Rationale

- Computational thinking (CT) represents important skills for young people and it is important to understand how to support and promote it.
- Scholars and theorists including Jenkins, Ito, Lessig, Manovich, and others have pointed to remixing and appropriation as an important skill, social and cultural practice, and literacy for young people.
- A major reason for excitement about remixing is that engagement with material created by others with different experience, skills, and knowledge is that exposure to others' material is a mechanism for learning.
- A major design goal of the Scratch online community is to promote learning of CT by encouraging users to remix each others projects and, through that process, become exposed to and learn more concepts computational concepts and learn them more quickly.
- Our study will examine the relationship between engagement in remixing and the level and speed that users demonstrate new computational thinking skills expressed in the projects they remix.

Objectives

General Objectives

- To test the theory that remixing or appropriation of code are a mechanism through which individuals can be exposed to and learn computational thinking skills.

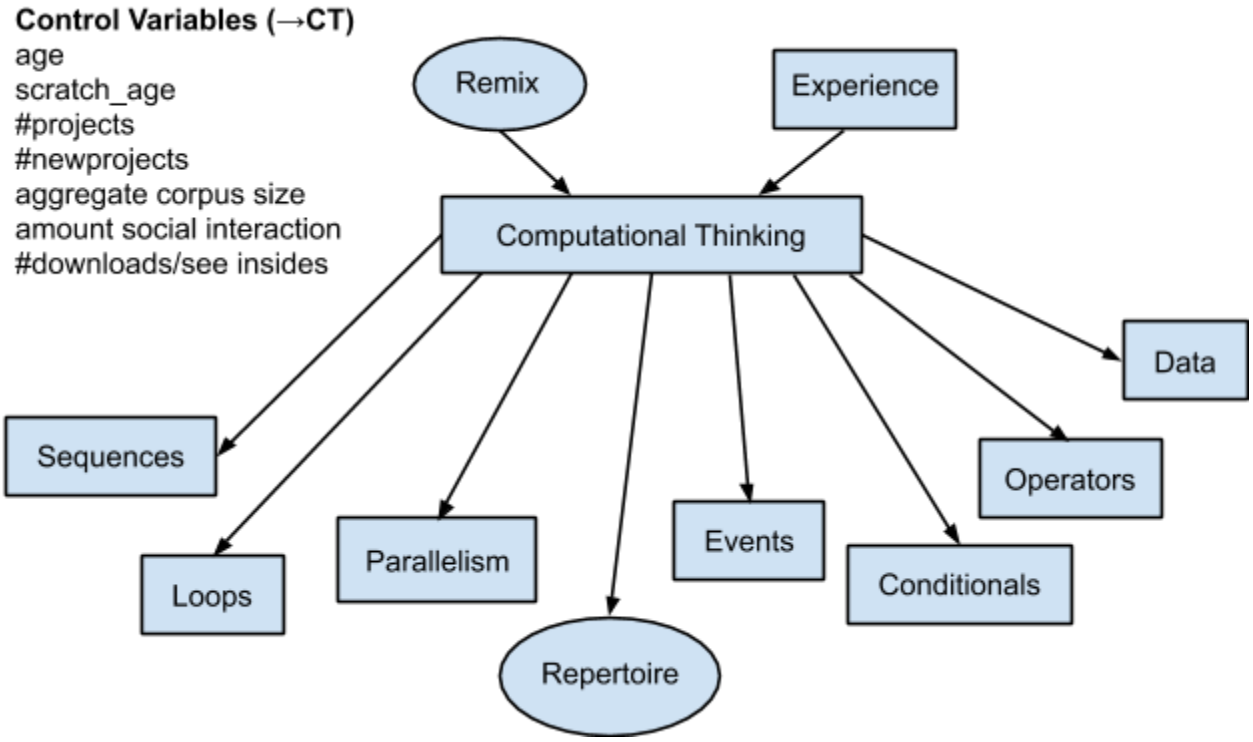
Specific Objectives

- To examine how *ceteris paribus* a user's block repertoire increases when the user remixes others projects that use unfamiliar blocks.
- To examine how *ceteris paribus* the likelihood of a user demonstrated CT concepts in their de novo projects increases when that users remixes projects that use those concepts.

Null Hypotheses

- There is no difference in the change in size of a users' block repertoire associated with a user remixing a project.
- There is no difference in the change in size of a users' block repertoire associated with a user remixing a project compared to the same user uploading an additional de novo project.
- A user remixing a project is not associated with an increase in the likelihood of that user demonstrating a CT concept.

Conceptual Model/Diagram



Measures

Measures of learning (*New frameworks for studying and assessing the development of computational thinking*)

Concepts	**Measure**
Sequences	Long list of blocks (not very accurate). Blocks that are longer than a particular threshold (e.g., longest stack in project is > 66% of projects). Contains any sequences of code. 1 (most users) or 0
Loops	Uses looping blocks (e.g., **Forever** block)
Parallelism	Parallel stacks with same “hat” block.
Events	Uses “**when <>”** hat blocks
Conditionals	Uses conditional blocks (e.g. “if” block)
Operators	Uses operator blocks (e.g. “+” or “or” blocks)

Data	Uses variable or list blocks
General	Repertoire of blocks in de novo projects (e.g., number of unique blocks used in de novo projects at least once).

Dummy Tables

MISSING

Threats to Validity and Limitations

Threat: Users CT measures may increase for reasons unrelated to, but correlated to remixing.

- Add control variables to control for effects of other variables.
- Find out/limit changes in CT (e.g., blocks added to repertoire and/or new demonstrated CT concepts) that are reflected in antecedent/remixed projects.

Threat: Lack of process information. For example, (a) code may be copy-pasted manually and (b) users may have have help from someone sitting beside them (see Brennan and Resnick).

- (a) Make demonstration threshold multiple positives in different projects.
- (a) Create an alternate version of block repertoire or a threshold for demonstration of a CT concept that requires that we “find” the block/concept in two different contexts (i.e., blocks are surrounded by different blocks).