How Do You Introduce Computing in an Engaging Way?

Experience with computers between boys and girls has equalized, but boys continue to have greater knowledge of computing and programming *concepts* than do girls. Not so in biology, chemistry, or mathematics, where both boys and girls are encouraged to provide evidence of proficiency when they apply to college. High school study of these subjects familiarizes students with the content and concepts, and gives them confidence. The result is that women's undergraduate completion rates have neared parity in these disciplines.

Because IT study is elective in almost all K-12 schools, developing relevant and interesting assignments that appeal to a broader audience is recommended for:

- fostering a climate where the non-predisposed can belong both academically and socially
- recruiting students who are not predisposed to pursuing computing
- exposing fundamental computing concepts to inexperienced learners

Is prior programming experience required for students to be successful in an IT program? Most undergraduate departments would say no. That is, experience with programming is not the same as expertise in problem-solving, algorithmic thinking, or computing theory. Yet research shows that introductory courses and their embedded assignments work better for students who have *some* experience with programming.

Research also shows that students with programming experience are more confident and more successful in introductory courses than are their inexperienced peers. Students with lower grades or less confidence are less likely to persist in an IT major. What is more, when introductory courses have limited opportunities for talking to other students (e.g., collaborative learning), inexperienced students have little information on which to judge whether they belong academically in the major. Hence more women than men switch out of IT majors (most often to other sciences or mathematics).

MAKING IT MEANINGFUL

Educational researchers emphasize the importance of linking educational materials and curricular programs to students' existing knowledge and experiences. When class syllabi list topics and assignments that focus on unfamiliar concepts with limited, if any, relationship to a student's life experience or interests, she or he is unlikely to take that class. High school curricula contribute to low enrollments in college computing because, under the existing educational policy of election, computing is rarely required in secondary schools. This means that students are likely to have a narrow and inaccurate view of what IT study involves, what careers are possible, or what kind of people "do" IT. Given the very small proportion of females who study computing in high school, females are less likely to choose IT in college.

The challenge to educators at all levels is to develop engaging assignments and curriculum that can appeal to a variety of students with different learning styles, interests, socio-cultural backgrounds, and abilities, while maintaining the rigor of the discipline. Putting the concepts of computing in appealing contexts and building on existing competence can both reduce entry barriers and level the playing field for those with limited experience.

Creative assignments that teach algorithmic thinking while also calling on students' existing knowledge or interests, may serve to both recruit and retain students. When experienced and inexperienced students use non-computer-based assignments to learn computing concepts, they quickly realize that their peers with programming experience are not necessarily better at algorithmic thinking, just more experienced with programming. Building confidence through relevant and interesting assignments is a promising practice for motivating student enrollment and retention.

RESOURCES

Lecia Barker and William Aspray, "The State of Research on Pre-College Experiences of Girls with Information Technology." In McGrath Cohoon, J. and W. Aspray (Eds.) *Women and Information Technology: Research on the Reasons for Under-Representation*. Cambridge, MA: MIT Press, 2006.

Joanne McGrath Cohoon and William Aspray, "A Critical Review of the Research on Women's Participation in Postsecondary Computing Education." In McGrath Cohoon, J. and W. Aspray (Eds.) *Women and Information Technology: Research on the Reasons for Under-Representation*. Cambridge, MA: MIT Press, 2006.

ncwit.org **Authors | Lecia Barker and J. McGrath Cohoon**

NCWIT Strategic Partners: NSF, Microsoft, and Bank of America; NCWIT Investment Partners: Avaya, Pfizer, Merck, Turner Broadcasting System, Inc., and AT&T

Storytelling (Case Study 1)

An Engaging Way to Introduce Computing

K-12 Education | **Undergraduate**

Learning to program with Alice is an innovative approach to teaching and learning introductory programming and other computing concepts. Beginning students, including middle and high school students and undergraduates, use the Alice programming environment to populate a virtual world with 3D models of objects (e.g., people, animals, vehicles, and more).

Formal assessment of this approach has been performed in several college and university environments. In published results, Alice is reported to be a successful intervention technique for students who have less mathematics preparation and/or programming experience. When these students used Alice first, their average grade was a 3.0 GPA in CS1 – comparable to the grades of their peers with greater mathematics backgrounds and prior programming experience. Without Alice, these "at-risk" students earned an average 1.2 GPA in CS1.

