Knitting and Computing as an Entryway to Computer Science

Stephanie Yang, Teachers College Columbia University, syy2114@tc.columbia.edu

Abstract: This paper describes an exploratory study interviewing novice knitters and programmers regarding their crafts, using their own projects as artifacts to generate discussion about their respective practices for comparison and contrast. Potential cross-discipline lessons and applications are considered, in particular the impact of qualities such as physicality and representational diversity.

Introduction

This paper seeks to relate computer science to existing pastimes – specifically, knitting – in order to broaden views of what computer science entails. This research is motivated by dominant attitudes around computing, which influence learners' ability to identify within the field. Turkle and Papert describe this as a "discrimination in the computer culture that is determined…by ways of thinking that make them reluctant to join in" (1990, p. 132). Supporting multiple, alternative views of computer science may help learners identify as computer scientists.

Knitting specifically has a strong conceptual overlap with programming (Craig, Petersen, and Petersen, 2012) due to the historical ties between the textile and computing industries (Bratich and Brush, 2011; Monteiro, 2017). However, crafting practices "constitute…structures of activity that are in themselves guided by structural, material, and ideological constraints" (Nasir and Hand, 2008, p. 176). The realm of possibility within a craft is therefore strongly influenced by the nature and context of the practice itself, whether it be textiles or computing.

Here I describe an exploratory study regarding knitting and programming. I conduct and qualitatively analyze interviews with both knitters and programmers to compare and contrast how each think about their work. I then consider how those differences might allow one field to utilize new ideas and perspectives from another.

Methods

In the course of this study I conducted semi-structured cognitive clinical interviews with all participants in an exploratory study to investigate the following research questions:

- What are the relations between how people think about the practices of knitting and programming? What are the similarities and differences?
- What new things can we learn about the specific domains of programming and knitting by looking at these similarities and differences across the domains?

The study was conducted at a major Northeastern university. Participants were asked while arranging the interview to bring either a knitting pattern they had used or a computer program they had worked on that had not been written for academic or work purposes, which was used as part of the interview. The interview focused on comprehension and compositional practices in each condition, both in general and regarding the participant's artifact. Each participant's experience and history with their given domain were also discussed.

Analysis of the interview transcripts used a coding scheme based on literature regarding computational thinking, systems-level thinking, and making/crafting. It was largely based on Weintrop et al.'s (2016) and Brennan and Resnick's (2012) frameworks for computational thinking practices, with additional concepts from Craig, Petersen, and Petersen's (2012) observations on relations between knitting and computer instructions.

Results

This study found new differences in how knitters and programmers discussed their respective practices. Many of these differences were driven by the nature, history, and unique affordances characteristic of each practice, supporting the idea that these creative processes are influenced by the nature of their respective activities.

The majority of knitters interviewed expressed preference against use of symbolic charts or textual abbreviations in knitting patterns despite their use alongside textual instruction in many patterns. This contrasts against common computer programming practices, whose instructions often heavily rely on symbols. Clarity of instruction was important to both disciplines, but knitters' standards for clarity varied with respect to format.

Developmental time cost appeared to significantly influence willingness to experiment. Many knitters felt comfortable projecting and adapting patterns (e.g., extending a base pattern to an entire hat), but claimed less desire to experiment with knitting than crochet because knitting errors took much longer to correct. Programmers project and adapt code as well (e.g., external libraries, APIs, code from previous projects), but have the advantage of stable rapid prototyping that knitters lack. This suggests that easier iteration may encourage experimentation in physical crafts and demonstrates how a practice's affordances affect composition and troubleshooting.

A third relates to level of detail in relation to composition. Because common knitting stitch patterns correlate closely with fine-grained instruction, knitters may describe knitted artifacts in terms of larger patterns with minimal information loss. Programming instructions, however, have greater possible variety in execution towards achieving the same outcome (e.g., ways to sort a list) and so description focuses more on fine detail.

Finally, practitioners of each draw upon online and in-person resources differently. Knitters were more likely to consult friends and communities for help, while programmers relied on the Internet more often. This may relate to knitting's history as a social activity (Bratich and Brush, 2011) as well as programming's virtual nature.

Discussion

Comparing two different disciplines is not a straightforward task, a complexity reflected in the difference between reading practices here. When focused on comprehension knitters and coders preferred quite different symbol systems, due in part to the materiality of the domain and associated time costs. Even where concepts overlap, seemingly similar crafts have different affordances (Buechley and Perner-Wilson, 2012). It is important to consider where specific qualities – physical or virtual, flexible or rigid – may influence overall practice.

Comparing the two fields, it is evident that practices in one activity might inform thinking and learning in the other. Knitters may be more free to experiment with if they had rapid prototyping methods like those of software, cutting down the time cost associated with error and sample-making. On the other hand, the representational diversity in knitting patterns applied to computing education might be more palatable for those struggling with a one-size-fits-all abstract approach to programming education (Turkle and Papert, 1990).

Designing accommodations to different styles of writing and interpreting computer code may be one possible step towards making programming and computer science education more accessible to learners who struggle with dominant teaching methods, enabling those who feel out of place to explore alternative approaches to learning. Fostering conversation between practitioners in both fields might bring about positive innovations in either or both crafts, although it would be necessary to keep in mind the different affordances of each.

Conclusion

In order to expand common views about computer science and where it can be applied, it is useful to demonstrate that computer science can be applied to and connected with a wide variety of different fields and practices. Here I build on existing relationships between programming and textiles in order to examine how both knitters and programmers think about their practice by interviewing them and discussion their own projects.

Major takeaways from this study are that the nature and qualities of both knitting and programming are tied to the lived practice of each. Programming, being textually based and allowing for rapid iteration, has different affordances compared to knitting despite conceptual overlap on the surface. Given the connection between the two fields, however, finding more ways to bridge the two may expose new ways of looking at and understanding computer science that may encourage students to see themselves as potential users of computer science in nontraditional ways. Future work may consider how to do so while integrating the strengths of each practice.

References

Bratich, J. Z., & Brush, H. M. (2011). Fabricating activism: Craft-work, popular culture, gender. *Utopian studies*, 22(2), 233-260.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*, Vancouver, Canada (Vol. 1, p. 25).

Buechley, L., & Perner-Wilson, H. (2012). Crafting technology: Reimagining the processes, materials, and cultures of electronics. *ACM Transactions on Computer-Human Interaction* (TOCHI), 19(3), 21.

Craig, M., Petersen, S., & Petersen, A. (2012, February). Following a thread: knitting patterns and program tracing. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 233-238). ACM.

Monteiro, S. (2017). The fabric of interface: Mobile media, design, and gender. MIT Press.

Nasir, N. I. S., & Hand, V. (2008). From the court to the classroom: Opportunities for engagement, learning, and identity in basketball and classroom mathematics. The Journal of the Learning Sciences, 17(2), 143-179.

Turkle, S., & Papert, S. (1990). Epistemological pluralism: Styles and voices within the computer culture. Signs: Journal of women in culture and society, 16(1), 128-157.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25(1), 127-147.