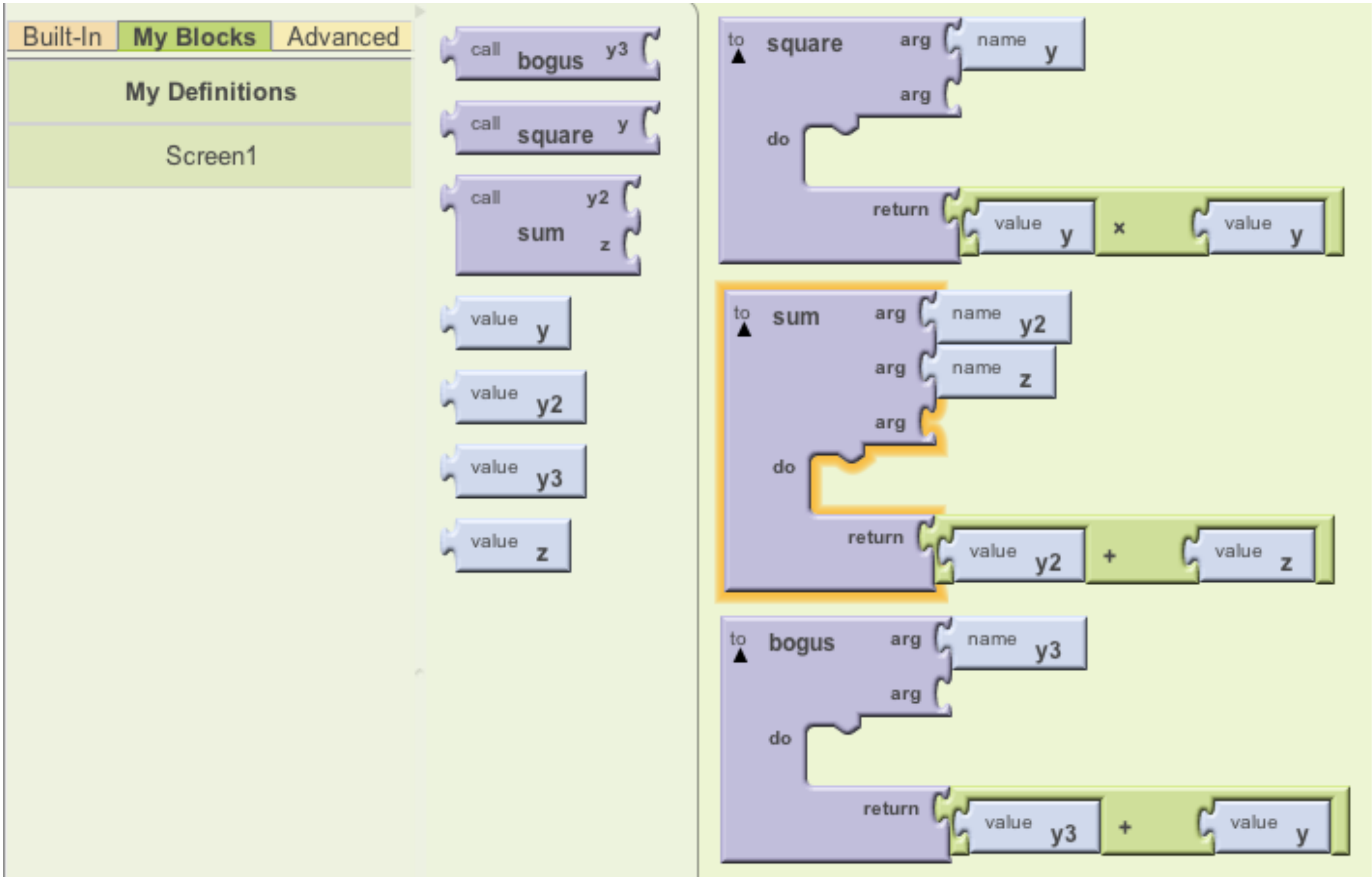




AI2

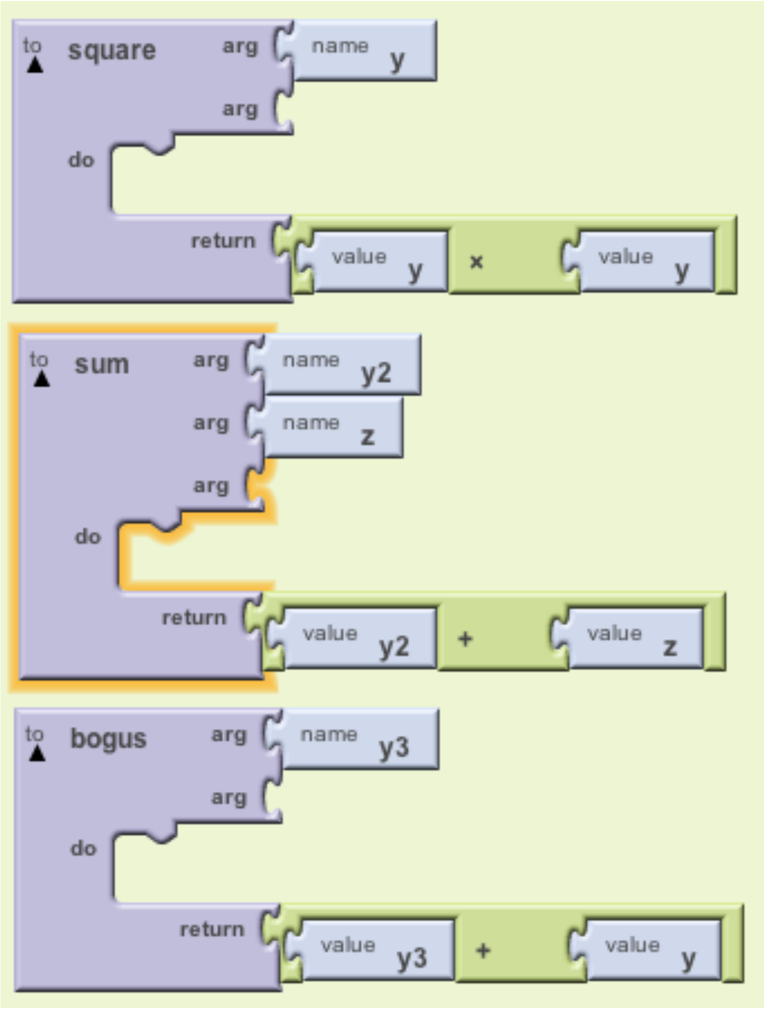

when AccelerometerSensor1 .AccelerationChanged
xAccel yAccel zAccel
do set Label1 . Text to get xAccel
✓ xAccel
yAccel
zAccel

