Learners' Perspectives on Learning Programming from Interactive Computer Tutors in a MOOC

Ruiqi Shen
Department of Informatics
New Jersey Institute of Technology
Newark, New Jersey, USA
rs858@njit.edu

Michael J. Lee
Department of Informatics
New Jersey Institute of Technology
Newark, New Jersey, USA
mjlee@njit.edu

***Abstract*—With the increasing demand for technology workers, more people are now learning to code. Many of these learners are turning to Massive Open Online Courses (MOOCs) due to their low cost and accessibility, especially compared to attending in-person courses. However, little is known about who these users are, and what they think about the instruction provided by these systems. In this study, we conducted a qualitative analysis on 218 responses (stemming from 62 questions) on Quora about a popular MOOC called Codecademy. We found that 1) learners are primarily composed of beginners, 2) Codecademy is good at delivering web development/front end courses, 3) an interactive environment increases engagement, and 4) learners largely criticize courses as being too rigid and impractical. Based on these findings, we discuss the importance of interactive computer tutors as programming instructors and propose design features that can potentially improve learning with MOOCs.**

***Index Terms*—interactive computer tutor, Codecademy, programming education, learners' view, grounded theory, Quora**

I. Introduction

The demand for technology workers continues to rise around the world. Nowadays, many people are becoming self-taught programmers, learning how to code on their own using books and online resources. Supporting the large numbers of self-taught programmers is the increasing availability of Massive Open Online Courses (MOOCs) [1]. As an alternative to physical, in-person courses to learn programming, MOOCs provide a cheaper (often free) and more accessible means to learn programming, while serving a large scale of learners [2].

Since learning programming is largely a learning-by-doing effort [3], and instant feedback is important for learners to make progress [4], some MOOCs employ techniques to support these features to help their users. Farrell et al. were among the first to introduce this kind of system to teach programming, describing a two-component system: a "problem solver" (which can interpret learners' code and provide instant feedback) and an "advisor" (which provides guidance to learners throughout the learning process) [5]. We use this description to define *interactive computer tutors* (ICTs) in this paper. The interactivity of ICTs amplifies the process of learning-by-doing, and thus accelerates the learning [6]. Despite their popularity, most commercial based ICTs such as Codecademy are underexplored in the research literature, especially from the learners' perspectives.

978-1-7281-6901-9/20/$31.00 ©2020 IEEE

Codecademy is a popular MOOC providing free, online, interactive lessons for a variety of programming topics [7]. The inclusion of an interactive learning environment is one of its highlights, where both tutorials and editable coding panels are displayed in one interface. In its system, the window on the left panel serves as the "advisor", which provides textual tutorials, hints and guidance, and both the windows in the middle and on the right serve as the "problem solver", in which learners can type the code and get the instant feedback. Although some studies have investigated the effects of integrating Codecademy into traditional classes [8]–[11], fewer works have examined the perspective of these types of systems from the larger population of self-taught programmers who choose ICTs to learn. For example, despite its large userbase, we know little about the actual users of Codecademy and how they view its instruction. Who are the users? Why do they choose it? What do they think about it? Reporting on these measures has the potential to provide invaluable insights for building effective educational tools for a broad and diverse audience, having implications for online pedagogy, user experience design, and computing education research.

Since we are among the first to probe self-taught learners' specific views about learning coding through Codecademy, we decided to adopt a grounded theory approach to our analysis [12]. This approach will let the data surface important elements during analyses, which will result in the high-level observations that are related to the integrated ICT within Codecademy. After analyzing 218 review, we found that learners perceived it that 1) it is mainly designed for beginners learning a new language, 2) it is good at delivering programming skills that can be visualized (e.g., web development), 3) interactive environment increases engagement, and 4) courses are not practical. The findings suggest that sites using ICTs can be effective programming teaching tools (which can partly address the shortage of human teachers [13]). Our contributions include: 1) among the first to specifically analyze learners' perceptions of an ICT-enabled MOOC; 2) suggestions for researchers studying how users interact with ICT-enabled MOOCs; 3) design recommendations for features that can improve learning experience within ICT-enabled MOOCs.

