Using Graphical Programming to Introduce All to Programming

Eric A. Freudenthal, Computer Science Department, University of Texas at El Paso
efreudenthal@utep.edu
Submitted to CSTA

We have developed a "Media-Propelled" approach to engaging students in Computational Thinking (iMPaCT) suitable for high school & college math, science and programming classes.

iMPaCT utilizes a learner-centric approach which accelerates engagement with programming and algorithm design. iMPaCT's lessons are motivated by graphical rendering problems that build rather than rely upon mathematical maturity.
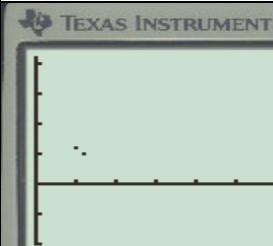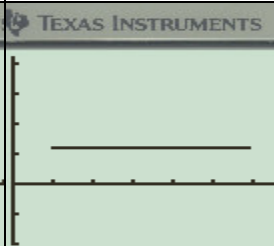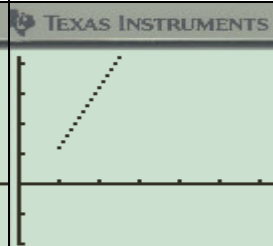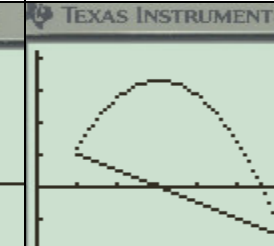
A key aspect of iMPaCT is its single-session introduction to programming that enables it to be incorporated into middle- and high school enrichment programs, high-school and college math courses, and as the basis of several college-level programming courses. iMPaCT has been taught using interactive language systems such as Jython or the BASIC interpreter within graphing calculators common in high school and college math classrooms.

Like conversational foreign language courses, our lessons are designed to minimize cognitive load by delaying systematic examination of vocabulary and syntax. In iMPaCT, students are initially introduced to simple and easily extended idioms that reveal key concepts within easily understood problems.

The first three columns of Table 1 illustrate much of the initial 45 minutes session's progression where students learn and practice:
- The assignment of values to named variables and simple algebraic operations using the computer's (or calculator's) interactive mode,
- The use of a named function within a program (such as DOT of Table 1) that draws dots on a raster display, and
- Iteration as a solution to the problem of drawing of first horizontal, and then inclined lines.

Table 1. Programs DOT, LINE1, LINE2, and CURVE from a calculator-based iMPaCT course. As students expect, the graph's origin is lower-left.

At each step, students are challenged to extend the programs in a manner that exposes key concepts. For example, modifying
- LINE1 to draw a variety horizontal line segments motivates discovery of iteration range and stride, and the independence of the x and y variables, and
- LINE2 to draw a variety of lines with varying slopes and translations that are less steep, lower and parallel, or maybe even perpendicular.

Depending on course objectives, subsequent lessons can focus on the exploration of math and/or programming skills. For example, program CURVE of Table 1 may be examined during a lesson motivated by the challenge of rendering a non-linear form, thus exposing students to the relationship between functions and their rates of change. Later, the students are lead to recognize that this approach to generating curves can also be employed to animate (and understand the principles underlying) the trajectory of a thrown object.

By focusing on problems with relevance outside of programming, iMPaCT can be implemented as an enrichment activity within conventional math, science, and engineering courses and specialized introductory programming courses. For example:

- We are examining introducing programming to all incoming freshmen through the embedding of iMPaCT into required math courses.
- iMPaCT has been embedded into a required introductory course in the C language intended for electrical engineering students that teaches students core programming concepts using electrical simulation problems in Python. After this preparation, these students master C far more quickly than in a conventional course.

Information on iMPaCT is posted to the web: http://cs.utep.edu/mpct.   Dr. Freudenthal is interested in feedback from faculty who incorporate this approach. He also has funding for faculty interested in contributing to iMPaCT.