

Attraction and Retention of Computer Science Majors through Experimentation and Zero-Entry Fee

Thesis Proposal

Introduction

Attraction and retention of Computer Science majors is a pressing concern as numbers of computer science majors continue to dwindle despite demand for educated computer scientists grows. From informal interviewing, it becomes clear that most professional computer scientists and computer science majors chose that path because they were introduced to computer science by someone else. ~~(Reference pending.)~~ The dependence on a mentor indicates that there is an initial hurdle, an entry fee if you will, that must be overcome before someone will choose computer science as a major. This entry fee may be in the form of overcoming misconceptions about computer science ("It's only for geniuses and bookworms", "It's too difficult to do", "Computer Science is for nerds."); it may be simply a lack of knowledge of what is needed to get started. Part of the entry fee is also commonly the time it takes to set up a compiler and IDE. The aim of this project will be to address the hurdles of the misconceptions about computer science and the initial setup and attempt to, as completely as possible, lower the entry fee to encourage choosing computer science as a major.

Method

Creation of a website that will encourage people to learn to programming

We propose the creation of an experience-based tutorial website. This website will utilize media (text and streaming video) and a new application written for this project; an in-browser compiler that the user can use to experience

programming as they learn. The tutorials will take the novice from their first program to more advanced programming concepts with example programs that will encourage the user to experiment and grow.

The site will also feature a gallery of programs submitted by users of the site and encourage submissions by the community. Visitors will be able to search a categorized list, view [or edit](#) the source for these programs, and run them. During this phase, usage and popularity of the pages being visited will be monitored to determine how many of those who start the tutorials finish the tutorial series. Formal polls of visitors will gauge whether these tutorials have changed their minds about choosing Computer Science as a major.

By removing the need to set up a compiler, a user will be able to play with programming without a setup entry fee. They can then later determine if they want to load a compiler to pursue programming and computer science further. Formal polls will also determine if more people are choosing Computer Science as a major. [Again, examples of the kinds of questions would be helpful.]

A upon registration to the site a first time user will be encouraged to take a short poll to determine their aptitude and level of expertise, personality type, background and demographic, as well as their interest in computer science as a major. ~~Three-One~~ [months](#) after launch and again ~~6-months~~ [three months](#) after launch, ~~the~~ [users-base](#) will be polled again to determine how far they have managed to progress and whether

their interest in computer science has changed. ~~[Likely do polls based on time since individual started, rather than polling everybody at the same time. Three months may be too long, as you will have no information from those that don't make it that far. I'm guessing MOST won't still be active three months out.]~~ From this, we will determine the numbers for the following statements:

- x% of those who started with our approach indicated they intended to take a formal computer science class.
- Those who were of demographic A tended to have a higher success rate with our approach. Those who were of demographic C did least well in terms of enjoyment and completion.
- y% of the people, who made it past lesson c, finished all lessons. The lessons were ordered from easiest to most difficult. The highest loss in retention occurred between lessons a and b.
- One of the lessons in the series was more like a typical computer lesson in that XXX. We found that students were (or were not) motivated to complete that lesson. We think that is because...
- We will also consider the effect of personality type, gender, and previous computer experience.

Online Compiler Specifications

Development of an online compiler built on Adobe Alchemy technology to compile C/C++ code into AS3 that can be run in browser.

We propose the creation of a browser based text editor that will link to an adobe alchemy [\(or similar\)](#) compiler running on a remote system which would then return the C/C++ code compiled ~~to a flash file that could then be~~ run in a [browser](#) window ~~below the editor window~~. When the program is being run, the code will be

saved and editor window will be inactive. When the editor is being used the program execution will stop.

The application will be developed so as to allow the preloading of source. This function will be used in tutorials. A partial program will be loaded with tutorial like lesson, questions, or instructions in a side frame.

The initial pass will not include any [automated](#) grading, aside from the user's own experience. ~~if the lessons are being reviewed in an educational environment then a teacher would be allowed access to the edited code. Later iterations of the website can feature automated grading of the edited code. [Do you mean actual grading or just feedback on the quality of the code? How will this be automated? This could be a very difficult part.]~~

Poll questions

Sample initial poll question

Upon registration new users will be encouraged to answer some questions. The questions will be as follows:

1. Rate your level of programming experience:
 - a. Never programmed before.
 - b. Very little programming.
 - c. Some programming.
 - d. Experienced, but not in C/C++.
 - e. Experienced.
2. Right now, what do you think of going to school (college) for computer science?
 - a. Don't think I could do it.
 - b. Not interested.
 - c. Might be interested.
 - d. Interested.

- e. Very interested.
 - f. Already in school for Computer Science.
3. Have you ever tried learning programming before? Please describe the method you used.
 4. If you have ever taken a Briggs-Meyer test, what was your personality type? (Option for don't know and a link to a free online test if they'd like to explore.)
 5. Are you:
 - a. Male
 - b. Female
 - c. Prefer not to answer.
 6. What country/region are you in?

a list of lessons and an option of "None".)

3. Which lesson do you remember being the most fun at the time?
4. Right now, what do you think of going to school (college) for computer science?
 - a. Don't think I could do it.
 - b. Not interested.
 - c. Might be interested.
 - d. Interested.
 - e. Very interested.
 - f. Already in school for Computer Science.
5. Suggestions for improvement? (Open form.)

Post lesson questions

After each lesson/tutorial on the site the following questions will be asked:

1. Did you think this lesson/tutorial was:
 - a. Too easy.
 - b. About right.
 - c. Too hard.
2. Suggestions for improvement. (Open form)

~~3~~One and ~~three~~6 month follow up questions

After a period of time those registered on the website will be sent an email asking them to fill out another poll with the following questions:

1. How many lessons/tutorials have you completed?
2. Which lesson do you remember being a hard one to do at the time? (Followed by