II. Background

Learning interactively from a computer system has been an educational practice for a long history [14]–[16]. According to Anderson's ACT theory [17], the acquisition of cognitive skills points out a direction for instruction method, in which students are presented with both declarative instruction and a series of guided practices [14]. This theory is a foundation for Cognitive Tutor, a particular type of intelligent tutoring system. In 1984, Farrell et al. used this model to build an interactive computer tutor that teaches LISP programming language [5].

There are some successful examples of using ICTs to teach programming. Because mastering programming skill requires extensive trial and error [3], programming-related ICTs often provide opportunities for users to learn-by-doing. Lee et al. designed a debugging game that effectively engages a wide range of users in learning introductory programming concepts [18]. Brusilovsky et al. applied ICTs in formal classroom settings to teach SQL programming, showing that complementing teachers' lessons with ICTs help to increase learners' motivation [19].

Due to the growing number of programming learners in the world, commercial systems, such as Codecademy, Treehouse, and DataCamp, which teach programming interactively are becoming increasingly popular. Shen et al. interviewed 20 programming learners, and compared the learning experience from both ICTs and human teachers [20]. They found that, experienced learners tend to prefer learning from ICTs and beginners tend to prefer human teachers' guidance. However, there is limited research about these systems. Who are the users, why do they choose to learn from these systems, and how well do these systems achieve their missions? This paper addresses these questions.

III. Method

Since there is limited research examining learners' perspectives of ICTs within MOOCs, we adopted a grounded theory methodology to explore this space [12], following the most recent framework suggested by Tie et al. [21], which comprises four main steps: purposive sampling, data collection, coding, and grounded theory.

We collected data from Quora, a popular question-and-answer website where users post questions and others respond, either factually or in the form of opinions [22]. We chose to use Quora as our data source because: 1) it has over 50,000 users subscribed to the topic "Codecademy," which represented a large number of Codecademy users around the world; 2) most Codecademy-related questions' responders have their real names and background information displayed, and 3) responses are typically authoritative and have high-quality, building on a reputation system among users [23], [24].

The data collection and analysis were conducted simultaneously, following these steps: 1) search using the keyword "Codecademy" on Quora's homepage, 2) select the "Questions" type on the sidebar to filter specifically for questions, 3) sequentially examine each question by its default displayed order (Quora uses a proprietary algorithm to display its results by relevance) and decide whether it fits our inclusion criteria (see below), 3) if it meets our criteria, go to the question page to review the answers, 4) record relevant answers into our data analysis tool called MAXQDA, 5) assign codes to snippets of the answers using the inductive analysis approach [25] (a method that allows findings to emerge from the themes inherent in raw text data [26]). These steps were completed iteratively until we reached data saturation [27], meaning that no more new codes emerged when examining questions and answers. Saturation occurred at 140 answers (57 questions), and to ensure reliability, we continued to analyze 5 more questions with a total of 68 more answers.

One researcher did the initial data collection and analysis, and another researcher verified the data and codes. Our inclusion criteria required a question to be learner-focused (e.g. "Is Codecademy an effective way to learn how to program?") rather than company/application-focused (e.g. "How will Codecademy monetize?"). Another selection criterion is that the answer should be informational enough to extract codes. For example, a response such as "*Codecademy is good.*" would not be selected, because words such as "good" and "bad" was too general and did not point to a specific feature or audience. In contrast, a statement such as "*Codecademy is good for beginners.*" would be selected because it gives a specific reason to why Codecademy was regarded as good.

IV. Results

In total, we analyzed 62 questions and 218 answers. Through inductive analysis, we consolidated this data down to 3 themes and 13 codes. Between two researchers, the intercoder reliability came out to 0.89, and the intercoder agreement to 0.93 [28]. In the following subsections, we will describe our results using representative excerpts from our Quora answers about Codecademy. Although we had access to the Quora users' real names and additional background information through their online profiles, we intentionally refer to them as S1-S218 throughout this paper to give them a degree of anonymity.

A. Who Are the Users?

We identified three types of Codecademy users: 1) self-taught learners, which include beginners (101/218) and intermediate learners (19/218), 2) teachers who use it as a supplementary tool to teach students (3/218), and 3) companies who use it to train staff (1/218).

