

Finding Productivity in Design Task Tinkering

Gina Quan and Ayush Gupta, University of Maryland, 082 Regents Drive, College Park, MD, 20742
gquan@umd.edu, ayush@umd.edu

Abstract: This poster looks at students' productive tinkering in a design activity. We analyze the emergence of a student-generated task and how the process of tinkering supported students in making progress in the task. In design activities, systematic, planned approaches are often valued over tinkering, a process which shortcuts that kind of analytical thinking. We argue that tinkering has value as a productive engineering practice and that it can support more systematic approaches.

Introduction: Can Tinkering Be Sense-Making?

Tinkering is an ad-hoc approach to a problem and involves the practice of manipulating objects to characterize a particular system in an exploratory way, often with the goal of achieving practical success (e.g. Turkle & Papert, 1991; Berland et al., 2013). Tinkering contrasts more deliberate sense-making towards achieving conceptual understanding of *how* some phenomenon works. Some researchers see tinkering as unproductive because of the lack of clear aims, or because it does not prioritize conceptual learning (Law 1998; Yeshno & Ben-Ari, 2001). Others view it as productive for students' learning and for generation of novel solutions (Turkle & Papert, 1991; Berland et al., 2013; Roth, 1996). In this poster, we speak to this tension on the productivity of tinkering for novice designers and programmers. We claim that tinkering, or ad-hoc sense-making, can play a productive role in making progress towards design-activity goals.

Method: Data Context, Selection, and Analysis

Our data comes from videotaped in-class group-work and interviews with a pair of high school students in a 2-week summer outreach program on physics. As part of the program, students learned to program Arduino (microcontroller) controlled robot-tanks (henceforth, Arduino-bot). The tanks could be programmed to move (via Arduino) and were fitted with basic distance and light sensors. Students spent the first week of the program working 1-2 hours per day on self-paced design tasks in pairs. The design task under consideration in this poster is one in which students had to program the Arduino-bot to make it turn if there was any obstacle in the tank's path (using the Arduino and the distance sensor for obstacle detection). Students were provided with a variety of resources, including sample code. The classroom was staffed by two instructors and several volunteers. Due to limited resources on the project, one pair was filmed each day, and two follow-up interviews were conducted. This analysis focuses on the fourth day of program, in which the two students, Hazel and Silver (pseudonyms), generated a task and attempted several strategies to solve it. We focus on this task because of the multiplicity of strategies employed and students' continued engagement in the task.

In this poster, we will present a fine-grained analysis of video data to characterize students' epistemic activities during the task. We look for evidence of particular student goals through speech, gestures and actions using interaction analysis (Jordan and Henderson, 1995). We also analyze students' talk as evidence of the knowledge students are drawing on (Parnafes & diSessa, 2013) and what they find productive. This analysis is intended to show that tinkering *can* be productive, and not necessarily that it is always productive.

Preliminary Analysis: An Example of Productive Tinkering

Our focal episode begins when Hazel and Silver had just completed a task in which they were asked to make the Arduino-bot move forward until it detected an obstacle/wall, and then make a right turn. Hazel and Silver initially did not program the Arduino-bot to stop after turning right, so after the Arduino bot turns right it starts running over the keyboard until Silver grabs it. In response, they decide that they should make the robot stop after turning right. This goal of making the robot stop was not assigned in class, but many contextual events contributed to its reinforcement. First, we see the goal emerge through constraints of the physical space: had Silver not grabbed the bot, it would have run off the table. When Silver suggested modifying the task to stop the bot, Hazel quickly took up the task and offered suggestions of how to make it stop. This goal is reinforced by their interaction with other groups who sought input from Hazel and Silver on how to make it stop. The classroom culture of student ownership over the project task also played a role in their comfort in modifying the task statement - no group asked the teacher for permission to do this. In that sense, that the problem statement is emergent is some indication that students are in an exploratory mode in working with the robots. What follows in the next several minutes is Hazel and Silver tinkering through a variety of strategies to make the robot stop. We argue that such tinkering was productive for Hazel and Silver, and that it reflects authentic engineering practice.

