National Center for Women & Information Technology

PROMISING PRACTICES

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

How Do You Introduce Computing in an Engaging Way?

with Case Study 1

K-12 Education | Undergraduate

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**