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  Ralph Morelli  Dave Wolber  Larry Baldwin
Wellesley College  Trinity College  U. of San Francisco  BIRC

Hal Abelson  Shay Pokress  Josh Sheldon  Fred Martin  Mark Sherman  Karen Roehr
MIT  Trinity College  U. of San Francisco  University of Massachusetts  BIRC

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
AI2 Demo: Add TextToSpeech Component in Designer
AI2 Demo: Specify Button Behavior in Blocks Editor

When Button1 is clicked, call TextToSpeech1 to speak a message.

Message: "Welcome to Rhode Island College"
AI2 Demo: Connect to Android Device (Live Development)

App Inventor

Android Device
AI2 Demo: Add SpeechRecognizer Component in Designer
AI2 Demo: Blocks Using SpeechRecognizer
AI2 Demo: Live Development Test of SpeechRecognizer App

“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
No Texting While Driving App

Daniel’s code, translated into App Inventor 2

USF CS107: Computing, Mobile Apps, and the Web
USF CS107 Spring 2013 Portfolios

https://sites.google.com/site/appinventorcoursecourse/students-spring-2013
Trinity CPSC 110 Computing with Mobile Phones
http://turing.cs.trincoll.edu/~ram/cpsc110/portfolios.html

Screen1

Bantam App

Learn ASL

Practice Fingerspelling
[2]

Recognize the Flashing Word

Answer:

Score:

Submit

Next

Back To Main

Start Timer
End Timer

Seconds
Minutes

0
0

Calculate Initial Location
Latitude: Longitude:

Academic Resources
On Campus
Sports
Food
Home

Use My Location

Mobile Computational Thinking in App Inventor 2
RIC 4/10/2014
Trinity College: Tree Height Calculator

http://notes.hfoss.org/index.php/TreeCalc
Trinity College: Commodity Tracker App for Haiti

http://notes.hfoss.org/index.php/Haiti_Commodity_Collector
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
Mobile Computational Thinking in App Inventor 2

Birding Buddy
RIC 4/10/2014
Wellesley CS117 Inventing Mobile Apps

galleries of location based-apps and web-service apps

Exchange Bus Buddy

Wellesley Fresh-O-Meter

StoryBook
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 visited 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: 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.
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.
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

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

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

www.csprinciples.org

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**

![Diagram showing how to reference an event parameter in AI Classic](image)

**AI2**

![Diagram showing how to reference an event parameter in AI2](image)
More Event Parameters

AI Classic

AI2
Procedure parameters in AI Classic
Procedure parameters in AI2

AI Classic

AI2
Local variables in AI2

AI Classic: only global vars

AI2: includes local vars
Performing actions before returning value
All together now
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
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)
Better Error Handling (Johanna Okerlund)

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

Soon, the error will appear on the block causing the error:
Better Error Handling (Johanna Okerlund)

Error messages can appear on multiple blocks until the errors are fixed:
Better Debugging: Watch (Johanna Okerlund)
Conversion Between Blocks and Textual Code

(Karishma Chadha)
Thank You! Questions?