



ISSN: 1545-679X

Information Systems Education Journal

Volume 4, Number 87

<http://isedj.org/4/87/>

October 2, 2006

In this issue:

Using Alice in a Computer Science Survey Course

Jill Courte
Miami University
Hamilton, OH 45011 USA

Elizabeth V. Howard
Miami University
Middletown, OH 45042 USA

Cathy Bishop-Clark
Miami University
Middletown, OH 45042 USA

Abstract: Students in an introductory computer science survey course for non-majors used the 3-D interactive programming environment Alice for their programming module. With Alice, students create animated worlds using a variety of objects and easily accessed programming constructs. Alice has previously been used successfully to provide a gentle introduction to computer programming in a fun and enjoyable way, and has been shown to increase retention in computer programming courses. However, use of Alice appears to be limited as there is little to be found in the literature regarding its use in information technology courses. This lack of data prompted a study to determine if Alice is a suitable tool to introduce computer programming to non-majors in a friendly and inclusive way. This paper reports on the use of Alice with approximately 100 students enrolled in a non-majors computer science course. Students were surveyed about their attitudes toward computer programming before and after using Alice. Descriptive data indicate that Alice does increase student enjoyment and promote positive attitudes toward programming.

Keywords: Alice, pair-programming, attitudes, introductory programming

Recommended Citation: Courte, Howard, and Bishop-Clark (2006). Using Alice in a Computer Science Survey Course. *Information Systems Education Journal*, 4 (87). <http://isedj.org/4/87/>. ISSN: 1545-679X. (Also appears in *The Proceedings of ISECON 2005*: §2133. ISSN: 1542-7382.)

This issue is on the Internet at <http://isedj.org/4/87/>

The **Information Systems Education Journal** (ISEDJ) is a peer-reviewed academic journal published by the Education Special Interest Group (EDSIG) of the Association of Information Technology Professionals (AITP, Chicago, Illinois). • ISSN: 1545-679X. • First issue: 8 Sep 2003. • Title: Information Systems Education Journal. Variants: IS Education Journal; ISEDJ. • Physical format: online. • Publishing frequency: irregular; as each article is approved, it is published immediately and constitutes a complete separate issue of the current volume. • Single issue price: free. • Subscription address: subscribe@isedj.org. • Subscription price: free. • Electronic access: <http://isedj.org/> • Contact person: Don Colton (editor@isedj.org)

2006 AITP Education Special Interest Group Board of Directors

Stuart A. Varden Pace University EDSIG President 2004		Paul M. Leidig Grand Valley State University EDSIG President 2005-2006		Don Colton Brigham Young Univ Hawaii Vice President 2005-2006	
Wendy Ceccucci Quinnipiac Univ Director 2006-07	Ronald I. Frank Pace University Secretary 2005-06	Kenneth A. Grant Ryerson University Director 2005-06	Albert L. Harris Appalachian St JISE Editor	Thomas N. Janicki Univ NC Wilmington Director 2006-07	
Jens O. Liegle Georgia State Univ Member Svcs 2006	Patricia Sendall Merrimack College Director 2006	Marcos Sivitanides Texas St San Marcos Chair ISECON 2006	Robert B. Sweeney U South Alabama Treasurer 2004-06	Gary Ury NW Missouri St Director 2006-07	

Information Systems Education Journal 2005-2006 Editorial and Review Board

Don Colton Brigham Young Univ Hawaii Editor		Thomas N. Janicki Univ of North Carolina Wilmington Associate Editor		
Samuel Abraham Siena Heights U	Tonda Bone Tarleton State U	Alan T. Burns DePaul University	Lucia Dettori DePaul University	Kenneth A. Grant Ryerson Univ
Robert Grenier Saint Ambrose Univ	Owen P. Hall, Jr Pepperdine Univ	Jason B. Huett Univ W Georgia	James Lawler Pace University	Terri L. Lenox Westminster Coll
Jens O. Liegle Georgia State U	Denise R. McGinnis Mesa State College	Therese D. O'Neil Indiana Univ PA	Alan R. Peslak Penn State Univ	Jack P. Russell Northwestern St U
Jason H. Sharp Tarleton State U		Charles Woratschek Robert Morris Univ		

EDSIG activities include the publication of ISEDJ, the organization and execution of the annual ISECON conference held each fall, the publication of the Journal of Information Systems Education (JISE), and the designation and honoring of an IS Educator of the Year. • The Foundation for Information Technology Education has been the key sponsor of ISECON over the years. • The Association for Information Technology Professionals (AITP) provides the corporate umbrella under which EDSIG operates.

© Copyright 2006 EDSIG. In the spirit of academic freedom, permission is granted to make and distribute unlimited copies of this issue in its PDF or printed form, so long as the entire document is presented, and it is not modified in any substantial way.

Using Alice in a Computer Science Survey Course

Jill Courte
courteje@muohio.edu
Miami University
Hamilton, Ohio 45011 USA

Elizabeth V. Howard
howardev@muohio.edu

Cathy Bishop-Clark
bishopcu@muohio.edu
Miami University
Middletown, Ohio 45042 USA

Abstract

Students in an introductory computer science survey course for non-majors used the 3-D interactive programming environment Alice for their programming module. With Alice, students create animated worlds using a variety of objects and easily accessed programming constructs. Alice has previously been used successfully to provide a gentle introduction to computer programming in a fun and enjoyable way, and has been shown to increase retention in computer programming courses. However, use of Alice appears to be limited as there is little to be found in the literature regarding its use in information technology courses. This lack of data prompted a study to determine if Alice is a suitable tool to introduce computer programming to non-majors in a friendly and inclusive way. This paper reports on the use of Alice with approximately 100 students enrolled in a non-majors computer science course. Students were surveyed about their attitudes toward computer programming before and after using Alice. Descriptive data indicate that Alice does increase student enjoyment and promote positive attitudes toward programming.

Keywords: Alice, pair-programming, attitudes, introductory programming

1. INTRODUCTION

Alice is a 3-D interactive programming environment developed at Carnegie Mellon University to provide a gentle introduction to computer programming (Cooper, Dann, and Pausch, 2000). Students often struggle with introductory programming concepts, and non-majors courses provide a particular challenge as students often have negative attitudes toward computer programming in general. Alice has been used to help students with little or no problem-solving experience learn to program in an enjoyable way (Cooper, Dann, and Pausch, 2003), so it

appears to be very appropriate for use in an introductory course. A screen shot of a very simple Alice program is shown in Figure 1. A typical Alice program will contain objects such as those shown which can be programmed to move and interact with each other in order to create an animated story.

Alice is not a general purpose programming language. It is intended to help students gain foundation skills such as algorithmic thinking and an understanding of basic programming constructs such as decisions and looping. As Cooper et al. point out, students in information technology courses are often

deficient in the skills needed to break down and solve unique problems in a step-by-step manner (Cooper, Dann, and Pausch, 2000). Even those with experience in advanced math courses struggle with this, so an environment such as Alice that seeks to promote algorithmic thinking while hiding the complexities of syntax and program construction is ideal to introduce programming. As shown in Figure 2, students use Alice to manipulate objects in simple English terms rather than typical computer language syntax. In this case, a student programmer would program the robot to move by simply selecting the method *kangarooRobot.turn*.

seems to lend itself well to using Alice in classes that promote teamwork.

This paper reports the experience of using Alice with non-majors in an attempt to positively affect student's attitudes toward programming. Alice was used in 5 sections of the course with two different instructors in the spring semester. Approximately 100 students were enrolled in the 5 sections, and data was collected from 64 students. Students were surveyed about their attitudes before and after using Alice.

2. THE COURSE

Student Profile

The course was offered at two regional campuses of a medium-sized university. There are approximately 5000 students between the two campuses and all are commuter students. Almost all of the students work and have outside responsibilities.

Very few of the students have any previous exposure to programming, and for many, the course is their first meaningful exposure to technology. Most are in their first of second year of college and are predominately female.

Course Description

The course is intended to be a broad look at technology, computer science, and its effects on society and is part of the math, logic, and formal reasoning block of a liberal education curriculum. It is a semester-long course held in a computer classroom, often taken in lieu of mathematics courses by those not majoring in information technology or computer science.

A substantial part of the course involves computer programming at various levels. Prior to using Alice, languages such as Basic, Visual Basic, or JavaScript were used exclusively in this course to teach high-level programming constructs. Those languages require technical skill and some commitment to master, and so have not always been well-received by students in this particular course.

In addition, in courses such as this there is not enough time for students to learn programming well enough to produce programs of any consequence. Most non-major stu-

