

“Doing Physics” and “Doing Code”: Students’ Framing During Computational Modeling in Physics

Luke D. Conlin, Salem State University, lconlin@salemstate.edu
Nicole M. Hutchins, Vanderbilt University, nicole.m.hutchins@vanderbilt.edu
Shuchi Grover, Looking Glass Ventures, LLC, shuchig@cs.stanford.edu
Gautam Biswas, Vanderbilt University, gautam.biswas@vanderbilt.edu

Abstract: There is an increasing need to integrate computing into science classrooms in mutually reinforcing, or synergistic ways. We have designed a block-based computational modeling environment and curriculum, C2STEM, with the aim to support students’ synergistic learning of computing and physics through explorations that reflect scientific practices. Prior work has established positive learning gains in both physics and computing for students using C2STEM, but what does synergistic learning look like in the moment? Do students see themselves as “doing physics,” “doing code,” or both? Do they see themselves as exploring like scientists, or as following rules? To address these questions, we examine video and log data of students from two classrooms as they build computational models. We find moments of synergy where students’ framing shifts fluidly back and forth between “doing physics” and “doing code”. We discuss implications for research and for instruction.

Keywords: Computing education, science education, interaction analysis, framing

Introduction

Science classrooms should give opportunities for students to learn by doing science, i.e., learning the concepts while also engaging in the practices of science. Computational modeling is increasingly recognized as an essential scientific practice across multiple disciplines. As such, there is an increasing need to integrate computational thinking (CT) into science classrooms. Integrating CT into science can be challenging (Basu et al., 2016). For instance, there is the risk that learning computing may distract from learning science, and vice versa. There is also the risk that when scaffolding students’ learning of computing, providing too much guidance can inadvertently take students out of a mindset of exploration and more into rule-following.

We have designed a block-based computational modeling environment and curriculum called C2STEM, to support students’ synergistic learning of CT and physics. We have adopted a *learning by modeling* approach, where students learn how to computationally model by using C2STEM to construct computational models of physics scenarios. Prior work has established positive learning gains in both physics and computing for students using C2STEM (Hutchins et al., 2019), but what does synergistic learning look like in the moment? Do students see themselves as “doing physics,” “doing code,” or both? Do they see themselves as exploring like scientists, or as following rules? To address these questions, we examine video and logfile data of students as they build computational models during two classroom implementation studies. We use the analytical construct of framing to describe a students’ sense of what kind of activity they are engaged in.

Framing

Framing is a person’s sense of what kind of activity they are engaged in. It is their rarely spoken but ever-present answer to the question “What is going on here?” (Goffman, 1974). A person’s framing involves a set of expectations and contextual assumptions of how to proceed in any given context (Tannen, 1993). A physics student’s framing of an activity, e.g., as a “discussion” or as “completing the worksheet” is dynamic and ongoing (Scherr & Hammer, 2009). A students’ framing of an activity is signaled by what kind of resources they bring to bear in the moment (Hammer, Elby, Scherr, & Redish, 2005).

To find evidence of students’ framing, we can analyze their conversations to see what resources they draw from moment-to-moment. For instance, students’ framing could be signaled by the kinds of warrants they use to back up their claims (Bing & Redish, 2009), or by a sudden shift in vocal register and tone (Tannen & Wallat, 1987). Their framing could also be signaled nonverbally, by their use of gesture (Scherr, 2009) or by sudden collective shifts of posture (Scherr & Hammer, 2009).

Computational modeling presents unique challenges for the analysis of students’ framing. For instance, often students do not talk when they are coding individually; at times we only have access to their actions within the modeling environment. Moreover, video data may miss important aspects of framing that show up in these coding actions if their screens are obscured or too small for cameras to see.

