Integrating Computer Science into K-12 Education

Francine Wolfe
Benedictine College
Atchison, KS

Mentor:
Susan Rodger
Duke University
Durham, NC

ABSTRACT

Adventures in Alice programming is a project at Duke University designed to integrate computer science concepts into K-12 schools using a 3D animation tool in a wide variety of subjects. This is the third summer of the program. Teachers from across North Carolina and some from out of state attended our workshops. We taught them the main ideas of how to use Alice. We also discussed with them different ways it could be used to teach topics in various subjects.

1. INTRODUCTION

With the rapidly growing field of technology, the demand for Computer Science majors is constantly increasing. There is also a call for an increase in understanding of technology for people in any field (1). The number of students majoring in Computer Science has been declining significantly. This decline has been even greater for women (2). One of the main reasons for this decline is the misconceptions and stereotypes that students have about Computer Science. A survey of primarily middle and high school students showed that 24% believed that computing is mainly about writing programs and 22% thought most computing jobs do not involve working with people (3).

Students start thinking about possible career choices at a very young age. Because of these stereotypes, and the lack of knowledge of what Computer Science is really like, the possibility choosing it as a major is eliminated very early on (1). The primary goal of using Alice in K-12 is to expose students to programming in an exciting new way in order to change or prevent these stereotypes.

Alice is an educational tool which students can use to make 3D animations while learning the basics of computer programming (4) (5). Alice has several ready to use 3D objects so that students do not have to go through the tedious task of creating them. It also uses a drag and drop interface which virtually eliminates syntax errors while programming. Each line of code is structured like an English sentence which makes it easy to understand. It also has an object-oriented feel which prepares students for programming in languages like Java. It has even been shown to help students with learning advanced programming concepts such as recursion (6). Alice is especially helpful for novice programmers because the visual element allows the student to see exactly what is happening as the computer executes each line of code.

With the strict curriculum requirements of different school districts, it is often difficult to have a separate class in which Alice is taught. For this reason, we are looking at ways that Alice can be used in other subjects. I focused particularly on how it can be used in Math and Spanish classes.

2. PREVIOUS WORK

Alice has been shown to be a good introduction to programming for students with weak backgrounds in Math, problem solving, and computing. Learning Alice as an introduction can teach these students how to create algorithms and visualize what code does (7). Results have shown that these at-risk students who have taken a course using Alice will perform as well in an introductory programming course as other students with stronger backgrounds (8). In this sense, Alice levels the playing field for students being introduced to programming.

Alice can also be a great visualization tool for other subjects too. Studies have shown that using digital 3D visualization tools can increase the effectiveness of regular teaching methods (9).

The Adventures in Alice Programming Project started in 2008 at six different sites. At the Duke University site, there was a 3-week workshop for teachers and one-week camps for students that summer (1). Alice was taught to both students and teachers. The participants were taught how to create worlds using certain concepts and also allowed free time to work on their own projects. Worlds created by the middle school students were collected in order to analyze which concepts they used in their own worlds. Several students used many of the concepts that we had taught them.

In the summer of 2009, the project continued. Three week-long workshops were held to teach the introductory concepts of Alice to K-12 teachers.

3. WORKSHOPS

This summer, we hosted two week-long workshops in which we taught primarily middle and high school teachers how to use Alice. Each workshop had about 20 participants. They came from a wide variety of disciplines, including Science, Math, Technology, and Business. Most of the week was spent teaching them, step by step how to create different worlds in Alice. Each of these worlds focused on at least one new concept. The teachers were also given free time to work on a world which we had given them certain requirements for. We also spent some time presenting the worlds we had developed to show them how Alice could be used in various subjects. The teachers were very engaged and also wanted to know more about what you can use Alice to do.

We also held two two-day workshops. These were for teachers who had already attended an Alice workshop in the past but wanted to learn more advanced concepts. These workshops were much smaller, with only a few people at each. The teachers were mainly asking about specific worlds they wanted to build. We gave them some ideas and tried to help them build them. We also made note of what sort of things they wanted to use Alice for. At the first advanced workshop, a lot of the teachers wanted to learn how to make more games. At the second workshop, they were more interested in subject-specific worlds, such as Math and Biology.

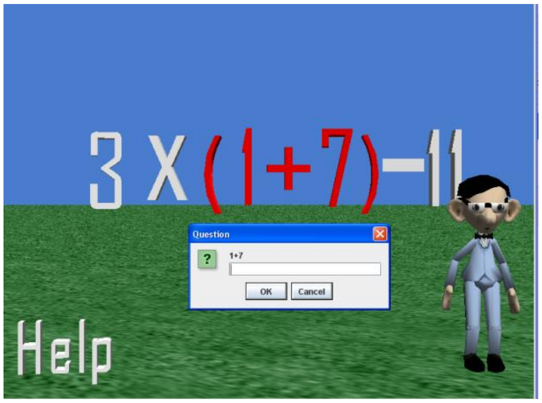
4. DEVELOPING WORLDS AND TUTORIALS

When we did not have workshops, we spent our time developing teaching materials. We made several example worlds. We made tutorials for some of them to show people how to make the worlds. The two subjects that I focused on were Math and Spanish. I also made two games since many of the teachers at one of the workshops expressed interest in learning how to use certain gaming concepts in Alice.

4.1 Math

When brainstorming what Alice worlds I could create, I thought that Math would be a good subject to visualize using Alice.

I made an example world for students who are learning the order of operations. A long complex equation is displayed. They are then taken through step by step how to solve it. The player must click on each operator in the correct order. After each one, they must type in the solution to that part of the problem as shown in Figure 1 below.



1+7

Figure 1: Order of Operations

The solution then replaces that part of the equation (Figure 2 below) and this process is repeated until the final solution is reached.

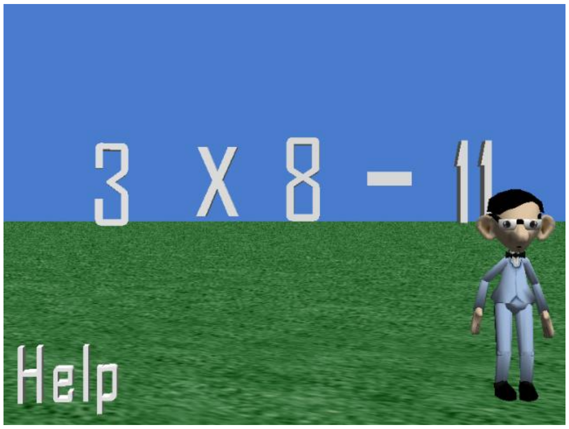
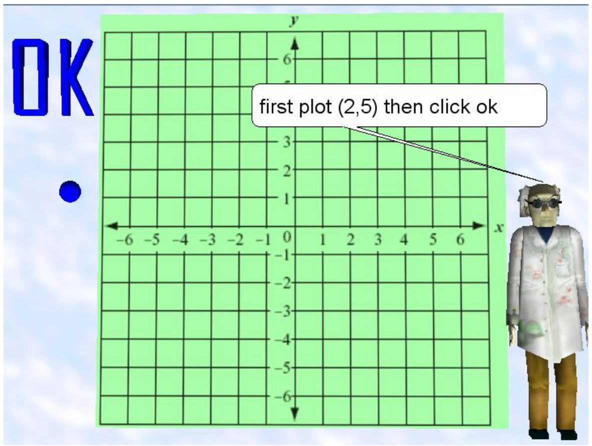


Figure 2: Order of Operations Solving

The player can also click on the help button at any time to display the order of operations. This would be a good exercise for students because they can visualize the problem as they solve it. Also, if they make a mistake, they are told so immediately so as to avoid making a minor mistake along the way that affects the final answer.

The second Math world I created required the player to plot points on a graph. I got this idea from a similar world that a teacher built at one of the previous workshops. I thought it was a very good idea, but when I looked at the code, I realized that it was very inefficient. It had invisible objects at each point on the graph where you could click. This not only makes it very tedious when creating the world, but also makes the file size of the world very large since each object in an Alice world takes up memory.

I decided to create a new Alice world which did the same thing in a more efficient manner (see Figure 3 below). The player has to click and drag points onto the graph at the correct coordinate. They then press the ok button to check it.

