Beginner CS Manual Usability Study

Abstract

The collaborators of this project recognize that computer science is a very complex and intimidating area of study. Sharing this knowledge motivated us to create a simple beginner's programming manual. This manual aims to introduce basic Python programming to middle school students interested in computer science. Possessing these skills before high school benefits students' ability to solve problems and gives them experience working in tech. The advancement of this manual will consider the most optimal way to yield such benefits.

Introduction

This manual serves as an introductory computer science course which explains basic concepts, programming structures, and syntax to middle school students or even beginners interested in programming. The manual consists of modules that cover basics of the Python language. Python is a very concise and readable language which we found most appropriate to present to beginners. If students desire to learn other less-abstracted programming languages after completing our course, at least they have an understanding of Python which can perform just as well.

Learning the basics of computer science helps not only those directly involved in the field, but many other professions! Completing this manual gives students a basic understanding of computer science, and the opportunity to strengthen their problem solving skills and attention to detail. Specifically, the ability to examine a problem and break it down into smaller parts to find steps to a solution. These are valuable skills that middle school students can apply later in high school, college, their careers, and their lives!

Process Description

Our group definitely "began with the end in mind" since we collectively had a vision of what a well constructed beginner's programming manual looked like. As computer science college students, we wanted to construct a manual we would have wanted in middle school. We kept things simple and focused on the very basics like variables, conditional statements, functions, etc. This covers everything students need to build simple programs. Including more complex structures and concepts makes the course longer and more difficult to understand.

The process of creating the manual went rather smoothly. The first task after the initial project pitch was to decide the specific content to include in the manual, and how to order it. After that, we posted the modules on a Kanban board in GitHub. Each group member was responsible for at least one module. During the completion of these modules we split up further responsibilities. Some modules took longer to complete, but those who had more free time began constructing the website and editing the GitHub. My individual responsibilities included the functions module, the final project, and ReadMe. This was slightly more than the average workload only because I felt confident with what I contributed.

Visibility also functioned as a significant part of the creative process. All of the content produced was posted and visible, so the other group members had the opportunity to give feedback and ask questions. Most communication occurred in workshop class, or over group text. Staying on the same page ultimately led to the success of the manual.

Being proactive was my biggest takeaway from this project. I complete a majority of my work alone, so I feel wired to do things “my way”. Working in a group meant that everybody had a say, and feedback was always present. We established goals and communicated throughout the process. We also were able to “sharpen the saw” by sharing skills with each other. Some of us went into the project without knowing how to use a Kanban board or create a website from scratch, but now we all have experience with such. I could even say we also sought first to understand, then to be understood. I cannot speak for the other group members but I had to do a little research and refresh on Python syntax. Since this project involved teaching, it was especially important that we understood the content taught. Looking back at it, we learned a lot as a group just by trying to help others learn what we know.

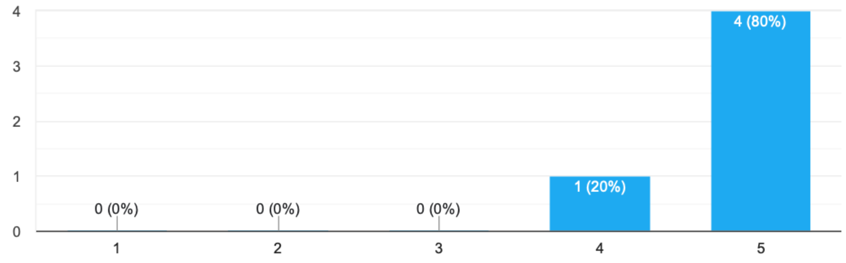
Usability Study Description

The test tool used for this project usability study consisted of 8 questions, either scalable or optional short answer. Each question intended to evaluate some aspect of the manual. Some focus on Don Norman’s design principles like affordances and consistency, while others focus on content and aesthetics. For example, “How easy/difficult was this manual to navigate?” or “How would you describe the visual aesthetic of this manual?” Feedback to these types of questions determines where exactly the project needs editing.

The subjects chosen for the survey were based on knowledge of computer science. Unfortunately, I could not contact any middle school students, so the sample consisted of adult subjects aware of the intended audience. 2 computer science majors, 1 with little programming experience, and 2 with no programming experience; 5 total. I wanted my sample to capture different perspectives. The programmers evaluate the content they are familiar with, while the non-programmers provide a beginner’s point of view.

Results and Recommendations

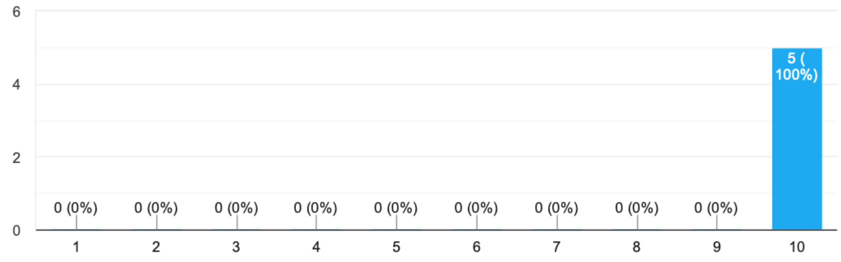
1. How easy/difficult was this manual to navigate?



2. Did you come across any confusing or unclear instructions or explanations in this manual? If yes, where?

- No, everything is clear and thorough
- No

3. On a scale of 1-10, how consistent did you find the modules of this manual? (content, clarity, language, etc.)



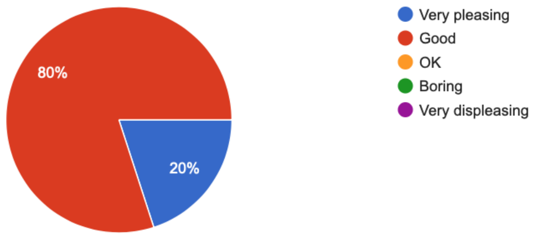
4. Do you think the modules in this manual are ordered appropriately? If not, what would you change?

- I would put the "Where can I use these skills?" section after the rock, paper, scissors part
- I don't know python but the concepts seem to go from least complex to most complex which is good.
- Yes, good as is

5. Would you recommend a different color scheme or font for the manual? If yes, explain.

- Nope, I think it fits well
- I like the way it is now.
- No
- No, good as is

6. How would you describe the visual aesthetic of this manual?



7. Did you find anything unnecessary in the manual that could be cut? If yes, what?

- Nope, nothing
- No

8. Any additional questions/comments?

- Nice website for beginners
- Maybe fix the menu bar stretching into the pages
- I like it, short and simple.

From these data, I looked further into the responses from questions 4, 6, and 8. The others received little to no feedback conveying a need for change; the subjects appear to favor the content and clarity of the module. Question 4 mentions the order of the content. One of the responses suggested switching the last 2 modules, and I agree with them. In a revision, the final project would occupy module 7, and the "CS and Beyond" would act as our concluding module. Question 6 focuses on aesthetics. 4 out of the 5 respondents said the manual was "good" aesthetically rather than "very pleasing" which is the goal. This makes sense since our manual is very simple and without graphics. Given more time, we could add more design and character so the overall manual is more eye-catching. Lastly, question 8 serves as a place for any miscellaneous comments or concerns. One subject mentioned how the menu bar stretches over the modules. Making the menu options a little smaller would be an easy solution to respect this concern.

Appendix

Project Deliverable
GitHub Repository
Usability Test Tool