Cocode: Co-learner Screen Sharing for Social Translucence in Online Programming Courses

Jeongmin Byun
KAIST
Daejeon, South Korea
jmbyun@kaist.ac.kr

Alice Oh
KAIST
Daejeon, South Korea
alice.oh@kaist.edu

Jungkook Park
KAIST
Daejeon, South Korea
pjknkda@kaist.ac.kr

CHI '20 Extended Abstracts, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6819-3/20/04.
DOI: https://doi.org/10.1145/3334480.3383154

Abstract
Online courses are popular among learners of programming, but many learners have trouble completing the courses. A common approach to increase learner engagement is to provide co-learner presence via chat and forums. In this work, we present Cocode, an online learning system where learners can share their presence without any explicit action; their normal learning activities would signal co-learner presence. Cocode is a web application for online programming courses that shows other learners' code editors and running screens in the programming environment to the learners while working on exercises. Results from our between-subject studies show that learners with Cocode are more engaged and work on more programming exercises compared to the learners using the system without social features.

Author Keywords
Education; programming course; online course; social translucence

CCS Concepts
•Social and professional topics → Computing education; Computer science education;

Figure 1: Co-learners shown in the social box of Cocode. Each of the small boxes in the social box shows either (A) a live code editor of a co-learner when the learner edits the code or (B) a live running screen when the code is running.

Introduction

Online programming education provides better access to affordable, convenient, and time-saving ways to learn to program. However, like all other online courses, online programming courses have very low completion rates, as low as 5-10%, according to previous research [1].

To overcome this, many online programming courses feature multiple hands-on programming exercises in web-based programming environments. These environments allow learners to write and run code on any web browser and get immediate feedback on their code from automated grading scripts. With high interactivity, we can expect improved engagement and learning gains from the learners [5, 3].

However, one major difference remains between online and offline classes, and that is the constant visibility of other learners in an offline classroom. By physically being near one another, learners can perceive their co-learners' learning behaviors without explicit actions such as asking a question. Previous studies show that providing learners *mediated illusion* of co-learners can engage them without direct interactions among the learners [4]. Therefore we designed a system, Cocode, that makes co-learners visible in the online programming classes. In online programming courses centered around hands-on programming exercises, learner behaviors are unobtrusively visible to the system. Cocode simply visualizes this information to other learners and thus does not require learners to take additional explicit actions.

Figure 2 shows an overview of a programming exercise page in Cocode. Learners can write and run code on a web browser. At the same time, learners can see their co-learners' screen in the social box (A) and communicate with them using the chat box (B). Figure 1 also shows the social box of Cocode.

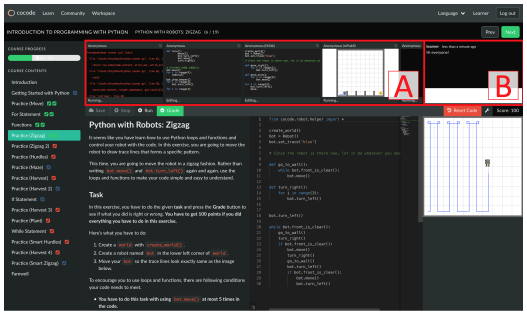

Figure 2: An overview of Cocode. On the top of usual tools for programming exercise, there are (A) social box that shows code editors and running screens from co-learners and (B) chat box provides co-learner presence to the learners.

For evaluation, we compared Cocode to an online programming environment without the aforementioned social features, and we have observed that learners try to solve more programming exercises in the course.

The Learning System

Cocode is a standalone web application that hosts online programming courses with hands-on exercise problems and supports learners to write and run code on a web browser. Figure 2 shows an overview of a programming exercise page in Cocode. A sample course hosted with Cocode is available at http://bit.ly/cocode-chi-2020.

Like Codecademy or Khan Academy, Cocode allows learners to walk through programming exercises with this user interface. However, unlike other online programming classes, Cocode shares learner's behaviors with other learners.

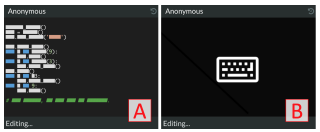

Figure 3: Co-learners shown in the social box of Cocode with the code editor social opacity set to (A) *"high"* or (B) *"opaque"*. With "high" setting, the learner can see other learners' code editors, but the code is illegible. With "opaque" setting, the learner can see other learners' state description with an icon only.

Implementation

Cocode is a standalone web application based on Python Django framework and Vue.js. Unlike other tools for web-based programming courses, Cocode allows users to write and run Python code without any support from the server by utilizing Brython that live-transpile Python code to JavaScript code and to run them on the client-side web browser. In this way, we could provide the co-learner screen sharing feature without putting additional loads to our servers.

In the social box, we introduce the concept of social translucence [2] by showing co-learners' code editors and running screens on top of typical tools for web-based programming education. When co-learners are editing the program code, their code editors are shown in the social box. Learners are not allowed to scroll the co-learners' code editors, so only about 20 lines of the code around the cursor from each learner is visible in the social box. The code editor turns into a running screen when that learner runs the code. Next to the social box, there is a chat box where learners can talk with the co-learners. With these features, we tried to replicate the environment of the offline programming class in the online class.

