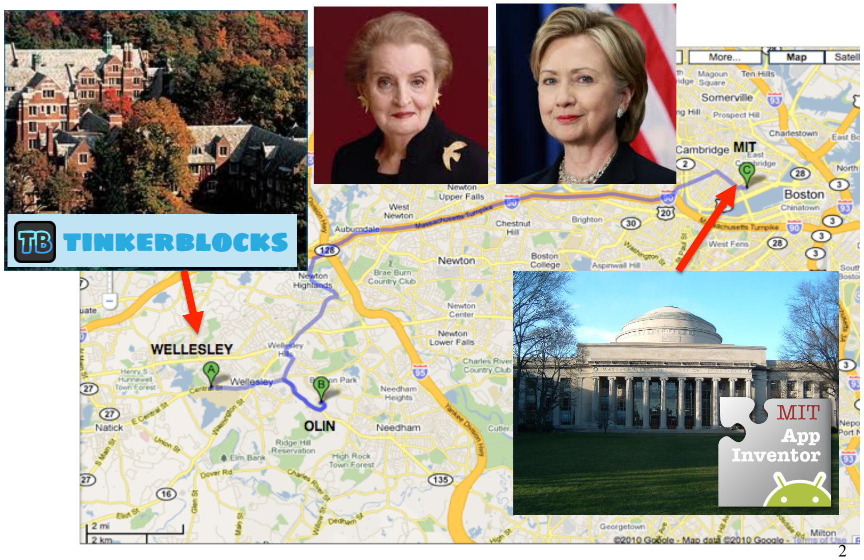
Democratizing Programming with Blocks Languages

Franklyn Turbak
Wellesley College Computer Science Dept.

Distributed Multimedia Systems (DMS 2015)
Vancouver, August 31, 2015

Wellesley & MIT



Talk Road Map

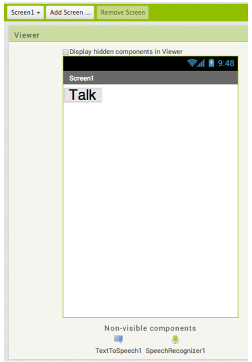
- Blocks demo: MIT App Inventor (AI)
- Democratizing programming with blocks: examples
- Lowering barriers with blocks
  - Syntax
  - Static semantics
  - Dynamic semantics
- Challenges in blocks programming
  - Usability
  - Learnability in blocks vs. text
  - Perception: blocks programming not “real”, maybe harmful
- Research questions

Talk Road Map

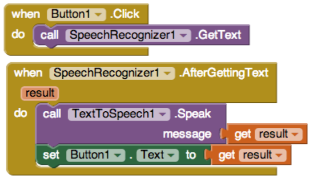
- Blocks demo: MIT App Inventor (AI)
- Democratizing programming with blocks: examples
- Lowering barriers with blocks
  - Syntax
  - Static semantics
  - Dynamic semantics
- Challenges in blocks programming
  - Usability
  - Learnability in blocks vs. text
  - Perception: blocks programming not “real”, maybe harmful
- Research questions

Simple App Inventor Example

Designer Window

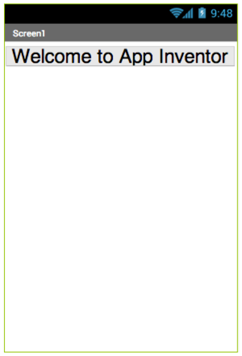


Blocks Editor



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

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

Android Device

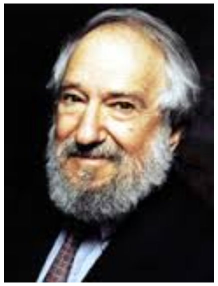


Talk Road Map

- Blocks demo: MIT App Inventor (AI)
- Democratizing programming with blocks: examples
- Lowering barriers with blocks
  - Syntax
  - Static semantics
  - Dynamic semantics
- Challenges in blocks programming
  - Usability
  - Learnability in blocks vs. text
  - Perception: blocks programming not "real", maybe harmful
- Research questions

Papert on Computers & Constructionism

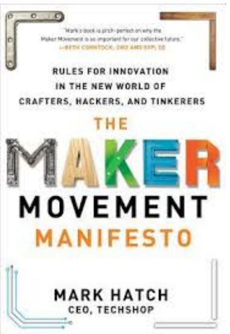
"In many schools today, the phrase "computer-aided instruction" means making the computer teach the child. One might say *the computer is being used to program* the child. In my vision, ***the child programs the computer***, and in doing so, both acquires a sense of mastery over a piece of the most modern and powerful technology and establishes an intense contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model building." *Mindstorms: Children, Computers, and Powerful Ideas* (*bolding mine*)

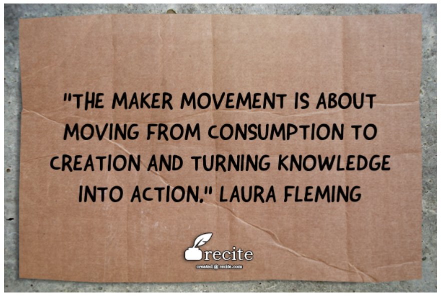


"The word **constructionism** is a mnemonic for two aspects of the theory of science education underlying this project. From constructivist theories of psychology we take a view of learning as a reconstruction rather than as a transmission of knowledge. Then we extend the idea of manipulative materials to the idea that **learning is most effective when part of an activity the learner experiences as constructing is a meaningful product**." *Constructionism: A New Opportunity for Elementary Science Education (bolding mine)*

Maker Movement

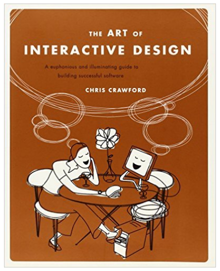
"But in this new world, you don't have to go bankrupt if you fail because you can fail small. **You can innovate as a hobby. Imagine that: a nation of innovation hobbyists working to make their lives more meaningful and the world a better place.** Welcome to the maker revolution."
— *Mark Hatch, The Maker Movement Manifesto: Rules for Innovation in the New World of Crafters, Hackers, and Tinkerers (bolding mine)*


"Mark's book is pitch-perfect on why the Maker Movement is so important for our collective future."
—BETH COMSTOCK, CMO AND SVP, GE
RULES FOR INNOVATION IN THE NEW WORLD OF CRAFTERS, HACKERS, AND TINKERERS
THE MAKER MOVEMENT MANIFESTO
MARK HATCH
CEO, TECHSHOP



"THE MAKER MOVEMENT IS ABOUT MOVING FROM CONSUMPTION TO CREATION AND TURNING KNOWLEDGE INTO ACTION." LAURA FLEMING

Democratizing Programming

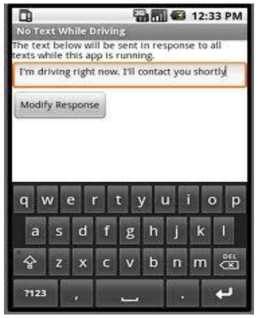


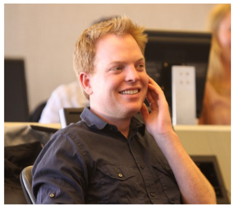
"What we need is a means of democratizing programming, of taking it out of the soulless hands of the programmers and putting it into the hands of a wider range of talents." *Chris Crawford, The Art of Interactive Design*

MIT App Inventor mission statement: The MIT App Inventor project seeks to **democratize** software development by empowering all people, especially young people, to transition from being consumers of technology to becoming creators of mobile technology.

