The CSTA Voice

Article Submission Template

Please submit by November 10, 2011 for the January 2012 issue.
Recommended content word count: 700-800
Topic: middle school CS curriculum project
Please initial your acceptance of the following:
All submissions become the property of CSTA and may be reproduced and used in whole, in part, or as edited, by any means and in any manner or media, at the sole discretion of CSTA.
Date____01/08/2012_____ Initials__Z.D.. MAE___

Your title suggestion: __ Middle School CS? Yes!_____
Author: ___Zachary Dodds, Mike Erlinger, and Elizabeth Sweedyk_____
Postal mailing address:___301 Platt Blvd. Claremont, CA 91711_____
A couple sentences to be used in the "About the author" block:____The authors teach in the computer science department at Harvey Mudd College. _____

Other Resources readers might find useful such as URLs or books, etc?__*please see the references*___

Are you aware of any upcoming events related to this piece or computer science in general readers might be interested in? (title, dates, location)____n/a____

Submit & questions to: Pat Phillips at cstapubs@csta.acm.org

Enter text below. Please do not include special formatting in your text.
Please tell us about the project: what motivated your work, who's involved, how it's going, what it looks like, when it might be available for broader distribution, etc.

---------- article ----------

"There's no recession here!" chirps an Apple Store employee, adding me to the bottom of untold screenfuls of names. In light of the diverse group who overflow the cavernous space, it's hard to argue with his contrived cheerfulness. Here, headscarves jostle with ballcaps; shirtsleeves and ink sleeves crisscross the displays whose flickers attract crowds. Nowhere is the American ideal of equal opportunity more fully realized than in consumption, and computation is at the top of many consumers' lists.

A generation ago, computation was a tool with relatively little impact on day-to-day life. That generational gap remains today -- not in the consumption of computation, however, but in its creation. As a result, computational enthusiasm and computational skills are unbalanced. The Department of Labor's anticipates this imbalance between the computational skills of our schools' graduates relative to the opportunities available [1]. The imbalance appears, too, in

who develops computational skills. The relative absence of women and ethnic minorities in computer science is as well documented [2] as it is clear without such studies. Computer science does not appear in curricula until late in high-school, if at all: at that point many students have been convinced that it is not something that *people like them* do.

How and when *should* computation appear in school? Certainly, computation *as a tool* should permeate K-12 education; in light of modern life, there's probably no way to stop this from happening, even if we wanted to. Yet for an initial exposure to computation *per se*, middle school offers us an opportunity to make computational creativity as much a part of students' identities as its consumption is already. At Harvey Mudd College we have undertaken two efforts to integrate middle-school teachers and students into the culture of creating computation. The first approach is through the design and deployment of educational games, an NSF project entitled *The Games Network: Games for Students, Games by Students*. [3] Our second effort is a middle-school curriculum named *Middle-years Computer Science*, or MyCS.

The Games Network addresses the misunderstandings students may have about the practice of computer science. Our approach is to engage middle school students in a semester-long software development project carried out with college-level computer science students. The deliverable is one of the modern-day CS artifacts that all students relate to: an educational computer game.

At the start of each semester, middle school teachers provide HMC's software development class with a list of learning objectives; e.g., items drawn from the 6th-7th Grade-level Content Expectations (GLCE). Each four-person software-development team chooses a GLCE they will target with their game. The middle school teachers and students are integrally involved throughout the development, providing bi-weekly written evaluations of the game's concept, storyline, and user interface. Middle school students also act as testers of game prototypes and beta releases. This first-hand involvement dispels CS stereotypes and offers insights into some of computer science's most compelling challenges.

For middle school teachers, participation does not require any specialized background. Yet it allows them to

- acquire media-rich, interactive learning tools designed specifically for their classrooms, at no cost
- engage their students in designing tools for their own learning
- support literacy efforts by providing an authentic context for communication between their students and college students/faculty

For CS professors who teach (or want to teach) game design and development, HMC's Games Network model allows them to

- enhance game projects by providing their students with a real customer who has an authentic need for game software
- cultivate a sense of social responsibility in their students and allow them to produce software that can have positive impact
- provide their students with the opportunity to serve as role models for younger students.

The design and testing of games provides a natural link with students' everyday experience of computation. Our middle-school CS curriculum, MyCS, pushes this connection further by making computation itself "the game." MyCS's goal is to develop the computational awareness and sophistication of larger numbers of middle-school students than existing curricula do. Though not truly exclusive to middle-school, MyCS does not shy from the turmoil of individual- and group-identity formation that energizes that age. Rather, MyCS seeks to build bridges amidst the family and societal pressures that influence the long process of self-definition. As students construct their many answers to “Who am I?” they necessarily also answer “Who am I not?” For its part, MyCS seeks to place CS in the former category more often than the latter.
Resources for engaging students with the "game of computation" abound, but many of those resources lack an overarching structure. For instance, the online games FactoryBalls [4] and LightBot [5] provide compelling, minds-on introductions to procedural thinking -- with the same requirement for precision and clarity that general-purpose computer programming demands. MyCS supplements these kinds of resources with our own Picobot [6], a Karel-like automaton that introduces an abstract language -- with all of the concomitant benefits and drawbacks -- into students' specifications.

MyCS's hands-on activities similarly convey the building blocks of reproducible procedures and abstraction. In one such exploration, student teams design a small Lego structure, which they then encode with a set of descriptive labels: the coordinates, shape, orientation, and color of each brick. Students proceed to represent those labels with a binary code, again of their own design. With the binary codes and their explanations, the teams cover their designs and swap their descriptions. Each pair then strives to build the structure specified by their classmates' codes. The activity can have *telephone*-like results as amusing differences arise between original and rebuilt structures. In this case, however, students can explicitly track down the source of any errors – and correct them – whether within the encoding, context-building, context-interpretation, or decoding phases of the process. These experiences map immediately to executable software, regardless of architecture or language.

To bring curricular structure to these kinds of activities, MyCS builds on the broad shoulders of Exploring Computer Science, ECS [7]. UCLA's innovative effort to democratize computer science for 10th- through 12th-grade students, ECS has enjoyed a successful deployment in high schools throughout the greater Los Angeles area. For its audience, MyCS shortens the ECS curriculum, emphasizing and expanding ECS's hands-on activities that ask students to create and shape computation. A pilot workshop, supported by Google and the National Science Foundation in summer 2011, brought together teachers of many different grades, CS undergraduates, and CS professors in order to create a first-draft MyCS curriculum [8], based on ECS. Three schools piloted the resulting one-semester offering in the fall of 2011, with an expanded effort slated for spring 2012.

Already, the experiment has demonstrated the many challenges that remain, as well as the rewards for grappling with them. One challenge is political: how should a curriculum like MyCS fit into the well-established technology courses already in existence? Our vision is that MyCS would supplement, not replace, those courses; MyCS can offer insights into computer science *as a discipline* to complement skill-development in specific (and rapidly changing) suites of computational tools. Details differ from school to school. In Claremont, for instance, one section of the district-required technology course has dedicated a day per week to MyCS materials.

These "Computer Science Fridays" have become a course highlight, popular with teacher and students alike.

A second challenge is communication: maintaining a high-bandwidth feedback loop among middle- and high-school teachers, college CS students, and instructors. A summer workshop [8] is only a starting point: we must cultivate resources such as student-staffed homework hotlines, out-of-the-box student demonstration days, and equipment-lending libraries that support curricular activities like ECS's computer-disassembly "scavenger hunt." Harvey Mudd College's experiments with these collaborations show that, when resources are available, teachers and students at the middle-school level express a demand for computer science comparable to the recession-proof pull of CS's artifacts.

Like its students, middle-school computer science is in the early stages of self-definition, weaving among the pros and cons of a bombardment of questions: "Should computational creativity be part of the curriculum?" "Should it be a choice or a requirement?" "How do we balance CS itself with its influential toolkits?" "Does CS detract from -- or can it, perhaps, enhance – middle-school students' journey of choosing who they are?" All of these questions remain open, to be sure, but we believe that efforts like MyCS and The Games Network can help students, teachers, and colleges answer them to the mutual benefit of all of those groups.

References

[1] Bureau of Labor Statistics, Occupational Outlook Handbook, 2009-2010.
[2] Margolis, J. et al. *Stuck in the Shallow End*. MIT Press, 2008.
[3] NSF's Broadening Participation award #1042472.
[4] FactoryBalls, at http://www.bartbonte.com/factoryballs/
[5] LightBot, at http://armorgames.com/play/2205/light-bot
[6] Parlante, N. et al. Nifty assignments. In *Proceedings, SIGCSE '10*. ACM, New York, NY, USA, 478-479, 2010.
[7] ECS's home page, at http://www.exploringcs.org/
[8] MyCS's home page, at http://www.cs.hmc.edu/~cs5grad/MyCS