Implementing the approach is supported by an extensive collection of curriculum and instructional materials. Sample course calendars, presentation notes, labs, projects, and test banks are included. An online community and Alice newsletter provide quick and easy access to online assistance.

TWO MAJOR LEARNING OUTCOMES FROM LEARNING TO PROGRAM WITH ALICE

1. Fundamental Concepts of Programming

Alice allows students to immediately visualize how their animation programs run, fostering understanding of the relationship between the programming statements and constructs and the behavior of their animations. Students learn the basics of computing by manipulating objects that are actors and scenes in a virtual world of their own creation.

2. Problem Solving and Logical Thinking

The traditional steps of problem-solving are applied through storytelling or task performance. Students use animation storyboards as design tools, creating a sequence of steps (in pseudocode) that they eventually implement, test, and revise. Students learn if-else and Boolean logic by creating interactive animations and simple games.

RESOURCES

Alice software: http://www.alice.org
Curriculum and instructional materials, workshops: http://www.aliceprogramming.net

Moskal, B., Cooper, S. & Lurie, D. (2004, March). *Evaluating the Effectiveness of a New Instructional Approach*. Paper presented at the meeting of SIGCSE 2004, Norfolk, VA.

The Alice Team: Randy Pausch (developer), Wanda Dann, Stephen Cooper, and Don Slater
Case Study Contributor: Wanda Dann, wpdann@ithaca.edu

NCWIT offers practices for increasing and benefiting from gender diversity in IT at the K-12, undergraduate, graduate, and career levels.
This case study describes a research-inspired practice that may need further evaluation. Try it, and let us know your results.

ncwit.org Authors | Lecia Barker and J. McGrath Cohoon

NCWIT Investment Partners: National Science Foundation, Avaya, Microsoft, Pfizer, and Bank of America

Unplugged (Case Study 2)
An Engaging Way to Introduce Computing

K-12 Education | Undergraduate

Computing is often a mystery: While people may know how to use computers, they rarely know what makes computers work. "CS Unplugged" uncovers the mystery by exposing students to computer science concepts, such as the nature of data or how data is sorted, but without the computer. The activities in "CS Unplugged" help to shatter the image of computing as long, lonely hours in front of an LCD screen by exposing learners to the kind of reasoning needed for inventing the next great ideas in computing.

"CS Unplugged" activities engage students in learning computer science concepts using hands-on activities. The activity described here, "Sorting Network," illustrates the structures used in parallel sorting networks, exposing learners to sorting, parallelism, and binary comparison through active, kinetic learning. In teams of six, students compare numbers (small or large) and follow simple logic.

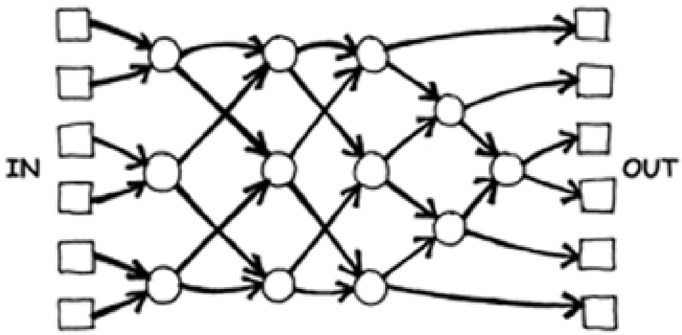
HOW DO YOU DO IT?

Start by drawing the layout to the right on the ground, using chalk on a pavement, masking tape for indoor surfaces, or electrician's tape on a tarpaulin. Each student on the team holds a card with a number on it (for the first time, use the numbers from 1 to 6). The goal is to get the numbers sorted into order.

Each student stands on one of the squares on the "in" side of the diagram. Students follow the arrow to step onto the first circle, where they meet another student and compare numbers. The student with the smaller number follows the arrow out on their left, while the student with the larger number follows the arrow out on their right.

Students continue following the arrows to each circle as another student steps to the circle, each time comparing numbers. The smaller always goes left and the larger goes right. Eventually they will reach the "out" side in sorted order. (The full lesson plan, "Beat the Clock: Sorting Networks" can be found on the website described below.)