MIT App Inventor

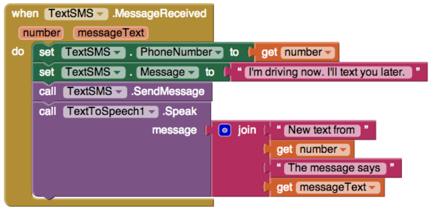
No Texting While Driving App


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
Modify Response

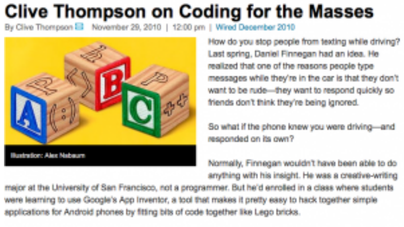


Daniel Finnegan, English Major, developed the app in Dave Wolber's USF course *CS017: Computing, Mobile Apps, and the Web*

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
```



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?

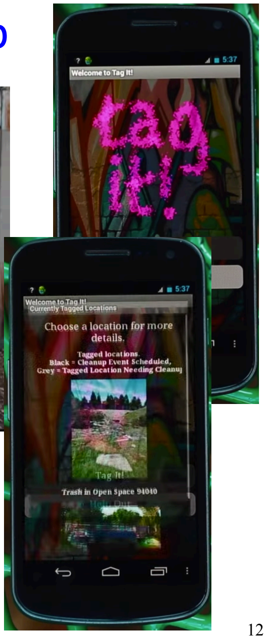
Normally, Finnegan wouldn't have been able to do anything with his insight. He was a creative-writing major at the University of San Francisco, not a programmer. But he'd enrolled in a class where students were learning to use Google's App Inventor, a tool that makes it pretty easy to hack together simple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

App To Track Feral Hogs

Alabama's Lawrence County High School students used App Inventor to build an app that tracks feral hogs, which were causing economic damage to their community. Their app won a prize of $100K in technology for Samsung's 2012 Solve for Tomorrow contest.

http://www.forbes.com/sites/samsung/2013/11/25/high-school-students-battle-wild-hogs-with-stem-solutions/

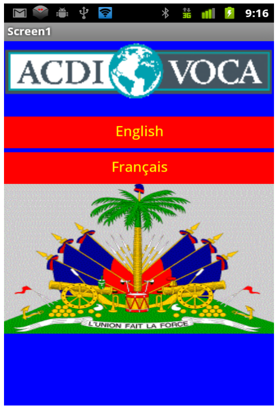
Trash & Graffiti Cleanup App

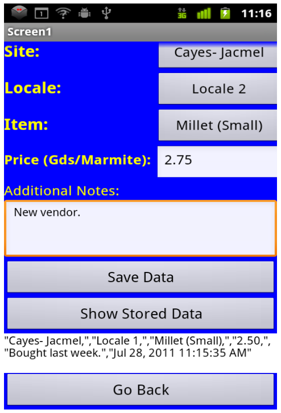

5:37
Welcome to Tag It!
tag it!
Welcome to Tag It!
Currently Tagged Locations
Choose a location for more details.
Tagged locations.
Black = Cleanup Event Scheduled,
Grey = Tagged Location Needing Cleanu
Tag It!
Trash in Open Space 94040

East Palo Alto girls created an app to tag the location of trash and create an event for cleaning it up. This app ranked highly in the Technovation Challenge competition.

http://appinventor.mit.edu/explore/stories/east-palo-alto-girls-create-app-clean-graffiti-trash.html

Commodity Tracker App for Haiti





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"

Developed using App Inventor as part of Trinity College's Humanitarian Free and Open Source Software (HFOSS) project.

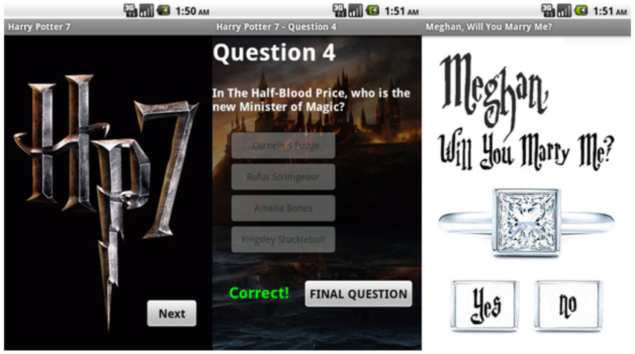
http://notes.hfoss.org/index.php/Haiti_Commodity_Collector

App to Destroy Mines Safely

Chris Metzger, United States Marine Corps Staff Sergeant, used App Inventor to create an app that helps other Marines destroy weaponry captured in the field. It calculates the amount of explosives necessary to safely destroy captured ammunition and mines.

http://appinventor.mit.edu/explore/stories/united-states-marines-use-app-inventor-field.html

Marriage Proposal App



In The Half-Blood Price, who is the new Minister of Magic?

Meghan, Will You Marry Me?

Hodgson didn't know how to develop an Android app. ... "How the heck was I going to build this thing?" he recalls thinking. "I tried a couple of other rapid development tools, but they really had too much of a learning curve to let me do it in the time-frame I had in mind." That is, until a friend recommended App Inventor, a tool for amateur Android devs created by Google Labs. "It allowed me, with no java knowledge, to quickly get this thing whipped up," Hodgson says.

http://www.fastcompany.com/1754193/google-love-story-man-builds-android-app-propose-girlfriend

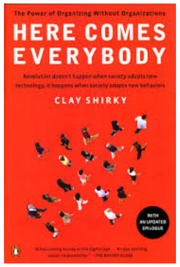
Clay Shirky on Situated Software vs. Web School (2004)

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

Leverage small groups

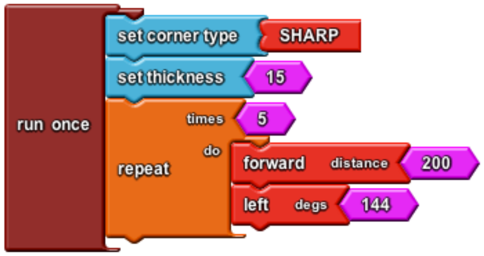
- local knowledge
- trust of other users
- publicly shame deadbeats in group purchase apps


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

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

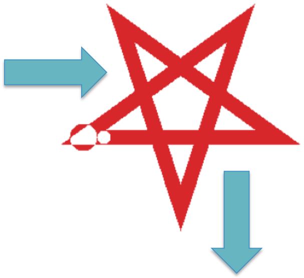
TurtleBlocks

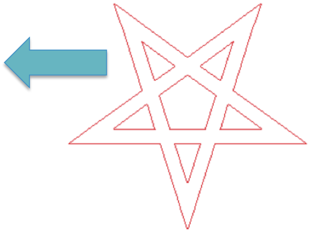
TurtleBlocks program