Code Editor Social Opacity Control

One concern with providing social translucence is that when learners are allowed to see the code of other learners, they can simply copy their code, which may reduce the learning gain of the learners. To mitigate this potential issue, Cocode provides a feature to increase the social opacity of the code editor of other learners. The offline classrooms inspired this feature, where learners can see others' screens but not necessarily be able to read the code on the screens because of the physical distance. In *high opacity* setting, Cocode hides the actual characters in the co-learners' code editors. We expect this feature to prevent the learners from copying other learners' code while providing co-learner presence. And when it is configured to *opaque*, Cocode shows co-learners current state description with an icon only. For example, "editing" or "running". Figure 3 shows the examples.

Asynchronous Co-learner Presence

Cocode shows co-learner screens on the social box in the user interface while learners work on hands-on programming exercises in online classes. However, if there are only a few learners online at a time, social features will not be as useful as when there are many co-learners.

To provide enough co-learner presence regardless of the number of online learners at the time, we save behavior logs from the learners on the server and replay them (like a video clip) for online learners. This happens when there are only a few learners online so that there are always at least a certain number of co-learners visible in the social box.

Evalution

We conducted a between-subject study to see if the group of learners using Cocode are more engaged to take the course compared to the control group. We collected 102 non-rewarded participants to take our course using either one of the systems. They are all volunteers who visited our website from a blog post that introduces websites for *coding practice*, and they all answered to our pre-survey before the study. All participants who answered to be *experienced* or *very experienced* in programming, or have more than two years of programming experience are filtered out from the study. We randomly assigned 51 learners each to the groups.

We calculated the averaged number of programming exercises a learner participated in. We considered a learner participated in an exercise when the learner edited and saved the code in the exercise at least once.

From the result, we observed that learners with Cocode participate significantly more in both optional and non-optional exercises compared to the learners in the control group. For non-optional exercises, the experiment group participated in **4.14** (SD=4.31) out of 11 exercises on average while the control group participated in 2.16 (SD=3.25) ($p \approx .01$ from a standard independent 2 sample T-test). For optional exercises, the experiment group participated

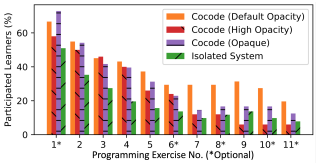

Figure 4: Percentage of learners participated in each of the exercises. More learners from Cocode default group is participating compared to the learners without Cocode in all exercises, and in most of the cases, the participation rate of the group with Cocode in high opacity or opaque settings are between the rate of Cocode default group and the group without Cocode.

Course Materials

The course used in our study consists of 17 hands-on programming exercises borrowed from the "Introduction to Programming" course at KAIST.
Exercises cover the basic programming concepts such as iteration, functions, and conditional statements. Learners use a specially designed library that controls a robot in a two-dimensional world on the screen.

	Default	High	Opaque	Isolated
Participated (Non-optional)	**4.14**	2.86	3.40	2.16
Participated (Optional)	**2.31**	1.72	1.92	1.12
Finished	**3.18**	2.16	2.25	1.39

Table 1: Average number of participated or finished programming exercises for groups of learners with Cocode (default), with Cocode (high social opacity code editor), with Cocode (socially opaque code editor), and without Cocode, respectively.

in **2.31** (SD=2.16) out of 6 exercises on average while the control group participated in 1.12 (SD=1.63) ($p < .01$).

Code Editor Social Opacity
We also conducted a study with 48 and 51 participants to take our course using high opacity and opaque settings, respectively. The result from this study shows that when high opacity and opaque settings are used, learners participate in less exercises compared to the group with Cocode default setting, but more exercises than the control group using the non-Cocode isolated system.

Figure 4 shows the percentage of learners participated in each of the optional and non-optional exercises, and Table 1 shows that in all cases, learners participate in or finish the most number of exercises when using Cocode with default settings.

Conclusion

We introduced Cocode, a programming learning system that shows co-learners' code editors and running screens. Unlike other attempts to increase interactions in online classes, Cocode provides co-learner presence to the learners without any explicit action from learners. Cocode also provides asynchronous co-learner presence to the learners when there are no online co-learners at the time. We evaluated Cocode by comparing Cocode to the system without social features, and the result shows that learners with Cocode participate in more programming exercises during the study.

Acknowledgements

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (2017M3C4A7065962).

REFERENCES

[1] Doug Clow. 2013. MOOCs and the funnel of participation. In *ACM LAK*. ACM, 185–189.

[2] Thomas Erickson and Wendy A Kellogg. 2000. Social translucence: an approach to designing systems that support social processes. *ACM TOCHI* 7, 1 (2000), 59–83.

[3] Catherine P Fulford and Shuqiang Zhang. 1993. Perceptions of interaction: The critical predictor in distance education. *American Journal of Distance Education* 7, 3 (1993), 8–21.

[4] Charlotte N Gunawardena and Frank J Zittle. 1997. Social presence as a predictor of satisfaction within a computer-mediated conferencing environment. *American journal of distance education* 11, 3 (1997), 8–26.

[5] Daniel CA Hillman, Deborah J Willis, and Charlotte N Gunawardena. 1994. Learner-interface interaction in distance education: An extension of contemporary models and strategies for practitioners. *American Journal of Distance Education* 8, 2 (1994), 30–42.