The exercise can be extended in a number of ways. For example, students could be timed to discover how quickly they can complete the sorting. For this, use larger numbers so it is hard to see where you are supposed to end up. And there are many questions to ponder: What if the smaller one goes to the right each time? How would you design a layout for sorting three numbers? Thirty numbers? Does it work backwards? Can you design a smaller layout to find the smallest number?



WHAT COMPUTING CONCEPTS DO STUDENTS LEARN?

When three pairs of students are comparing numbers at the same time, it takes much less time than comparing only one pair of numbers at a time. This "Sorting Network" demonstrates parallel computation, one of many ways that computer scientists have devised to sort data quickly. Instructors tell students that they have just learned about the computing concepts behind computer applications with which they are familiar, such as alphabetical lists of files, etc.

Initial evaluations of sessions involving this activity and others show that children gain a better appreciation of what Computer Science is about, and girls in particular respond positively to the logic and problem solving. More detailed international evaluations are underway.

RESOURCES

For more information on this activity and a pdf of the complete teacher's version, see http://csunplugged.com.
Please see NCWIT's Computer Science-in-a-Box: Unplug Your Curriculum, http://ncwit.org/resources.res.box.cs.html.

Case Study Contributor: Dr. Tim Bell, tim.bell@canterbury.ac.nz

Authors | Lecia Barker and J. McGrath Cohoon

NCWIT Investment Partners: National Science Foundation, Avaya, Microsoft, Pfizer, and Bank of America

Meet Them Where They Are (Case Study 3)

An Engaging Way to Introduce Computing

K-12 Education

To educate girls about information technology and potential careers in the field, the Girl Scouts, Hornets' Nest Council in North Carolina runs a Girls are I.T. program, sponsored by a National Science Foundation research grant. Through its two key components – an educational website and a mobile technology bus—the program aims to increase access to technology by meeting girls where they are, both geographically and experientially. Since its inception, the bus has reached over 5,200 girls, many of whom are in rural locations with limited exposure to technical experiences.

The Mobile Technology Classroom features 12 workstations designed to showcase technology and technology careers in ways that tend to appeal to many girls. For example, four hands-on activities explore how technology helps people live better lives – (see detailed descriptions below). In each activity, girls are encouraged to imagine the future of technology based on the program component they've just completed. The girls then upload their ideas to www.girlsareit.org, a website that presents the history of technology and highlights women who have exciting IT careers.

EXAMPLES OF HANDS-ON ACTIVITIES OFFERED ON THE MOBILE TECHNOLOGY CLASSROOM

- **Nanotechnology**
  Girls explore how a nanodevice is built, what "nano" means, and how tiny nanodevices will be used in the future. Using laptop computers, they create four different nanodevices – light emitters, oscillators, mesh fabric, and DNA Scaffold.

- **Assistive Technology**
  Girls "see" and "talk" using computer software and hardware designed to assist the visually-, hearing- and speech-impaired. They begin to understand how technology aids those with disabilities, software's limitations in this area, and the need for continued progress.

- **HTML Webpage Design**
  Girls learn to create and edit a web page with HTML code. They then design a web page for their troop or for a local non-profit in need of a website.

- **Wireless Sensors**
  Girls operate an explorebot, similar to the Mars Rover. From their laptops, they see what the robot sees and maneuver through various terrains located in the back of the bus, learning how technology enables us to go places that may not be safe for humans. The missions include New Species Discovery, Earthquake Search and Rescue, and Shipwreck in the South Seas.

While this unique program might be difficult to replicate, educators can increase girls' access to IT through curriculum that adapts several key components:

- Use hands-on activities that solve real-life problems and/or call on girls existing knowledge and interests.
- Build in strategies for reaching girls with limited access to technology (e.g. remove geographical or other logistical barriers).
- Develop all-girl activities that are collaborative.

RESOURCES

For more information about this program see the Girls are I.T. website, www.girlsareit.org or contact Heather Doyle, hdoyle@hngirlscouts.org.
For more information about other Girl Scouts of the USA technology programs see www.GirlsGoTech.org.