```
run once
  set corner type SHARP
  set thickness 15
  repeat times 5
    do forward distance 200
       left degs 144
```

turtle drawing



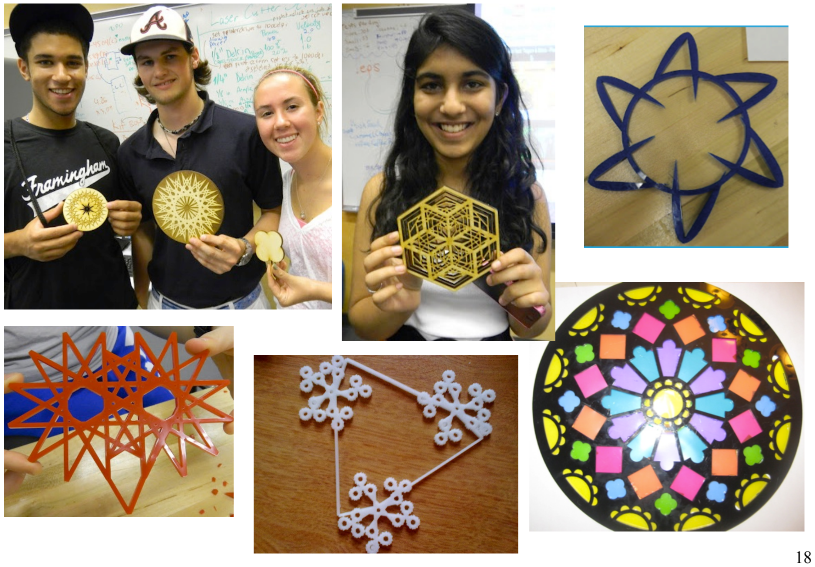


drawing boundary

acrylic

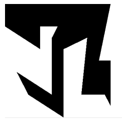
cardstock

TurtleBlocks Artifacts

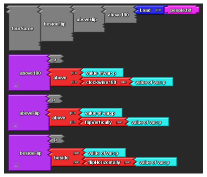


PictureBlocks: Sketching & Engraving

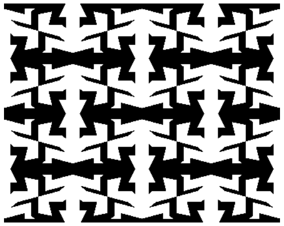
user sketch

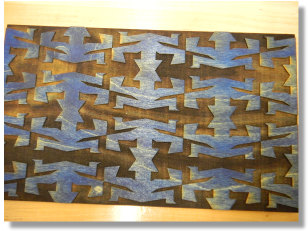


PictureBlocks program



resulting picture

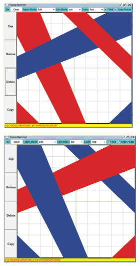


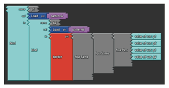


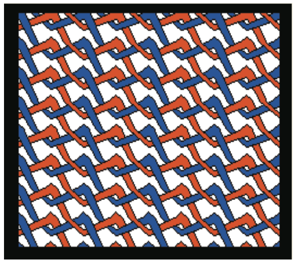
wood engraving

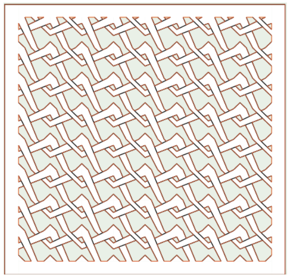
print from engraving

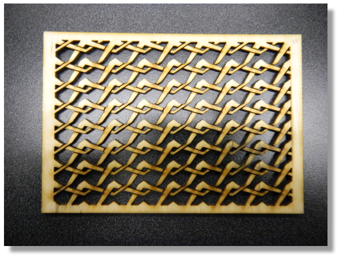
PictureBlocks: Engraving + Cutting







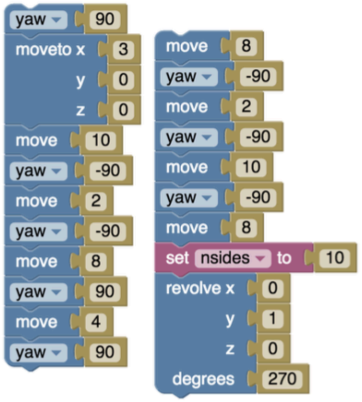




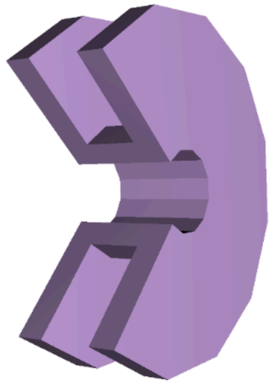
PictureBlocks Artifacts

Madeup: 3D Modeling with Blocks

Chris Johnson, University of Wisconsin
Peter Bui, Notre Dame

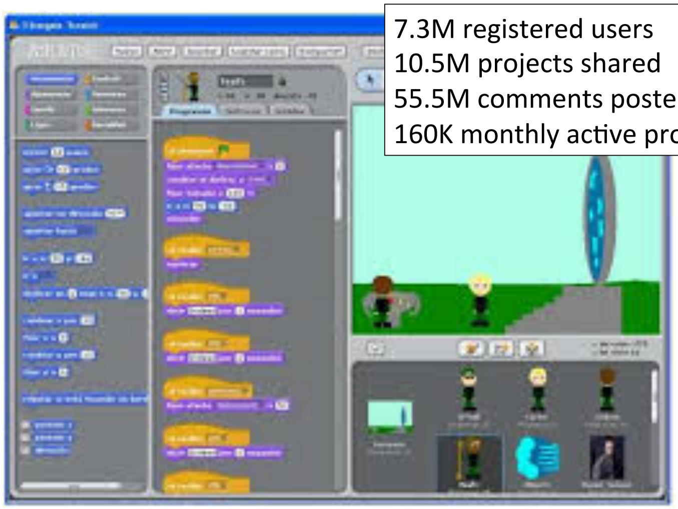


```
yaw 90
moveto x 3
       y 0
       z 0
move 10
yaw -90
move 2
yaw -90
move 8
yaw 90
move 4
yaw 90

move 8
yaw -90
move 2
yaw -90
move 10
yaw -90
move 8
set nsides to 10
revolve x 0
        y 1
        z 0
  degrees 270
```



Scratch

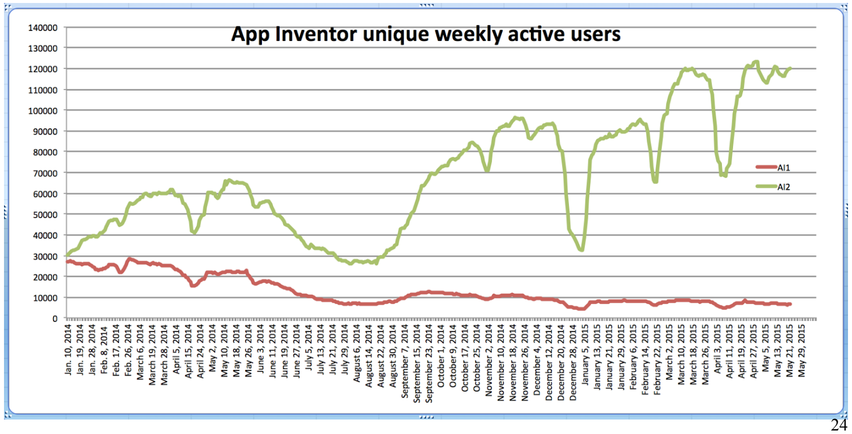
multi-media programs, animations, and games



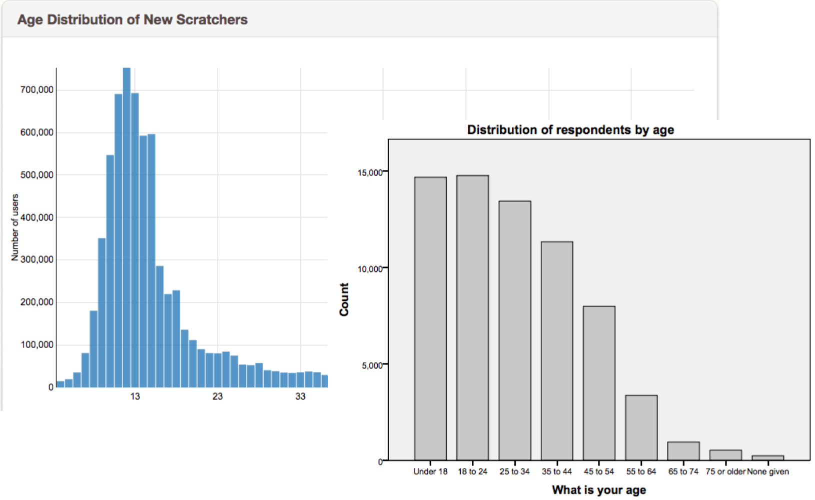
7.3M registered users
10.5M projects shared
55.5M comments posted
160K monthly active project creators

App Inventor Usage is Growing

- 3.3 million registered users
- 185 countries
- 8.9 million mobile apps created
- ~ 120K unique weekly users

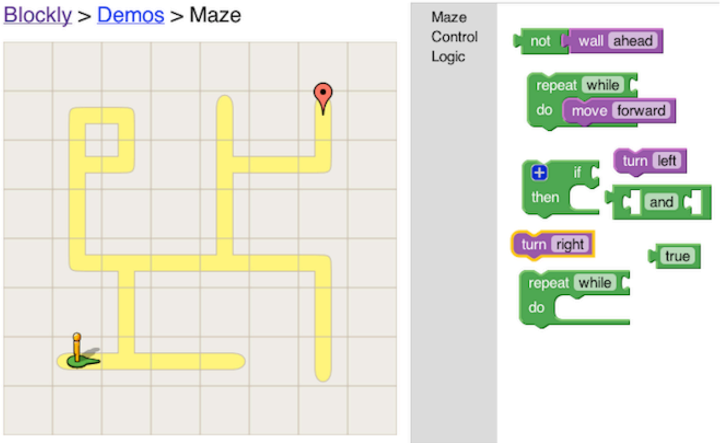


Age Distribution: Scratch vs. App Inventor



Blockly

Many blocks-based activities. Basis for early Code.org challenges. Many other blocks environments, including App Inventor, are based on Blockly.



Blockly > Demos > Maze

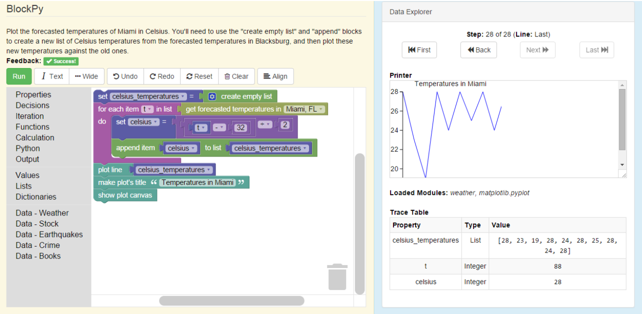
And many more …

Snap!: Scratch for Scheme, *Beauty and Joy of Computing curriculum* (Harvey, Monig, Garcia @ Berkeley)

StarLogo Nova: multi-agent simulations (Wendel et al @ MIT)

Alice: 3D storytelling and gaming environment (CMU)

BlockPy: Blocks-based version of Python for teaching data science (Bart, Tilevitch, Shaffer, Kafura @ Virginia Tech)



BlockPy

Plot the forecasted temperatures of Miami in Celsius. You'll need to use the "create empty list" and "append" blocks to create a new list of Celsius temperatures from the forecasted temperatures in Blacksburg, and then plot these new temperatures against the old ones.

Feedback: ✓ Success!

```
set celsius_temperatures = create empty list
for each item t in list get forecasted temperatures in Miami, FL
do  set celsius = t - 32 ÷ 2
    append item celsius to list celsius_temperatures
plot line celsius_temperatures
make plot's title "Temperatures in Miami"
show plot canvas
```

Data Explorer

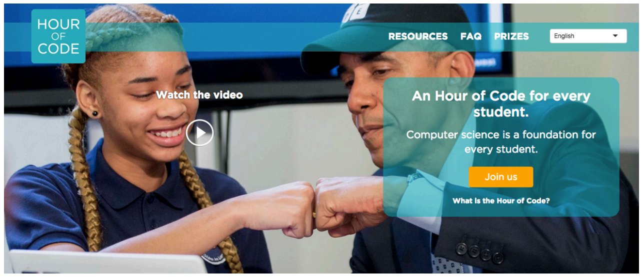
Step: 28 of 28 (**Line:** Last)

Loaded Modules: *weather, matplotlib.pyplot*

Trace Table

Property	Type	Value
celsius_temperatures	List	[28, 23, 19, 28, 24, 28, 25, 28, 24, 28]
t	Integer	88
celsius	Integer	28

Code.org Hour of Code

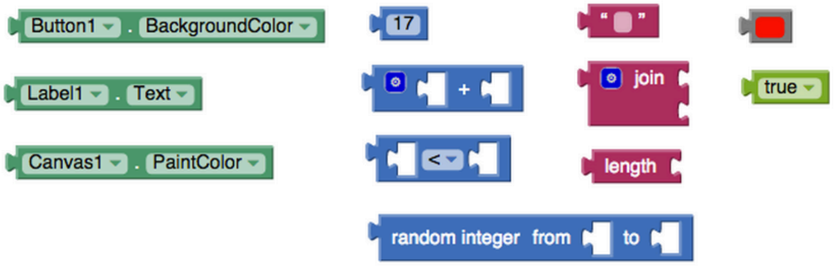

HOUR OF CODE
RESOURCES
FAQ
PRIZES
English
Watch the video
An Hour of Code for every student.
Computer science is a foundation for every student.
Join us
What is the Hour of Code?

- Dec. 2013:
  - 26M participants spend an hour programming in one of ~24 programming environments
  - 74% of these use one of the 5 blocks languages
    - Code.org exercises based on Blockly
    - Scratch
    - App Inventor
    - Tynker
    - Hopscotch
- Dec. 2014 and beyond: claim > 100M participants total

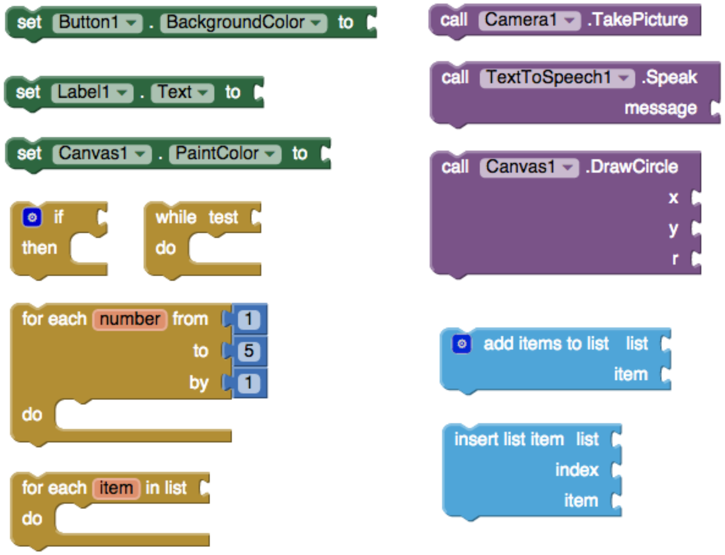
Talk Road Map

- Blocks demo: MIT App Inventor (AI)
- Democratizing programming with blocks: examples
- Lowering barriers with blocks
  - Syntax
  - Static semantics
  - Dynamic semantics
- Challenges in blocks programming
  - Usability
  - Learnability in blocks vs. text
  - Perception: blocks programming not “real”, maybe harmful
- Research questions

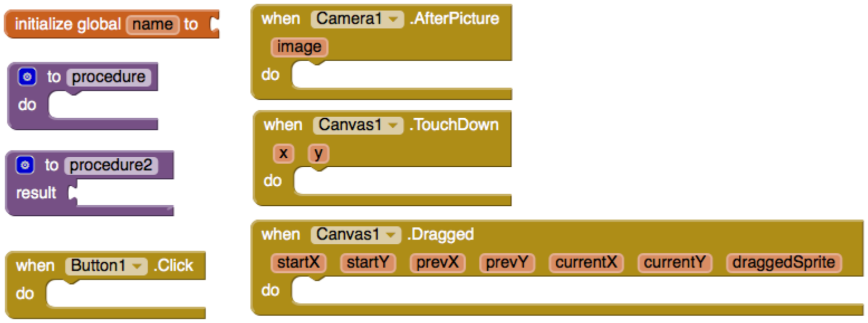
AI Syntax: Expressions



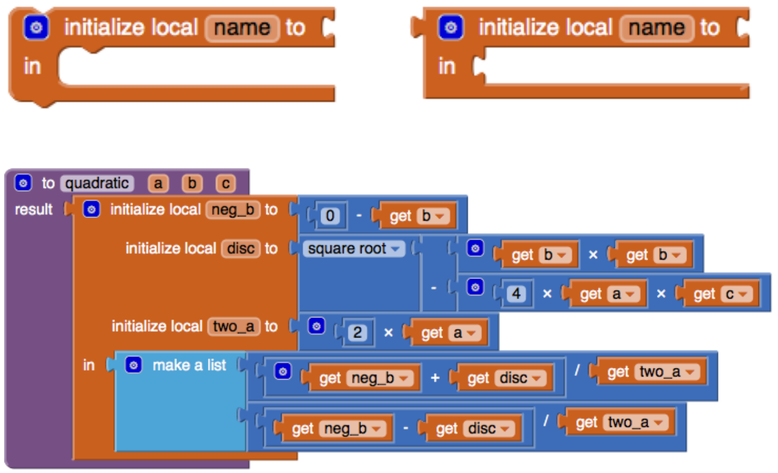
AI Syntax: Statements



AI Syntax: Top Level Declarations

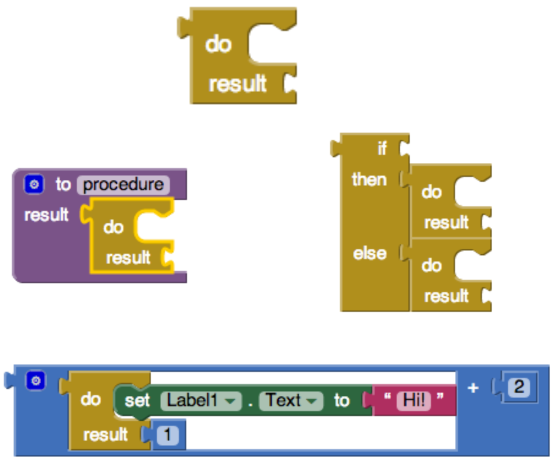


AI Syntax: Local Variable Declarations



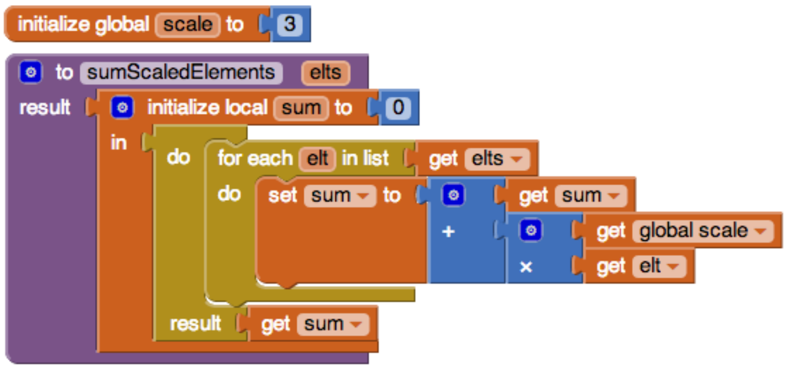
```
initialize local name to
in

