AVAVA/VEER

ALICE *in the Real World*

Tom Parker

A computer application promotes programming knowledge and allows students to create their own worlds through mathematical problem solving.

As a fifth-grade mathematics teacher, I try to create authentic problem-solving activities that connect to the world in which my students live. I discovered a natural connection to my students' real world at a computer camp. A friend introduced me to Alice, a computer application developed at Carnegie Mellon, under the leadership of Randy Pausch. This computer application was designed to create interest in computer programming for middle school girls.

After my third-grade son spent a week using the Alice application at camp, he returned home with a new love for computer programming and problem solving. Although Alice has a storytelling version designed for middle school students, the director of the computer camp was convinced that the original Alice application, designed for high school and college students, could be used with students as young as third grade. After seeing my son's experience, I was convinced that I needed to introduce this application to my fifth-grade students.

GETTING TO KNOW ALICE

What is Alice, and why should you learn about this application? Alice is an innovative computer environment specifically designed to introduce students to programming. It allows students to create a three-dimensional world in which designers pose their own stories and problems to solve while also evaluating which solutions are most effective. In a created computer world that is safe and controlled, students can manipulate objects and characters (object-oriented programming) and immediately see the results of their efforts.

Although Alice is used by high schools and colleges, it is incredibly user-friendly, in part because it relies on a drag-and-drop interface to create a virtual world. If you can drag, drop, and move objects on your computer, you and your students can solve problems and receive immediate feedback. Users can explore variables, algorithms, rates of change, and probability in a three-dimensional world that can be manipulated similar to a video game. In the classroom, students will explore problem solving and math concepts in a real world that they create and control.

What first intrigued me about Alice was how well the programming environment differentiates problem solving. The same application that challenges a ten-year-old can challenge a college freshman studying computer science. Another productive by-product of its use addresses students who often struggle with traditional problem-solving activities: Working in this medium, many students often respond with the most creative and intricate solutions in their programs.

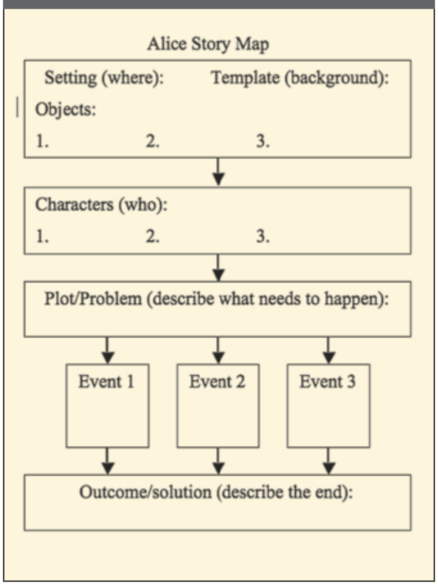
Students experience real ownership and make natural connections when they are the authors of scenarios laden with real-world problems that they must solve. The more students try to solve the initial problems in their stories, the more problems occur, change, and evolve into unforeseen obstacles that ultimately enrich the virtual world that the students create. In the process, students employ and evaluate the effectiveness of many problem-solving strategies: guess and check, make a model, and find a pattern, to name a few.

When I first started using Alice, I considered myself a programming novice. *Learning to Program with Alice—Brief Edition* (Dann, Cooper, and Pausch 2007) gave me the background needed to introduce Alice in my classroom. The book is straightforward, and the information quickly allowed me to explore the application and become comfortable enough to design a math unit for my fifth-grade students. Two colleagues have since read the book and also teach using Alice. The book contains two step-by-step lessons that I completed and recommend; however, the application differentiates naturally to the comfort level of the user. Alice also has tutorials built in, which my students have used to teach me a few things. That is another advantage of Alice: The students and I learn from one another.

Short-term failure was accepted as a necessity for ultimate success.

The key to the unit's success in my classroom was a structure adapted from the book. By building on the idea of a storyboard created by computer programmers, I adapted a traditional language arts story map to align with Alice's structure (see **fig. 1**). The setting, background, and characters align with the actual drag and drop of objects. The problem of each student's story is resolved using Alice's methods and events to control the action of objects. *Methods* are small sets of directions that define how to perform specific tasks, whereas *events* initiate a series of instructions. The larger problem I faced was how to teach 20 fifth graders to use Alice in one week, without any additional classroom help.

Fig. 1 A story map from language arts class can be adapted for an Alice-related project.