Procedure parameters in AI Classic



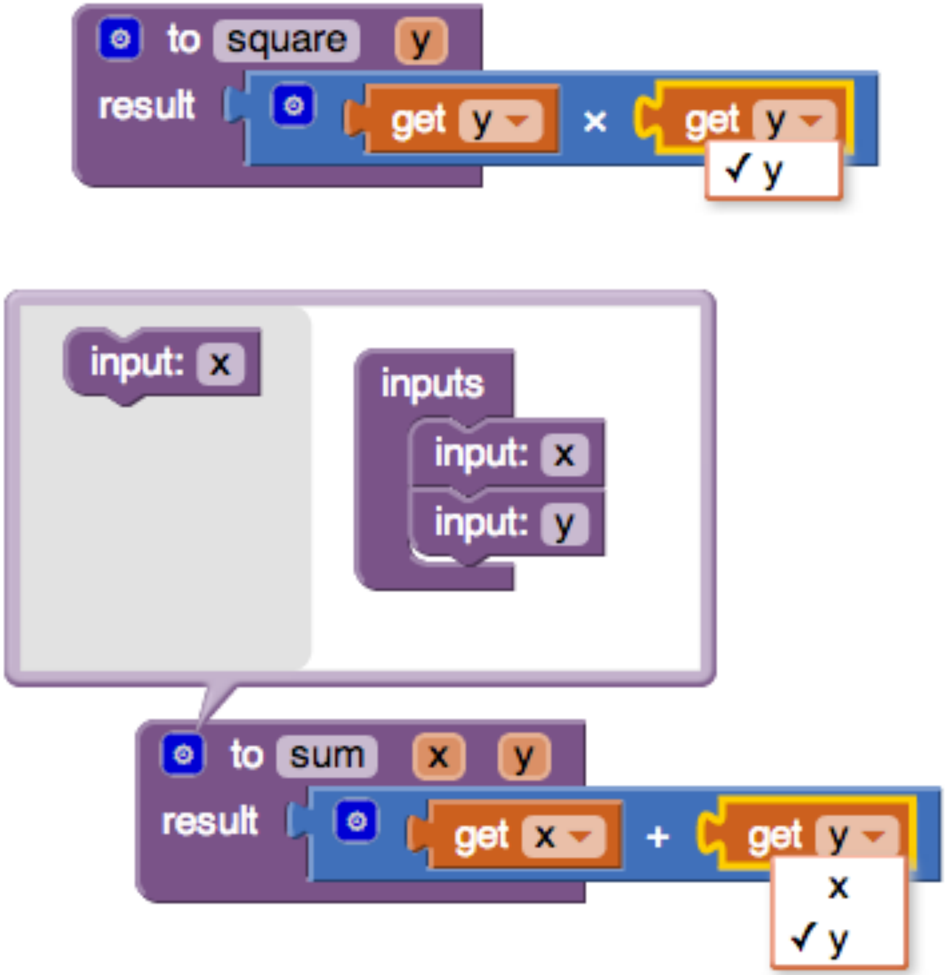
Procedure parameters in AI2

AI Classic



```
to square   arg  name y
            arg
do
return  value y  ×  value y

to sum   arg  name y2
         arg  name z
         arg
do
return  value y2  +  value z

to bogus   arg  name y3
           arg
do
return  value y3  +  value y
```

AI2



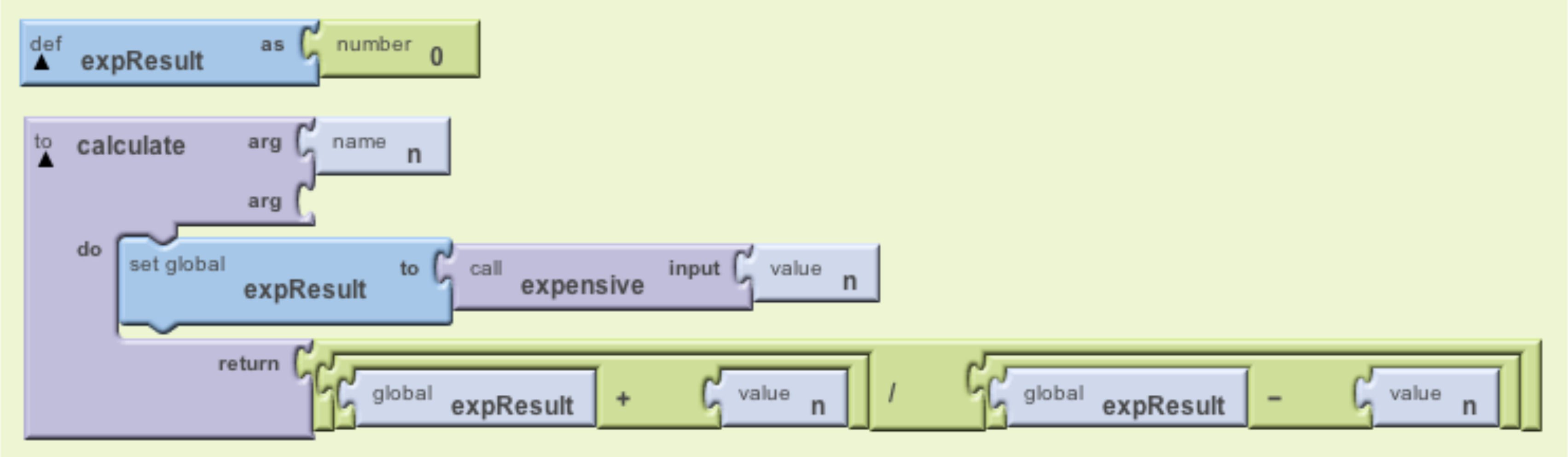
```
to square  y
result  get y  ×  get y
                    ✓ y
```

```
input: x

inputs
  input: x
  input: y
```

```
to sum  x  y
result  get x  +  get y
                    x
                    ✓ y
```