1) For Beginners: We found that most users view Codecademy as an ideal starting point to learn programming. Even experienced learners who expressed major criticisms about Codecademy tended to admit that it does a good job in teaching programming to beginners. S117, a web developer for many years, wrote: "*I do recommend the [Codecademy] tutorials for complete beginners. Though a few things may be misleading, the tutorials provide the best information (I have come across) for complete beginners to learn from. As you advance, you'll inevitably find other sources more useful. But initially they provide a pretty good foundation for web*

development newbies." Another example is S6, a programming beginner when he started to use Codecademy 5 years prior, who stated: "*As an English teacher who had no technical background at all, I really enjoyed learning and writing code on that website. It was this website that helped me step into the world of code.*"

2) Experienced learners: We also found that some experienced learners use Codecademy as a tool to refresh their knowledge or learn the basics of a new language. For example, S65 wrote: "*Codeacademy tends to work well for experienced programmers who want to brush up on some syntax or want to learn a new language.*" Although this review seems more an opinion than an experience, we have S30, who had been programming for over 30 years, shared his personal experience: "*From the personal perspective, every now and then I go to those sites when I'm curious about a language I don't know. [...] I find sometimes that's the fastest way to learn a new skill. It doesn't matter how long you've been programming, you will always be a newbie at something.*"

3) Others: Besides self-taught learners, there are other users who use Codecademy as a training tool, such as teachers and companies. Although limited comments were found, we believe it is noteworthy to mention because it suggests a broader usage of ICTs. For example, S87 was a middle or high school teacher—he used Codecademy to encourage interest in coding for his students. He commented: "*Codecademy feels more like an interactive game that you experiment with [...] at our school, we used Codecademy in our intro course to get kids' feet wet with the whole process of programming.*" Somewhat similarly, companies use it to train their staff. S90 thought that Codecademy is good for beginners that have some coding experience. Tools like it can be effective for training teams.

B. Why do they use it to learn?

Learners report that they find Codecademy helpful because it delivers basic content (93/218) in a structured (25/218) way, which allows them to pick up the skill(s) quickly and easily (24/218). Moreover, the interactive learning environment (35/218) arouses learners' interests and increases engagement (33/218). We also encountered many discussions about learning a specific language or skill from it, such as web development skills (61/218) and Python (17/218).

1) Learning basic content in a structured way: Providing content in a structured way appears to be important to attract users, especially beginners. For them, it is often difficult to determine what to learn, and to identify the essential parts of the material. Therefore, it is the educators' (or tools') job to carefully provide scaffolding highlighting these points [29]. This structure allows learners to build up their knowledge incrementally. Based on the learners' answers to others' questions, Codecademy does a good job of providing this structured content, especially compared to searching for scattered information online. For example, S114 said: "*I found it to be slightly more structured in terms of progressing from lesson to lesson, and Codecademy had some repetition and slight jumps in difficulty level between some exercises.*"

2) Learning quickly and easily: While this structured content is particularly useful to beginners, learning basic content quickly and easily is also helpful for experienced learners. Codecademy breaks down the materials into smaller chunks followed with exercises, which are easier to understand and follow than a larger amount of information at once. S93, a beginner, wrote: "*The course, say, JavaScript, is well designed in terms of the difficulty, how easy for a beginner to learn and apply.*" Similarly, S139, an experienced programmer (who was very critical of Codecademy based on his response), mentioned that he used Codecademy often to learn new languages: "*I use Codecademy A LOT. The last time I used it was for PHP. [...] I sat down and finished the course in about an hour and a half. This made it much easier for me to jump right in and read the code I was working on.*"

3) Interactive environment increases engagement: Many users mentioned that Codecademy keeps them engaged with the content, attributed not only to the structured and easy-to-follow content, but also because of the interactive learning environment. S50, an intermediate programmer, wrote: "*Interactivity makes the entire learning process quite engaging and, for many people, less boring than just copying code from a book/tutorial. It can also provide great help in "breaking the wall" that generally scares people who are approaching programming.*" In addition, gamified feature is another source of engagement, as S180 commented: "*The tutorials are made in a way that is interactive, conversational and has some gaming elements that keep you moving forward. You will appreciate the idea that they are really trying to help people who wants to learn to code but too afraid to take first step to do it.*" Similarly, S15 said:"*The best part about Codecademy, which makes it superior to other types of learning, is the interactive way in which they teach code. You are forced to write the code as you go along the lessons rather than sitting on your couch and passively reading a fat, old book.*"