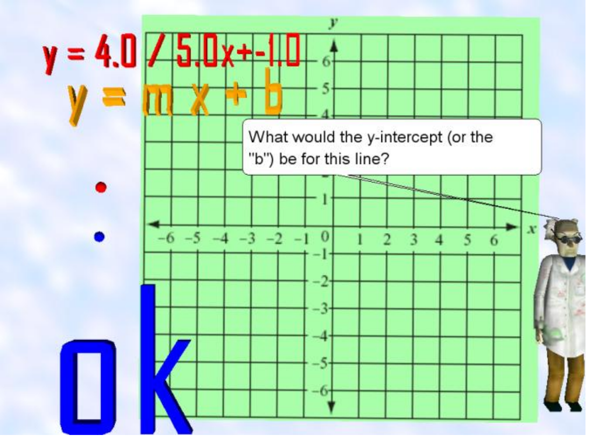


first plot (2,5) then click ok

Figure 3: Plotting Points

There is only one invisible object in the world, a pair of axes, which are aligned with the axes on the grid. To check what coordinates the points are at, Alice' s built-in distanceAbove and distanceToTheRightOf functions were used to compare where the points were in relation to the axes. The program uses a formula to convert the units in Alice to units on the graph. The coordinates which the player is required to plot are stored as variables. This makes it very easy if a teacher wanted to change the points or make them random. Since I thought this was a concept that many Math teachers could use, I made a tutorial that shows how to build this world.

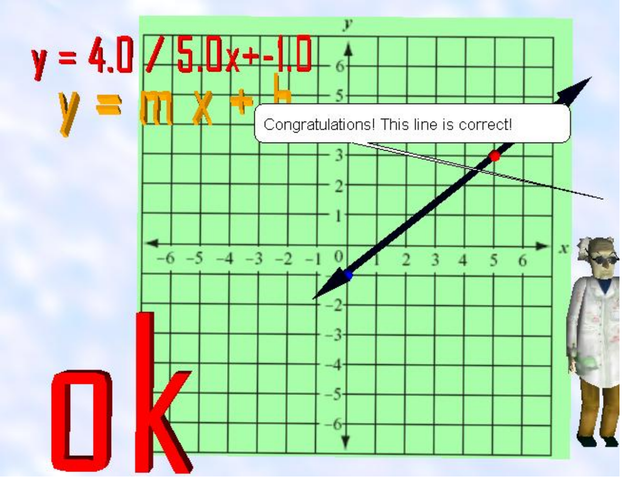
There are several different math worlds that could be created using this concept of how to check the coordinates of a point. I made two example worlds just to show the teachers how to expand on this concept. The first world, shown in Figures 4 and 5, shows the student how to plot a line. First, they are giving an equation for the line in slope-intercept form. The teacher in the world gives a short explanation of how to use an equation to plot a point. He then asks what the y-intercept of the line should be.



What would the y-intercept (or the "b") be for this line?

Figure 4: Plotting a Line - Finding the Y-intercept

The player must type in the answer. If they answer correctly, they must click on the first point and move it to the y-intercept and click ok. Then the teacher asks what the slope is and the player must type it in. The player then has to plot what the next point should be using the slope. If both points are plotted correctly, the line is drawn, which goes through the two points.



Congratulations! This line is correct!

Figure 5: Plotting a Line - Finished

The world can be modified to plot as many lines as the teacher chooses. They can also easily change the equations which are stored using variables. This automatically changes the equation that is displayed, what numbers must be typed in, and what points must be plotted. They can also change how much instruction is given for each equation. If fewer instructions are wanted, the teacher does not explain how slope intercept form works at the beginning, and instead of having to type in the numbers, the player is only required to plot the two points of each line. In the sample world, more instruction is given for the first line and the second one has less.

I also made a second example using the coordinate checking method. It teaches students how to use a scatter plot to predict future data. It starts out with a biker riding down the street, as shown in Figure 6. The player must click on the clock four times during the ride to collect data. Each time the clock is clicked, the time and the biker' s current position are recorded and displayed on the screen.

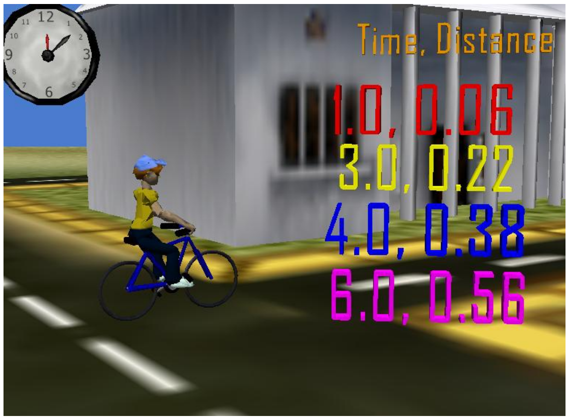


Figure 6: Scatter Plot - Collecting Data

Then a graph is displayed, on which the player is required to plot each of the four points. Once all four of the points have been correctly plotted, the teacher says that he is curious how far the biker could travel in 10 seconds. He then explains that you could use a best fit line to guess. He then draws a best fit line, as shown below in Figure 7. The player is then asked to type in their guess using the new line. He also asks how long it would take to travel one block.

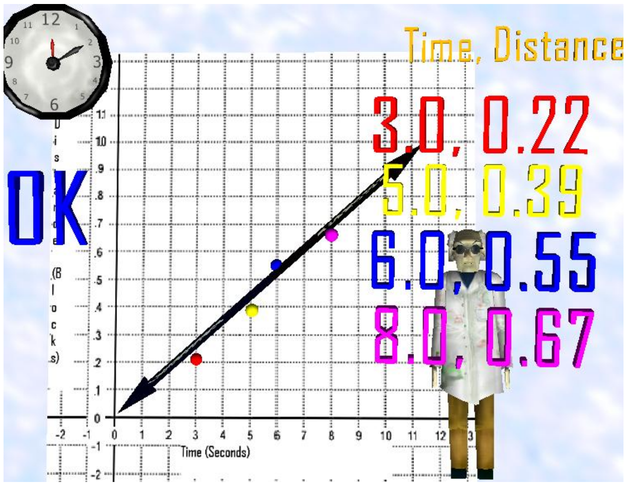


Figure 7: Scatter Plot - Making a Best Fit Line

This world could be very helpful for learning about best fit lines since it is highly visual. The student gets to actually see the bike ride as opposed to just reading the story problem in a textbook. Having the student click on the timer to record points would also hopefully make him or her feel more engaged. The student could play the world several times, recording different points each time, and see that the same final answer is always achieved.

I made one more Math world using an idea I got from a book called *Math Doesn’ t Suck* (10) which explains math concepts in a way that is relevant and attractive to middle school girls. The example I chose uses bags of pearls and a scale to explain how to solve an Algebra problem (shown in Figure 8 below). A girl in a prom dress is in the world to make it more appealing to girls. She explains that a bag of an unknown number of pearls represents x, and the two sides of the scale represent the two sides of the equation. She says that the goal is to have one bag of pearls on one side and some number of loose pearls on the other side to determine how many pearls are in each bag. She then solves the equation step by step while taking pearls off of each side at the same time.

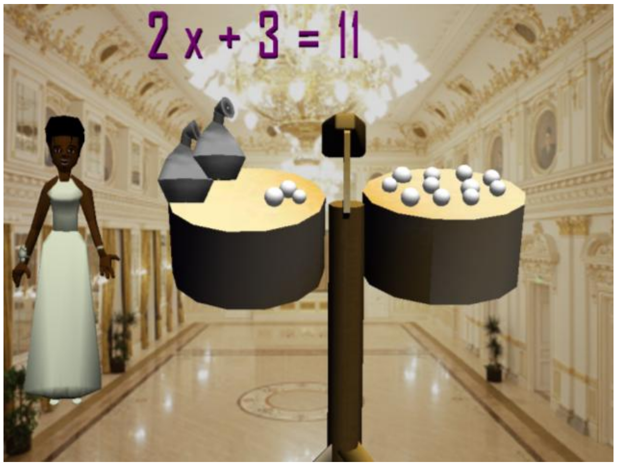


Figure 8: Algebra Pearls

I think that this world could be very helpful for girls struggling with Algebra. Pearls are very feminine so it is easy for the girls to relate to the story. Also, it is very helpful for the visual learners to see each step as it happens.

4.2 Spanish

The second subject I chose to make Alice worlds for was Spanish. I made a Spanish cooking show using Alice. This game could be used to practice vocabulary from the food unit of a Spanish class. The girl in the world gives instructions, in Spanish, of how to make banana bread (see Figure 9 below). The player must click and drag each ingredient to the bowl in the correct order.

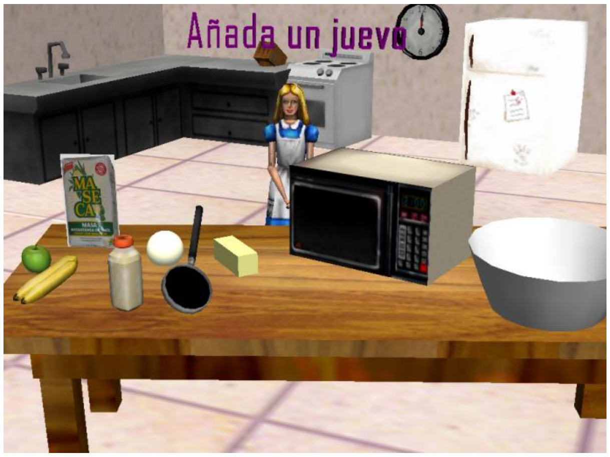


Figure 9: Cooking Show - Adding Ingredients

The concepts used in this game could be used in any game in which the player must perform actions in a certain order. I made a tutorial which explains how to build this world in either English or Spanish.

The world that is built using the tutorial is a simple version of the game which only uses three ingredients. I also created a more advanced version of this world which uses more ingredients and also includes a second part where the user must set the table (see Figure 10 below). The second part of the world uses vocabulary for food as well as items needed when setting the table for a meal. The advanced world uses most of the vocabulary from the food section of a textbook (11) which I used to get ideas from.

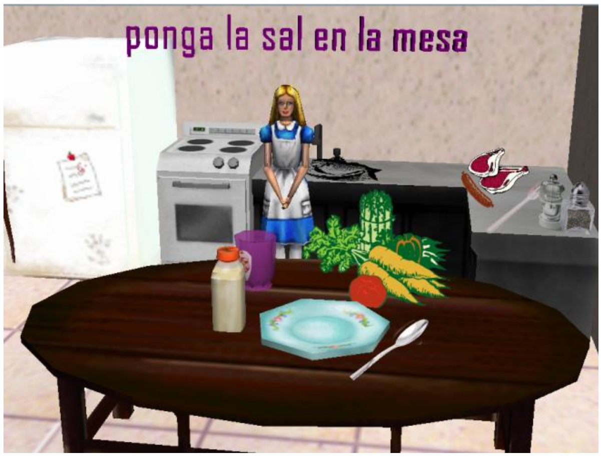


Figure 10: Cooking Show - Setting the Table

4.3 Games

I wanted to create a tutorial which showed how to make a simple game in Alice. I decided to make a game of catch, where the goal is to catch a ball three times (Figure 11 below). The ball is pitched several times at a random spot each time. The user can use the arrow keys to move the glove to try to catch the ball. The main concept that this tutorial taught was how to use a score. It also used some collision detection to check whether the ball is close enough to the glove to catch it.

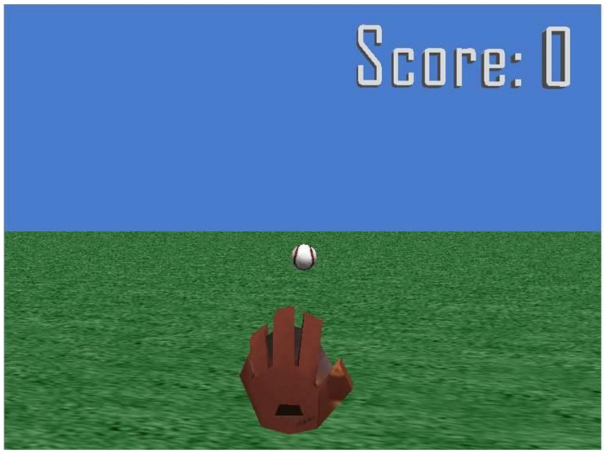


Figure 11: Catch Game

At one of the workshops, several of the teachers said that they wanted to learn how to make more advanced game worlds. I decided to make a game similar to Mario Bros (see Figure 12 below). The player uses the arrow keys to control a ninja. He must jump over obstacles and he can jump to hit boxes which will release a random object: scissors, a bug, or a potion. Touching the scissors or the bug will kill the ninja, but he can jump on top of the bug to kill it. Picking up a potion gives you an extra life. This means that if you have picked up a potion when you die, you start from the beginning again. If you have no potion, the game is over. The screen is always moving to the right and the ninja will always stay within view.

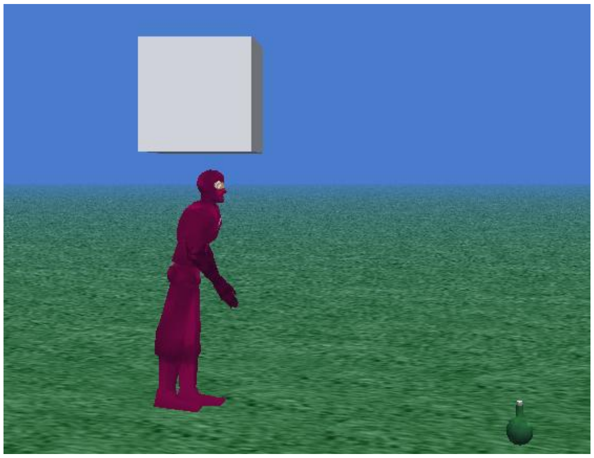


Figure 12: Ninja Game

This game was too complex to make a full tutorial so instead I made a description page which explained how the difficult concepts were implemented. For example, collision detection was used when the ninja hits the box to release an object, touching the various objects, and keeping the ninja from moving through objects or off of the screen. If someone sees the game and is curious how it was made they can look at the description and get a fairly good understanding. They should also be able apply those concepts to a different game they might want to make.

5. CONCLUSION

The goal of the Adventures of Alice Programming project is to integrate programming into K-12 education so students will have a better understanding of Computer Science. Alice is an engaging tool for teaching algorithmic thinking and basic programming concepts. Its visual element makes it highly attractive to young students while helping them understand the code process.

At the workshops, teachers from a wide array of subjects were taught how to use Alice. Some intended to teach Alice to their own students, while other just wanted to use it as a visual aide for teaching. Potential lesson plans were discussed as well. Most of the teachers were very excited about using Alice in the classroom.

I created several educational worlds focusing on Math and one on Spanish. Topics taught in the Spanish worlds included order of operations, using the coordinate system, slope-intercept equations, scatter plots, and algebra. The Spanish world enforces food vocabulary. I also created two game worlds. Tutorials were made for some of these worlds. Teachers can use these tutorials to learn the concepts used and hopefully come up with ideas for other ways to use Alice in various subjects.

References

1. **Rodger, Susan, et al.** Engaging Middle School Teachers and Students with Alice in a Diverse Set of Subjects. *Technical Symposium on Computer Science.* Bilkent, Ankara, Turkey : SIGCSE 2009. pp. 271-275.

2. **Dyck, Lana.** Connecting Computer Science and the K to 12 Classroom. DREU Report : CRA-W, 2009.

3. **Joseph A. Cottam, Samantha S. Foley, and Suzanne Menzel.** Do Roadshows Work? Examining the Effectiveness of Just Be. Milwaukee, Wisconsin : SIGCSE 2010. pp. 17-21.

4. Alice. [Online] July 2010. Alice.org.

5. **Dann, Wanda, Cooper, Stephen and Pausch, Randy.** *Learning to Program with Alice.* 2nd Edition. Upper Saddle River : Pearson Education, 2009.

6. **Dann, Wanda, Cooper, Stephen and Pausch, Randy**. Using Visualization to Teach Novices Recursion. Canterbury, England : ITiCSE 2001, July 2001. pp. 109-112.

7. **Dann, Wanda and Pausch, Randy.** Alice: a 3-D Tool for Teaching Programming Concepts. Ramapo, NJ : CCSCNE, April, 2000.

8. **Moskal, Barbara, Lurie, Deborah and Cooper, Stephen.** Evaluating the Effectivenss of a New Instructional Approach. Norfolk,VA : SIGCSE 2004. pp. 75-79.

9. **van Langeveld, Mark and Kessler, Robert.** Digital Visualization Tools Improve Teaching 3D Character Modeling. Milwaukee, WI : SIGCSE 2010. pp. 82-86.

10. **McKellar, Danica.** *Math Doesn't Suck.* New York, New York : Penguin Group, 2007. pp. 244-246.

11. **Estella Gahala, Patricia Carlin, Audrey Heining-Boynton, Ricardo Otheguy, Barbara Rupert.** *¡En Español!* Evanston, IL : McDougal Little, 2000. pp. 356-379.

12. **van Langeveld, Mark and Kessler, Robert.** Digital Visualization Tools Improve Teaching 3D Character Modeling. Milwaukee, WI : SIGSCE 2010.

13. **Lurie, B. Moskal and D.** Evaluating the Effectivenss of a New Instructional Approach. Norfolk,VA : SIGSCE 2004.

14. **Joseph A. Cottam, Samantha S. Foley, and Suzanne Menzel.** Do Roadshows Work? Examining the Effectiveness of Just Be. Milwaukee, Wisconsin : SIGSCE 2010.

15. **Dyck, Lana.** Connecting Computer Science and the K to 12 Classroom. DREU Report : CRA-W, 2009.