initialize local name to
in

to quadratic a b c
result  initialize local neg_b to 0 - get b
        initialize local disc to square root (get b × get b) - (4 × get a × get c)
        initialize local two_a to 2 × get a
        in  make a list  (get neg_b + get disc) / get two_a
                         (get neg_b - get disc) / get two_a
```

AI Syntax: Performing actions before returning value



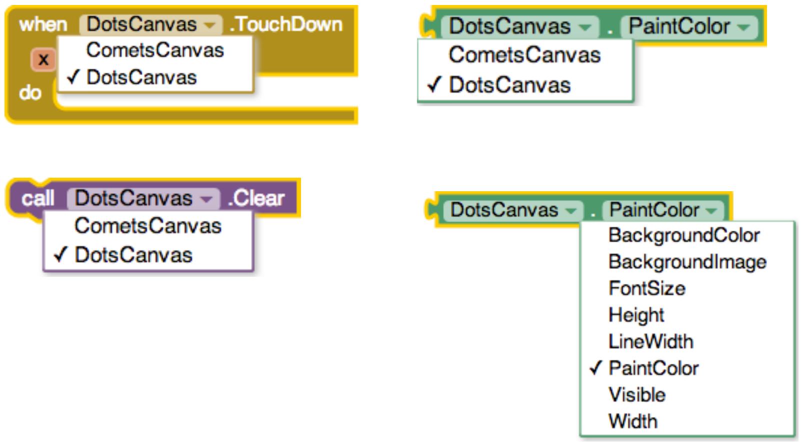
```
do
result

to procedure
result  do
        result

if
then  do
      result
else  do
      result

do      set Label1.Text to "Hi!"
result  1
+ 2
```

AI Syntax: 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
```

Drop-Downs Reduce Errors & Viscosity

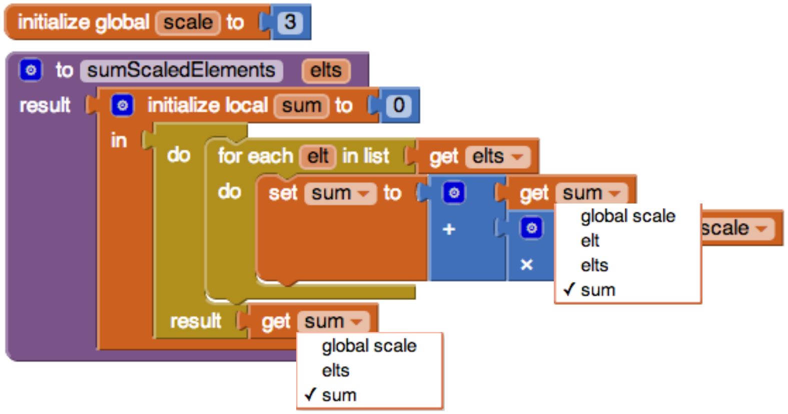


Talk Road Map

- Blocks demo: MIT App Inventor (AI)
- Democratizing programming with blocks: examples
- Lowering barriers with blocks
  - Syntax
  - Static semantics
  - Dynamic semantics
- Challenges in blocks programming
  - Usability
  - Learnability in blocks vs. text
  - Perception: blocks programming not "real", maybe harmful
- Research questions

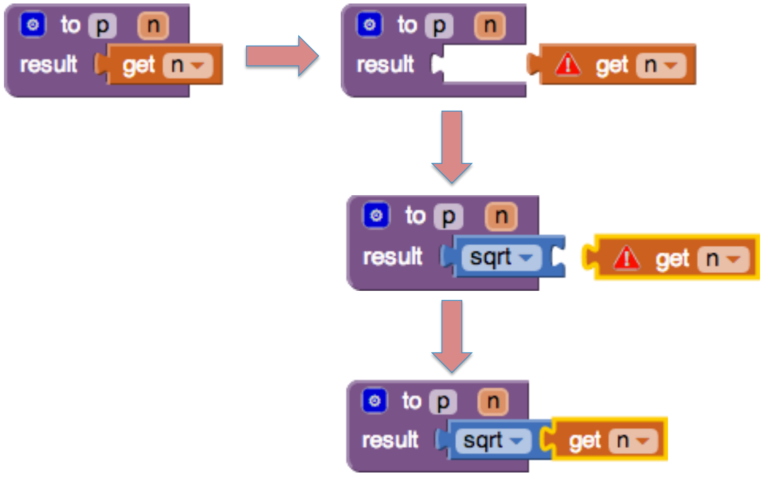
Name Scoping in AI

- 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 references



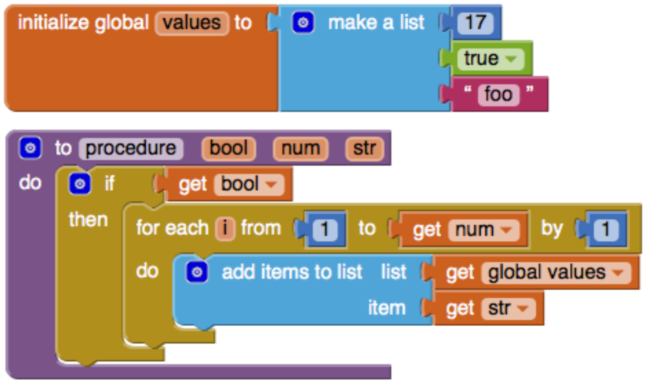
```
initialize global scale to 3

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
            result get sum
```

Handling Unbound Names



What About Types?

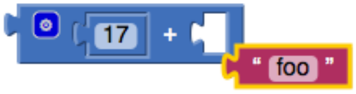
App Inventor is dynamically typed, so there's only one plug shape:



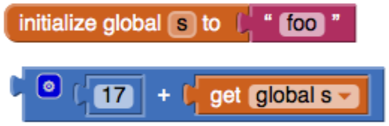
```
initialize global values to  make a list  17
                                          true
                                          " foo "

to procedure bool num str
do  if get bool
    then  for each i from 1 to get num by 1
          do  add items to list  list get global values
                                 item get str
```

Simple “Soft” Static Type Checking

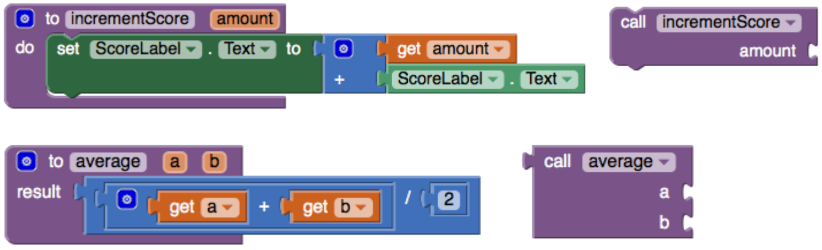
Type errors at block connection time are prohibited by “repulsion”



Dynamic type errors can be hidden by variables:

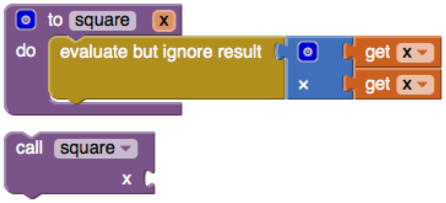


Distinguishing Void and Fruitful Procedures



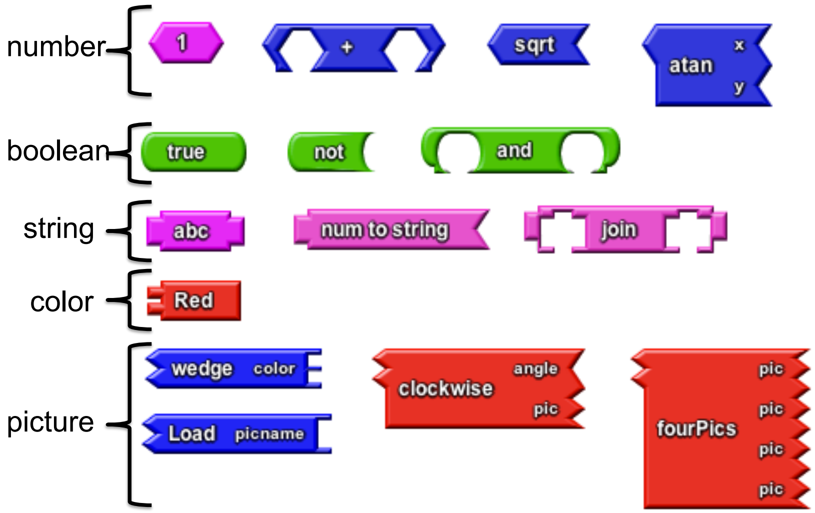
Python function gotcha

```
>>> def square (x):
...   x * x
...
>>> square(5)
>>>
```

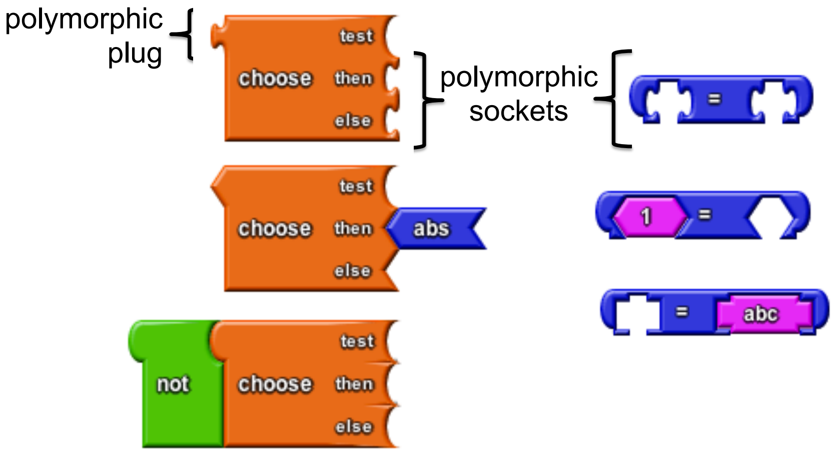


Connector Shapes in PictureBlocks

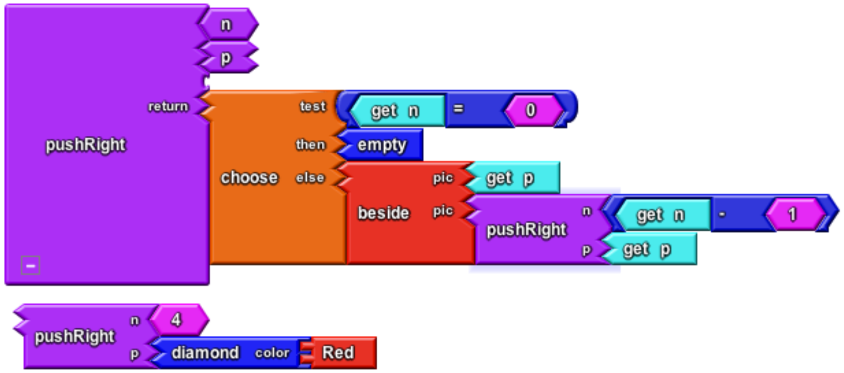
(Similar to types-as-shapes in StarLogo TNG)



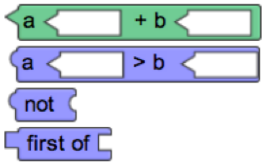
Polymorphism in PictureBlocks



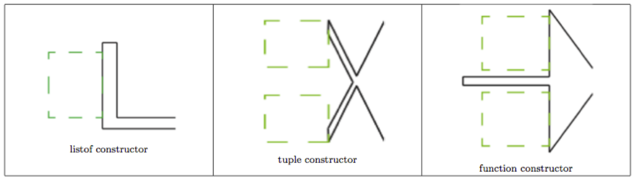
pushRight: Complete Declaration and Call

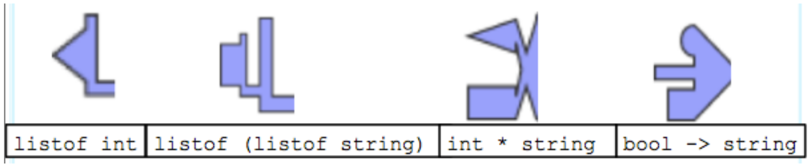


Type Blocks



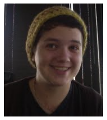
Marie Vasek '12
Wellesley

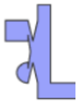




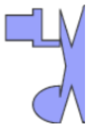
`listof int` `listof (listof string)` `int * string` `bool -> string`

Type Blocks: More Examples

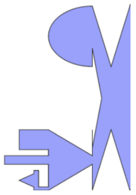




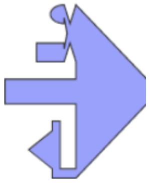
`listof (string * boolean)`



`(listof string) * boolean`

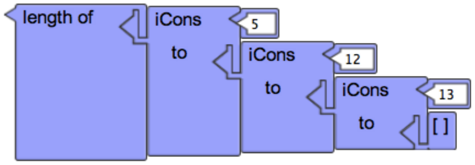


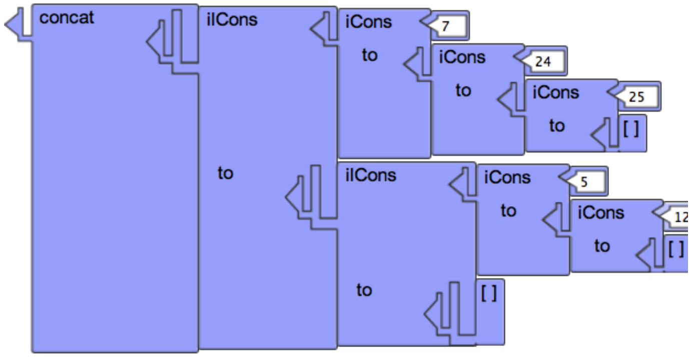
`boolean  * (string -> listof number)`



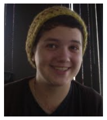
`(boolean  * string) -> (listof number)`

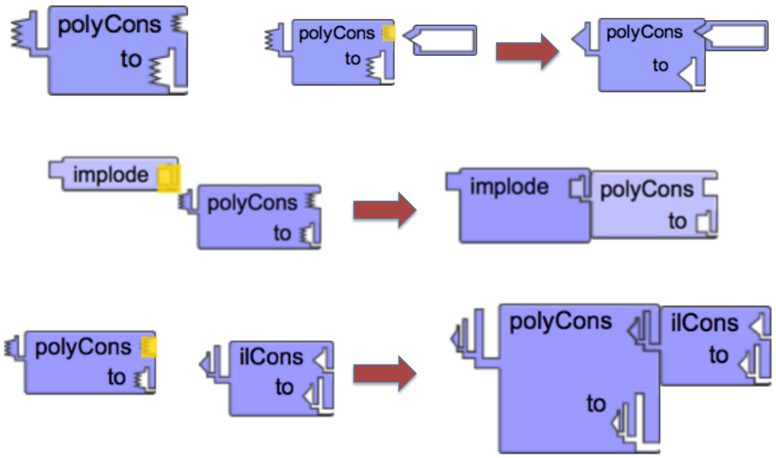
Type Blocks: Lists





Type Blocks: ML Style Universal Polymorphism



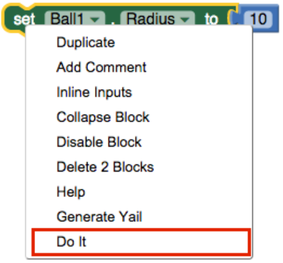


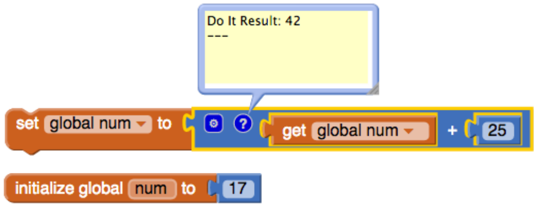
Talk Road Map

- Blocks demo: MIT App Inventor (AI)
- Democratizing programming with blocks: examples
- Lowering barriers with blocks
  - Syntax
  - Static semantics
  - Dynamic semantics
- Challenges in blocks programming
  - Usability
  - Learnability in blocks vs. text
  - Perception: blocks programming not "real", maybe harmful
- Research questions

