ENGAGING PROSPECTIVE TEACHERS AND STUDENTS IN PROGRAMMING ACTIVITES

Author, Author, and Author
Affiliation

We describe a coding lesson enacted by prospective secondary math and science teachers with students ages 11-13. We discuss challenges of introducing this content to learners for whom it is unfamiliar and the pedagogical skills teachers needed to effectively address coding activities with students.

Wing (2010) states that computational thinking involves solving problems, designing systems, and understanding human behavior. Computational thinking is increasingly necessary in our technological world, yet according to the United States Bureau of Labor Statistics there are proportionally fewer computer science graduates now than in the 1980s. The skills used in coding prepare students for learning challenging and abstract concepts of upper level mathematics. Cognitively demanding problems (Smith & Stein, 1998), those without an obvious procedure, which require students to invent novel strategies, can support developing computational thinking. Coding requires students not only to develop a solution to a problem, but also communicate that solution in a specific way such that it can be carried out by a computer. This study aims to understand how to effectively implement a lesson for 11-13 year-olds focused on developing computational thinking and coding.

Participants were 7 prospective secondary math and science teachers (PTs) and 12 students, ages 11-13. When we enacted the lesson with PTs, our goal was to familiarize them with the lesson and to reflect on aspects that might challenge students. The lesson relied on the "use-modify-create" progression (Lee et al., 2001) to support students' learning, foster creativity, and give teachers a structure for scaffolding student problem solving. First, students interacted with a computer to learn how it was coded, which involved tinkering and a self-led exploration to describe actions the computer could execute. Then they modified and revised instructions that had been written for the computer to create their own project, in turn developing their computational thinking. We found that students were independent and confident during the "use" portion of the activity, and that support from the PTs was important during the "modify" and "create" portions because students struggled to have purpose to their tinkering and to persevere in solving problems. We also identified specific PT pedagogical skills needed for enacting a computational thinking lesson: questioning to help students clearly describe the computer's actions in mathematical language, attending students to connecting actions to code to identify causes of and relationships among computer actions, and making a plan before translating it into a computer algorithm.

Acknowledgements

The research reported here was supported in part by National Science Foundation Award #XXXXX. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32–37.

Stein, M. K., & Smith, M. S. (1998). Mathematical tasks as a framework for reflection: From research to practice. *Mathematics teaching in the middle school*, (4), 268–275.

Wing, J. M. (2010). Computational thinking: What and why? Available at https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf