Not Just for Kids: Blocks Programming Language Design and Implementation in MIT App Inventor

Franklyn Turbak
Wellesley College Computer Science Dept.

Williams College CS Colloquium Talk
October 24, 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   University of Massachusetts Lowell

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

- MIT App Inventor (AI) demo
- Some notes on blocks languages in general
- AI blocks language design and implementation
  - Syntax
  - Static Semantics
  - Dynamic Semantics
  - Pragmatics
- Future work
Talk Road Map

- MIT App Inventor (AI) demo
  - Some notes on blocks languages in general
  - AI blocks language design and implementation
    - Syntax
    - Static Semantics
    - Dynamic Semantics
    - Pragmatics
  - Future work
Talk Road Map

- MIT App Inventor (AI) demo

- Some notes on blocks languages in general

- AI blocks language design and implementation
  - Syntax
  - Static Semantics
  - Dynamic Semantics
  - Pragmatics

- Future work
Blocks Languages are Growing in Popularity

**Scratch**: multi-media programs, animations, and games

**MIT App Inventor**: apps for Android smartphones

**StarLogo Nova**: multi-agent simulations

---

**Blockly**: Many blocks-based activities; Basis for MIT AI, main code.org challenges.

**code.org’s Hour of Code**: >20M participants, >75% blocks PLs
Negative Responses to Blocks Languages

I have never met a student who cut their teeth in any of these languages and did not come away profoundly damaged and unable to cope.

I mean this reads to me very similarly to teaching someone to be a carpenter by starting them off with plastic toy tools and telling them to go sculpt sand on the beach.

Not one thing they learn will bear any piece of resemblance to real work. All you're doing is teaching them misimpressions of what the job is, and tricking them out of having meaningful formative experiences.

http://blog.acthompson.net/2012/12/programming-with-blocks.html

These are not proper programming languages, anyone with half a brain knows that, but why deny those who can't or don't want to 'code' the opportunity of being creative with these tools and learning some logic skills along the way.

http://blog.acthompson.net/2012/12/programming-with-blocks.html

Working with actual code writing instead of a drag & drop interface prepares children better for the real world.

http://www.playcodemonkey.com/
So they currently see this:

when it is really this:

Why do they see it this way? Because they grew up on this:

Yes, it is colorful and newfangled, but it still gets jobs done. Not all of them, but a bunch of them.
I would like to express my utmost appreciation for your product. I'm teaching several pre-CS courses for gifted youth at Junior-high school level (7th-9th grades) as well as CS and software engineering at high school (10th – 12th grades) including Android development in Java. It is really amazing that in AppInventor, 7th grade students (with about 50 hours prior experience in Scratch) can do in 6 hours what 12th grade students take about 200-300 hours to achieve in Java (and this is after studying CS and Android development for about 700 hours). AppInventor goes way beyond the 80:20 principle (80% of the utility in 20% of the effort) – it is more like 60:5 (60% of the functionality, for less than 5% of the effort) which makes it much more fun, and opens up a lot of space for creativity.

Yossi Yaron, Israeli teacher
Talk Road Map

○ MIT App Inventor (AI) demo

○ Some notes on blocks languages in general

○ AI blocks language design and implementation
  • Syntax
  • Static Semantics
  • Dynamic Semantics
  • Pragmatics

○ Future work
AI Blocks Syntax: Expressions

- Button1 . BackgroundColor
- Label1 . Text
- Canvas1 . PaintColor
- 17
- " "
- `true`
- `join`
- `length`
- `random integer from` to `to`
AI Blocks Syntax: Statements

- `set Button1.BackgroundColor to`  
- `set Label1.Text to`  
- `set Canvas1.PaintColor to`  
- `if` and `then`  
- `for each item in list do`  
- `while test do`  
- `call Camera1.TakePicture`  
- `call TextToSpeech1.Speak message`  
- `call Canvas1.DrawCircle x y r`  
- `call Canvas1.DrawCircle x y r`  
- `add items to list list item`  
- `insert list item list index item`
AI Blocks Syntax: Top Level Declarations

initialize global name to

to procedure
do

to procedure2
result

doi

when Camera1 .AfterPicture
  image
  do

when Canvas1 .TouchDown
  x y
  do

when Canvas1 .Dragged
  startX startY prevX prevY currentX currentY draggedSprite
  do
AI Blocks Syntax: Local Variable Declarations
Performing actions before returning value