App Inventor: DoIt

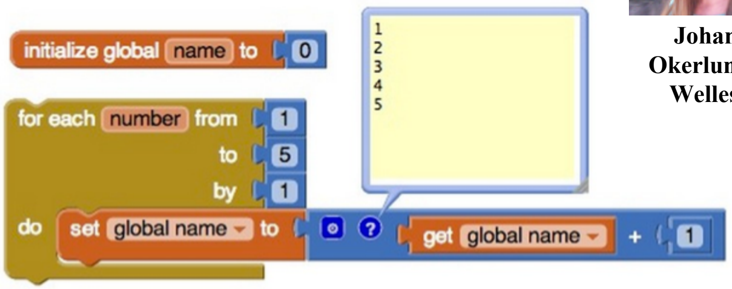
Simple form of interactivity/liveness found in many blocks environments (as well as interpreter text-based languages).





```
Do It Result: 42
---
```

Better Debugging: Watch



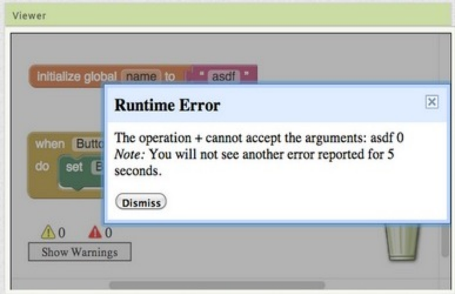
```
1
2
3
4
5
```

Johanna Okerlund '14 Wellesley

Emery Gerndt Otopalk is currently working on a trace feature for watching all blocks after a breakpoint

Better Error Handling

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

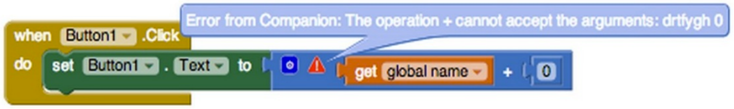


Runtime Error

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

Dismiss

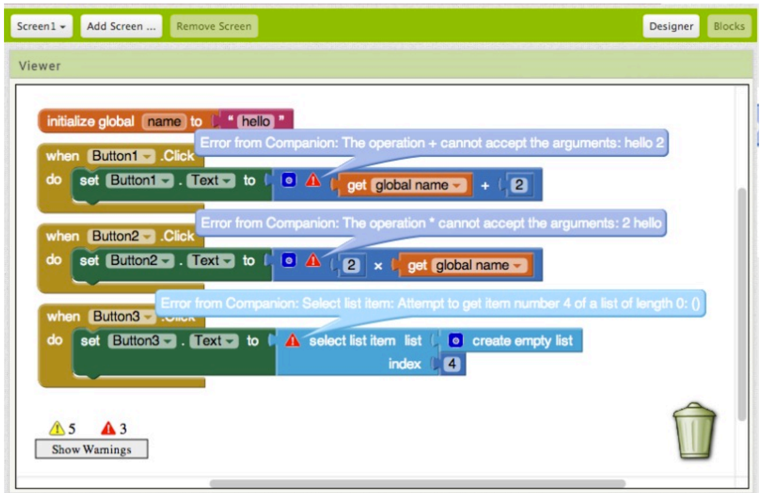
Soon, the error will appear on the block causing the error:



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

Better Error Handling

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

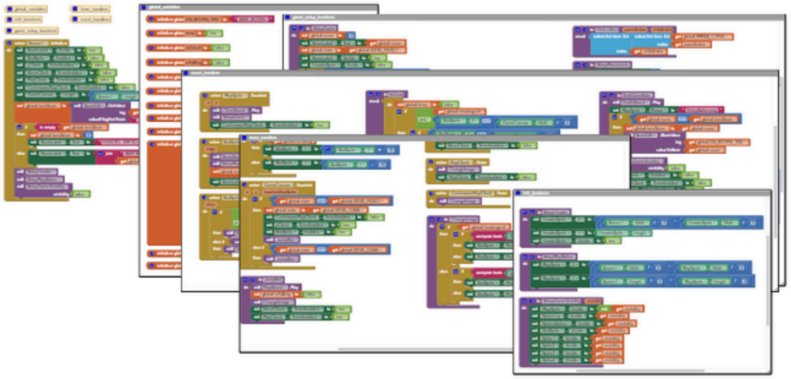
Error from Companion: Select list item: Attempt to get item number 4 of a list of length 0: ()

Talk Road Map

- Blocks demo: MIT App Inventor (AI)
- Democratizing programming with blocks: examples
- Lowering barriers with blocks
  - Syntax
  - Static semantics
  - Dynamic semantics
- Challenges in blocks programming
  - Usability
  - Learnability in blocks vs. text
  - Perception: blocks programming not “real”, maybe harmful
- Research questions

Usability: Current Work in AI

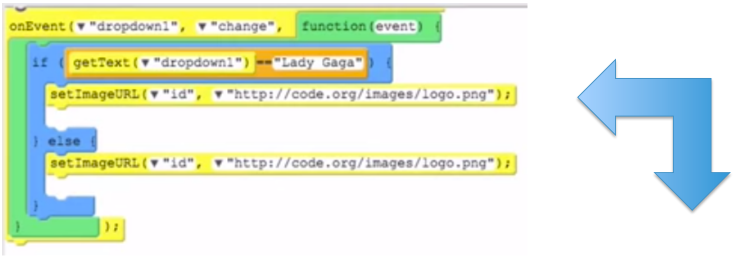
- Folders



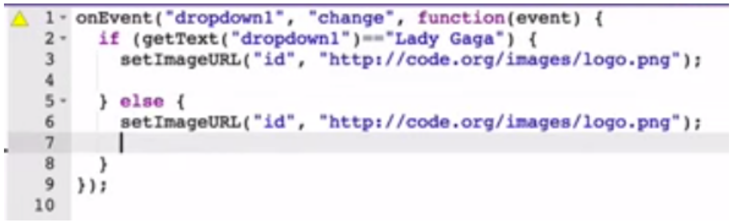
- Searching for blocks on workspace
- Zooming

Usability: Droplet's Isomorphic Blocks/Text Conversion

Used in PencilCode and Code.org's AppLab JavaScript curriculum



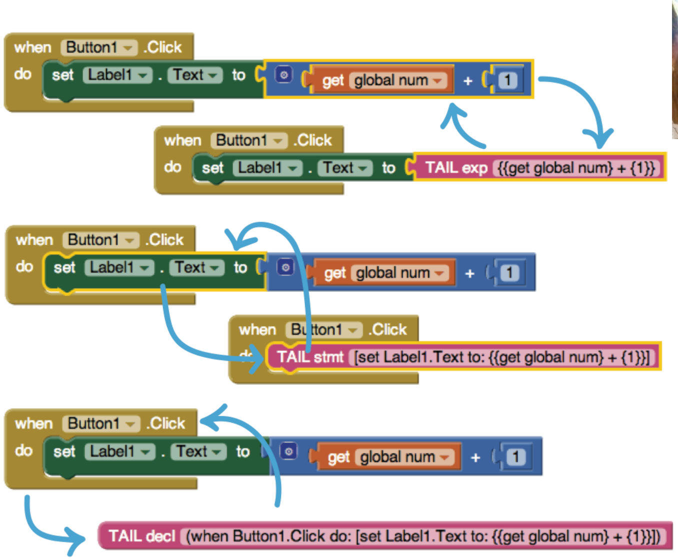
```
onEvent(▾"dropdown1", ▾"change", function(event) {
  if ( getText(▾"dropdown1") == "Lady Gaga" ) {
    setImageURL(▾"id", ▾"http://code.org/images/logo.png");
  } else {
    setImageURL(▾"id", ▾"http://code.org/images/logo.png");
  }
} );
```



```
onEvent("dropdown1", "change", function(event) {
  if (getText("dropdown1")=="Lady Gaga") {
    setImageURL("id", "http://code.org/images/logo.png");

  } else {
    setImageURL("id", "http://code.org/images/logo.png");

  }
});