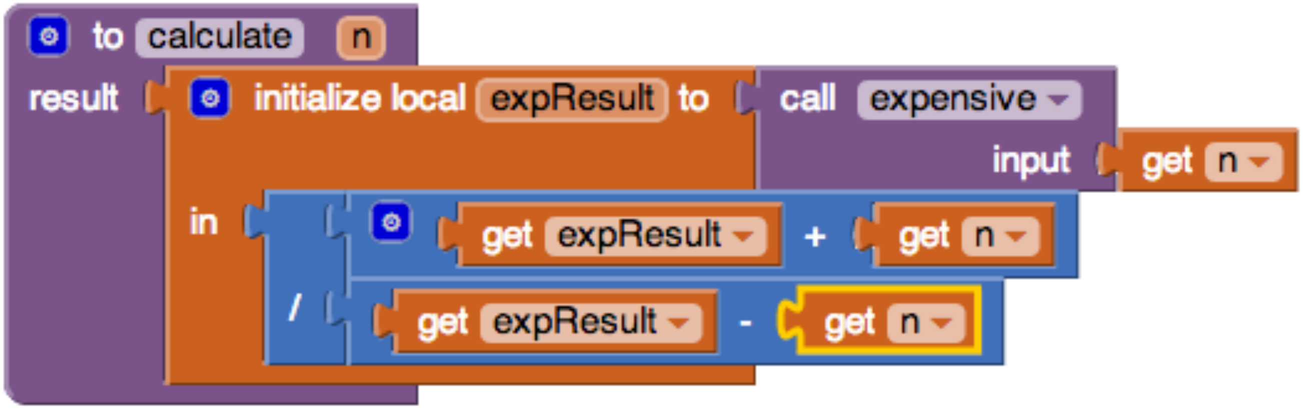
Local variables in AI2

AI Classic: only global vars



```
def expResult as number 0

to calculate arg name n
             arg
do   set global expResult to call expensive input value n
return global expResult + value n / global expResult - value n
```

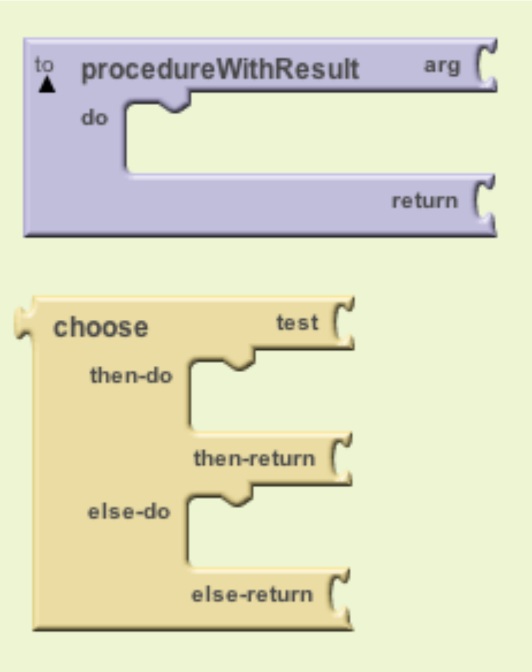
AI2: includes local vars



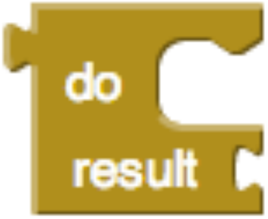
```
to calculate n
result  initialize local expResult to call expensive
                                        input get n
        in   get expResult + get n
             / get expResult - get n
```

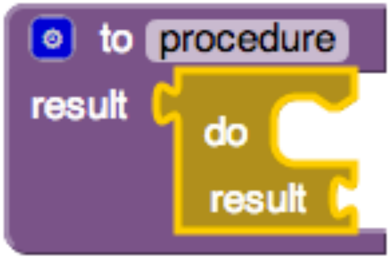
Performing actions before returning value