4) Learning a specific language: There were many positive comments about using Codecademy to learn web development skills, mostly about HTML, CSS, PHP, and JavaScript. Learners particularly liked that users were able to quickly see the output of their code through an interactive window. S110 articulates this in his comment: "*For early-stage learning, I preferred Codecademy for the following reasons: 1) Really quick feedback loops - I knew whether I was coding right/wrong almost immediately because the coding and preview screen were built in. 2) And I thought they should have started with HTML5 and CSS as the first part on 'Intro to Programming' as it's more tangible.*"

C. Complaints about Codecademy

We also surfaced complaints about Codecademy. The codes we extracted are: not practical (49/218), and too rigid (15/218).

1) Not practical: The analyses of Quora questions and answers revealed that there are some learners (22.5% of respondents) who are confused after they finished Codecademy

courses, because they did not know how to apply their skills to actual practice/work. These learners believed that this was primarily due to the lack of work on practical projects during the learning process, and the content concentrating on aspects such as syntax, rather than on problem solving skills. S17 said: "*It [Codecademy] is a good place to learn basic syntax. But, if you really want to be a good programmer, you need to solve real world programming problems and gain through that experience.*" S70 commented: "*I felt there was too much emphasis on syntax and mechanics and not quite enough focus on learning how to think through problems.*" Finally, learners such as S163 said that Codecademy was too helpful: "*It [Codecademy] is, in my opinion, too helpful, so people don't learn how to debug/answer their own questions, which is a critical skill to be a good professional.*"

2) Too rigid: Codecademy offers structured tutorials where learners have to follow a strict, predefined path. While some learners found it helpful as we mentioned above, other learners thought that this rigidity was unhelpful, citing that some content was repetitive and boring. S23 commented: "*In my opinion it [Codecademy] is too boring and relaxed. Just repeating language constructs after a template won't help you learn a language or technology since you will forget everything you've learned that way in a week.*"

V. Discussion

To summarize our findings, we found that: 1) learners are primarily composed of beginners, but also include experienced programmers, 2) Codecademy is good at delivering web development/front end courses, 3) an interactive environment increases engagement, and 4) learners largely criticize courses as not being practical. Because Codecademy is a popular, well known MOOC [30], these findings may be useful to educators, researchers, and even regular users about using/improving interactive learning environments and ICTs. In this section, we discuss our findings, and share our thoughts on future research directions and design features that can improve the ICT learning experience.

Although we found that beginners are the largest group using Codecademy, intermediate programmers also use it for learning new skills. Beginners often do not know where to start of what they have to learn. They need some kind of scaffolding to help them succeed [29]. However, experienced learners have higher learner autonomy. While beginners may enjoy a highly structured curriculum, experienced learners may prefer more freedom to pick up whatever they want to learn quickly. We saw this reflected in the complaints learners had about Codecademy. For example, they pointed out that learning from Codecademy might not be practical and also too rigid. Perhaps these MOOCs and ICT systems can detect (or simply ask) first-time users about their current skill level, experience, or preferences, and customize the learning experience to attract more users and keep them engaged [31]. Also, giving learners more experience with practical skills, such as working on realistic projects and more problem solving skills, such as debugging [8], [32], can further improve learners' perceptions of these types of MOOCs and ICTs. Further experiments can be done to explore what specific design features will benefit different types of learners.

VI. Limitations and Future Work

Our goal for this study was to surface patterns about a specific educational tool (i.e., Codecademy) from a group of known users (i.e., from Quora) qualitatively [33]. We did this manually by iteratively looking through responses about Codecademy-related questions, but future work may benefit from further analyses through automated or semi-automated means such as text analyses using natural language processing.

Although we had two researchers analyzed and verified the codes with a high degree of agreement, there may have been ambiguities or misinterpretations given that our core data were one-time, asynchronous posts/responses to a stranger's online question. For example, the categorization into "beginners" and "experienced learners" were based on keywords. Some of the keywords and phrases we used to identify beginners included "beginner," "novice," and "first start[ing] to learn." Keywords and phrases for experienced learners included "experienced programmer" and "been programming for years." However, we did our best to be systematic and consistent with our classifications. In future work, we can recruit Codecademy (or similar MOOCs with ICTs) users for semi-structured interviews or focus groups, where we can ask questions and follow-up questions based on the findings from this study as a starting point.

Our data represented eight years of questions and answers. Even though Quora lists questions by relevance to the search query, there may have been some questions or answers that are obsolete. In addition, although our questions and answers came from actual Codecademy users, the data was collected outside the context of Codecademy itself. This may have introduced some bias into our sample, limiting the generalizability of our findings, as the types of people who are open to asking and answering questions on an open online forum might not be representative of the larger population of Codecademy users. To address these issues in future studies, we could send out an online questionnaire to current users of Codecademy. The results from this study can serve as a good source of information for the questionnaire design.

VII. Conclusion

In this paper, we explored users' perceptions of online interactive computer tutors. We did so by using a grounded theory approach to systematically analyze questions and answers from Codecademy users from Quora. We collected 218 answers (stemming from 62 questions), and extracted 3 themes and 13 codes from them. Based on these findings, we discussed the importance of interactive computer tutors as programming instructors and proposed potential follow-up studies and design features that can potentially improve the MOOC learning experience using interactive computer tutors.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation (NSF) under grants DRL-1837489 and IIS-1657160. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] L. Pappano, "The year of the mooc," *The New York Times*, vol. 2, no. 12, p. 2012, 2012.

[2] T. Tang, S. Rixner, and J. Warren, "An environment for learning interactive programming," in *Proceedings of the 45th ACM technical symposium on Computer science education*, 2014, pp. 671–676.

[3] C. Alario-Hoyos, C. D. Kloos, I. Estévez-Ayres, C. Fernández-Panadero, J. Blasco, S. Pastrana, and J. Villena-Román, "Interactive activities: the key to learning programming with moocs," *Proceedings of the European Stakeholder Summit on Experiences and Best Practices in and Around MOOCs, EMOOCS*, vol. 319, 2016.

[4] H. Keuning, J. Jeuring, and B. Heeren, "Towards a systematic review of automated feedback generation for programming exercises," in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 2016, pp. 41–46.

[5] R. G. Farrell, J. R. Anderson, and B. J. Reiser, "An interactive computer-based tutor for lisp." in *AAAI*, 1984, pp. 106–109.

[6] N. Tillmann, J. De Halleux, T. Xie, S. Gulwani, and J. Bishop, "Teaching and learning programming and software engineering via interactive gaming," in *2013 35th International Conference on Software Engineering (ICSE)*. IEEE, 2013, pp. 1117–1126.

[7] Z. Sims and C. Bubinski, "Codecademy," *URL= http://www. codecademy. com*, 2011.

[8] M. J. Lee and A. J. Ko, "Comparing the effectiveness of online learning approaches on cs1 learning outcomes," in *Proceedings of the eleventh annual international conference on international computing education research*, 2015, pp. 237–246.

[9] R. B. Figueroa Jr and E. M. Amoloza, "Addressing programming anxiety among non-computer science distance learners: A upou case study," *International Journal*, vol. 9, no. 1, pp. 56–67, 2015.

[10] J. H. Sharp, "Using codecademy interactive lessons as an instructional supplement in a python programming course," *Information Systems Education Journal*, vol. 17, no. 3, p. 20, 2019.

[11] M. Olsson and P. Mozelius, "On design of online learning environments for programming education," in *Academic Conferences and Publishing International ECEL*, 2016, pp. 12–24.

[12] A. Strauss and J. M. Corbin, *Grounded theory in practice*. Sage, 1997.

[13] M. Guzdial, "Limitations of moocs for computing education-addressing our needs: Moocs and technology to advance learning and learning research (ubiquity symposium)," *Ubiquity*, vol. 2014, no. July, pp. 1–9, 2014.

