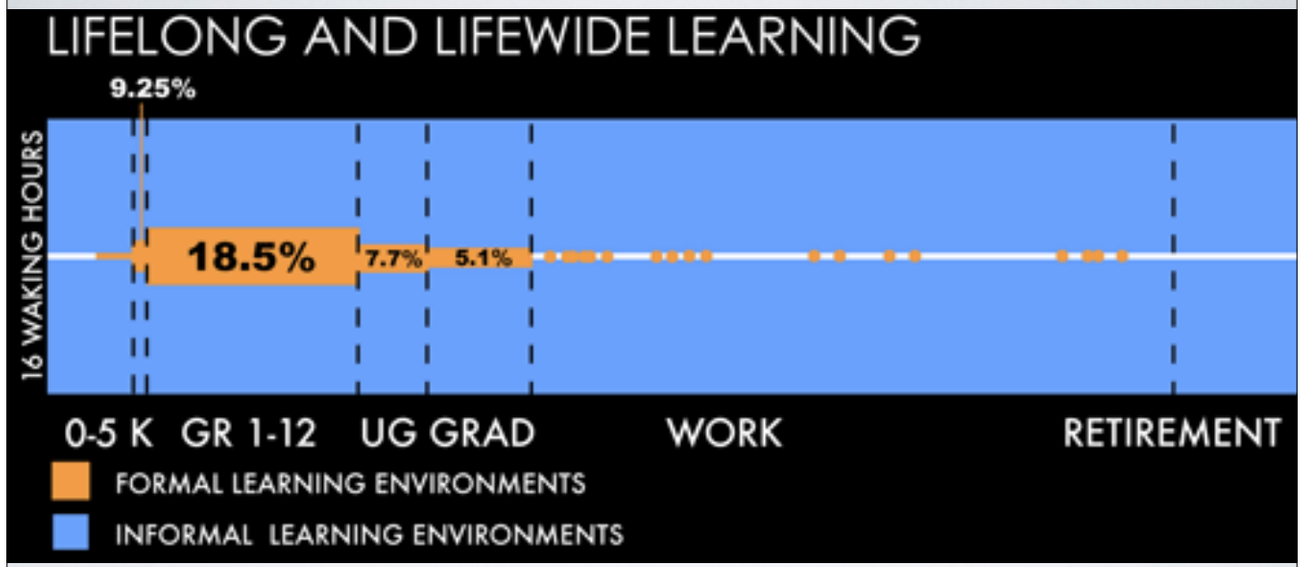
HOW WE DEVELOP SKILLS BY MAKING APPS

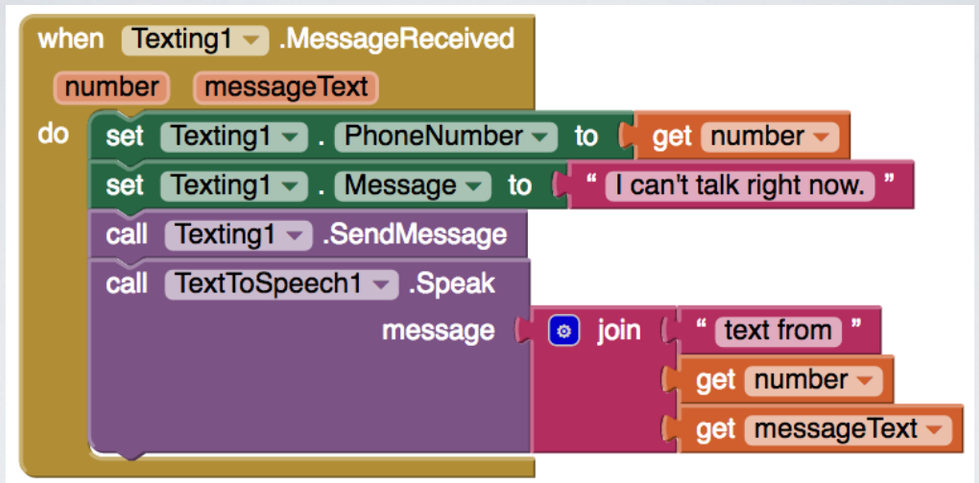
Progression of Computational Thinking Skills
Demonstrated by App Inventor Users

Benjamin Xie, MIT App Inventor
May 2016



LIFE Center: Stevens, R. Bransford, J. & Stevens, A., 2005

When we think about where learning happens, our minds wander to our favorite (or least favorite) lecture hall or classroom. But the truth is, even at the peak of our grammar school days, we will spend most of our life outside the classroom. This graph depicts the amount of time you will be in a spend in a formal learning environment (in yellow) and outside of one (in blue). This thesis goes into understanding how people learn when they are in the “Sea of Blue.”