```

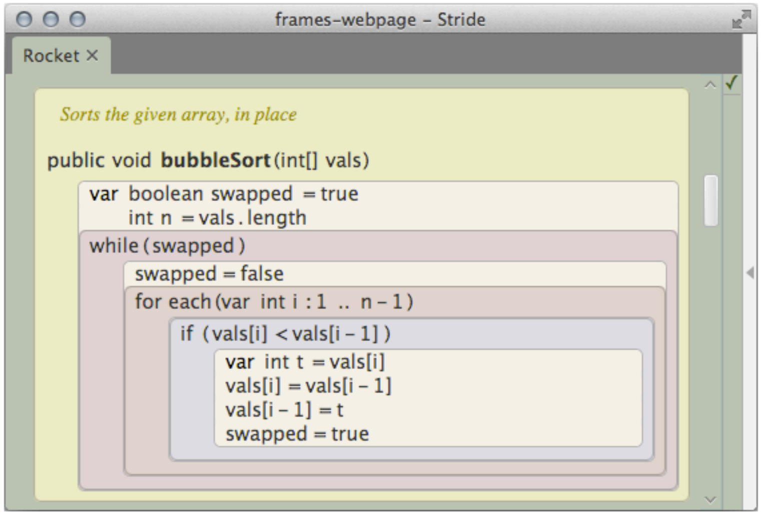
AI: Conversion Between Blocks and Text

Karishma Chadha '14 Wellesley



```
when Button1▾ .Click
do  set Label1▾ . Text▾ to  get global num▾ + 1

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

when Button1▾ .Click
do  set Label1▾ . Text▾ to  get global num▾ + 1

when Button1▾ .Click
do  TAIL stmt [set Label1.Text to: {{get global num} + {1}}]

when Button1▾ .Click
do  set Label1▾ . Text▾ to  get global num▾ + 1

TAIL decl (when Button1.Click do: [set Label1.Text to: {{get global num} + {1}}])
```

Usability: Greenfoot's Frame-based Editing



```
Sorts the given array, in place

public void bubbleSort(int[] vals)
    var boolean swapped = true
        int n = vals.length
    while (swapped)
        swapped = false
        for each (var int i : 1 .. n – 1)
            if (vals[i] < vals[i – 1])
                var int t = vals[i]
                vals[i] = vals[i – 1]
                vals[i – 1] = t
                swapped = true
```

Talk Road Map

- Blocks demo: MIT App Inventor (AI)
- Democratizing programming with blocks: examples
- Lowering barriers with blocks
  - Syntax
  - Static semantics
  - Dynamic semantics
- Challenges in blocks programming
  - Usability
  - Learnability in blocks vs. text
  - Perception: blocks programming not "real", maybe harmful
- Research questions

Learnability in Blocks vs. Text

- Lewis: Logo vs. Scratch study (SIGCSE 2010)
  - Few significant differences between Logo-first and Scratch-first
  - Scratch-first did better on conditionals
  - Logo-first had more confidence as programmers.
- Weintrop and Wilensky: Snap! vs Java (IDC 2015)
  - Blocks easier to read and compose than text
  - Blocks perceived as less powerful, more verbose, inauthentic
- Problem: Nonisomorphic languages
  - Weintrop and Wilensky Commutative Assessment on blocks vs. text in isomorphic languages (ICER 2015) is promising approach
  - Matsuzaka taught Java with blocks environment isomorphic to text (SIGCSE 2015). Students perceived text as more “real”.

Talk Road Map

- Blocks demo: MIT App Inventor (AI)
- Democratizing programming with blocks: examples
- Lowering barriers with blocks
  - Syntax
  - Static semantics
  - Dynamic semantics
- Challenges in blocks programming
  - Usability
  - Learnability in blocks vs. text
  - Perception: blocks programming not “real”, maybe harmful
- Research questions

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

Working with actual code writing instead of a drag & drop interface prepares children better for the real world. http://www.playcodemonkey.com/

Mark Sherman’s Response

Mark Sherman
UMass Lowell

So they currently see this:

when it is really this:

Yes, it is colorful and newfangled, but it still gets jobs done. Not all of them, but a bunch of them.

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

More Positive Feedback

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

- Blocks demo: MIT App Inventor (AI)
- Democratizing programming with blocks: examples
- Lowering barriers with blocks
  - Syntax
  - Static semantics
  - Dynamic semantics
- Challenges in blocks programming
  - Usability
  - Learnability in blocks vs. text
  - Perception: blocks programming not "real", maybe harmful
- Research questions

Some Research Questions

- 2D blocks workspaces:
  - What are good ways to search, navigate, and organize them?
  - Do they confer any advantages over linear text?
- How can debugging & visualization of dynamic execution for blocks environments be improved?
- What tools can improve collaborative development of blocks programs?
- How can we do programming on the devices themselves? (Existing examples: microApps, Pocket Code, Touch Develop.)
- Can any blocks affordances improve productivity in mainstream languages?
- What does big data analysis say about learnability/usability of blocks vs. text notations and transitioning from blocks to mainstream languages?
- What role do the following "nonblocks" aspects play in learnability and usability of blocks languages: web-based environments, cloud-based storage, high-level abstractions, sharing/remixing communities, liveness.

App Inventor Development Team

Hal
Abelson
MIT

Andrew
McKinney
MIT

Jeff
Schiller
MIT

Paul
Medlock-Walton
MIT

Jose
Dominguez
MIT

Mark
Friedman
Google

Sharon
Perl
Google

Liz
Looney
Google

Neil
Fraser
Google (Blockly)

Franklyn
Turbak
Wellesley College

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

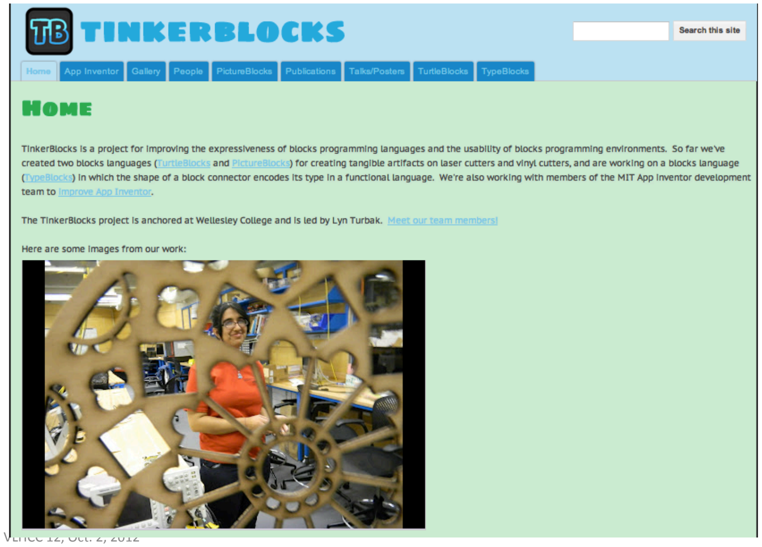
Hal Abelson Shay Pokress Josh Sheldon
MIT

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.

Wellesley TinkerBlocks Students

Questions?



Home

TinkerBlocks is a project for improving the expressiveness of blocks programming languages and the usability of blocks programming environments. So far we've created two blocks languages (TurtleBlocks and PictureBlocks) for creating tangible artifacts on laser cutters and vinyl cutters, and are working on a blocks language (TypeBlocks) in which the shape of a block connector encodes its type in a functional language. We're also working with members of the MIT App Inventor development team to Improve App Inventor.

The TinkerBlocks project is anchored at Wellesley College and is led by Lyn Turbak. Meet our team members!

Here are some images from our work: