Progression of Computational Thinking Skills Demonstrated by App Inventor Users

Benjamin Xie

Dept. of Electrical Engineering & Computer Science, MIT App Inventor

OBJECTIVE

Understand how people learn by creating apps

How do people develop **computational skills** creating with App Inventor?

What is the typical **order** for acquiring computational concepts in App Inventor?

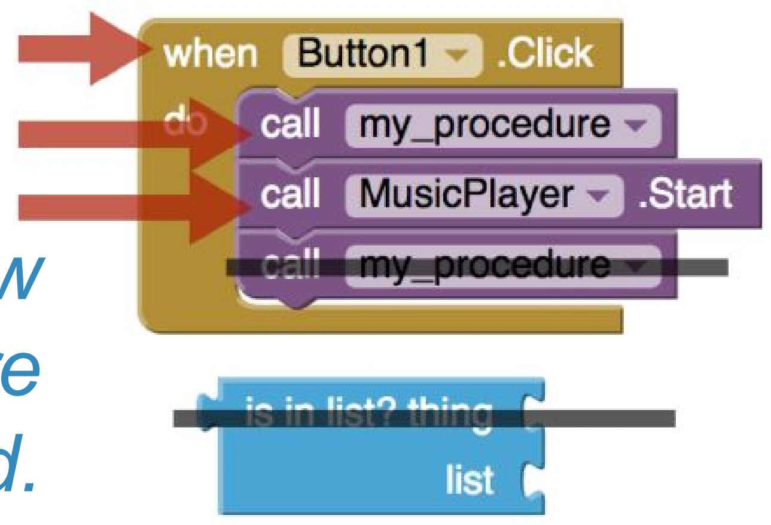
What is the **relationship** between developing skills to program with App Inventor and skills to use computational concepts?

DATA

AI Projects from 10.5k users

First 20 projects of random users who created ≥ 20 projects (total: 211,420 projects)

Analyzing **Blocks** in each project
- # of block types
- # of events responded to

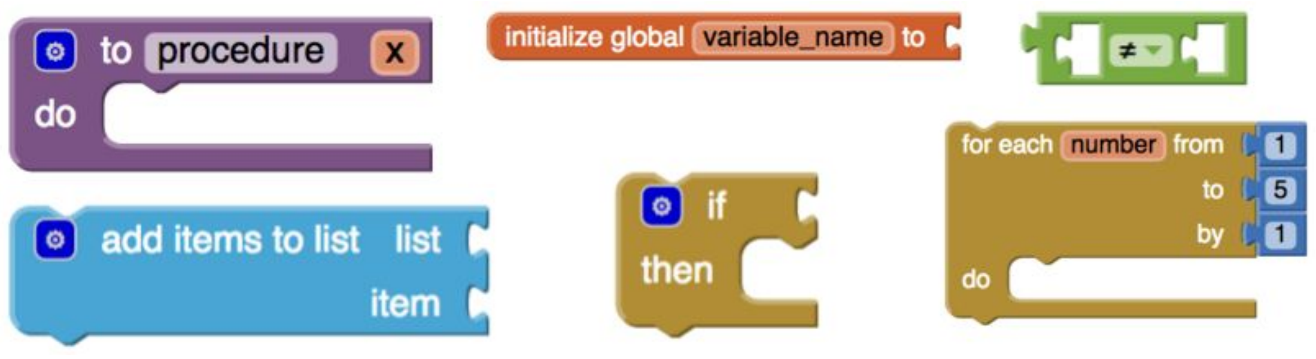


Example of how block types are counted.

BACKGROUND

Computational concepts are generalizable

Computational concept (CC) blocks related to procedures, variables, logic, loops, conditionals, lists



Example of Computational Concept (CC) Blocks

METHOD

Experiment 1: Measure the breadth and depth of capability to use computational concepts

Objective: analyze progression of project sophistication (breadth, depth of capability) as users create more projects

For each project:
Breadth of capability: number of ***new*** block types introduced
Depth of capability: ***total*** number of block types used

Separate CC-blocks and non-CC blocks and analyze each set.
- CC-blocks: generalizable skills (computational concepts)
- non-CC blocks: domain-specific skills (app functionality)

Experiment 2: Cluster users and analyze centroids

Objective: Understand common learning patterns by clustering users, qualitatively inspecting users that represent population (centroids)

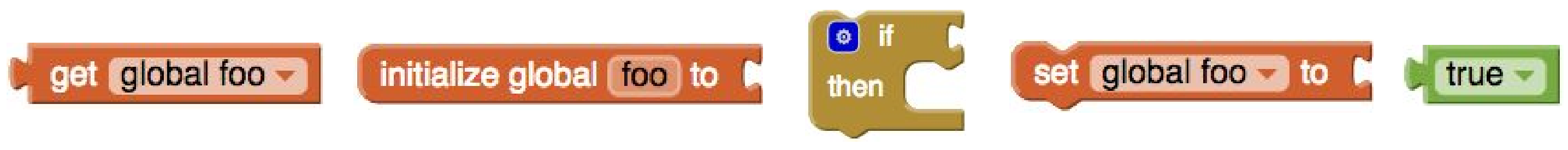
Cluster by CC blocks used in each project for each user. CC blocks are IDF weighted so less frequently used blocks have greater weight/relevance.

Look at 3 users nearest to center of cluster and identify similarities in order CC blocks introduced

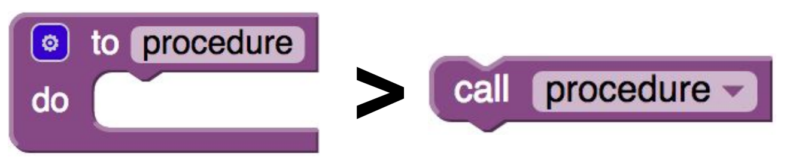
RESULTS

Frequency of Computational Blocks

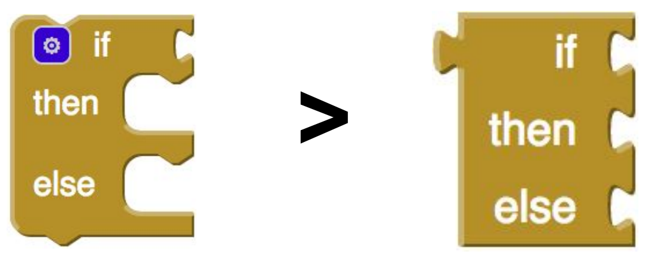
Most Common CC Blocks



More procedures are defined than called



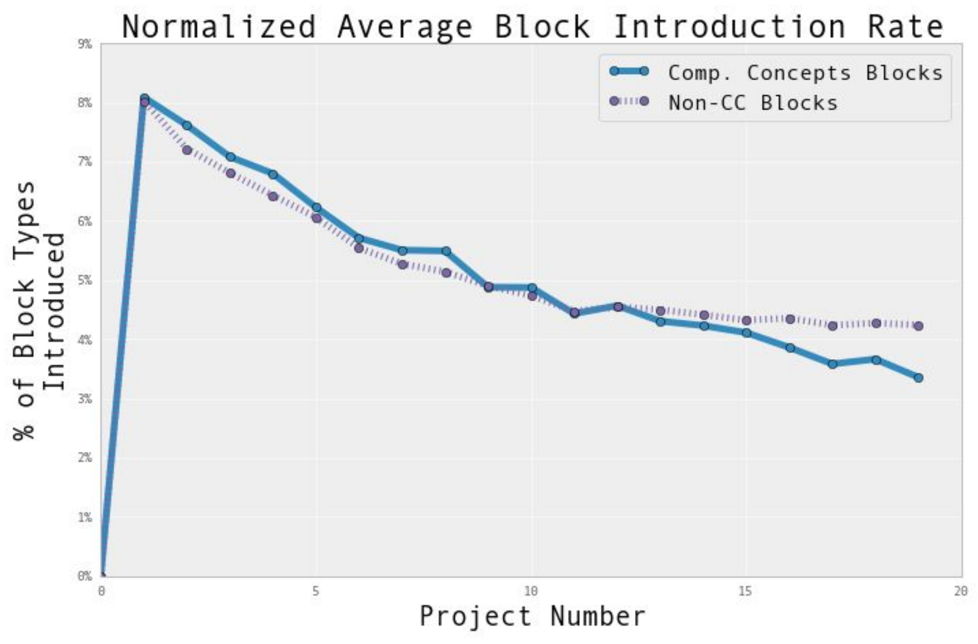
Statements are used more than expressions

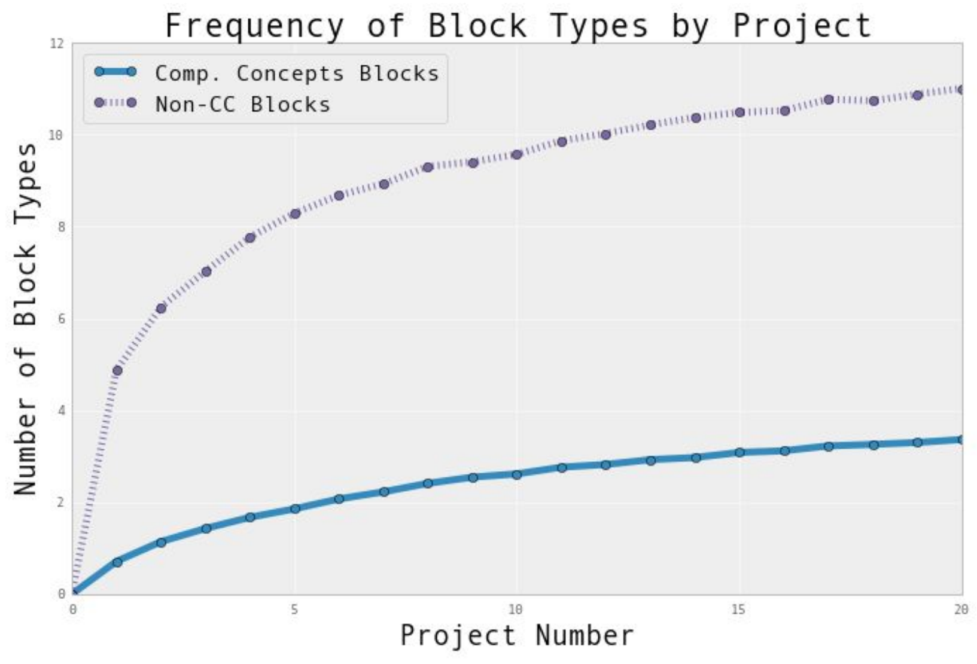


Progression of Breadth, Depth of Capability

Breadth decreases

Learners use fewer ***new*** blocks as they create more apps



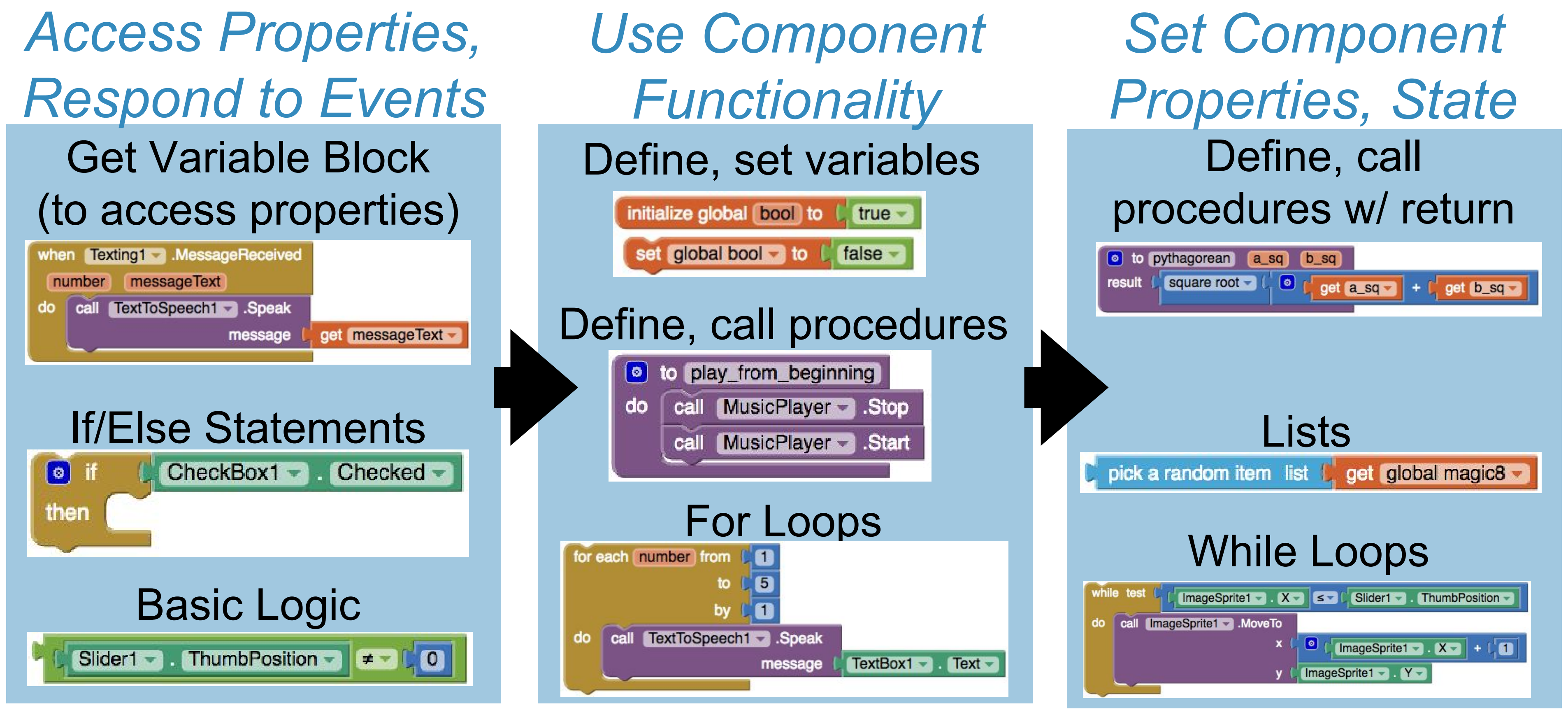


Depth increases

Learners use more blocks as they create more apps

Complexity Measure

Perceived complexity of computational concepts relative to component usage


when Texting1 .MessageReceived
number messageText
do call TextToSpeech1 .Speak
message get messageText
if CheckBox1 . Checked
then
Slider1 . ThumbPosition ≠ 0
initialize global bool to true
set global bool to false
to play_from_beginning
do call MusicPlayer .Stop
call MusicPlayer .Start
for each number from 1
to 5
by 1
do call TextToSpeech1 .Speak
message TextBox1 . Text
to pythagorean a_sq b_sq
result square root get a_sq + get b_sq
pick a random item list get global magic8
while test ImageSprite1 . X ≤ Slider1 . ThumbPosition
do call ImageSprite1 .MoveTo
x ImageSprite1 . X + 1
y ImageSprite1 . Y

Access Properties, Respond to Events
- Get Variable Block (to access properties)
- If/Else Statements
- Basic Logic

Use Component Functionality
- Define, set variables
- Define, call procedures
- For Loops

Set Component Properties, State
- Define, call procedures w/ return
- Lists
- While Loops

CONTRIBUTIONS

Quantitatively analyze the progression of using computational concept skills

Teachers can develop curriculum with App Inventor's event-based environment in mind

Students can monitor and measure their progression of learning

Researchers can quantitatively measure progression of skill in blocks-based environments

Student image by Michal Kučera