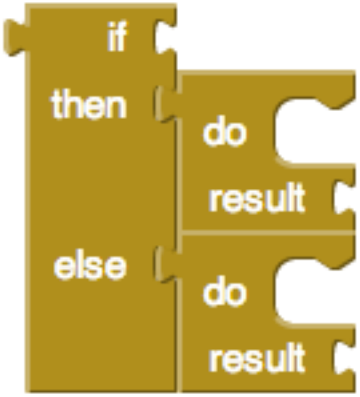
AI Classic

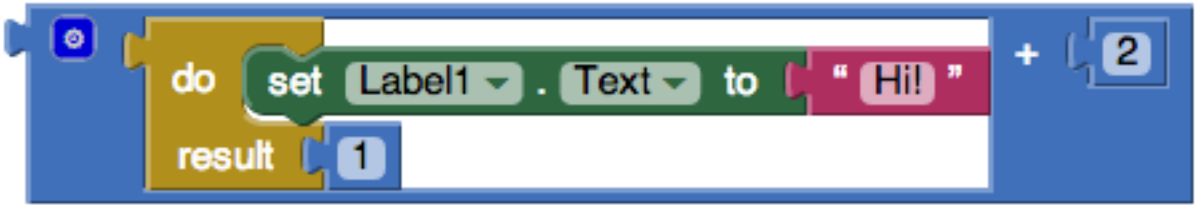


AI2

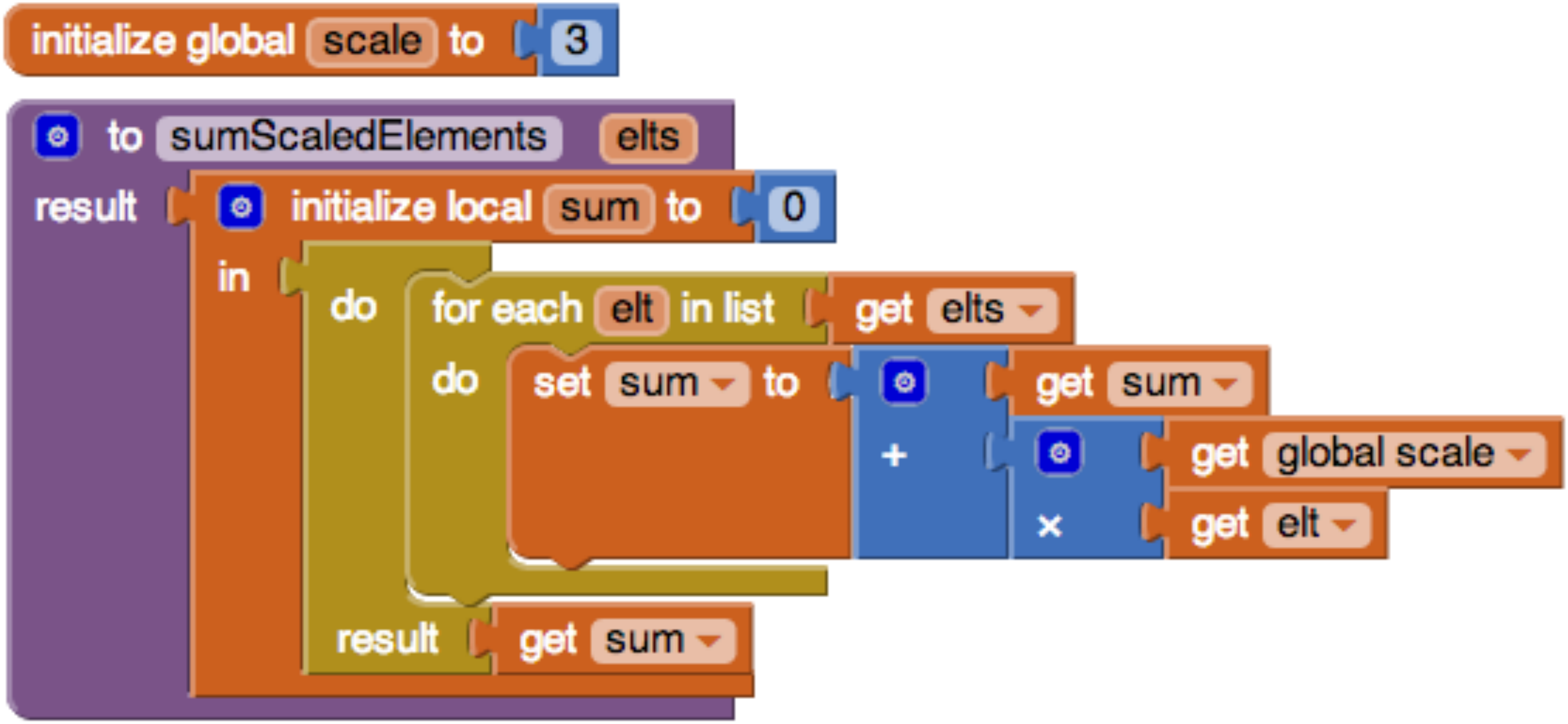








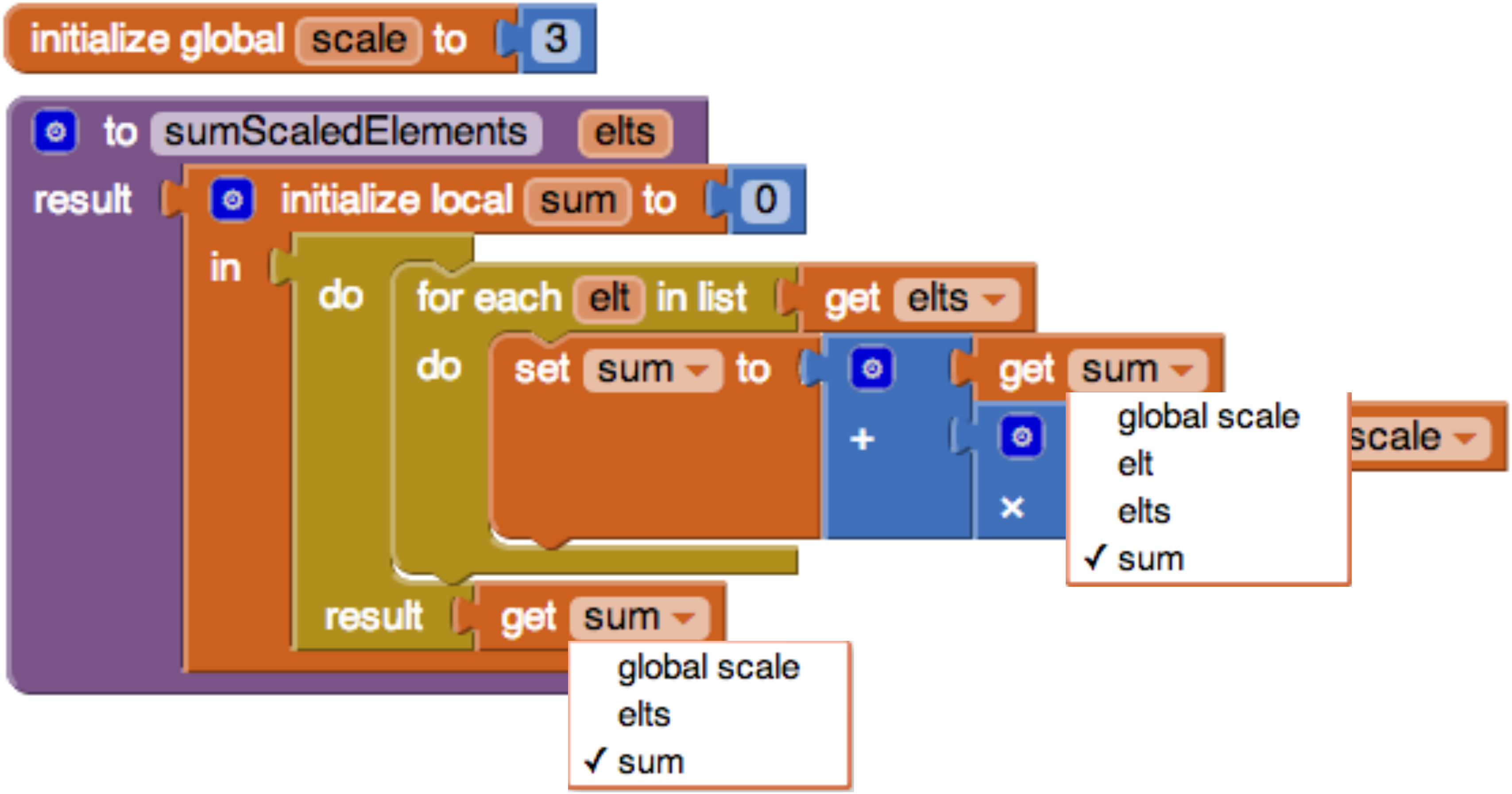
All together now



```
initialize global scale to 3

to sumScaledElements elts
result  initialize local sum to 0
        in  do  for each elt in list  get elts
                do  set sum to  + get sum
                                  × get global scale
                                    get elt
            result  get sum
```