DAY 1: MEETING ALICE

The authors of *Learning to Program with Alice* emphasized the importance of breaking commands into smaller pieces (solve a simpler problem), writing and revising instructions (make an organized list), and repeating successfully written program code (follow a pattern). In response, I decided to approach computer programming as a problem-solving unit. At the macro level, the students create a problem for their story and solve it using Alice. Success at the micro level, with students actually programming stories, would require an in-depth process involving trial and error (guess and check) to continually improve their programs. Each student would thus have to learn the fundamentals of programming, laying the foundation for related languages (i.e., Java, the focus of high school AP coursework).

The unit began with a PowerPoint® presentation that quickly summarized the aforementioned fundamentals, as well as introduced the adapted story map. I also broke down the components of the application through two self-made Alice guides and provided a page of directions on which students could take notes. When I wrote the unit, I thought these directions would be instrumental to the unit's success; ultimately, two pillars for success surfaced that defined the success for the unit.

The first pillar for success was a general procedure for all questions and concerns in the classroom: "Ask three, then me." No matter the subject being discussed in my room, my students are taught from day 1 of class to ask their table mates, then neighboring students, before they raise their hands for help with a question. For this unit, I instituted a variation on the jigsaw method, in which each group is expert on a topic and group members disperse to share their knowledge with other groups. The variation for this unit established the become-an-expert strategy: Every time I taught a student how to solve a problem, he or she was then responsible as the resident expert to teach or share the skill to his or her table group.

The second pillar for success reflected the spirit of the late Randy Pausch, who gave the penguin award in his classrooms at Carnegie Mellon for the student who tried something large and failed. (The award is named for the penguin that has to take a risk and be the first to dive into an ocean filled with predators.) I told my students that the only way to get better at working with Alice was to keep trying, even anticipating struggles in the short term. From their failures, they would learn how to create amazing programs. I told them I believed in them and that they had to experiment and fail often to reap the rewards of success. They did not disappoint me.

After the introductory PowerPoint, the students and I created a sample Alice world together. They then replicated the world on their laptops while I explained each step on the SMART Board™. We explored dragging and dropping objects; created methods and events to control objects in their world; and added a few bells and whistles, such as moving objects (changing the vehicle) and working with functions.

Fig. 2 Alice methods consist of drag-and-drop commands, generating natural-language-like programming statements. These commands can include functions, mathematical equations, and references to other objects in the constructed world.

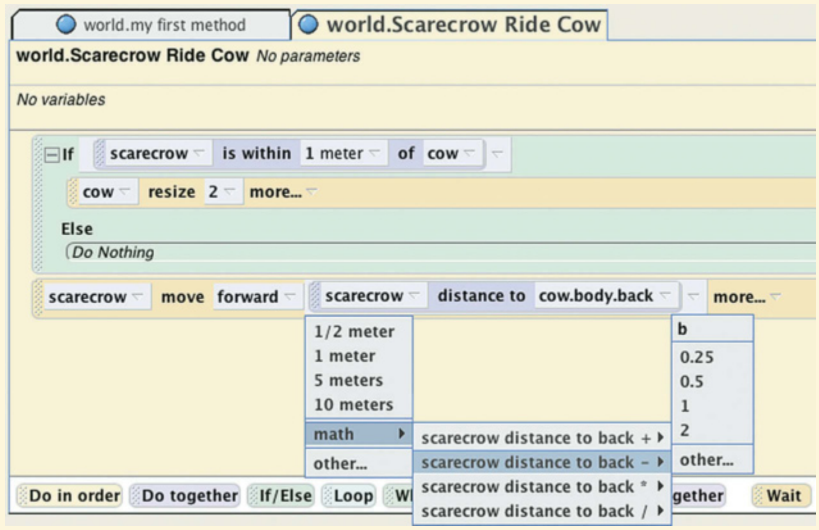
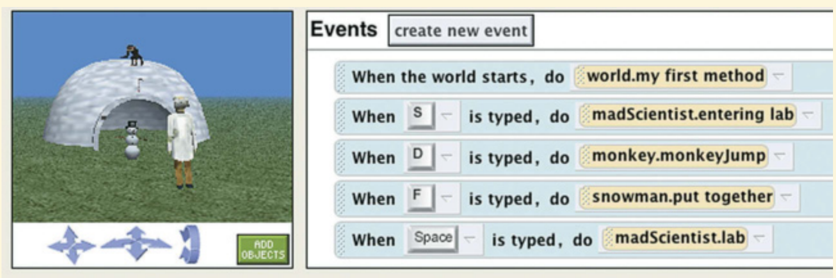


Fig. 3 Alice events allow students to control methods by touching different keys on the keyboard.

