

# Developing Novice Programmers' Self-Regulation Skills with Code Replays

Benjamin Xie  
Institute for Human-Centered  
Artificial Intelligence, McCoy Family  
Center for Ethics in Society  
Stanford University  
Stanford, California, USA  
benjixie@stanford.edu

Jared Ordon Lim\*  
The Information School  
University of Washington, Seattle  
Seattle, Washington, USA  
jorlim7@uw.edu

Paul K.D. Pham\*  
The Information School  
University of Washington, Seattle  
Seattle, Washington, USA  
pkdpham@uw.edu

Min Li  
College of Education  
University of Washington, Seattle  
Seattle, Washington, USA  
minli@uw.edu

Amy J. Ko  
The Information School  
University of Washington, Seattle  
Seattle, Washington, USA  
ajko@uw.edu

## ABSTRACT

Learning programming benefits from self-regulation, but novices lack support for developing these skills of cognitive control. To support their development, we designed Code Replayer, an online tool that enables novice programmers to practice programming and then replay their coding process to reflect and identify process improvements. To evaluate the impact of replaying code on self-regulation, we conducted a formative qualitative evaluation with 21 novice programmers who used Code Replayer to practice writing code. We found that after watching code replays, participants more frequently interpreted problem prompts and planned their solutions, two crucial self-regulation behaviors that novices often overlook. We interpret our results by focusing on two focal points in the design of code replays as a programming self-regulation intervention: interpreting pauses in replays and ensuring replays of struggle are more informative and less detrimental.

## CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; • **Social and professional topics** → **Computing education**.

## KEYWORDS

self-regulation, process data, keystroke logs, metacognition, qualitative methods, computing education,

### ACM Reference Format:

Benjamin Xie, Jared Ordon Lim, Paul K.D. Pham, Min Li, and Amy J. Ko. 2023. Developing Novice Programmers' Self-Regulation Skills with

\*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*ICER '23 V1, August 7–11, 2023, Chicago, IL, USA*

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9976-0/23/08...\$15.00

<https://doi.org/10.1145/3568813.3600127>

Code Replays. In *Proceedings of the 2023 ACM Conference on International Computing Education Research V.1 (ICER '23 V1)*, August 7–11, 2023, Chicago, IL, USA. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3568813.3600127>

## 1 INTRODUCTION

Programming is cognitively demanding because it requires special notation, uses abstractions, and lacks direct manipulation, making programs difficult to inspect [4]. This burden is especially high for novice programmers, who are still learning the syntax of programming languages while also learning core concepts in computing [65].

One skill that helps alleviate the burden of programming is *self-regulation*. Effective self-regulation helps a programmer monitor their cognitive processes as they understand a problem, search for analogous problems, search for potential solutions, identify goals, implement a solution, and evaluate their solution [32]. However, novices are often unaware of the need for self-regulation, its relationship with programming, or find it too taxing to simultaneously learn programming-specific knowledge as well as self-regulation skills [11, 49]. One explanation for this lack of awareness is that novice programmers lack scaffolded opportunities to develop these skills [11, 33, 35, 36].

Existing approaches to developing self-regulation skills focus on explicit instruction from an expert instructor or an emphasis on specific self-regulation behaviors. Prior work has explored explicit instruction to develop self-regulation skills through personalized tutoring (e.g. [32]) and live-coding demonstrations [44, 52, 55, 58], but these approaches require interactions with instructors who have programming and self-regulation expertise. Other work has explored using explicit prompting, such as cuing novices to understand the problem prompt prior to writing code [36, 49]. These approaches assume a formal learning environment such as an introductory computing course. However, people often learn programming in informal and online settings [5, 54]. Therefore, existing approaches to teaching self-regulation in programming may not transfer to informal and online learning contexts.































