LEARNINGconnections

Coding in the Elementary School Classroom

My goal as an educator is to lead my students to new ways of thinking, understanding, and communicating in a digital world. I know that if I teach my students basic coding, they will be on their way to learning those skills and have a good time doing it. But I didn't expect this breathless comment to spill out of the mouth of one of my third graders after I taught her to use Scratch. "I just started coding, and it is like a new world!" she said. "It is amazing because you get to almost go into someone else's brain and teach them based on how they think."

When I heard those words, I realized my students were learning much more than just HTML. Since starting my journey with the Coding is Common to the Core initiative in the Paradise Valley Unified School District in Phoenix, Arizona, I have looked for innovative ways to implement the ISTE Standards for Students (formerly the NETS•S) and the Common Core State Standards (CCSS) while teaching my students to code.

I started the year with Scratch, giving my students design-based challenges to familiarize them with the basic blocks. For example, after a lesson on idioms, I had the students create an animation in Scratch to illustrate an idiom. They unleashed their imagination while demonstrating understanding and mastery of a concept.

Next, they would reflect on and analyze the thinking they used while programming. Here's how one of my fourth graders described his experience using Scratch:

> I am working on an idiom project —a taste of your own medicine. It was pretty challenging because most of the sprites they had were not what I needed so I had to create my own and my own background and fix things up and reprogram it to make it better. I had to make my mad scientist disappear. When I was trying to do that I tried to make my own block and typed in "disappear," but when I clicked on that for him, he just stayed there. Then I had to make him move 240 steps so you didn't see him anymore, but you could still see his hand. I'm trying to make him "disappear," but it's hard because I tried to get the background to go over him, but that didn't work either (so)… I'm looking around Scratch to see how I can program it to set itself up all over again.

Real-World Debugging Experience

Next, I gave my students a series of debugging challenges, beginning with some created by MIT's Scratch Ed team. This gave them exposure to the process of solving glitches, and it simulated the experiences of computer scientists working in the real world. Being put in a situation where they were "stuck" spontaneously led my students to a paired-programming approach that involved communication and collaboration with their peers. Instead of struggling independently on complex tasks, they asked one another for ideas and leads to solve the challenges.

For a performance-based assessment, I started a studio in Scratch called Mak's Debuggers (although one student recently suggested I change it to Mak's Exterminators), where my students created and submitted their own debug-it programs for their classmates.

Scratch also allows students to tinker with existing projects. Using the remix feature, they can redesign another student's project, adding their own touches and personality while learning from others in an informal but authentic way.

As the students worked on their Scratch projects, I also taught some basic computer science concepts, such as binary numbers, so that students would better understand how computers work and refine their computational thinking. Beginning with a mini-lesson on binary numbers, I asked them to connect what they learned to the eight standards for mathematical practice in the CCSS.

By Janice Mak

A student created this animation in Scratch to illustrate the idiom "Hold your horses!"

Here's how one of the third graders responded:

> In binary numbers, I use math practice number 6 (attend to precision) when doing code. Binary numbers are the digits 0 and 1. Code for binary numbers is doubling one number starting with one and writing them in 0 or 1. You see, trying binary numbers with cards makes it easier. We did it in class once. First, our teacher made us cards that say 1, 2, 4, 8, and 16. Cards that are showing are 1. Cards that are not are 0. Then our teacher, Mrs. Mak, said a number and we had to put the cards face-down if they did not equal the number Mrs. Mak said. For example, if all the cards were showing and we had to make the number 12, then we would flip over 16, 2, and 1 so that only numbers that are showing are 8 and 4 because if you add them together, you get 12. So in binary, it would be 01100. You have to attend to precision when doing binary numbers.

For more ideas on teaching about computer science "unplugged," I have found csunplugged.org and learn.code.org to be outstanding resources.

Robotics Teach Problem Solving

NXT Mindstorms, Lego's programmable robot kits, is another powerful tool that fosters critical thinking, problem solving, and collaboration. I began by setting up a series of challenges for my students, such as having the robot travel accurately in a straight line from a starting point to the finish line. We then moved on to more complex robot obstacle courses involving mazes and figure eights.

It was amazing to see my students ask one another questions, engage in mathematical reasoning, and even argue over the best way to complete the challenge. They pulled out rulers, measuring tapes, and calculators. Then they debated the merits of programming the robot using number of rotations, time in number of seconds, or number of degrees.