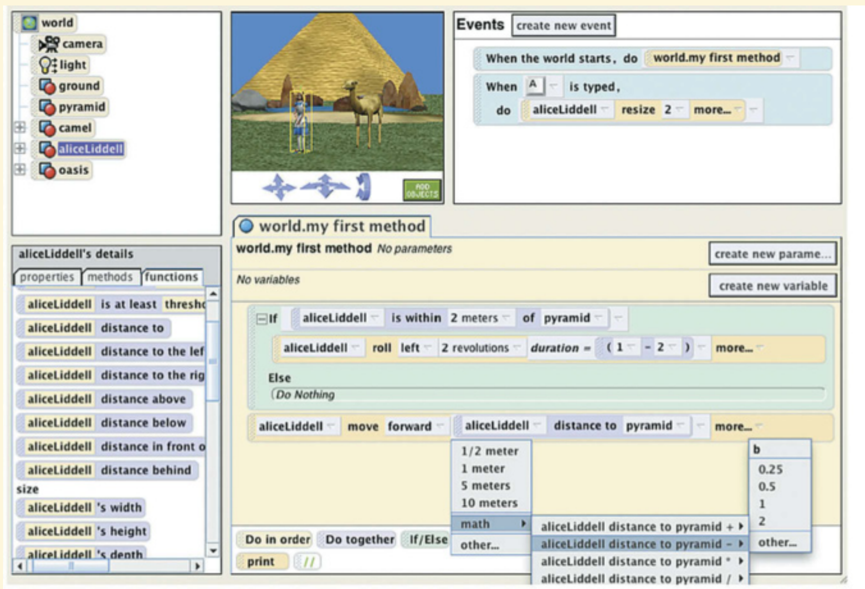


DAY 2: CONTROLLING OBJECTS

The next day focused primarily on methods (see **fig. 2**), since controlling individual objects was the first step in creating an Alice world. Each step in this process was laden with problem solving and mathematics. For example, after a new object was added to the world, we clicked on the object to explore the range of methods (or actions) that each object could make. At first, this seemed simple enough, but the students had to use estimation, guess and check, and patterns to write the directions that would successfully move their object as they wished around the coordinate space.

For example, if a character wanted to ride a horse, the student had to explore which direction the horse would go and what distance. This seemed simple enough, but students soon discovered the problem of estimating distance: Objects rarely ended up exactly where they wanted them. Most students used the guess-and-check strategy. For advanced students, Alice provides actual algorithms, functions, and logic expressions to better control objects. Several of my fifth graders were very capable of using if-then statements with mathematical functions (if x is within 5 meters, then y turns green)

Fig. 4 Advanced students can use equations and functions in their Alice worlds to better manipulate the actions of objects.



to create a scenario in which characters reacted mathematically to the world around them.

Additionally, each character and object had a true center inside its three-dimensional image, and the parts of each object could be ordered to move sequentially or simultaneously with other actions. Much laughter resulted from an object and its parts moving in ridiculous ways. Through constant trial and error, students gradually learned how to use estimation to create the desired method to solve their world's problem. With my guidance, advanced students explored algorithms to gain more control. Students could use three-dimensional coordinates (x, y, z) to pinpoint locations of objects anywhere in the created world.

By the end of this twenty-minute process, the students were chomping at the bit to start. I released them to spend the remaining class time exploring Alice on their own. The next day, students created a storyboard problem on the basis of their initial perception of Alice's capabilities.

DAY 3: CREATING EVENTS

The day began with the introduction of events (see **fig. 3**). *Events* are actions that allow the user to execute an action or method with the touch of a key. For example, a student programmer can create the instructions that make a rabbit jump up and flip before returning to the ground. This action can be triggered by an event. For example, when the user's world is played (started), touching the F key activates the jump and flip of the rabbit.

The students were thrilled that they could control their world similar to a video game but quickly learned that each event creates new problems for their program. However, with a lot of estimation using guess and check, students were able to repeat successful patterns and create a world that ran smoothly and could be controlled by their classmates. At this point, I introduced logic expressions and functions to make objects react to other objects.