195 COUNTRIES
4.7 MILLION USERS
14.7 MILLION APPS MADE

MIT App Inventor is an online visual blocks based programming environment where users with users can build fully functional Android applications. The visual blocks programming language removes syntax concerns so people who are new to computer science can focus on learning the semantics. Here, you see blocks for an app that responds to texts received and reads them aloud.

Although the intention of App Inventor is to each programming by creating apps (constructionism), it is extensible enough such that

VISION

My vision or long term objective is to understand how people learn computational thinking skills in informal settings and connect this learning to classroom experiences. For my thesis, I want to understand how developing domain-specific App Inventor skills relates to developing transferrable computational thinking skills. I also want to understand how these learning behaviors vary by different types of users.

STEPS

- Data
- Computational Concept Blocks
- Demonstrated Skill

I take steps towards this vision by quantitatively modeling the demonstrated skill of users under two dimensions: breadth and depth of capability.

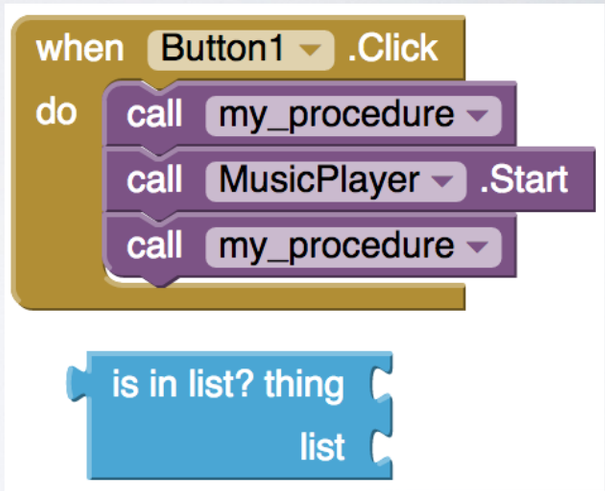
DATA: PROJECTS FROM LONG-TERM USERS

- Users with at least 20 projects
- Looking at blocks
- Interested in skill progression across projects

top 1.4% of users
looking at blocks (ignoring components)
interested in progression of skill by user across projects

COUNTING BLOCK TYPES

- Prevent double-counting
- Ignoring non-functional blocks

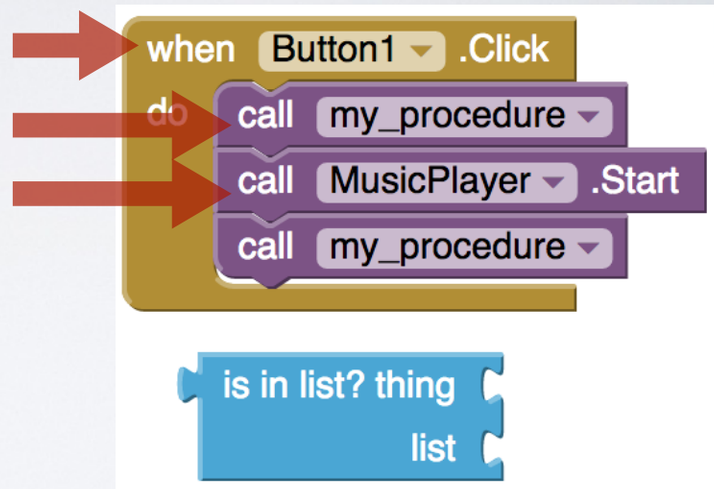


- counting block types more robust than counting blocks
- example: alice and eve have apps with equivalent functionality except alice uses a procedure and eve doesn't. alice: < blocks, > block types
- ignore blocks with no function

(example)

