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

App Inventor server → user projects

Build server → YAIL to JVM compiler

YAIL for whole project

Project .apk

Incremental YAIL

Retvals

Android device

YAIL interpreter in AI2 Companion

App Inventor environment on web browser
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))
```

Dolt with return value:
Watch

initialize global name to 0

for each number from 1 to 5 by 1

do set global name to get global name + 1
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

App Inventor Browser  Rendezvous Server  App Inventor Companion

6-character code

has companion answered?

nope

has companion answered?

scan code

(hash(code), IP)

OK

check hash, get Companion IP

(hash(code), IP)
Two-way WiFi communication via HTTP

App Inventor Browser

web server on App Inventor Companion

YAIL1

any values?

OK

YAIL2

[watchval1]

OK

any values?

[watchval2, screenchangeval, errorval]

Run YAIL1

watchval1

watchval2

screenchangeval

errorval

Queue YAIL2

Run YAIL2
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.