STORYBOARDS AND BEYOND

After several introductory lessons, students were given three days to put their problem-solving skills to work. To enact their storyboard's problem, students were required to use at least three characters and three objects as the foundation of their project, or plot. The problem had to be resolved by three events controlled by the keyboard. Some students created problems containing dialogue (speech bubbles) between characters, whereas others created a more mechanical world in which the keyboard controlled a group of planes landing on an aircraft carrier. Although most of the storyboards lacked details, most had enough basic story structure to make me comfortable allowing students to dive into their program.

As students developed new questions about how to work within the Alice application, we explored their problem at their computer, careful to keep my hands off their keyboard. When their problem was solved, I asked them to share what they had learned. Students whose big risks were rewarded with success projected their program on the SMART Board to share with the class.

Differentiated instruction with Alice is also very natural. Actions such as moving objects can be done at a basic level using estimation or more precisely when equations are inserted into their methods (see **fig. 4**). Because the unit was predicated on risk taking using guess-and-check strategies, students were willing to explore more complex programming commands intended for high school and college-level students.

Short-term failure was accepted as a necessity for ultimate success. At the same time, students never felt satisfied with their end product and instead explored more ways to add diversity and control to their Alice worlds. Many students even mastered the process of creating flight-simulator-like controls for multiple objects in their worlds.

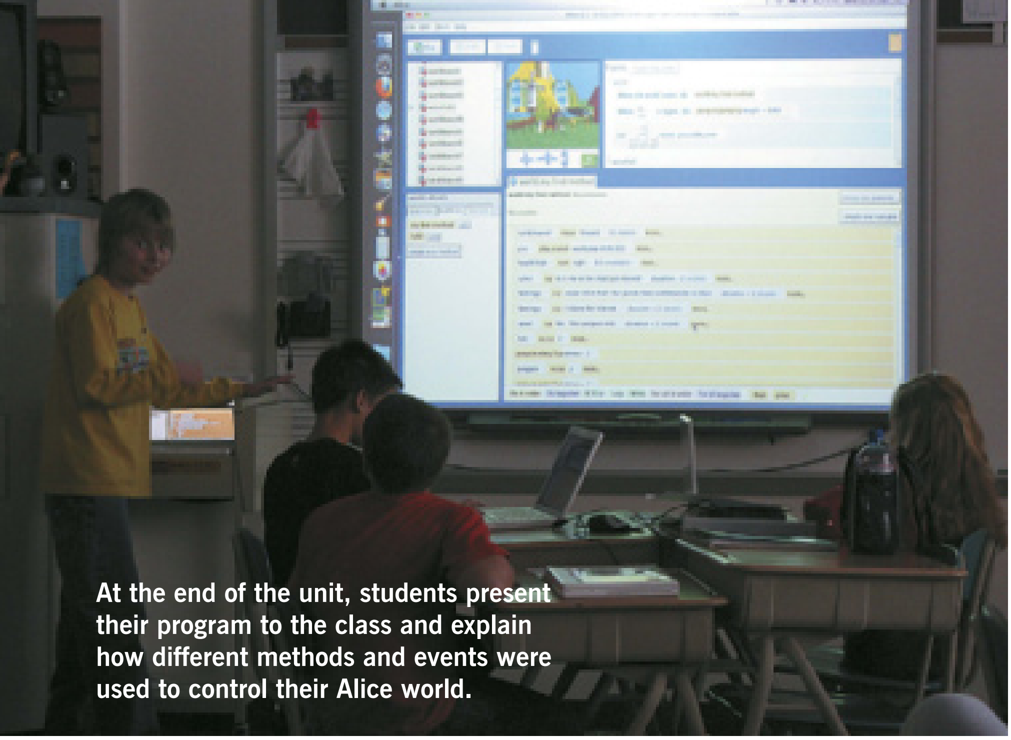
IMPRESSIVE RESULTS

By week's end, student projects exhibited a lot of diversity. Those who were strong in the language arts inserted speech bubbles to tell their stories, whereas students with stronger visual and spatial skills created programs with complex movements of objects. The greatest surprise was in viewing perhaps the most complex and successful programs, which were created by several learning-supported students. Although they struggled with writing, their ideas flowed naturally in Alice's world of drag-and-drop commands and objects. Ironically, those students who were used to high scores and grades tended to struggle the most at first because they were hesitant to try something in which they could flat out fail. After some initial failures while taking small risks, and with some encouragement, these students became comfortable with a process that lacked a correct path for success.

> ***Has technology been neglected as an effective medium for problem solving?***

We ended the unit with a day of presentations in which students projected their finished worlds on the interactive board and demonstrated to their peers how they controlled the objects in their Alice world. I invited three administrators to this event to demonstrate how their investment in technology was reaping dividends. As the students confidently demonstrated worlds full of complex objects that could be controlled on command, the administrators' surprise was obvious. Not knowing what to expect, they witnessed students who had mastered complex problem-solving skills beyond what I think most would consider possible for a diverse fifth-grade classroom. When I requested an upgrade for our aging laptops, my students were rewarded with the next generation of Apple products, which served us well with the next generation of Alice.

This year marked the third round for my Alice unit; it was met with similar enthusiasm and success. During the unit, I read *Out of Our Minds: Learning to be Creative* by Ken Robinson (2001), a book recommended by a local professor. Robinson described the educational system in the West lagging behind a world that



At the end of the unit, students present their program to the class and explain how different methods and events were used to control their Alice world.

TOM PARKER

demands greater innovation and creativity. He argued that students need to be presented with multiple media with which to express their creativity because, although everyone is creative in some way, our educational system lacks authentic opportunities for students to find the medium through which to express that creativity and understanding. Has technology been neglected as an effective medium for problem solving?

AFTER ALICE

I also read "Enhancing K–12 Education with Alice Programming Adventures" (Rodger et al. 2010) in which several Duke University professors drew attention to the decline of math, science, and computer programming interest in schools. In response, they produced a series of summer workshops and a resource website (http://www.cs.duke.edu/csed/alice09/) to help teachers use Alice in their classrooms. Although several disciplines were addressed, the article highlighted creative uses of Alice and math that were not addressed in my article. In particular, several examples were shared in which teachers created their own interactive Alice worlds for students to explore. In each world, students had to solve problems that reinforced recently learned concepts, such as fractions, geometry, and place value. However, unlike traditional two-dimensional activity sheets, the students had to create their answers in a three-dimensional world.

Rodger et al. hoped that similar creative uses of Alice will work their way down to middle school and the upper-elementary grades to help restore students' enthusiasm for math, science, and computer programming. In so doing, today's classrooms can help students connect their real-world interests in technology with a passion for creative problem solving.

As teachers, we expect students to transfer what they have learned in class to the real world. However, do we enable learning that will connect to the real world of this technologically savvy generation? The International Center for Leadership in Education (2008) created the Rigor/Relevance Framework® to guide curriculum, assessment, and instruction toward more rigorous and relevant instruction. The framework suggests that teachers pursue imparting knowledge and learning that reach the pinnacle of Bloom's taxonomy in activities that apply knowledge to unpredictable situations in the real world. The expansive growth of technology is a testament to the need to keep pace with an unpredictable world. Could it be that virtual worlds, such as Alice, provide the gateway into the real world in which our students now live?

Note: The latest version of Alice, as well as supporting information, can be downloaded for free at http://www.alice.org.

BIBLIOGRAPHY

Dann, Wanda, Stephen Cooper, and Randy Pausch. 2007. *Learning to Program with Alice—Brief Edition.* Upper Saddle River, NJ: Pearson Prentice Hall.

International Center for Leadership in Education. 2008. *Rigor/Relevance Framework.* http://www.leadered.com/rrr.html.

National Council of Teachers of Mathematics (NCTM). 1989. *Curriculum and Evaluation Standards for School Mathematics.* Reston, VA: NCTM. http://www.nctm.org/fullstandards/previous/currevstds/5-8s1.asp.

Robinson, Ken. 2001. *Out of Our Minds. Learning to be Creative.* Oxford, UK: Capstone Publishing.

Rodger, Susan H., Maggie Bashford, Lana Dyck, Jenna Hayes, Liz Liang, Deborah Nelson, and Henry Qin. 2010. "Enhancing K–12 Education with Alice Programming Adventures." Conference for Innovation and Technology in Computer Science Education (ITiCSE), June 2010, Bilkent, Ankara, Turkey.

Tom Parker, tparker@nwsd.org, teaches fifth-grade mathematics at Northwestern Elementary School in Albion, Pennsylvania. He is interested in real-world problem solving and the integration of technology into the curriculum.

Ed. note: After you have downloaded and tried Alice with your students, tell us what they learned by sending an e-mail to **mtms@nctm.org** and typing "Readers Write" in the message line.