# Live programming of mobile apps in App Inventor

Jeff Schiller Hal Abelson Jose' Dominguez Andrew McKinney *MIT*  Franklyn Turbak Johanna Okerlund *Wellesley College*  Mark Friedman

Google

## Outline

- Goal: Be able to create fun and useful Android apps with minimum coding.
- Demo
- What we mean by live programming
- App Inventor Architecture: YAIL, Kawa, Forms, Companion

#### Demo

- Ball fling & bounce
  - Ball Flung event -- is active immediately (without a run button!)
  - Add bounce handler -- bounces immediately, even if stuck at edge.
  - Doit to enlarge ball radius while bouncing
  - Add timer (will reinitialize interface) to add trail.
  - Remove some parts of handler to show error.
- Dave Wolber Raffle App?

#### Architecture





Android Phone (USB)

#### Yail Example

;; Screen1

(do-after-form-creation

(set-and-coerce-property! 'Screen1 'Title "Screen1" 'text))

;;; Canvas1

(add-component Screen1 Canvas Canvas1 (set-and-coerce-property! 'Canvas1 'BackgroundColor #xFF00FFFF 'number) (set-and-coerce-property! 'Canvas1 'Width 200 'number) (set-and-coerce-property! 'Canvas1 'Height 300 'number)) ;;; Ball1

(add-component Canvas1 Ball Ball1 (set-and-coerce-property! 'Ball1 'X 46 'number) (set-and-coerce-property! 'Ball1 'Y 27 'number))

(define-event Ball1 Flung(\$x \$y \$speed \$heading \$xvel \$yvel) (set-this-form) (set-and-coerce-property! 'Ball1 'Speed (lexical-value \$speed) 'number) (set-and-coerce-property! 'Ball1 'Heading (lexical-value \$heading) 'number))

#### Doit

#### Dolt with ball example:



YAIL sent to Companion:

(process-repl-input 186

(set-and-coerce-property! 'Ball1 'Radius 10 'number))



#### Watch

;



## **Multiple Screens**

- Demo with multiple screens?
  - Press button in app to go to screen2; screen 2 blocks show up in browser.

#### **Liveness and Changes in Designer**

- ;

## **Browser/Device Configurations**

- Connect to device via wifi
- Connect to device via USB
- Connect to emulator

#### **Establishing WiFi communication**



#### **Two-way WiFi communication via HTTP**

#### web server on App Inventor Browser App Inventor Companion natio. 🏭 Branavat Tania 🤌 stager Duton Du ur IP Address is: 192.168.1.20 Î 1 A0 Sov Nation YAIL1 any values? Run YAIL1 OK watchval1 [watchval1] YAIL2 watchval2 Queue YAIL2 OK screenchangeval any values? errorval Run YAIL2 [watchval2, screenchangeval, errorval]

## **Companion Security**

- Companion is "safe" to have on the phone. It will not listen to the network without user input. Malware can start it, but cannot get it to do anything
- App Inventor connections are not encrypted, so there exists a risk that an intruder can introduce commands to the phone, but only during a live development session.

## Influences on our notion of liveness

- Lisp REPL, Smalltalk
- System figures out what it has to update on edits
- Our decision makes things easier for most users most of the time, but not always correct and sometimes annoying.

## **Future Work with AI Live Development**

- Improve fidelity (handle corner cases better).
- Re-work network architecture to better handle "networks" where two local devices cannot talk to each other (like at a hotel).

## **Future Work in App Inventor**

#### • Textual representations

- TAIL (consistent with Live development)
- Java Bridge (inconsistent with live development)

#### Demo

- Ball fling & bounce
  - Ball Flung event -- is active immediately (without a run button!)
  - Add bounce handler -- bounces immediately, even if stuck at edge.
  - Doit to enlarge ball radius while bouncing
  - Add timer (will reinitialize interface) to add trail.
  - Remove some parts of handler to show error.