COUNTING BLOCK TYPES

- Prevent double-counting
- Ignoring non-functional blocks

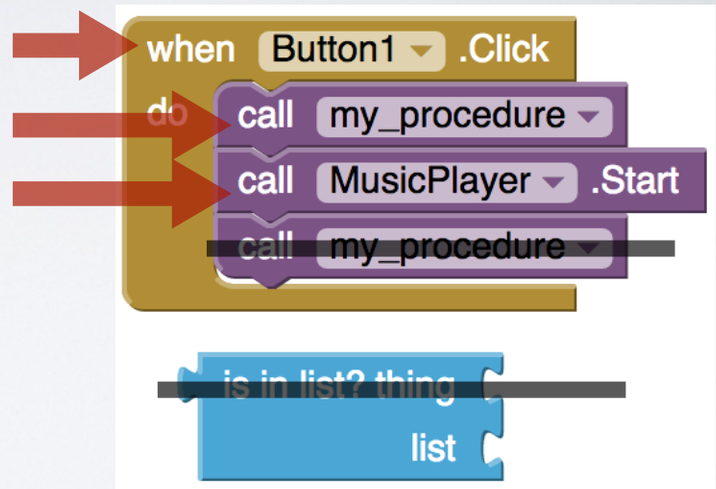


- counting block types more robust than counting blocks
- example: alice and eve have apps with equivalent functionality except alice uses a procedure and eve doesn't. alice: < blocks, > block types
- ignore blocks with no function

(example)

COUNTING BLOCK TYPES

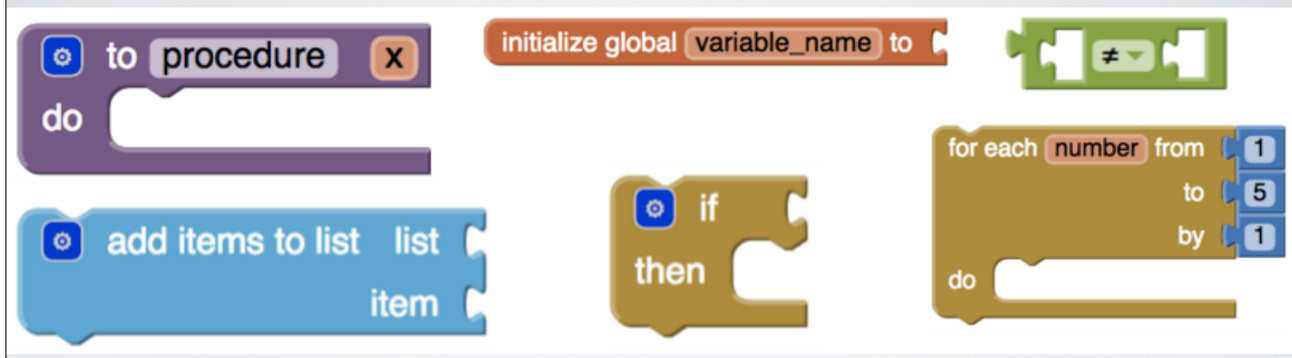
- Prevent double-counting
- Ignoring non-functional blocks



- counting block types more robust than counting blocks
- example: alice and eve have apps with equivalent functionality except alice uses a procedure and eve doesn't. alice: < blocks, > block types
- ignore blocks with no function

(example)

COMPUTATIONAL CONCEPTS ARE TRANSFERABLE



DEMONSTRATED SKILL: **BREADTH + DEPTH**

- BREADTH: Number of **new** block types used in project
- DEPTH: **Total** number of block types used *in* project

breadth: number of types of blocks used for the first time by that user
depth: total number of block types used in project