In the first few minutes of trying to make the robot stop, the girls engage in the rapid testing of several

solutions to the problem. Here, tinkering supported extended engagement in the activity. They first try to change the digital settings of the motors, not knowing that the settings they were changing correspond to the motors' direction, not speed. This strategy leveraged knowledge that was productive in a previous task, in which rewriting a digital output from HIGH to LOW turned an LED off. We see this as tinkering because it does not attempt to draw out underlying principles of how the system works - they even verbally acknowledge that they do not understand what individual functions do. However, we see it as productive in that they notice the similarity between tasks (switching off motor versus LED) and draw on that prior experience. Next, the epistemic nature of their activity changes, as they try to make sense of how motor treads work (now extracting new information about the new system). Their dialogue and gestures give evidence that they see that turning a motor off will cause a corresponding tread to stop moving. After their first modification does not work, they spend a couple of minutes checking a reference guide and then search on Google for the solution, practices which were encouraged in class. We see this utilization of available resources, without getting too fixated on one resource, as productive trial-and-error. Later they isolate and execute individual lines of code (commenting out the other lines), to generate some knowledge about what individual commands do. These rapid trial-and-error activities occur in the span of about seven minutes.

One could argue that instead of tinkering Hazel and Silver should have systematically parsed the code to make sense of it; they would have had better task success and better learned Arduino programming through that. We contend this notion. Their activities reflect a recognition of the variety of resources at their disposal and a systematic walk through the resources to try and achieve their goal. They also reflect a certain level of judgment and resource management: they did not get fixated on any path, but quickly judged if the path would be productive, and if not, they switched tactic. These are all important skills in the design process, and within authentic engineering practice are not trumped by the value of systematic processing of the code.

We don't, however, deny that line-by-line processing of code is also a valuable epistemic practice. Indeed, when the other strategies don't make the bot stop in about seven minutes, Hazel and Silver begin synthesizing their observations into an understanding of what lines of code do. Hazel remarks, "So we know that, that HIGH was backwards." We then see Hazel parsing each line of the code in terms of its functionality, "So it takes a reading. It calls it for one second. It goes forward." However, it was while tinkering with the variables in the code that they explicitly problematized their lack of understanding of certain functions. This awareness could have contributed to their getting to parsing the code to make sense of the functions. Tinkering can thus give students a sense for where they might focus a more deliberate analysis later.

Discussions and Implications

Though tinkering may not lead to generalizable content learning, we argue that it has value as an engineering disciplinary practice. Tinkering produces useful knowledge for troubleshooting, and points out gaps in understanding, thus potentially prompting future systematic unpacking of ideas. Given the small sample size of present work, future work needs to be done to continue building on the productive role of tinkering in design activities. Work also needs to be done to understand how to encourage this kind of productive tinkering.

References

- Berland, M., Martin, T., Benton, T., Petrick Smith, C., & Davis, D. (In Press). Using Learning Analytics to Understand the Learning Pathways of Novice Programmers. *Journal of the Learning Sciences*.
- Jordan, B., & Henderson, A. (1995). Interaction analysis: Foundations and practice. *The Journal of the Learning Sciences*, 4(1), 39-103.
- Law, L. C. (1998). A situated cognition view about the effects of planning and authorship on computer program debugging. *Behaviour & Information Technology*, 17(6), 325-337.
- Parnafes, O. & diSessa, A. A. (2013). Microgenetic learning analysis: A methodology for studying knowledge in transition. *Human Development*, 56(1), 5-37.
- Roth, W. M. (1996). Art and artifact of children's designing: A situated cognition perspective. *The Journal of the Learning Sciences*, 5(2), 129-166.
- Turkle, S., & Papert, S. (1991). Epistemological pluralism and the revaluation of the concrete. *Constructionism*, 161-192.
- Yeshno, T., & Ben-Ari, M. (2001). Salvation for bricoleurs. In *Proceedings of the Thirteenth Annual Workshop of the Psychology of Programming Interest Group*, Bournemouth, UK (pp. 225-235).

Acknowledgements

We thank Eric Kuo and Chandra Turpen for reviewing earlier drafts of the manuscript. We thank members of the UMD Physics Education Research Group for discussions.