**AI Classic**

**AI2**
All together now
Conversion Between Blocks and Text

when Button1 .Click
do set Label1 . Text to \text{get global num} + 1

when Button1 .Click
do set Label1 . Text to \text{TAIL exp} \{{\text{get global num}} + 1\}

when Button1 .Click
do set Label1 . Text to \text{TAIL stmt} \{\text{set Label1.Text to: {{get global num} + 1}}\}

when Button1 .Click
do set Label1 . Text to \text{TAIL decl} \{\text{when Button1.Click do: [set Label1.Text to: {{get global num} + 1}}\}]
Talk Road Map

- MIT App Inventor (AI) demo
- Some notes on blocks languages in general
- **AI blocks language design and implementation**
  - Syntax
  - **Static Semantics**
  - Dynamic Semantics
  - Pragmatics
- Future work
Name Scoping in AI

- Globals are in a separate namespace
- Indentation visually highlights area of name scope
- Drop-downs list only names in scope.
- Inner names can shadow outer ones
- Changing declared names automatically consistently changes all
Handling Unbound Names
Other Drop-Downs Reduce Errors & Viscosity
Distinguishing Void and Fruitful Procedures

Python function gotcha

```python
>>> def square(x):
...    x * x
...    ...
>>> square(5)
```
What About Types?

App Inventor is dynamically typed, so there’s only one plug shape:
Simple “Soft” Static Type Checking

Type errors at block connection time are prohibited by “repulsion”

Dynamic type errors can be hidden by variables:
Digression: Connector Shapes in PictureBlocks

- **Number**
  - 1
  - +
  - sqrt
  - atan

- **Boolean**
  - true
  - not
  - and

- **String**
  - abc
  - num to string
  - join

- **Color**
  - Red
  - wedge
  - clockwise

- **Picture**
  - Load
  - fourPics
Polymorphism in PictureBlocks

polymorphic plug

choose then else

polymorphic sockets

choose then abs

not choose then else

1 =

= abc
pushRight: Complete Declaration and Call
Continued Digression: Type Blocks

Marie Vasek ‘12
Wellesley

listof constructor
tuple constructor
function constructor

listof int  listof (listof string)  int * string  bool -> string
Type Blocks: More Examples

listof (string * boolean) * boolean  

(listof string) * boolean  

boolean  * (string -> listof number) 

(boolean  * string) -> (listof number)
Type Blocks: Lists
Type Blocks:
ML Style Universal Polymorphism
Back to AI: List Mapping

Python:

```python
>>> nums = [5, 2, 17, 8]

>>> map(lambda x: x*2, nums)
[10, 4, 34, 16]
```

App Inventor doesn’t have first-class functions, but can finesse mapping:
Experimental Higher-Order List Operators in AI

Soojin Kim ’15 Wellesley
Loop-based List Processing
List Processing With Higher-Order Operators

```
reduce list
mapping each item to
by combining item and answerSoFar

result

make new list from
keeping each item passing

test is a number? get item

make new filtered list from

get global originalList

get item x 2
```

Blocks PLDI in MIT AI
Nondestructive vs. Destructive List Ops In Python

```python
>>> elts = [19, True, "foo", 23, "bar", 17, False]

>>> elts.sorted()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'list' object has no attribute 'sorted'

>>> sorted(elts)
[False, True, 17, 19, 23, 'bar', 'foo']

>>> elts
[19, True, 'foo', 23, 'bar', 17, False]

>>> elts.sort()

>>> elts
[False, True, 17, 19, 23, 'bar', 'foo']
```
Nondestructive vs. Destructive Sorting In AI

make new sorted list from

sort existing list

changes existing list
makes new list

make new sorted list from

sort existing list

changes existing list
makes new list
Other Nondestructive vs. Destructive List Ops In AI

