Interactive Code Walkthroughs

SIIP Exploration Proposal

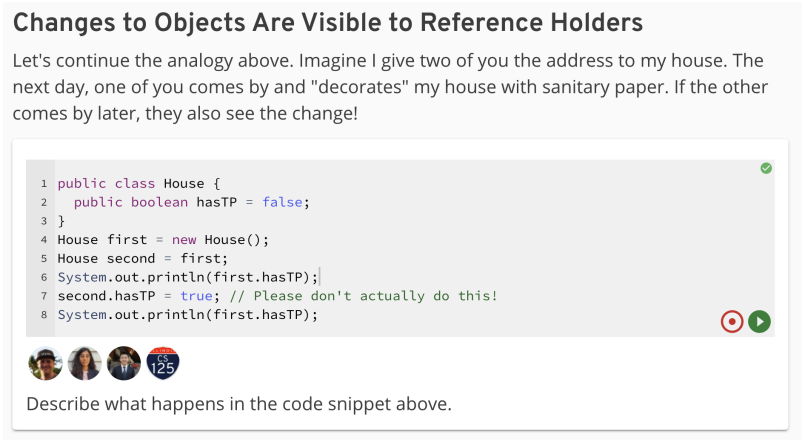
Geoffrey Challen, Teaching Associate Professor, Computer Science
Michael Nowak, Teaching Assistant Professor, Computer Science
Tiffani Williams, Teaching Professor, Computer Science

Problem Description

Live coding exercises are an important and commonly-used component of introductory computer science courses. It provides the instructor a chance to demonstrate how to formulate a problem solving approach, how to transfer that high-level structure to a code outline, how to flesh out the specific functions and lines required to complete the solution, and how to evaluate the results throughout the process. As students watch the code being manipulated, they have the chance to hear the instructor's thought process verbalized, helping them learn how to learn to *think* about algorithmic challenges, not just write code required to solve them.

In a traditional lecture-style classroom live coding would be done in front of the student audience. However, this format already has several weaknesses. Programming examples can be difficult to read even from a short distance, much less across a large classroom. Video capture of live coding can be even more difficult to read. And recordings do not adequately capture the code or the environment used by the instructor, making it difficult or impossible for students to perform their own experiments as they follow along. As a result, while live coding is frequently used effectively in the classroom, similar exercises can be less effective when incorporated into online materials available for asynchronous use.

During preparations to move "CS 125: Introduction to Computer Science" to a fully asynchronous format for Fall 2020, we developed a new interactive code walkthrough component that not only addresses the weakness of live coding but also opens up exciting new opportunities for collaborative course creation and student evaluation. A full demo is available on the course website. Instead of capturing code content as video, we record all changes to the embedded in-browser editor and synchronize them with an audio track. As a result, students can pause the walkthrough at any point and modify and run the code themselves. When they resume the walkthrough, the code and audio pick up where they left off. During replay, students also have access to the same execution backend as the person who recorded the example. The result is more interactive than a video recording, lower bandwidth, and integrates perfectly with the text and other interactive components that comprise our daily lesson format. We also use Otter.ai to immediately provide synchronized captions.



Recording new interactive walkthroughs is also done in a way that facilitates staff contributions. As shown in the screenshot at right, walkthroughs are recorded in the browser, and at the exact point on the page where the walkthrough will appear to students, helping contributors understand the walkthrough's pedagogical context. Recording a walkthrough is as simple as pushing a single button and then typing and talking. Recordings can be reviewed before they are uploaded, deleted by creators and course instructors, and rated by students once they have completed viewing that walkthrough.

These interactive walkthroughs now play a central role in how we teach CS 125. In Spring 2021, 76 daily lessons contained 203 instructional code walkthroughs slots, or roughly 3 per lesson. 105 staff members had together recorded 574 walkthroughs, meaning that, as shown above, most slots contain explanations from multiple staff members with different perspectives. Most walkthroughs are short—with a median length of 4.5 minutes. But other worked examples are longer, up to almost 20 minutes in some cases. We also provide a solution walkthrough for each lesson's daily homework problem.