Name scoping in AI2

- Globals are in a separate namespace
- Indentation visually highlights area of name scope
- Can change any variable value, including procedure inputs
- Inner names can shadow outer ones



```
initialize global scale to 3

to sumScaledElements elts
result  initialize local sum to 0
        in  do  for each elt in list  get elts
                do  set sum to  +  get sum
                                   ×  scale
            result  get sum
```

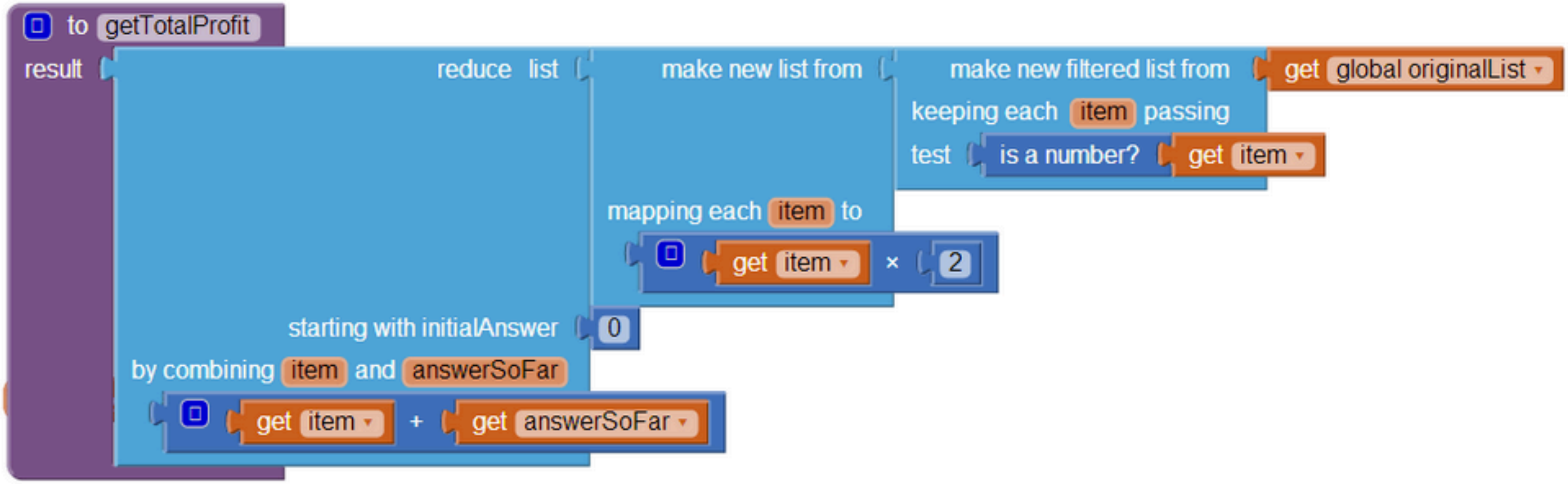
Talk Overview

- App Inventor 2 Demo
- App Inventor App Examples
- Situated Computing & Mobile Computational Thinking
- App Inventor 2 & Mobile Computational Thinking
- Looking Forward

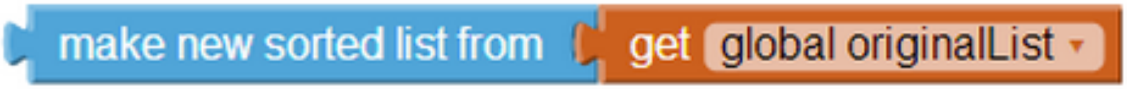
AI2 Coming Improvements

- AI1 to AI2 conversion
- Higher-order list operations (sort, map, filter, reduce)
- Better error handling and debugging
- Conversion between blocks and textual code
- Backpack for copying blocks code
- Dictionary datatype
- Background processes?
- Easier cloud data?

Higher-order List Operations (Soojin Kim)



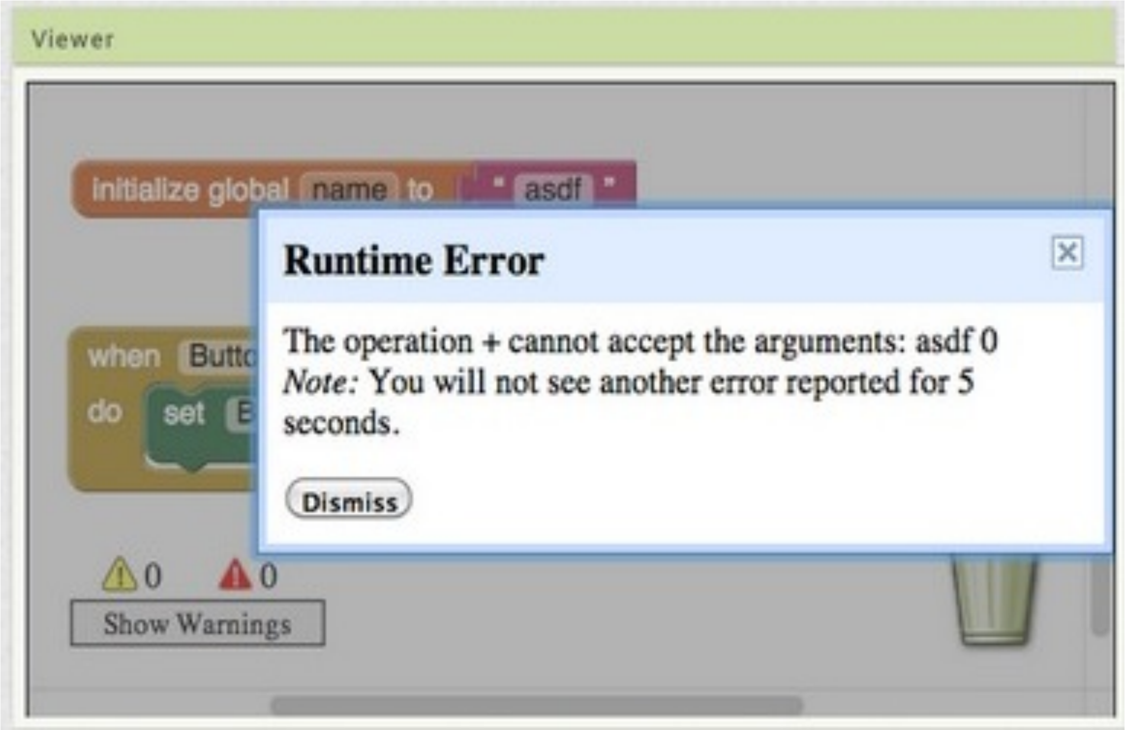
```
to getTotalProfit
result  reduce list  make new list from  make new filtered list from  get global originalList
                                         keeping each item passing
                                         test  is a number?  get item
                     mapping each item to
                       get item × 2
        starting with initialAnswer  0
        by combining item and answerSoFar
          get item + get answerSoFar
```



```
make new sorted list from  get global originalList
```