- Make new list from mapping each **item** to
- Change existing list mapping each **item** to

- Changes existing list
- Makes new list

- Make new filtered list from keeping each **item** passing test
- Changes existing list
- Makes new list

Blocks PLDI in MIT AI
Talk Road Map

- MIT App Inventor (AI) demo
- Some notes on blocks languages in general
- **AI blocks language design and implementation**
  - Syntax
  - Static Semantics
  - **Dynamic Semantics**
  - Pragmatics
- Future work
The AI Event Model is Accessible for Simple Tasks …
… but can be Confusing for more Complex Ones

This program for playing all recordings in a list does \textbf{not} work:
Correctly Playing all Recordings

```blocks
initialize global index to 0
when PlayAllButton . Click
do set global index to 1
call playNextRecording

do to playNextRecording

if get global index > length of list list
global recordings
then set PlayAllButton . Text to “Play All”
else set PlayAllButton . Text to get global index

set Player1 . Source to select list item list
index get global recordings

call Player1 . Start

when Player1 . Completed
do set global index to get global index + 1
call playNextRecording
```
Talk Road Map

- MIT App Inventor (AI) demo
- Some notes on blocks languages in general
  - **AI blocks language design and implementation**
    - Syntax
    - Static Semantics
    - Dynamic Semantics
    - **Pragmatics**
- Future work
AI Live Development Architecture

App Inventor server

Build server

YAIL to JVM compiler

YAIL interpreter in AI2 Companion

incremental YAIL

Android device

App Inventor environment on web browser

Browser code

User projects

YAIL for whole project

Project .apk

Retvals
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 Examples

YAIL sent to companion:

```
(process-repl-input 186
 (set-and-coerce-property! 'Ball1 'Radius 10 'number))
```

Dolt with return value:

```
set global num to
get global num
initialize global num to
Do It Result: 42
```
Better Error Handling

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

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

- Initialize global `count` to 0
- For each `number` from 1 to 5 by 1
- Do set `global count` to
- Get `global count`
- Add 1
Talk Road Map

- MIT App Inventor (AI) demo
- Some notes on blocks languages in general
- AI blocks language design and implementation
  - Syntax
  - Static Semantics
  - Dynamic Semantics
  - Pragmatics
- Future work
Future Work

- More flexible event handling
- Non-local returns
- More faithful live development
- User studies: what works and what doesn't?
- Blocks, text, and in-between
- Component development kit
- Blocks-based version of ML or Haskell

Want to join me? Email fturbak@wellesley.edu
Thank You! Questions?

appinventor.mit.edu

www.tinkerblocks.org
Establishing WiFi communication

App Inventor Browser  Rendezvous Server  App Inventor Companion

6-character code

has companion answered?

nope

has companion answered?

(hash(code), IP)

check hash, get Companion IP

scan code

(hash(code), IP)

OK

@ IP
Two-way WiFi communication via HTTP

App Inventor Browser

web server on
App Inventor Companion

Run YAIL1
watchval1
watchval2
screenchangeval
errorval

Queue YAIL2

Run YAIL2

YAIL1
any values?
OK
YAIL2
[watchval1]
OK
any values?
[watchval2, screenchangeval, errorval]
Another Conversion Example

```
TAIL decl (to <quadratic> <a> <b> <c> result: {{{0} - {get b}} + {sqrt {{{get b} * {get b}} - {{4} * {{get a} * {get c}}}}}}}
```
TurtleBlocks program

- **run once**
- **set corner type**: SHARP
- **set thickness**: 15
- **repeat**
  - **times**: 5
  - **do**
    - **forward** distance: 200
    - **left** degress: 144

turtle drawing

cardstock

Block PLDI in MIT AI

acrylic

57

drawing boundary

Williams, October 24, 2014
TurtleBlocks Artifacts
PictureBlocks: Sketching & Engraving

user sketch

PictureBlocks program

resulting picture

print from engraving

wood engraving
PictureBlocks: Engraving + Cutting