Despite the challenges, there is a need for more research on students' process of computational modeling in STEM classrooms. Some researchers have used video screen captures to examine students' learning process while coding in STEM classrooms as well as logfiles of students' clicks within the system (e.g., Grover et al., 2017). This work has mostly focused on identifying students' learning of specific coding skills or concepts in closed coding contexts. If we are to integrate computing into STEM classrooms in ways allow students to engage in scientific practices, we need learn the processes by which students engage with the computational modeling environments. Do the students see themselves as doing science? Is coding enhancing their learning of physics, or distracting from it? In this paper, we present a preliminary analysis of students' framing while constructing computational models of physics scenarios. We address two primary research questions:

1. Do students frame their activity as “doing physics”, “doing coding” or both?
2. Do students frame their activity as exploring, or as step-following?

Methods

The data we present here come from two different studies of students working with C2STEM in public high school classrooms. To investigate students' framing, we analyze a combination of video data of the classroom, video screen capture with audio, and logfiles of students' clicks within the system.

Students' framing can be studied via a top-down perspective based on a coding scheme, or a bottom-up approach that lends insight into key moments (Scherr, 2009). These two approaches are not contradictory, but complementary (Grover, et al., 2017). Given the preliminary stage of this analysis, we are starting with the bottom-up approach by selecting key moments to analyze for insights into students' learning processes. We present two such cases to illustrate how to look for evidence of students' framing while coding.

To analyze students' framing from a bottom-up perspective, we first made content logs of the video data, noting times of key transitions (e.g., from building to testing code). We then focused on a subset of video clips that contained evidence of synergistic learning. We iteratively constructed narrative interpretations of students' framing through collaborative viewing and analytic memos.

Instructional context

Case 1 (Amy) is drawn from a semester-long study conducted in 5 sophomore physics classrooms (n ~100 students) in a large urban public high school. Half of the physics classes used C2STEM in place of their laboratory experiments during their kinematics and force units. All of the instruction was delivered by the students' regular classroom teacher. To introduce students to the block-based programming environment, we used a faded scaffolding approach, where they first completed a step-by-step tutorial before being introduced to more open-ended instructional and challenge tasks. Students showed significant learning gains pre- to post-instruction on assessments physics and computational thinking (CT) concepts and practices. This suggests that students learned both physics and computing in mutually reinforcing ways. But what does synergistic learning look like in the moment? Are students seeing themselves as doing science, or doing code?

To examine these questions, we selected students for case studies who showed large pre-post gains on both physics and CT, and for whom we had good video and audio data. We looked through their video screen captures for moments of debugging, since synergistic learning often occurs when students are debugging their code (Grover et al., 2019). Debugging was identified when students' code behaved in an unexpected way and they edited and tested their code. We present the results from an episode in which Amy is debugging her code during a 2D kinematics task.

Case 2 is drawn from a second study we conducted in 5 physics classes in a different large urban high school. The instructional context was different in several ways. It was a charter high school with a focus on technology, so more students had prior coding experience, with many students reporting having used Scratch (a block-based language similar to Snap! used in C2STEM). The study involved mostly juniors (~50) on the last three days of the school year. This short unit on 2D projectiles was used as a review for the final exam.

The second study was a pilot to test a new mode of using C2STEM. In previous studies we had noticed that while students were learning physics and CT, they were not approaching many of the tasks with a spirit of exploration, a key ingredient of “doing physics”. We hypothesized that our faded scaffolding approach to introducing the programming environment may have inadvertently led students to frame their activity as step-following rather than exploration. For this study, we adapted C2STEM to encourage more exploration. To introduce the coding environment, we used a preparation for future learning approach (PFL; Schwartz & Martin, 2004), in which students were first given an open-ended task where they would invent their own solutions, to create a “time for telling” in which they would get direct instruction on how an expert might solve the problem.

We designed the environment to support exploration, but do they take it up this way? To explore this

question, we selected a group of four students to analyze for whom we had the best video and audio data. We analyze their interactions and their behaviors in C2STEM to analyze whether they were framing their activity as exploration. Since we suspect that framing can be heavily influenced by first impressions, we chose the very first coding activity to analyze.

Case 1: Doing physics and doing coding

In this episode, students are working on the task to build a model of a drone that would carry two packages at constant velocity to the right before dropping the first package onto a target on the ground, and a second package onto an elevated target.

Amy started with a program she had already built in a previous session. It was close to working, but had two main problems: (1) Package 1 did not quite reach the target, and (2) Package 2 did not move with the drone at all, but rather remained at rest in the air.

- 1 Amy: *(looking at another student's code) What code is that? Is that the drone?*
- 2 Student: *Uh, yeah and the packages.*
- 3 Amy: *This is what she- this is like what I did with Liz. Yeah the drone moves, but it doesn't like I- This is the one that's not moving. It drops it here.*
- 4 Student: *Well you gotta get all the packages to move*
- 5 Amy: *I think something is cancelled in the y... (inspects neighboring student's code, then adds "change x-position of package 2" block to her own code.)*

To analyze Amy's framing, we examine the resources she accesses over time. In debugging her code, sometimes Amy relies on physics as a warrant for why the package is not dropping ("Is the acceleration supposed to be zero?" "Something must be canceling in the y direction"). At other times she relies on coding to understand why it is not falling. She tweaks a particular block of code by adjusting its value and testing in a trial-and-error process. Amy's framing shifted fluidly back and forth between "doing physics" and "doing code". This debugging process was an example of synergistic learning, where learning physics and learning coding are mutually reinforcing each other.

Case 2: Exploration versus step-following

In this episode, the students were given almost no instructions, only an open-ended task: "EXPLORE to find a way to move the truck to the right with a constant velocity of 10 m/s." This was their first use of C2STEM.

Four students sitting at the same table worked independently on the challenge for 5 minutes, using their own laptops. After 5 minutes, the teacher checked in on their progress, asking if they had gotten the truck to move yet. Alan replied, "Yeah, we did". The teacher then spontaneously gave them a new challenge: Stop the truck at the stop sign.

- 10 Teacher: *Okay, try to get it to STOP hehe. See if you can get it to come up to the intersection and stop.*
- 11 ALAN: *It's going over the line, but, yeah.*
- 12 BERT: *Oh I know how. I think? Is it like this? ((turning his computer to the teacher))*
- 13 Teacher: *((walking up to look)) Do it again. ((Student plays it)). Wow! How'd you do that?!*
- 14 ALAN: *((to Guest Teacher)) I want to get it to stop RIGHT AT the stop sign.*
- 15 Teacher: *((To BERT:)) Did you do it by manually stopping the simulation?*
- 16 BERT: *No.*
- 17 Teacher: *((looking through the code)) You only have those like two lines of code! Repeat 10. OHHHh, that's clever! I see how he did it.*
- 18 GT: *Did you all have a similar approach?*
- 19 ALAN: *((To Teacher:)) I stopped mine right at the stop sign. Or tried to at least.*
- 20 Teacher: *((watching Alan's screen)) Yah. ((To Guest Teacher)) See, they did it by not messing with the velocity at ALL, just by changing the position a set number of times.*

Analysis of the logfile for the first 5 minutes of exploration revealed that Alan quickly got the truck to move by using a "repeat 10 times" control block with a "change x position by 10" inside (Figure 1a). Although he proclaimed success, the logfile analysis reveals that Alan never stopped refining and rebuilding his code, trying

to incorporate the velocity block. Alan tried and tested a total of six more combinations of blocks, including set x velocity to 2 and change x velocity by 2 (Examples given in Figure 1b).

When the teacher gave the new challenge to stop at the stop sign, Alan explored by tweaking his code to adjust where the truck would stop. He adjusted the value of the *change x position* block from 10m to 9.5m, and considered it mostly successfully (Line 19).

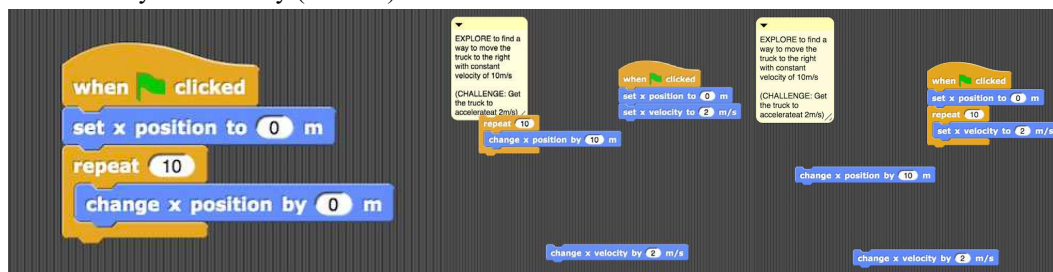


Figure 1. A few examples of Alan's attempts to move the truck, from the logfile "playback" of Alan's clicks.

Conclusions

We have examined students' framing while engaged in computational model building in STEM classrooms. We found that there are moments of synergy where students' framing shifts fluidly back and forth between "doing physics" and "doing code", particularly during times when they are debugging. We have also found that a PFL-based approach to instruction encourages exploration, despite little up-front instruction on the environment.

This work contributes a small but growing body of research on students' learning processes while coding in STEM classrooms. In analyzing students' framing, we have found evidence that students, at least at times, frame their activity as scientific exploration. This work highlights how analysis of classroom video, screencapture video, and logfile analysis can be triangulated to provide insight into students' learning processes in STEM classrooms that integrate computing.

Any large-scale attempts to integrate computing into STEM classrooms must be flexible enough to accommodate multiple modes of instruction. The present study demonstrates that C2STEM can be adapted to support multiple modes of instruction, including open-ended exploration, direct instruction, and faded scaffolding approaches. An important lesson is that how you set up coding instruction initially can have consequences for how students approach computational model building: as step-following, or as doing science.

References

- Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and Practice in Technology Enhanced Learning*, 11(1), 1–35.
- Bing, T. J., & Redish, E. F. (2009). Analyzing problem solving using math in physics: Epistemological framing via warrants. *Physical Review Special Topics-Physics Education Research*, 5(02108), 1–15.
- Goffman, E. (1974). *Frame analysis: An essay on the organization of experience*. Harvard University Press.
- Grover, S., Hutchins, N., Biswas, G., Snyder, C. & Emara, M. (2019). Examining Synergistic Learning of Physics and Computational Thinking through Collaborative Problem Solving in Computational Modeling. In *Proceedings of AERA, 2019*. Toronto, CA.
- Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., & Stamper, J. (2017). A framework for using hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming environments. *ACM Transactions on Computing Education (TOCE)*, 17(3), 14.
- Hammer, D., Elby, A., Scherr, R.E., & Redish, E.F. (2005). Resources, framing, and transfer. In J. Mestre (Ed.), *Transfer of Learning from a Modern Multidisciplinary Perspective* (pp. 89–120). Greenwich, CT: Information Age Publishing.
- Hutchins, N.M., Biswas, G., Maróti, M., Lédeczi, Á., Grover, S., Wolf, R., Blair, K.P., Chin, D., Conlin, L.D., Basu, S., and McElhaney, K. (2019). C2STEM: A system for synergistic learning of physics and computational thinking. *Journal of Science Education and Technology*, 1–18.
- Schwartz, D. L., & Martin, T. (2004). Inventing to prepare for future learning: The hidden efficiency of encouraging original student production in statistics instruction. *Cognition and Instruction*, 22, 129–184.
- Tannen, D. (1993). *Framing in discourse*. New York: Oxford University Press.
- Tannen, D., & Wallat, C. (1987). Interactive frames and knowledge schemas in interaction: Examples from a medical examination/interview. *Social Psychology Quarterly*, 50(2), 205–216.