Better Error Handling (Johanna Okerlund)

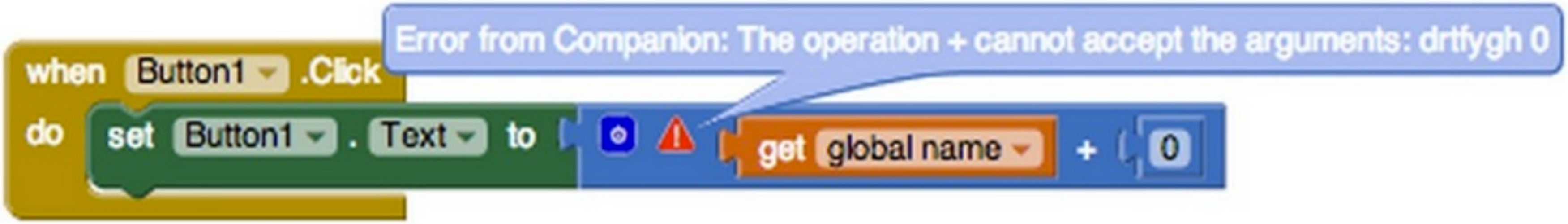
Currently, AI2 error window covers blocks and does not pinpoint block causing error:



Runtime Error

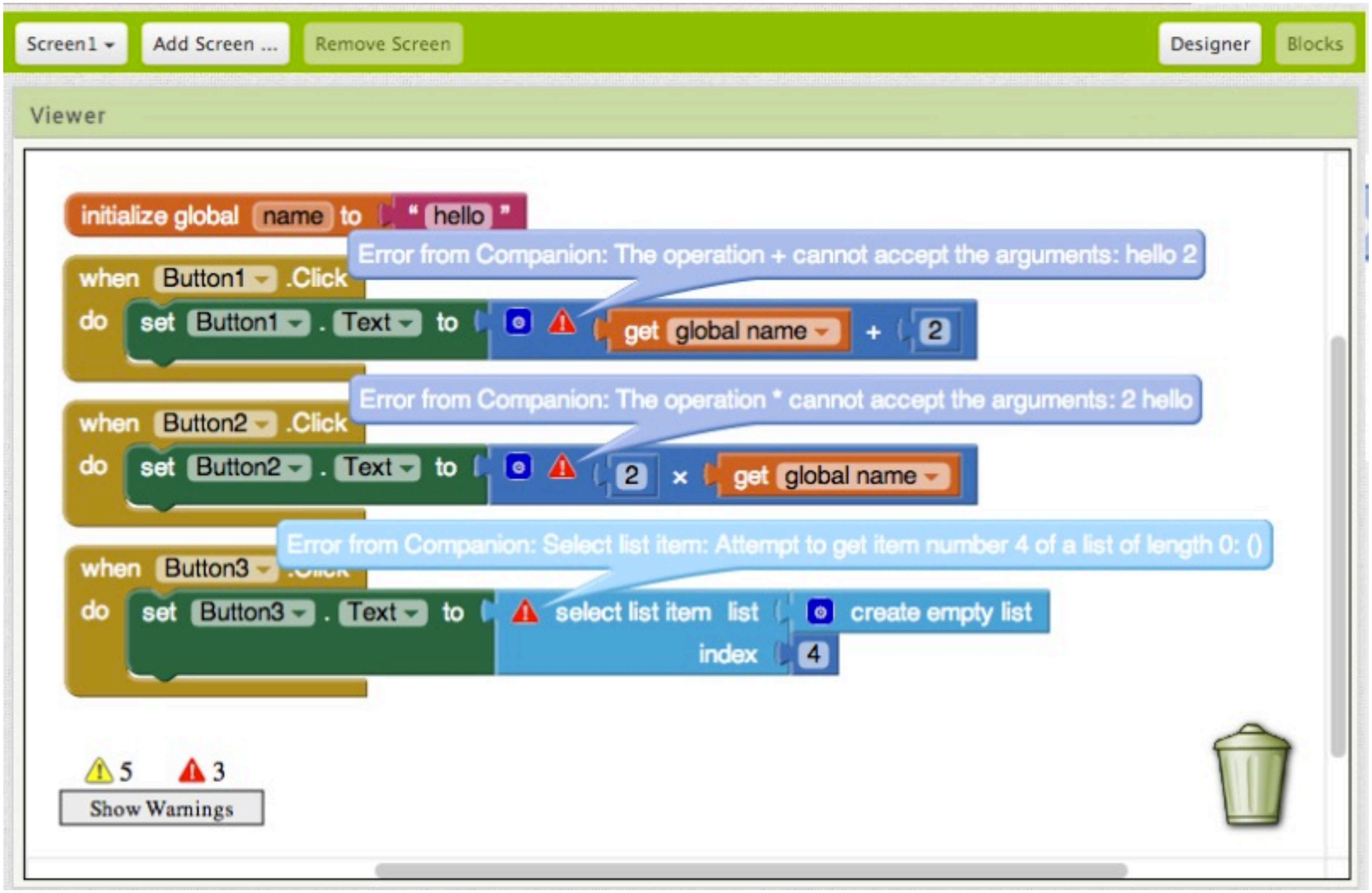
The operation + cannot accept the arguments: asdf 0
Note: You will not see another error reported for 5 seconds.