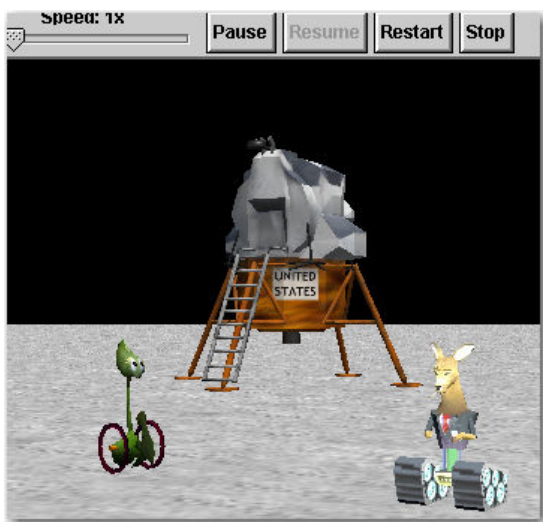


Figure 1 A simple Alice program

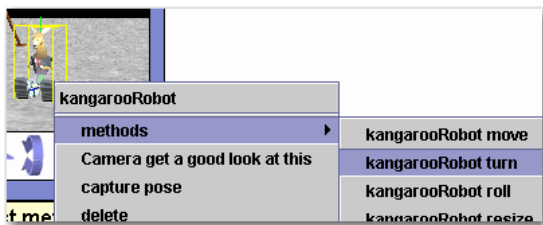


Figure 2 Alice commands

Alice also lends itself very well to student collaboration, as students readily seek others opinions about object selection, placement, and movement. The story-telling nature of Alice also promotes collaboration as students readily share and enjoy the results of their work. This collaborative aspect

dents become quickly frustrated with the syntax requirements of high-level languages and so gain and retain little knowledge about programming. For students seeking an introduction to computer science and programming, Alice appears to be an excellent replacement for more traditional computer programming languages.

3. PROGRAMMING WITH ALICE

Alice was used over the course of a week for approximately 3 hours class time. While this seems like a short amount of time, it is not unreasonable for a survey course. In addition, students completed a module on JavaScript following the time spent with Alice.

We were interested to know student's reactions to programming with Alice in general, and to determine if using Alice increased their satisfaction with computer programming. In addition, we wanted to find out if programming with Alice contributed to better attitudes toward programming or comfort working in the class.

The First Class Session

Pre-test survey Students first took a pre-test survey. This included demographic questions, questions to evaluate their attitudes toward programming, and questions to evaluate their knowledge of programming concepts. A five point scale was used, ranging from strongly agree (1) to strongly disagree (5) for the evaluation questions while a 12 point scale was used for the programming concepts.

Programming introduction Students then started their use of Alice with a very brief introduction to programming terminology. They then worked through three of the four tutorials provided with Alice at their own pace. The fourth tutorial which deals with creating a world "from scratch" and adding objects to it was not used and was replaced at the next session with a paper reference which described the basic process of starting a new world and adding an object.

Most students worked quickly through the tutorials with little problem. Those who had problems generally got stuck on the same thing – whether or not to add quotation marks around text when typing it into a prompt box. The nature of the tutorials to

require correct input before the students can go on allows problematic concepts to be addressed when they occur. As a result, mistakes are readily explained and corrected, reducing the possibility of students continuing without understanding what they are doing.

In contrast, units in which the students learn languages like JavaScript or Visual Basic often include a fair amount of copying code without really understanding the concepts. With Alice, verbal feedback and fewer questions from the students indicates that they assimilate and understand more as they proceed.

The Second Class Session

Alice program In the next class session, students created their own Alice programs from scratch using a paper reference that showed them the basics of creating a new "world", adding objects to it, and animating those objects.

As many students in this class can struggle with even the basics of using computers, we felt that a simpler reference that they could consult at any point while creating their own program would be more helpful than using a tutorial to learn these concepts.

Students completed their programs individually or in with partners of their choice.

Program Requirements Students were required to include the following elements in their programs:

- At least 5 objects
- Movement of at least 5 objects
- Interaction between objects
- Multiple events
- A set of actions in a loop
- A set of actions done together

Very few students asked questions and most completed their programs within the class period. One pair of students used the computer lab to finish their programs out of class due to problems with the Alice software as they worked in class.

Post-test survey Students then took a post-test survey at the beginning of the next class period. This was identical to the pre-test except they were also asked on the

post-survey to provide free-form comments on the experience.

4. RESULTS

Students indicated overall that they enjoy working with Alice. They enjoy creating and showing their programs to others. During the exercise, many students will call the instructor over to show off their efforts. Both instructors reported that the students were engaged and enthused throughout their experience using Alice.

Table 1 illustrates the positive impact of using Alice for the programming module.

Question	Pre	Post
<i>Generally I have felt secure about attempting computer programming</i>	19%	48%
<i>I have a lot of self confidence when it comes to programming</i>	17%	29%
<i>Generally I have felt secure about attempting computer programming</i>	19%	48%
<i>I am no good at programming</i>	26%	16%
<i>I like writing computer programs</i>	10%	27%
<i>Programming is enjoyable and stimulating</i>	11%	37%

Table 1 Positive Impact of using Alice

As can be seen, students felt more secure about attempting computer programming after using Alice, 48% post-test vs. 19% pre-test. Their confidence in programming increased overall, both when asked about increases in their confidence (29% post-test vs. 17% pre-test) and decreases in negative feelings about their ability to program (16% post-test vs. 26% pre-test).

