International Society of
the Learning Sciences

Supporting remote programming instruction with real-time collaboration and awareness tools

Stephanie Yang, Harvard Graduate School of Education, szhang@g.harvard.edu
Allan Jiang, Stanford University, jiangts@stanford.edu
Bertrand Schneider, Harvard Graduate School of Education, bertrand_schneider@g.harvard.edu

Abstract: Learning to program remotely can be difficult for novices due to a lack of adequate online communication and coding tools. The current paper empirically assessed the impact of real-time collaboration and awareness tools on improving remote programming learning and instruction. Novice programmers were randomly assigned to use either screenshare, a shared online programming notebook (with real-time collaboration), or a shared notebook with awareness tools (shared gaze, mouse and scroll position) with an instructor. Findings revealed that students learned better in the condition where they had access to both real-time collaboration and awareness tools. A discussion of the results reveals the differential benefits of shared gaze and shared cursor for instructors and students and explores the efficacy of sharing web-based eye-tracking data to assist programming instruction. A notable contribution of this work is an open-source chrome extension that enables real-time sharing of awareness tools for online collaborative programming notebooks.

Introduction

A rapid increase in online programming instruction has forced many CS educators to seek out novel tools for remote pedagogy. However, existing technologies for programming instruction, such as screen sharing, github and google docs are not adequately meeting student needs, particularly for novice programmers (Ying & Boyer, 2020). While a number of tools, such as real-time collaboration online notebooks and shared awareness visualizations, offer promise in improving learning, they have not been researched in instructional contexts for novice programmers. In the current study, novice programmers and instructors were either assigned to work through a remote lesson using screen share, a shared notebook (with real-time collaboration tools), or a shared notebook with a chrome extension that enabled awareness tools (shared gaze, shared mouse and shared scroll position). The primary goal of this research is to assess the impact of real-time collaboration tools and awareness tools in improving remote programming instruction, and the differential benefits of shared gaze and shared cursor for instructors and students.

Related Work

Instructors often utilize screen share to help students with their code in a remote setting (Kawash et al., 2021). Screen sharing affords many advantages for programming instruction. For example, when students are sharing their code, instructors can see what they are seeing, which provides a shared reference for discussion and can provide insights into what the student is thinking. However, instructors do not have proper tools to direct student attention or provide real-time code-input (Kawash et al., 2021). During learning, these communication barriers can overburden a students' cognitive load, which is already high for novice programmers.

Real-time collaboration tools for programming, which allow multiple users to edit the same code synchronously, can alleviate some of these issues. Instructors can enter a shared coding environment with a student, and both parties can contribute real-time code input. Several studies have shown the benefits of real-time collaboration tools during pair programming to facilitate peer debugging (Goldman et al., 2011). However, given the novelty of mainstream synchronous programming tools, to our knowledge, they have not been studied in the context of coding instruction and in relation to student learning gains.

While synchronous programming tools allow both instructor and student to work collaboratively on a piece of code remotely, they still lack many advantages afforded during in-person learning. For example, when an instructor and student are working side-by-side, the instructor can point with their finger to direct attention and read the student's nonverbal cues, such as their eye gaze and facial expressions. These can help the instructor infer the students' thought process and tailor their teaching accordingly (D'Angelo & Schneider, 2021). The ability to point at a location on the screen or to see the other person's eye gaze or scroll position is not afforded in traditional real-time collaboration tools for programming. However, recent advances in low-latency interfaces and eye tracking software can enable transmission of nonverbal cues in a remote setting. These are known as awareness tools. Sung et al., (2021) found that when instructors could remotely see a student's eye gaze, students learned more, and instructors were better able to predict their understanding. In a review of shared gaze technology, D'Angelo & Schneider (2021) explain that shared gaze enables joint-visual attention, which provides a common ground for collaborators to ensure they are on the same page. However, it is yet unclear whether other awareness tools, such as a shared cursor, can better convey these cues. Additionally, these previously mentioned studies have only assessed the effect of instructors visualizing student eye gaze. Given that student-directed awareness tools can be beneficial for learning (Sharma et al., 2016), the current study seeks to differentiate the benefits of awareness tools for students and instructors during a two-way sharing session. Lastly, to our knowledge, prior studies leveraging shared gaze have all used high-cost, external eye trackers, such as a Tobii Pro. These trackers are often expensive and not widely accessible to instructors and students in a real-world context. In the current study, shared gaze was enabled through webcam-based eye tracking, which is more low-cost and accessible. Understanding the benefits of sharing gaze data from a webcam is a first step in democratizing this technology for education.

Methods

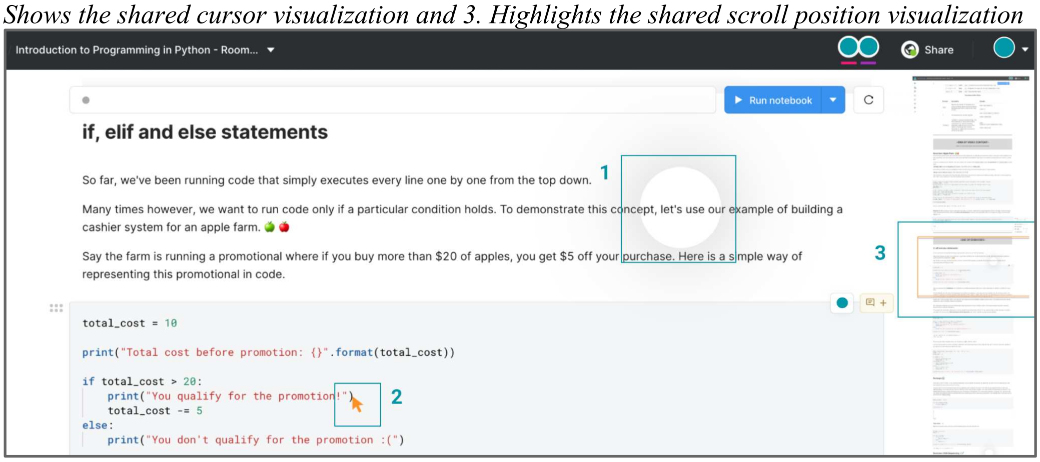
Students in the study were 39 English speaking adults between the ages of 19 and 43, who self-reported no prior programming experience. A total of 13 instructors taught the sessions and were recruited through a mailing list sent to teaching fellows for an introductory computer science course and through word of mouth. Each instructor participated in three one-on-one sessions with a student and experienced each condition once. Each one-on-one session was 90 minutes long and took place remotely on Zoom. Together, students and instructors worked through learning content in a Deepnote notebook (deepnote.com). Deepnote enables users to run iPython notebooks in a browser and supports real-time collaboration. The content of the notebook contained exercises and examples for the instructors to teach the learning content. Students were randomly assigned to one of three experimental conditions: Screenshare, Shared notebook and Shared notebook with Awareness Tools.

Based on the interviews with instructors and evidence from other studies (Ying & Boyer, 2020) screenshare is a widely prevalent tool for conducting remote programming instruction. Thus, this was chosen to be the control condition, and a comparison against more advanced instructional tools. In the **Screenshare condition**, instructors and students were given different instances of a notebook containing the same content. The session began with the student sharing their screen and the instructor looking on. The instructor and student were free to switch off screen sharing at any time as they felt convenient. The impact of real-time collaboration tools was assessed in a **Shared Notebook condition** where instructors and students worked on a shared notebook that updated edits and run results in real-time. Both instructors and students were only allowed to reference the shared notebook on their screen. In the **Shared notebook with Awareness Tools** condition, students and instructors worked on a shared notebook, like the previous condition. In addition, they were able to see each other's gaze, mouse and scroll position on the page (Figure 1). The other person's eye gaze was represented by a circle (50px radius), the other person's mouse position was indicated by an orange cursor, and the scroll position was represented by an orange rectangle on a mini-map representation of the notebook on the side of the page. These awareness tools were enabled by a chrome extension (github.com/jiangts/GazeCoder) that users enabled at the beginning of the session. The real-time gaze, cursor, and scroll position was broadcasted to the other user through our websockets server. We used the GazeCloudAPI (gazerecorder.com) for eye tracking, which uses computer vision to determine the position of a user's gaze on the screen.

At the beginning of each session, students watched a 10-minute video sourced from Udacity (udacity.com) which introduced them to the basic building blocks of python. After, they took a pre-test consisting of 6 questions to control for prior knowledge. Students were then introduced to their instructor, who was also in the Zoom room, and both were sent a link to the Deepnote notebook. In the conditions with awareness tools, instructors and students set up the chrome extension and calibrated their eye gaze. After, the students and instructors were given 45 minutes to complete as much of the notebook as possible. Instructors were encouraged to cover all the new content, even if they couldn't complete all the exercises in the notebook. At the end of the 45 minutes, students and instructors completed a survey about their experience, and students then took a 12-question learning post-test that closely mirrored the content of the pre-test, with an additional question for each original pre-test concept. This completed the students' participation. After instructors completed their three student sessions, they were interviewed about their experience teaching in each condition.

Student learning gains were measured using the pre- post-test taken before and after the learning session. The assessments contained multiple choice, short answer and coding questions, and assessed understanding of types, variables, if statements and for loops. Learning gains were calculated by subtracting the pre-test score from the post-test score. Additionally, the usefulness of the shared gaze and cursor tools were assessed based on responses from instructors' and students' post-experiment survey.

Figure 1
Screenshot of Shared Notebook with Awareness tools in Deepnote. 1. The shared gaze visualization, 2. Shows the shared cursor visualization and 3. Highlights the shared scroll position visualization



Results

A one-way ANCOVA was performed to compare the effect of the three different conditions on learning gains. The results revealed that there was a statistically significant difference in learning gains between at least two conditions ($F(2, 36) = 9.91$, $p < .001$). A post-hoc test found that learning gains were significantly different between the Shared Notebook with Awareness tools condition and Screenshare condition ($p = .002$, 95% C.I. = (1.87, 7.42)) and the Shared Notebook with Awareness tools and Shared Notebook condition ($p = .02$, 95% C.I. = (.61, 6.22)). There was no statistically significant difference in learning gains between the Shared Notebook and Screenshare condition ($p = .376$). In these analyses, we controlled for the age of participants based on prior evidence of a relationship between technological learning and age (Charness & Boot, 2009). Additionally, in our data, age was negatively and significantly correlated with learning gains ($r(39) = -.54$, $p < .001$).

In the post-experiment survey, students and instructors were asked to rate the shared gaze and shared cursor tools on a 5-point Likert scale in terms of the tool's usefulness for their learning/teaching. On average, students rated the shared gaze as "Not at all useful" (M=1.2, SD=0.4), and the shared cursor as "Moderately useful" to "Very useful" (M=3.4, SD=1.4) for their learning. Instructors rated seeing the students' gaze as "Moderately useful" (M=2.9, SD=1.4) on average, and the shared cursor tool as "Moderately useful" to "Very useful" (M=3.5, SD=1.2). In general, the instructors found the shared gaze to be more useful than the students, and both instructors and students thought the shared cursor was useful for their learning/teaching.

Discussion

In a one-on-one tutoring setting, students working in a shared notebook with an instructor, who could also see their eye gaze, mouse cursor and scroll position learned more than students who could only see the instructor's shared screen. No difference in learning gains was found between a student working with an instructor in a shared notebook and a student and instructor screensharing. These results suggest that real-time collaboration tools are not sufficient in supporting learning, and that the addition of awareness tools are necessary for significant learning gains. Preliminary analysis of instructor interviews uncovered potential reasons for these findings. When instructors only had access to real-time collaboration tools (Shared Notebook condition), they recalled that they could easily write out code for students and took advantage of this when making small syntax changes or writing example code to further explain a concept. However, instructors also lost affordances offered in the screenshare condition. For example, they could not see where the student was scrolled to or where their cursor position was. Instructor's recalled feeling "lost" or "blind" in this condition when the student was silent since they could not tell if they were thinking, zoning out, or stuck. In comparison, when instructor's had access to both awareness and real-time collaboration tools, the awareness tools compensated for cues afforded during screensharing, providing both grounding and feedback.

Generally, instructors rated the awareness tools as more useful than the students did. Notably, the instructors found the shared gaze tool to be "Moderately useful" whereas students thought it was "Not at all useful". One instructor reflected that the shared gaze conveyed "thinking", whereas the shared cursor conveyed "intent". Indeed, during a learning situation, it is more useful for an instructor to receive cues about what the student is thinking or confused about so they can tailor their instruction. On the other hand, it is less useful for a student to know what the instructor is thinking, but more useful to know their intent, and what they want the student to pay attention to. This could explain why instructors found the shared gaze to be more useful than students, and why students thought the shared cursor was more useful than the shared gaze.

A few limitations caveat the findings of this research. First, the small sample size may hinder the reliability of these findings. Additionally, since a less accurate web-based eye-tracker was utilized in the study, the results may not be generalizable to other instructor-student gaze-sharing scenarios.

Conclusion

In this paper, we assessed the impact of real-time collaboration and awareness tools on improving remote programming instruction. The results show that synchronous code editing tools can benefit remote programming instruction, but only when supplemented with awareness tools that aid the communication and grounding between students and instructors. Given the rising number of commercial synchronous code editors, such as Deepnote and Replit (replit.com), and the open-source nature of the chrome extension developed in the study, these findings have real-world applications to many CS learning settings, such as office hours or tutoring.

References

Charness, N., & Boot, W. R. (2009). Aging and information technology use: Potential and barriers. *Current Directions in Psychological Scienc*e, 18(5), 253-258.

D'Angelo, S., & Schneider, B. (2021). Shared Gaze Visualizations in Collaborative Interactions: Past, Present and Future. *Interacting with Computers*.

Goldman, M., Little, G., & Miller, R. C. (2011, October). Real-time collaborative coding in a web IDE. *In Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 155-164).

Kawash, J., Horacsek, J., & Wong, N. (2021). What We Learned from the Abrupt Switch to Online Teaching Due to the COVID-19 Pandemic in a Post-secondary Computer Science Program. In *CSEDU* (2) (pp. 148-155).

Sharma, K., Alavi, H. S., Jermann, P., & Dillenbourg, P. (2016, April). A gaze-based learning analytics model: in-video visual feedback to improve learner's attention in MOOCs. *In Proceedings of the sixth international conference on learning analytics & knowledge* (pp. 417-421).

Sung, G., Feng, T., & Schneider, B. (2021). Learners Learn More and Instructors Track Better with Real-time Gaze Sharing. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1), 1-23.

Ying, K. M., & Boyer, K. E. (2020, April). Understanding Students' Needs for Better Collaborative Coding Tools. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (pp. 1-8).