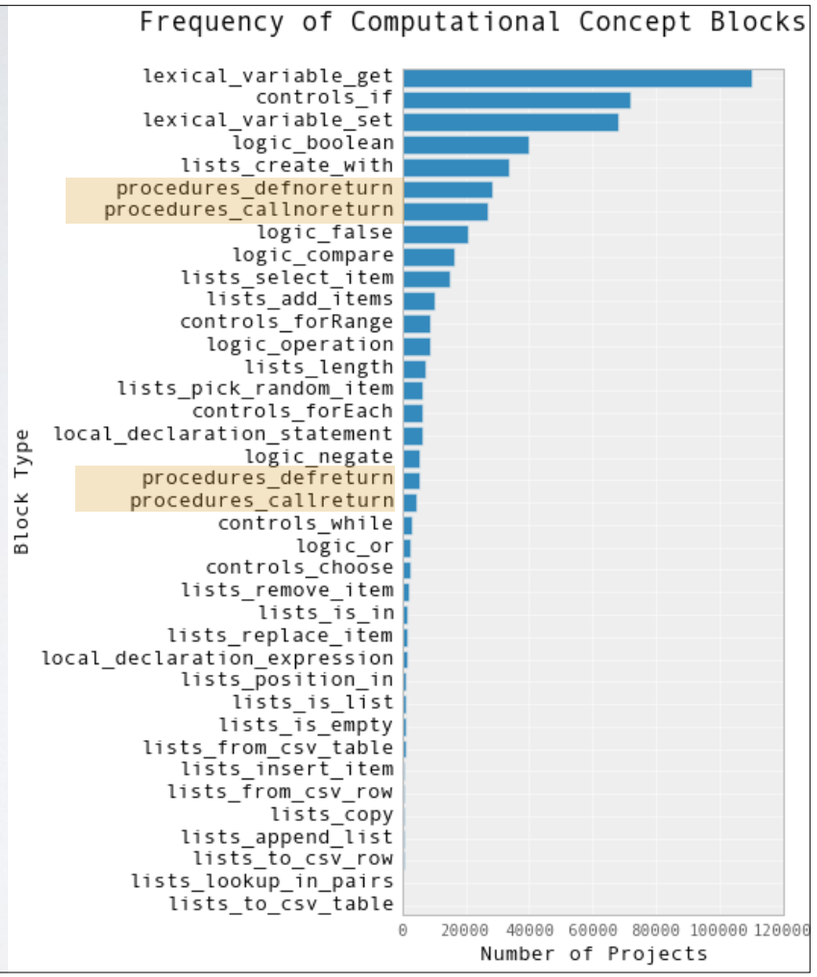
NEWS

- Frequency of Computational Concept Blocks
- Breadth
- Depth

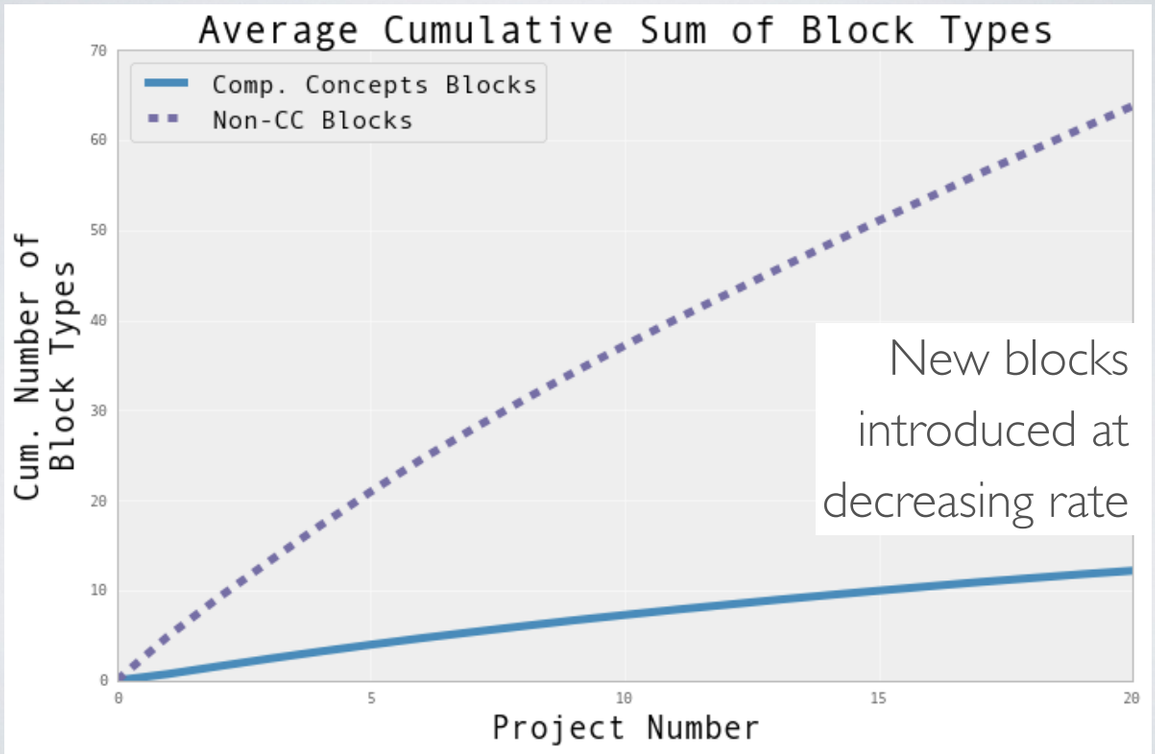
Frequency

CC BLOCK FREQUENCY

- | Procedure Def'n | > | Procedure Call |
- Least common: Local variables, advanced list operations, loops

