Rethinking How We Learn to Program

Patrick Ester - pxe120030@utdallas.edu

Problem

For all of the progress in computers and technology, the act of programming a computer remains largely the same. Text is entered into an editor, which is then run through a compiler or interpreter. Output is separate from input. IDEs, code folding, syntax highlighting, and the like augment the paradigm rather than alter it. As such, coding is structured in such a way that restricts its accessibility to novices, especially children. If coding is, as they say, the new literacy, why must it retain the tradition of opacity?

Pseudo Solutions

Efforts to promote coding like Scratch and Khan Academy are making strides to reimagine the way students learn to code. Scratch brings a graphical drag and drop approach, while Khan Academy provides live coding, code completion, mutable example videos, etc. However, they still miss the mark in several categories. The scripts in Scratch are pre-made and immutable. Users cannot decompose these commands to understand, alter, and remix their own. Khan Academy lessons are constrained by the limits of the JavaScript language. Curly braces, semicolons, draw loops, and the like are all obstacles to learning to code.

Project

This capstone will evaluate the Elm programming language as an instructional tool for novice programmers. The guidelines for suitability come from the writings of Seymour Papert, Bret Victor, and Doug Engelbart. Papert provides global guidelines for computers, kids, and learning; Victor suggests several characteristics for the tools and the language itself; and Engelbart advocates the relearning of computers to create a richer experience for the user.

Plan

I am having some trouble narrowing down the particulars of the project. It would be beneficial to build something in Elm to acclimate myself with the language. In addition, the object built should aid kids in learning to program.

Unresolved Questions

There are still many questions to be asked of this project.

- Should there be a physical component to the project?
- Should the project focus on concepts like conditionals, variables, etc. and be language agnostic?
- Is focusing on the Elm language distracting from the larger goal of rethinking the tools for teaching kids to program?
- What form should the programming aid take? A game? What steps can be taken to ensure it is a gender-neutral approach?

Deliverable

The capstone is a project in two parts. The first is the blog (www.patr1ck.net) that documents the learning process for Elm. If Elm is not to be used, then the blog will record the project in general. The second is the educational tool for learning programming. It is also important to share the findings of this project so that others can learn from and improve upon this project. The sharing will take place at a local level, such as EdCamp Dallas or at a meet-up of Dallas area private schools. The findings may make their way to a larger audience at the BigDesign Conference.

Feedback

I would appreciate any feedback you may have for me either on this form or sent to me - pxe120030@utdallas.edu. Thank you.

Summary

Please summarize my project to see if I am clearly expressing my idea.

Answers

What are some answers to the questions on the previous page?

Questions

Are there any additional questions that need to be asked?

Thoughts/Ideas

Do you have any other thoughts or ideas that would improve this project?