Our data shows that students make heavy use of the interactive code walkthroughs. 99% of enrolled students watched at least one walkthrough, with the median student viewing 154 (75%) of the instructional walkthroughs. Given that students arrive in CS 125 with varying levels of prior programming experience, this lesson component provides an effective way to personalize the experience to student needs, with students being able to pick and choose explanations that address questions or confusion and skip over ones on topics they already understand. It is also worth noting that our lessons also contain text and editable code examples which also reinforce the concepts being discussed.

Spring 2021 was the first semester that we opened up the walkthroughs for contributions from staff. At this point 104 non-instructor staff members have recorded 356 walkthroughs to supplement the primary explanations, and most lessons have both a full set from the primary instructor as well as supplemental explanations from course staff members. This allows us to provide backup content, creates an opportunity for staff to contribute in a direct and lasting way to the course, and diversifies the voice with which the course speaks.

Student Feedback

Initial feedback from students about the use of interactive walkthroughs in CS 125 has been overwhelmingly positive. 88% of Fall 2020 students completed the course evaluation and rated CS 125 as excellent. Many student evaluations specifically mentioned the effectiveness of the interactive walkthroughs as a pedagogical tool:

- Well structured lessons with videos and walkthroughs helped understand the concepts and see the process of setting up programming problems.
- I also like the instructional walkthroughs where we can see the professor's code as he writes it while he follows along with a voiceover.
- Along with this, the walkthroughs with just Geoff's voice and the code were really nice to see what was going on. I think this format was probably way better than lecture format, for a CS class.
- I think the daily walkthroughs were very thorough and helpful.
- He's very clear in what he says, the walkthroughs are an amazing piece of work
- The walkthroughs and the daily programming were very helpful. The code walkthroughs were easy to understand.
- The website also had interactive coding walkthroughs and it just made learning so much more accessible and original.
- He makes interactive walkthroughs which really helped.

Proposed Improvements

Given the demonstrated success of interactive code walkthroughs in CS 125, we propose to both expand the use of this new technology to several other courses and perform data collection to better understand and improve how students utilize these walkthroughs.

Specifically, first-year project goals will include:

1. **Refactor existing code to support new courses and programming languages.** Currently support for CS 125 walkthroughs is embedded in the CS 125 website codebase. We will separate this feature out as its own separate service, which can be then integrated into other online materials either via frontend framework integration or through web embedding. Collaborators Nowak and Williams have agreed to incorporate these walkthroughs into their own introductory courses, CS 128 and iCAN (Illinois Computing Accelerator for Non-Specialists), respectively. This will also require extend our current interactive playground environment to support C++ (CS 128) and Python (iCAN), a straightforward change that will also make our interactive walkthroughs able to support many other Illinois programming courses taught in these languages (CS 101, CS 105, CS 225, others).

2. **Explore student recordings for assessment and course contribution.** In addition to making recording capabilities available to official course staff, we will also enable students to upload their own recordings. These can be used in two ways. First, they may be useful for student assessment. Just examining the code that a student submits for a programming problem does not reveal their thought process. Requiring that they upload short walkthroughs explaining how they developed their solution will provide more information for staff either for the purposes of assessment or to identify conceptual misunderstandings and ensure that students understand the code they are submitting. Exemplary student submissions may be reused and later presented to other students alongside staff and instructor contributions.
3. **Continue to collect data to analyze usage of interactive walkthroughs and improve online offerings.** As described above, we are already collecting usage statistics about usage of each interactive walkthrough: How many watch it, what order do they examine them if more than one appears on a page, do they return for review before assessment, and how often do they stop to examine and experiment with the code during the walkthrough are some of the questions that we will explore. We are already using this data to help understand how to improve CS 125 and expect that it will also be useful for instructors of other courses.
   In addition, we are also interested in determining what the right order to present explanations to students is when more than one exists for a given concept. Currently the CS 125 website places the instructor walkthrough first, but this may not always be appropriate, particularly if another contributor has a much more effective explanation. We will use student usage data, explanation ratings, and student performance on nearby assessments to help interrogate this question.

Budget Request

Our budget request for 2021–2022 comprises $9,800[1] to support an undergraduate developer for 10 full-time weeks during Summer 2021 and then continuing as part-time (10 hours / week) through the Fall 2021 and Spring 2022 semesters.

[1] Calculated as (10 weeks / summer * 40 hours / week + 2 * 15 weeks / semester * 10 hours / week) * $14 / hour = $9,800.