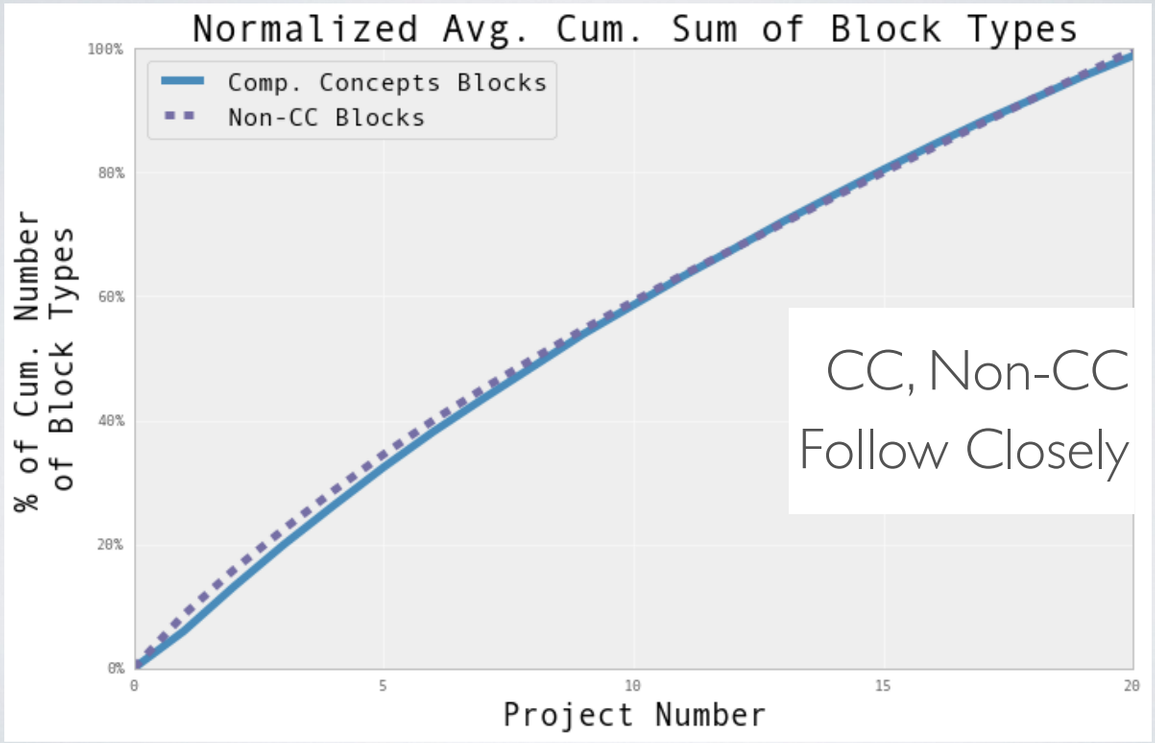


BREADTH



somewhat

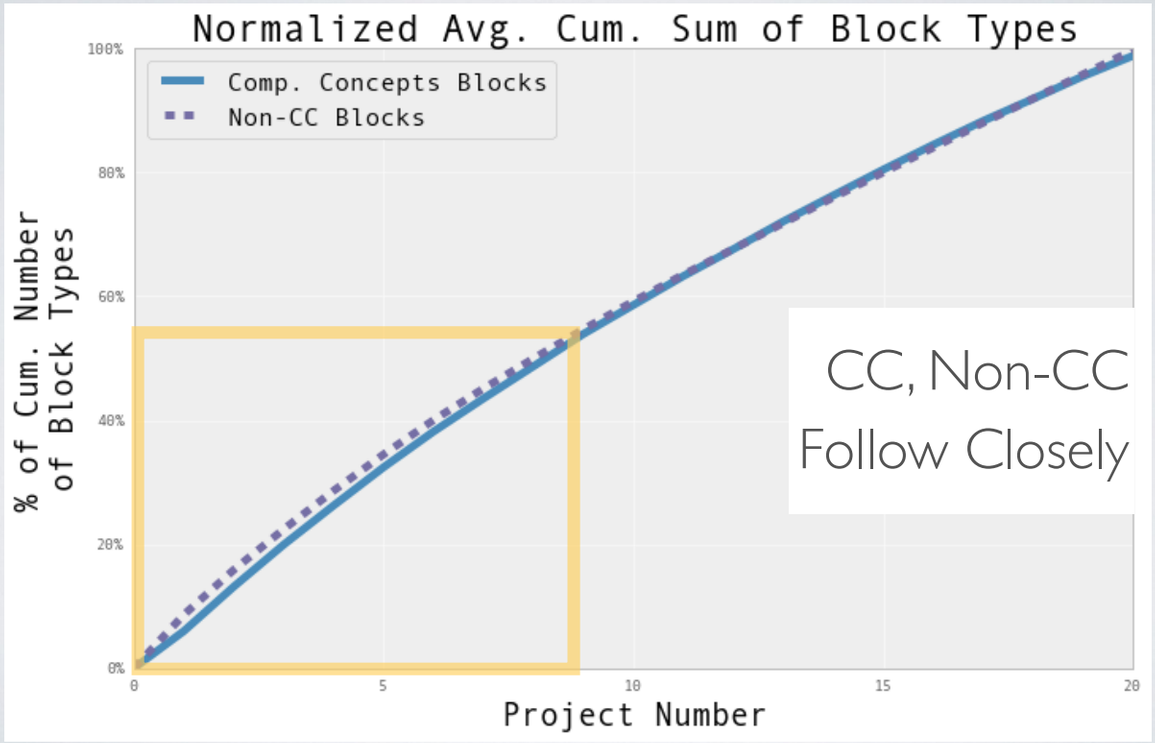
BREADTH



-CC and Non-CC follow very closely

People expand the breadth of their App Inventor knowledge and CC knowledge in similar patterns