Soon, the error will appear on the block causing the error:


Error from Companion: The operation + cannot accept the arguments: drtfygh 0

Better Error Handling (Johanna Okerlund)

Error messages can appear on multiple blocks until the errors are fixed:

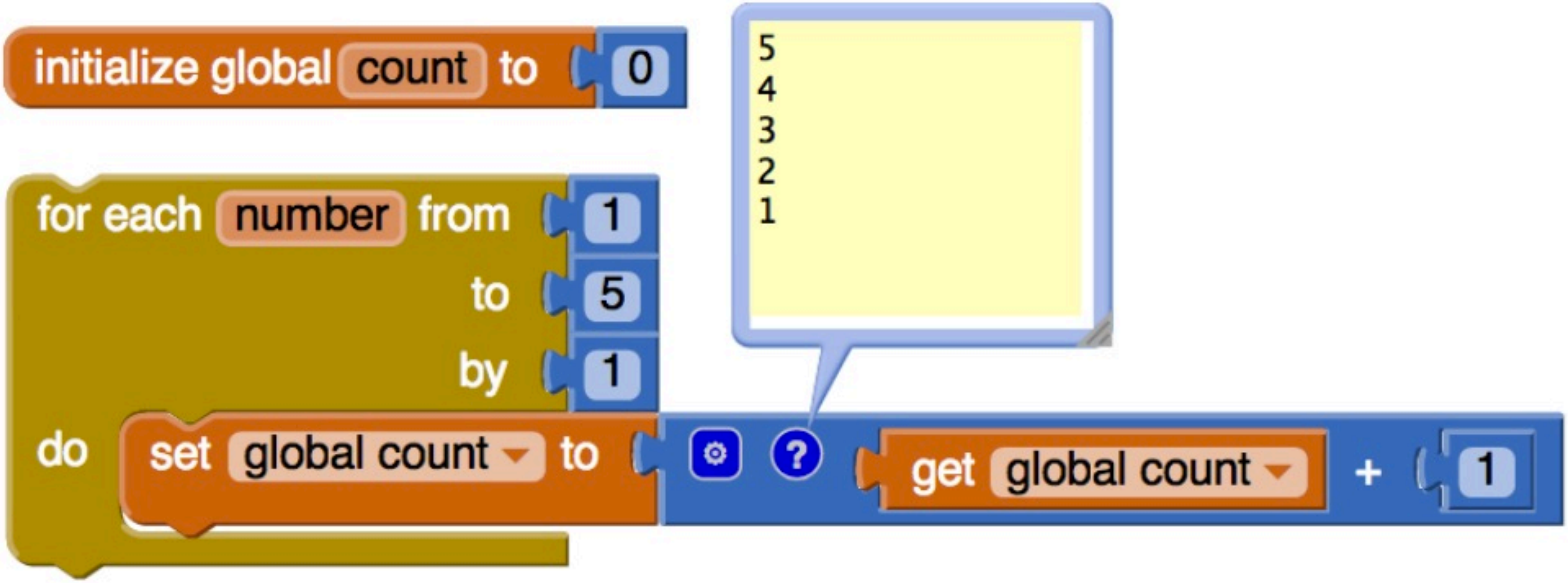


Error from Companion: The operation + cannot accept the arguments: hello 2

Error from Companion: The operation * cannot accept the arguments: 2 hello

Error from Companion: Select list item: Attempt to get item number 4 of a list of length 0: ()

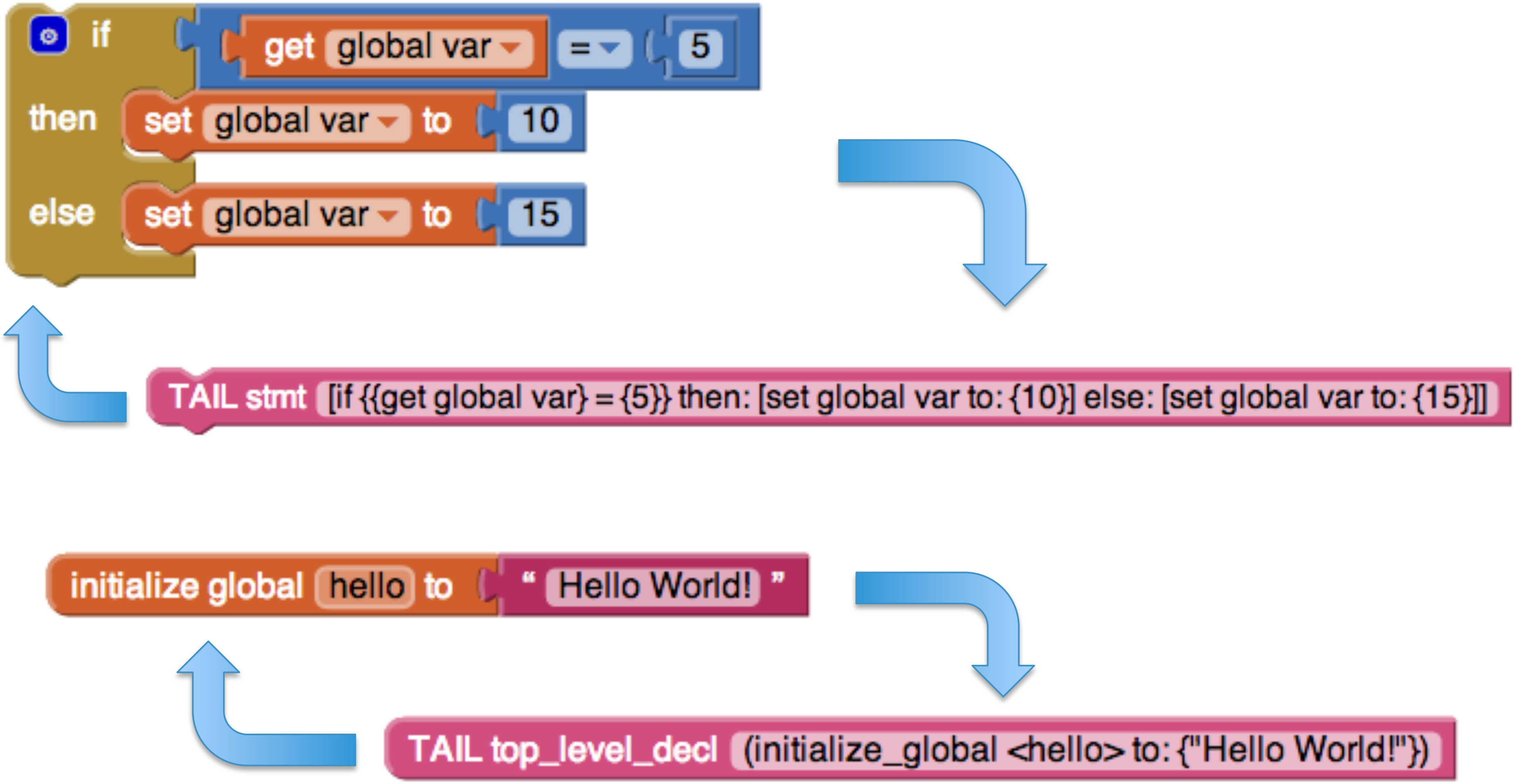
Better Debugging: Watch (Johanna Okerlund)



```
initialize global count to 0

for each number from 1
                  to 5
                  by 1
do  set global count to   get global count + 1
```

```
5
4
3
2
1
```

Conversion Between Blocks and Textual Code (Karishma Chadha)



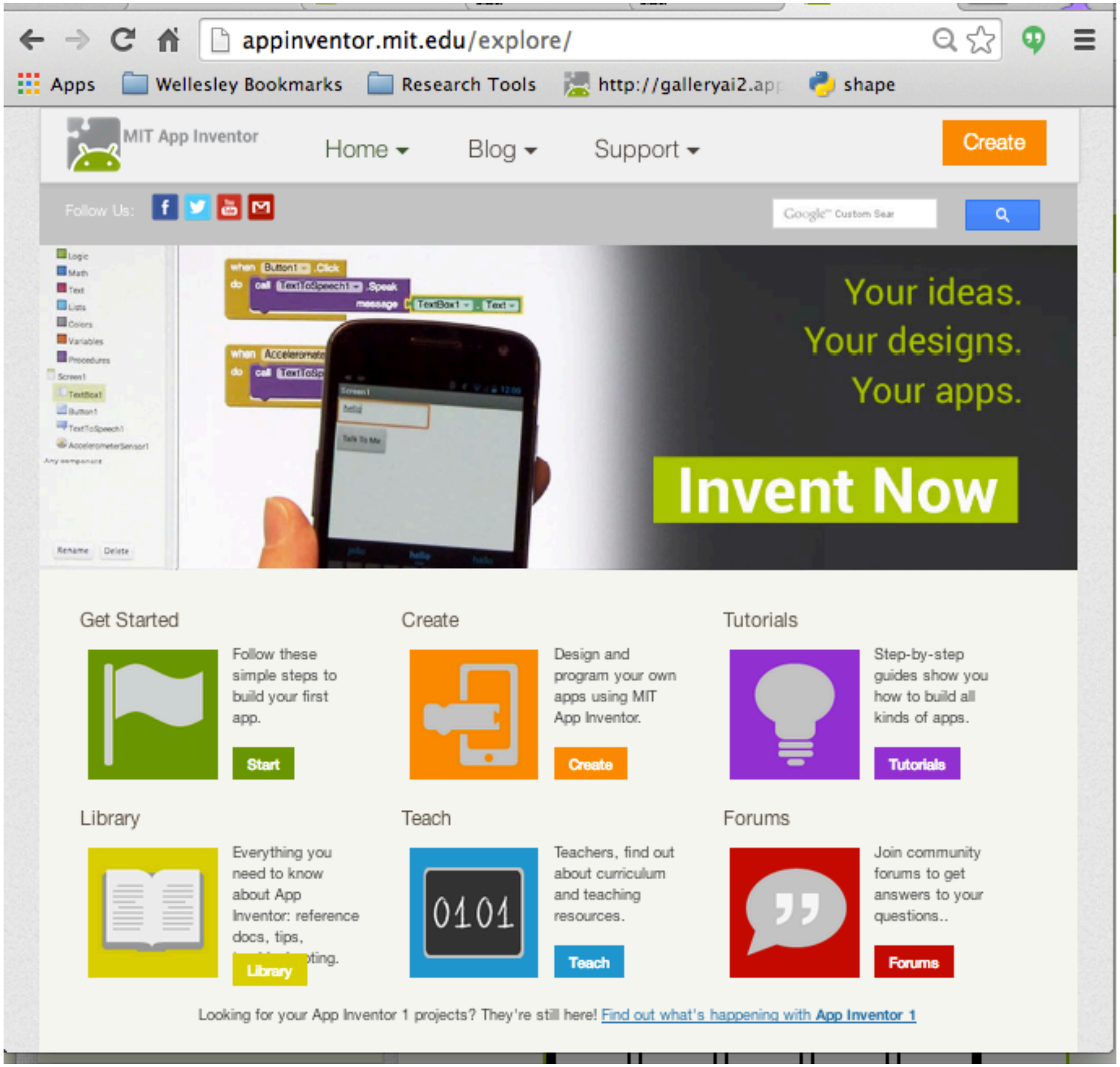
```
if   get global var = 5
then set global var to 10
else set global var to 15
```

```
TAIL stmt [if {{get global var} = {5}} then: [set global var to: {10}] else: [set global var to: {15}]]
```

```
initialize global hello to " Hello World! "
```

```
TAIL top_level_decl (initialize_global <hello> to: {"Hello World!"})
```

Thank You! Questions?



appinventor.mit.edu/explore/

Apps | Wellesley Bookmarks | Research Tools | http://galleryai2.app | shape

MIT App Inventor

Home ▾ Blog ▾ Support ▾ Create

Follow Us:

Google™ Custom Sear

Your ideas.
Your designs.
Your apps.

Invent Now

Get Started

Follow these simple steps to build your first app.

Start

Create

Design and program your own apps using MIT App Inventor.

Create

Tutorials

Step-by-step guides show you how to build all kinds of apps.

Tutorials

Library

Everything you need to know about App Inventor: reference docs, tips, oting.

Library

Teach

Teachers, find out about curriculum and teaching resources.

Teach

Forums

Join community forums to get answers to your questions..

Forums

Looking for your App Inventor 1 projects? They're still here! Find out what's happening with App Inventor 1