[14] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier, "Cognitive tutors: Lessons learned," *The journal of the learning sciences*, vol. 4, no. 2, pp. 167–207, 1995.

[15] V. Aleven, E. Stahl, S. Schworm, F. Fischer, and R. Wallace, "Help seeking and help design in interactive learning environments," *Review of educational research*, vol. 73, no. 3, pp. 277–320, 2003.

[16] T.-H. Tan and T.-Y. Liu, "The mobile-based interactive learning environment (mobile) and a case study for assisting elementary school english learning," in *IEEE International Conference on Advanced Learning Technologies, 2004. Proceedings.* IEEE, 2004, pp. 530–534.

[17] J. R. Anderson, "The architecture of cognition," 1983.

[18] M. J. Lee, F. Bahmani, I. Kwan, J. LaFerte, P. Charters, A. Horvath, F. Luor, J. Cao, C. Law, M. Beswetherick *et al.*, "Principles of a debugging-first puzzle game for computing education," in *2014 IEEE symposium on visual languages and human-centric computing (VL/HCC)*. IEEE, 2014, pp. 57–64.

[19] P. Brusilovsky, S. Sosnovsky, M. V. Yudelson, D. H. Lee, V. Zadorozhny, and X. Zhou, "Learning sql programming with interactive tools: From integration to personalization," *ACM Transactions on Computing Education (TOCE)*, vol. 9, no. 4, pp. 1–15, 2010.

[20] R. Shen, D. Y. Wohn, and M. J. Lee, "Comparison of learning programming between interactive computer tutors and human teachers," in *Proceedings of the ACM Conference on Global Computing Education*, 2019, pp. 2–8.

[21] Y. Chun Tie, M. Birks, and K. Francis, "Grounded theory research: A design framework for novice researchers," *SAGE open medicine*, vol. 7, p. 2050312118822927, 2019.

[22] S. Ovadia, "Quora. com: another place for users to ask questions," *Behavioral & Social Sciences Librarian*, vol. 30, no. 3, pp. 176–180, 2011.

[23] G. Wang, K. Gill, M. Mohanlal, H. Zheng, and B. Y. Zhao, "Wisdom in the social crowd: an analysis of quora," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 1341–1352.

[24] S. A. Paul, L. Hong, and E. H. Chi, "Who is authoritative? understanding reputation mechanisms in quora," *arXiv preprint arXiv:1204.3724*, 2012.

[25] J. Corbin and A. Strauss, *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage publications, 2014.

[26] D. R. Thomas, "A general inductive approach for qualitative data analysis," 2003.

[27] B. Saunders, J. Sim, T. Kingstone, S. Baker, J. Waterfield, B. Bartlam, H. Burroughs, and C. Jinks, "Saturation in qualitative research: exploring its conceptualization and operationalization," *Quality & quantity*, vol. 52, no. 4, pp. 1893–1907, 2018.

[28] J. L. Campbell, C. Quincy, J. Osserman, and O. K. Pedersen, "Coding in-depth semistructured interviews: Problems of unitization and intercoder reliability and agreement," *Sociological Methods & Research*, vol. 42, no. 3, pp. 294–320, 2013.

[29] M.-J. Laakso, T. Rajala, E. Kaila, and T. Salakoski, *Novice Learning*. Boston, MA: Springer US, 2012, pp. 2482–2483. [Online]. Available: https://doi.org/10.1007/978-1-4419-1428-6%5F1520

[30] J. B. Bacon, "When community becomes your competitive advantage," Jan. 2020. [Online]. Available: https://hbr.org/2020/01/when-community-becomes-your-competitive-advantage

[31] M. J. Lee and B. Ferwerda, "Personalizing online educational tools," in *Proceedings of the 2017 ACM Workshop on Theory-Informed User Modeling for Tailoring and Personalizing Interfaces*, 2017, pp. 27–30.

[32] R. Chmiel and M. C. Loui, "Debugging: from novice to expert," *ACM SIGCSE Bulletin*, vol. 36, no. 1, pp. 17–21, 2004.

[33] P. Ochieng, "An analysis of the strengths and limitation of qualitative and quantitative research paradigms," *Problems of Education in the 21st Century*, vol. 13, p. 13, 2009.