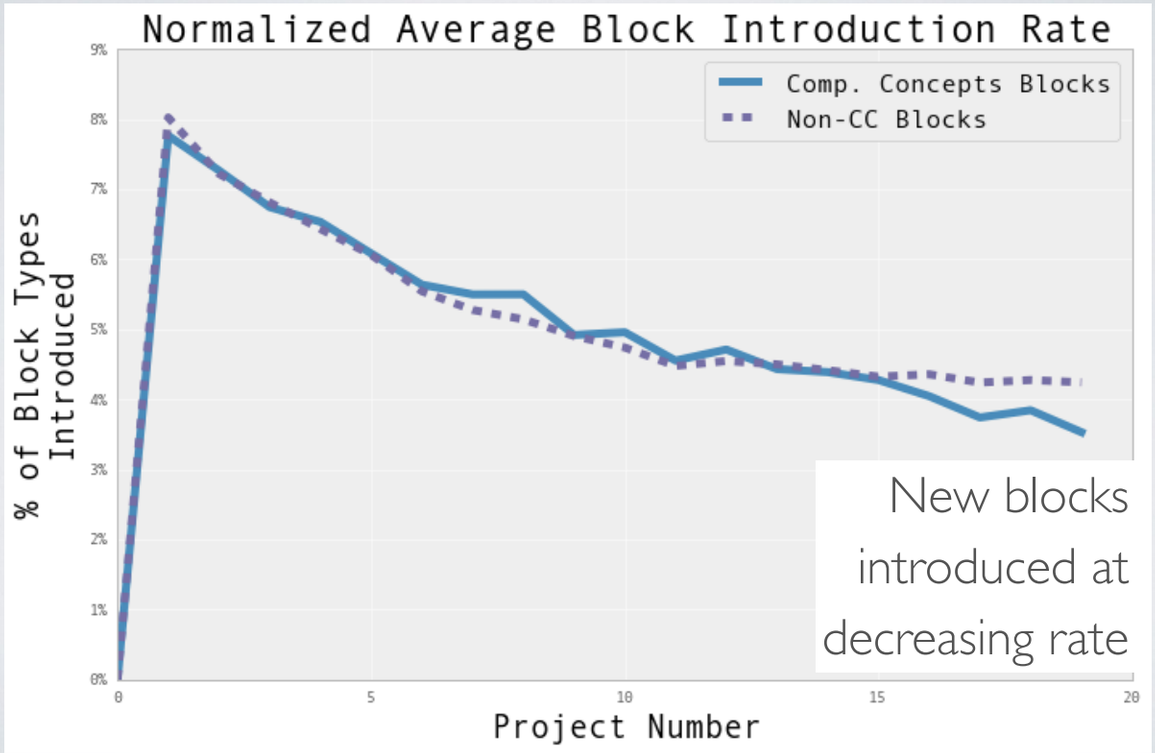
BREADTH



-CC and Non-CC follow very closely

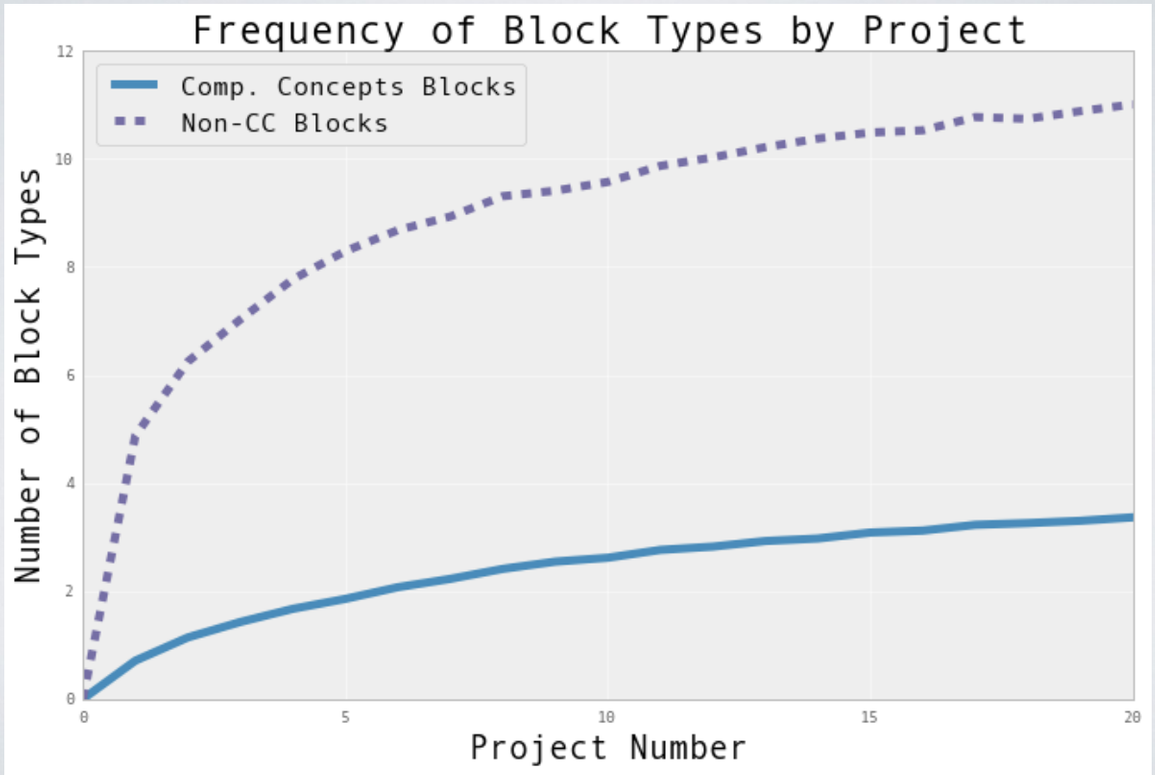
People expand the breadth of their App Inventor knowledge and CC knowledge in similar patterns