Case Study Contributors: Girl Scouts, Hornets' Nest Council, and Girl Scouts of the USA

NCWIT offers practices for increasing and benefiting from gender diversity in IT at the K-12, undergraduate, graduate, and career levels.
This case study describes a research-inspired practice that may need further evaluation. Try it, and let us know your results.

ncwit.org **Authors | Lecia Barker and J. McGrath Cohoon**

NCWIT Investment Partners: National Science Foundation, Avaya, Microsoft, Pfizer, and Bank of America

Teaching Programming and Language Concepts Using LEGOs® (Case Study 4)

An Engaging Way to Introduce Computing

K-12 Education | Undergraduate

Teaching Programming and Language Concepts Using LEGOs® is an innovative method for using LEGO® bricks to teach programming and other computing concepts to middle and high school students as well as to undergraduate freshmen in introductory computer science classes. In this assignment, individual LEGO® bricks are used to express a special-purpose programming language, integrating tactile and kinesthetic elements into the learning experience and helping to make abstract concepts more concrete.

The method has two main learning outcomes:

- *Language Specification* – The goal of the assignment is to be able to build LEGO® creations on a standard grid base plate. To build a creation brick by brick, it is necessary to specify the type of brick, its color, and its location on the base plate. The combination of colors and positions indicates a specific action. Students learn to develop and state a set of sequenced instructions, a critical skill for programming.
- *Bridge to Other Abstract Concepts* – This teaching method has been used to teach a variety of topics to different audiences, including freshman CS majors, K-12 students, and K-12 teachers. In each group the use of the language provides opportunities to discuss more abstract concepts, including CPU Simulation, Writing and Testing Programs, and Extending the Programming Language.

Informal assessment of these exercises has been positive. Participants enjoy working with LEGOs® as a means of exploring programming and processing concepts. In one case, 100% of the freshmen taking an introductory computer science course were engaged in the exercise: an unprecedented event, according to the instructor. In fact, 75% of participants volunteered positive comments about the LEGO® exercise in end-of-semester course evaluations. Currently, the exploration of partnerships with assessment specialists to help develop quantitative aptitude progress methods is underway.

The use of LEGOs® may "level the playing field." Students both with and without computer programming experience struggle with the assignment. When told that they have learned a central concept of computer programming, inexperienced students feel both successful and confident, in spite of not using the computer to "program." Because the LEGO® approach does not directly involve technology that can be seen as intimidating to students, this approach shows promise for increasing participation of diverse audiences.

Implementing this program is strikingly simple because it only requires LEGO® pieces and a basic understanding of how LEGOs® fit together. A base plate and different-shaped LEGO® blocks are easily acquired and a single base plate is sufficient for each participating student or team. Clear language specifications for each type and combination of LEGOs® should be established prior to the onset of the exercise to avoid confusion later; however, a knowledgeable instructor or moderator can quite easily provide some instruction and guidance for each exercise to each participating group.

RESOURCES

Computer Science Teachers Association Resource Site: http://csta.acm.org/Resources/Resources.html
Cynthia Hood and Dennis Hood, "*Teaching Programming and Language Concepts Using LEGOs®*." Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, 2005. Available from the ACM Digital Library.
The Educator's Reference Desk Lesson Plans in Computer Science: http://www.eduref.org/cgi-bin/lessons.cgi/Computer_Science
For free offline activities for teaching computing concepts, try "Computer Science Unplugged," located at: http://unplugged.com

NCWIT offers practices for increasing and benefiting from gender diversity in IT at the K-12, undergraduate, graduate, and career levels.

This case study describes a research-inspired practice that may need further evaluation. Try it, and let us know your results.

ncwit.org Authors | Lecia Barker and J. McGrath Cohoon

NCWIT Investment Partners: National Science Foundation, Avaya, Microsoft, Pfizer, and Bank of America

National Center for Women & Information Technology
PROMISING PRACTICES

Snap, Create, and Share with Scratch (Case Study 5)
An Engaging Way to Introduce Computing

K-12 Education | Undergraduate

WHAT MAKES SCRATCH SO ACCESSIBLE TO NOVICES?
Scratch is a free "media rich programming environment" in which novice programmers can quickly express their creativity while learning computational thinking. Developed by the Lifelong Kindergarten group at the MIT Media Lab, Scratch is used at both the K-12 and undergraduate levels to reduce the barriers created by a programming language's abstract syntactic and semantic rules. Instead, students "snap" together several categories of "building blocks" (e.g., statements, loops, variables) to quickly generate animations, games, and art. The building blocks only snap together if they are syntactically appropriate. Students can work both individually and in small teams.

