



Using Physical Computing Projects in Teaching Introductory Programming

Karen H. Jin

Department of Applied Engineering and Sciences
University of New Hampshire, Manchester, NH
Karen.Jin@unh.edu

Howard Eglowstein

Department of Applied Engineering and Sciences
University of New Hampshire, Manchester, NH
Howard.Eglowstein@unh.edu

Mihaela Sabin

Department of Applied Engineering and Sciences
University of New Hampshire, Manchester, NH
Mihaela.Sabin@unh.edu

Abstract

Physical computing engages students who are learning how to program through hands-on projects with tangible devices. Many of these projects are fun and artistic, but may disguise fundamental and non-obvious concepts such as variables, conditionals or complex control flow. Access to fun “gadgets”, e.g., sensors, motors, LEDs, etc., gets the students excited about projects, without focusing on specifically learning programming and computational problem solving techniques. This work presents the authors’ experience of teaching a physical computing workshop for middle school students using Adafruit Circuit Playground Express and Microsoft MakeCode. Project selection is based on a set of software and hardware features in order to choose the ones that best help students retain programming concepts.

Our Experience

- This work presents the authors’ experience of teaching a week long physical computing workshop for middle school students (5th-7th graders) with limited experience in programming.
- The Adafruit Circuit Playground Express, an all-in-one hobbyist board with embedded sensors, LEDs and USB support, is used as the hardware platform, in conjunction with the Microsoft MakeCode visual programming environment.
- Project-based instruction using sample projects with open-ended extensions.



How to Choose Projects

Our aim is to identify projects that are not only engaging and enjoyable to work on, but also effective in demonstrating abstract programming concepts. The initial project candidates were prototyped and they grouped based on several feature rubrics: software complexity, hardware complexity, software extensibility and hardware extensibility. Projects were then selected based on these rubrics. For example: projects with medium to high software complexity and low to medium hardware complexity are best suited for introductory projects. Projects with high software extensibility are more desirable over those with high hardware extensibility in helping students reinforce their programming skills.

Using Physical Manipulatives in Teaching Intro to Programming

- **The Good:** Manipulating physical objects makes it easier for students to relate to a programming task. It becomes more personal.
- **The Bad:** It can be hard to teach variables, loops and control concepts when the students are distracted by the gadgets.
- **and the Ugly:** Some students, not used to crafting, can get frustrated with issues not related to programming. They call it “hot” glue on purpose.