BREADTH

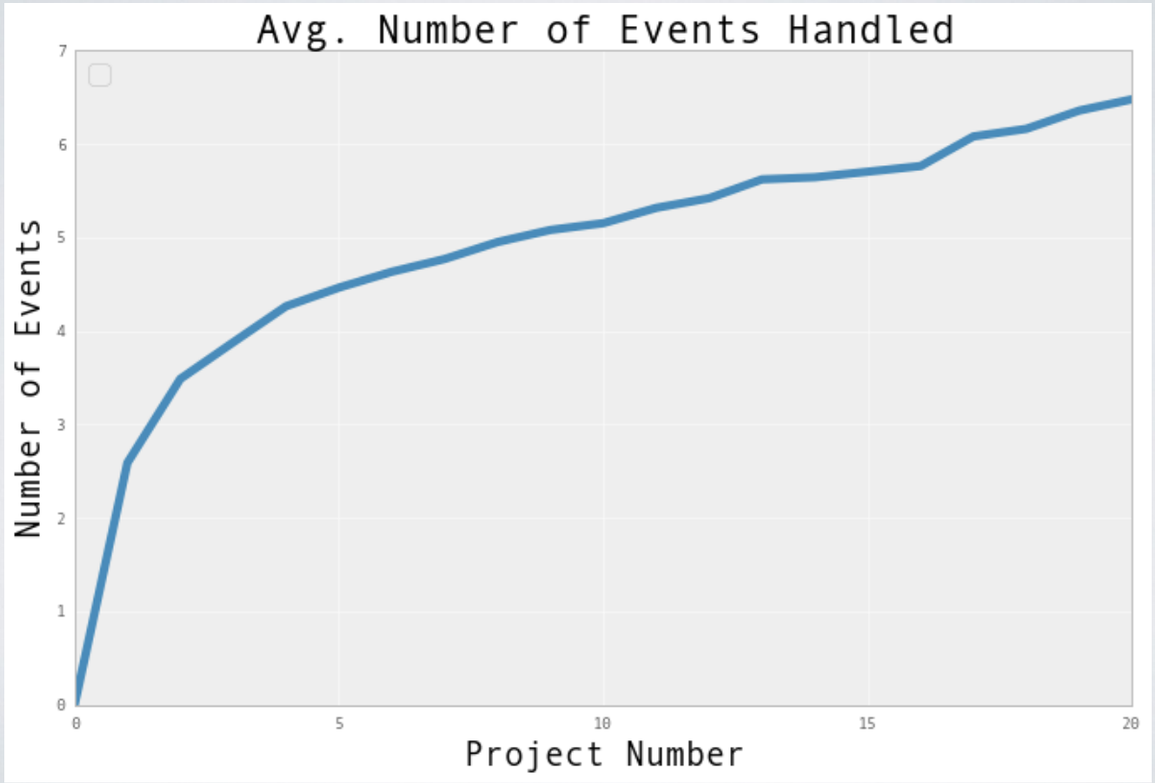


New blocks introduced at decreasing rate

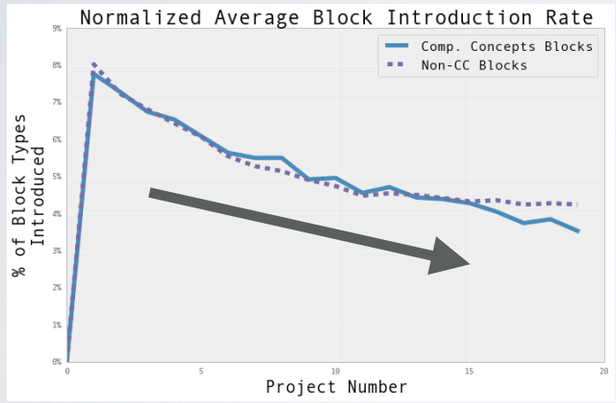
DEPTH

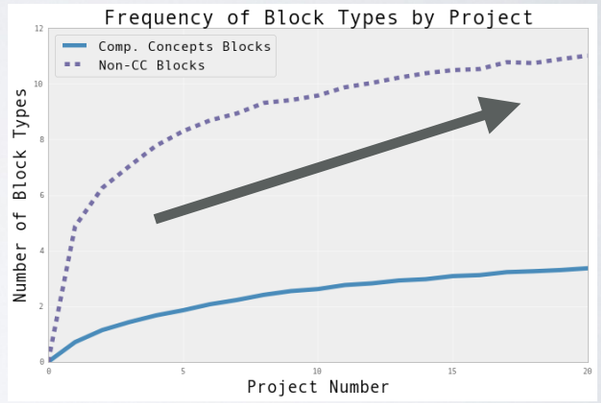

Frequency of Block Types by Project
Comp. Concepts Blocks
Non-CC Blocks
Number of Block Types
Project Number
0
2
4
6
8
10
12
0
5
10
15
20

DEPTH


Avg. Number of Events Handled
Number of Events
Project Number
0
1
2
3
4
5
6
7
0
5
10
15
20

BREADTH THEN DEPTH?





CONTRIBUTIONS

- Modeled demonstrated breadth and depth of App Inventor user capability
- Compared demonstrated capability of using computational concepts with using App Inventor functionality
- Identified common learning patterns among different types of users
- Compared results to Scratch