Scratch is effective as a learning tool because it incorporates several effective practices: it uses hands-on, active learning; it is visually appealing; it allows users to express their own creativity and to build on their own experiences; it gives immediate, understandable feedback; and it allows users to avoid syntax errors without focusing on minutiae, freeing them to focus on processes and concepts.

WHAT COMPUTING CONCEPTS DO STUDENTS LEARN USING SCRATCH?
Educational researchers at MIT Media Lab and University of California-Los Angeles studied Scratch scripts used in 425 programming projects created by 80 girls and boys ages 8-18 to determine which programming concepts they learned. The researchers found that all these projects used sequential execution and 90 percent used threads (multiple scripts running in parallel). About half of the projects included loops and user interaction and about a quarter included conditional statements and synchronization. A smaller set included Boolean logic, random numbers, and variables. The projects tended to include more of these concepts the longer students used Scratch.

ASSESSMENT OF SCRATCH AS TRANSITIONAL TOOL
Although Scratch was originally designed for ages 8-16, several universities are using Scratch in undergraduate courses, including Harvard, Rutgers, and College of New Jersey. Harvard researchers conducted a small classroom-based study on the use of Scratch for entry-level programming at the undergraduate level. The researchers used surveys to gather information about students' prior programming experience, their experiences with Scratch, and the ease of the post-Scratch transition into Java. Most students felt that Scratch positively influenced their ability to learn Java. Of the students who felt Scratch had no influence, all had prior programming experience.

SCRATCH COMMUNITY AND EDUCATOR SUPPORT
The makers of Scratch created a social network of sorts within the Scratch site. Users can post their project and remix others' projects; they can also discuss issues on the Scratch forum in several languages. More than 200,000 projects have been posted on the Scratch web site by novice programmers from around the world. The "top-loved" project has more than 23,000 views and 635 votes of "Love It." More than 26,000 projects have been remixed by other Scratch developers. The website also has a section especially for educators, with videos and other resources for getting started and ongoing support. Find out more here: http://scratch.mit.edu/.

SCRATCH-BASED ONLINE EDUCATIONAL COMMUNITIES

A number of educators have begun posting lesson plans and support materials to share with other teachers around the world. For example, Karen Randall, an elementary school teacher in Minnesota, has created a wiki (at http://wiki.classroom20.com/Scratch) where people can share Scratch materials. MIT Media Lab doctoral student Karen Brennan is creating an online community called ScratchEd, where educators will be able to share ideas, experiences, and curriculum plans with one another (to be launched later this year). Here are other sources of Scratch lesson plans and materials:

- http://nebomusic.net/scratch.html
- http://coweb.cc.gatech.edu/ice-gt/446
- http://www.learnscratch.org/
- http://www.lero.ie/educationoutreach/secondlevel/scratchlessonplans.html

RESOURCES

Malan, D.J., & Leitner, H.H. (2007). Scratch for budding computer scientists. *SIGCSE Bulletin (39)* 1, 223-227.

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with Scratch. *SIGCSE Bulletin (40)* 1, 367-371.

NCWIT offers practices for increasing and benefiting from gender diversity in IT at the K-12, undergraduate, graduate, and career levels.

This case study describes a research-inspired practice that may need further evaluation. Try it, and let us know your results.

ncwit.org

Author | Lecia Barker

NCWIT Investment Partners: National Science Foundation, Avaya, Microsoft, Pfizer, and Bank of America

Scalable Game Design for Middle School (Case Study 6)