Students also reported better attitudes about programming in general after using Alice. More students reported that they liked writing computer programs 27% post-test vs. 10% pre-test, and that programming in enjoyable and stimulating. 37% post-test vs. 11% pre-test.

This improvement in attitudes toward programming is consistent with findings by Moskel, et.al., who found similar, although smaller increases. (Moskel, Deborah Lurie, and Cooper 2004). However, their population was made up of students who had already expressed interest in becoming computer science majors, so greater increases in attitude for non-majors would be expected.

Not only did the data in the post test indicate a more positive attitude toward programming, this is also indicated by their comments. In response to the survey question *Please name three things that you enjoyed about the programming module using the Alice programming language*, we received comments such as:

"I enjoyed learning how to do it, making objects do different things, and being able to watch what I did."

"Being creative, Having a world at my finger tips, Being able to use my imagination."

"It was fun to create a program that you could watch, it felt like I was doing something entertaining, and it was a great learning experience."

"I enjoyed creating a world. I found that the creating events fun. I also enjoyed trying to make the objects do what I wanted them to do."

"...The elementary beginner steps, then going up the ladder after practice to the harder ones."

Of the students responding to the question, only two actively expressed negative feelings:

"I thought it was different, but I didn't like it to be honest."

"none"

Although we currently have no formal survey regarding comparisons to other languages, this is in sharp contrast to comments received when a more traditional language like Visual Basic or JavaScript is used in the class. In those cases students express much more frustration and seem to derive little pleasure from the experience.

This is consistent with others' experiences who report that students found Alice *extremely fun to use* (Cooper, Dann, and

Pausch (2000) and that *students had a highly positive experience* (Moskel, Deborah Lurie, and Cooper 2004).

5. CONCLUSIONS

Alice has been presented as a tool to increase enjoyment and learning, particularly for students with little or no programming experience. Our experiences bear that out, as well as indicate that it is a useful tool to introduce computer programming in a fun way to students who may have had no prior interest in programming. Using Alice also seems to improve attitudes toward programming in general. While Alice is not intended to be a general purpose programming language, it seems to serve very well to introduce basic algorithmic thinking and beginning programming constructs.

For a non-computing major, a service course in computing may be their only exposure to computer programming. Teaching a one to three week unit in Java or C++ can be difficult and frustrating. Alice provides an alternative. Our data suggest that students' attitudes toward programming can be improved by as little as a one-week session.

We are currently working on analyzing and publishing more formal results of this experience. We are also planning on research to more directly compare the use of Alice to more traditional programming languages in non-majors courses.

Overall, we believe that positive programming experiences in courses such as these can dispel negative attitudes toward information science as well as potentially increase the number of students interested in information science majors.

6. REFERENCES

- Cooper, Stephen, Wanda Dann, and Randy Pausch (2003) "Teaching Objects-first in Introductory Computer Science". Proceedings of the 34th SIGCSE technical symposium on Computer Science Education.
- Cooper, Stephen, Wanda Dann, and Randy Pausch (2000) "Alice, a 3-D Tool for Introductory Programming Concepts." Proceedings of the 5th annual CCSC north-eastern conference, pp. 107-116.
- Cooper, Stephen, Wanda Dann, and Randy Pausch (2000) "Developing Algorithmic Thinking with Alice." Proceedings of ISECON 2000, v17, pp. 506-539.
- Dann, Wanda, Stephen Cooper, and Randy Pausch (2004) "Making the Connection: Programming with Animated Small Worlds." Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education, Finland, pp 41-44.
- Dann, Wanda, Stephen Cooper, and Randy Pausch (2001) "Using Visualization to Teach Novices Recursion." Proceedings of the 6th annual conference on Innovation and technology in computer science education, pp 109-112.
- Dann, Wanda, Toby Dragon, Stephen Cooper, Kevin Dietzler, Kathleen Ryan, and Randy Pausch (2003) "Objects: Visualization of Behavior and State." Proceedings of the 8th annual conference on Innovation and technology in computer science education, pp 84-88.
- Kelleher, Caitlin and Randy Pausch (2005) "Stencils-based Tutorials: Design and Evaluation." Proceedings of the SIGCHI conference on Human factors in computing systems, pp 541-550.
- Moskel, Barbara, Deborah Lurie, and Stephen Cooper (2004) "Evaluating the Effectiveness of a New Instructional Approach." Proceedings of the 35th SIGCSE technical symposium on Computer Science Education, pp 75-79.