The most valuable discussions occurred when students did not experience immediate success. Rather than step in with the answer, I asked more questions to ignite their thinking. For example, when they saw that the robot traveled only about half the length of the course, they began to investigate by tracing back through their steps to see where their programming had gone awry.

They discovered the importance of being consistent with the systems of measurement. Instead of measuring the circumference of the robot's tires in inches, as they had done with the length of the course, they measured it in centimeters. These learning experiences make indelible impressions on my students as they persevered in solving problems collaboratively.

Meeting the ISTE Standards

Here's how this project met the ISTE Standards:

Standard 1: Creativity and Innovation. Students used coding on Scratch and other platforms to create a variety of projects that encapsulate classroom learning.

Standard 2: Communication and Collaboration. Students blog about experiences with coding, reflect on challenges and successes, and collaborate with one another to solve debugging and robotics challenges in pairs.

Standard 4: Critical Thinking, Problem Solving, and Decision Making. Students are engaged in authentic problem solving through a variety of programming "puzzles" that require students to be analytical decision makers to arrive at viable solutions. They need to be able to deconstruct a problem to understand its component parts and be able to identify patterns as well as cause-and-effect relationships.

Computational thinking is not just for an elite group of students who enroll as computer science majors in universities. I have seen every one of my students, regardless of gender or background, benefit from the thinking and collaboration that is inherent in what we do in the classroom. It flows seamlessly into the CCSS and ISTE Standards (see "Meeting the ISTE Standards"). The ultimate goal, after all, is to cultivate critical thinkers who can deconstruct problems into their component parts to solve them effectively, communicate their findings, and justify their reasoning with evidence.

In the words of one of my fourth graders:

> Coding is like doing crossword puzzles because one word in the wrong place affects the word that is going down or across that word. That relates to coding because if you do Scratch and put the wrong block, when you want to do something else, it turns into a disaster. When you do a crossword puzzle, at first it might be just experimenting and trying to fit the words in the squares. Same with coding. You have to experiment to see if the cat moves the way you want or not. At first, the crossword puzzle might look different or weird, but it turns out cool. With coding, it might look weird at first, but when it is finished, it is awesome!

Acknowledgments

I would like to thank my technology director Jeff Billings for his leadership and support in pioneering education into the third millennium.

—Janice Mak teaches third and fourth graders in Phoenix, Arizona, and is working on her doctorate in global science education. Her students inspire her to integrate critical thinking and creative problem solving through the meaningful use of educational technology. You can read her blog at supercodingpower.blogspot.com.

Tip

You've Got Mail—from Yourself—and an Easy Way to Track Goals

Problem: I'm working with my students on setting and meeting goals, but it can be difficult for them to stay on track. And I admit it can be difficult for me too. Is there a tool that can help with this?

Here's a solution: Keeping on top of goals can be a challenge for students and teachers alike. Regardless of whether your goals are personal or academic, it helps to have a system in place for keeping you headed in the right direction and reflecting on how you're doing. The website FutureMe.org is set up to allow you to send yourself emails in the future for the purpose of documenting your thoughts and feelings at a particular point in time. But you can also use this site to email yourself a pep talk or a reminder for goals that you've set. It's such a simple concept, and it can be a powerful way to check in with yourself.

I have used this site to deliver reminders at key times in order to get things done, and also as a self-checkin to ensure that I am meeting the goals I have set. Say, for example, that I have a goal to contact parents more regularly to share good news. That's not a task that requires a lot of to-dos, but it is something that I need to be reminded about regularly to make it a habit.

This functionality can also serve as a sort of IEP (individualized education program) for students. They can email their goals to themselves, and when the email arrives, they can respond to themselves with an explanation of how those goals were met or what they have yet to do to meet them.

In short, FutureMe.org is a great site with no learning curve. You can use it from any device to quickly take reflective and progressive action to make yourself a better teacher and learner.

Adam Bellow created eduClipper.net and eduTecher.net. He is a former English teacher, technology training specialist, and technology director. He delivered the closing keynote at the ISTE 2013. Follow him on Twitter @adambellow.

Have a tip to share? Email tips to senior editor Diana Fingal at dfingal@iste.org.