An Engaging Way to Introduce Computing

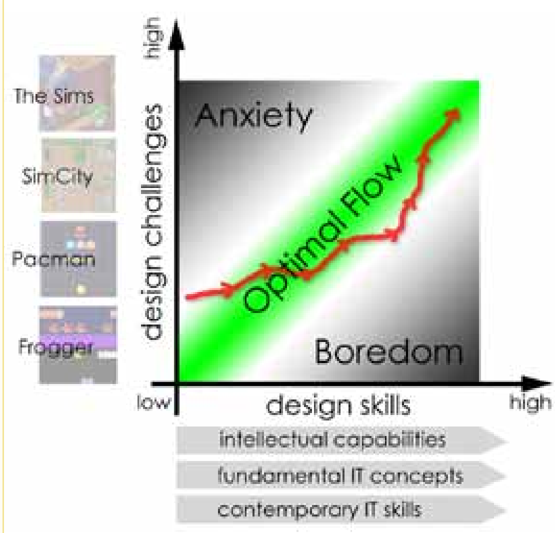
K-12 Education

It's not so easy to build and design a working video game, but a well-crafted learning environment makes it possible and interesting for many students. The middle school computing curriculum in Colorado's Boulder Valley School District (BVSD) uses Scalable Game Design to introduce computer programming in engaging ways and helps students develop IT skills aligned with ISTE'S National Educational Technology Standard of Creativity and Innovation.

In the very first lesson, students make their own Frogger-like game to publish on the web. Over the course of a one- to two-month module, students learn more sophisticated topics in order to create increasingly complex games and computational science applications. According to Len Scrogan, Director of Instructional Technology for BVSD, the results of the BVSD implementation include motivated students, engaged teachers, and excited parents.

THE THEORY BEHIND THE CURRICULUM

Scalable Game Design uses the premise that learning is most successful when students engage in tasks that are difficult enough to be interesting but not so difficult that they become frustrating. The psychological notion of "flow" can help manage this tension. This notion suggests that students learn best when in "optimal flow", where design challenges match design skills and anxiety is relatively low. In this stage, students are highly receptive to guided learning even if the topic appears too difficult. Scaffolding lessons this way helps students progress from simple arcade games to games that require sophisticated artificial intelligence.



Flow balances design challenges with design skills outlined by the National Academies of Science's Fluency with Information Technology (FIT).

EVALUATION: BOTH GIRLS AND BOYS MOTIVATED TO PROGRAM

AgentSheets, a scalable game design product, has been evaluated in two small studies for its effectiveness in motivating middle-school students to learn programming. In a summer elective course, 36 middle school boys and girls used AgentSheets to experiment with programming concepts and create games or animations. Interestingly, while most students expressed a desire to continue with AgentSheets, students with low-technology experience expressed a slightly stronger desire than those with high-technology experience. By the end of the course, girls and boys also expressed similar levels of desire to continue using AgentSheets. Another study using AgentCubes, a 3D simulation and programming tool developed by the creators of AgentSheets, found that all students were able to create a working 3D game in less than five hours. This study was conducted in an afterschool program that included girls, inner-city low-income students, and students in a U.S. technology hub. All students performed well in developing and troubleshooting their creations.

CHARACTERISTICS OF A SUCCESSFUL EDUCATIONAL PROGRAMMING ENVIRONMENT

- Accessible to students without prior programming experience
- Simple enough to make a working game in three hours or less
- Powerful enough to allow implementation of sophisticated artificial intelligence algorithms
- Works for game and computational science applications
- Transitions to traditional programming such as Java

So far, one product on the market combines these ingredients, AgentSheets. Originally developed at the University of Colorado, AgentSheets is available as a ten day free trial at www.agentsheets.com.

RESOURCES

For more information on AgentSheets and related resources, see www.agentsheets.com. AgentSheets is funded by NSF.

Ioannidou, A., Repenning, A. and Webb, D. (2008). *Using Scalable Game Design to Promote 3D Fluency: Assessing the AgentCubes Incremental 3D End-User Development Framework*. Paper resented at the 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, Herrsching am Ammersee, Germany.
Repenning, A, & Ioannidou, A. (2008). Broadening participation through scalable game design. *SIGCSE Bulletin (40)*1, 305-309.
Walter, S.E., Forssell, K, Barron, B, & Martin, C. (2007). Continuing motivation for game design. *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, 2735-2740.

Case Study Contributors: Alexander Repenning, Alexander.Repenning@colorado.edu, and Len Scrogan, len.scrogan@bvsd.org.

Authors | Catherine Ashcraft and Stephanie Hamilton

NCWIT Investment Partners: National Science Foundation, Avaya, Microsoft, Pfizer, and Bank of America

Globaloria: Students Designing Educational Games (Case Study 7)

An Engaging Way to Introduce Computing

K-12 Education

Connecting computing to social issues and "real-world" problems is important for increasing all students' interest, but especially girls' interest, in computing education and careers. Globaloria, created in 2006 by the World Wide Workshop, builds on this principle by involving students in collaborative teams that create video games around important educational and social issues. The program, operating in several states, is the country's largest social learning network of schools and community centers using a game design curriculum to develop students' digital literacies, computing knowledge, and global citizenship skills. Generally, students' participation in Globaloria ranges from three to five times per week, for 60-90 minutes per session, over two semesters. In Austin, Texas, one school has implemented Globaloria into its core curriculum for all students from 6th to 12th grade.

EVIDENCE-BASED PRINCIPLES FOR CURRICULUM. Grounded in constructionist learning theories, Globaloria employs the following key elements necessary for successfully engaging students in introductory computing:

- Using active, hands-on, creative, and open-ended learning activities
- Making explicit connections between computing and social issues
- Promoting collaborative teamwork and opportunities for student interaction
- Allowing ample opportunity for student self-reflection on accomplishments and future learning
- Providing students with opportunities to learn and gain feedback from experts

EVALUATION: INCREASING FEMALE ENROLLMENT AND HOME COMPUTING ACTIVITY. Based on a study of West Virginia classrooms, Globaloria has demonstrated initial successes when it comes to girls' involvement in computing courses. Female enrollment in Globaloria elective classes reached 33% in 2010-2011 and 37% in 2011-2012, exceeding the national average for computing courses (20-25%). In addition, initial pre- and post-test analysis revealed that participation in Globaloria classes increased middle and high school girls' *home* computing activities; importantly, many of these activities involved creating and adapting technologies. For middle school girls, it also decreased the gender gap in "thinking up an idea for a technology project" and "making computer games." These trends in home computing experience may be especially important given research showing that, while girls and boys have similar access to computers and computing at school, girls have less access at home than boys.

The core curriculum is customizable and aligned to language arts, math, and science standards, allowing Globaloria to be implemented in a variety of formats, including after-school programs, electives during the school day, and core academic courses.

To what extent these increases in girls' computing activities translate to increased interest in or plans to pursue computing requires further research, but initial feedback from participating girls is encouraging. Consider the following comments from middle and high school girls in the program:

"I thought this class was only for boys; I thought geeks only used computers, but then I really got to see the neat things about it… It's not like boys get to do this or girls get to do this; it's whoever puts their mind to it, their heart to it, and their time, they can do anything." (middle school female participant)

"Globaloria is… letting girls have an opportunity to have a career and make computer games." (middle school female participant)

"I do consider ourselves innovators…at 15-years-old Globaloria has given me a chance to learn computer science and be a computer scientist." (high school female participant)

Future research is being conducted to determine the pervasiveness of these trends and how girls' interest in and plans to pursue computing education and careers change over time. Because some Globaloria sites involve students in the curriculum over the course of several years, these sites offer particularly promising opportunities for exploring longitudinal trends.

RESOURCES

For more information on Globaloria see www.worldwideworkshop.org/programs/globaloria and www.worldwideworkshop.org/reports

Ashcraft, C., Eger, E., & Friend, M. (2012). Girls in IT: The Facts. National Center for Women & Information Technology.

Wu, Z., Ashcraft, C., DuBow, W., & Reynolds, R. (2012). *Assessing Girls' Interest, Confidence, and Participation in Computing Activities: Results for Globaloria in West Virginia.* National Center for Women & IT.

NCWIT offers practices for increasing and benefiting from gender diversity in IT at the K-12, undergraduate, graduate, and career levels.
This case study describes a research-inspired practice that may need further evaluation. Try it, and let us know your results.

ncwit.org Author | Catherine Ashcraft

NCWIT Strategic Partners: NSF, Microsoft, and Bank of America; NCWIT Investment Partners: Avaya, Pfizer, Merck, Turner Broadcasting System, Inc., and AT&T