Evaluating the Effectiveness of a New Instructional Approach

Barbara Moskal
Mathematics Department
Colorado School of Mines
Golden, CO 80401-1887
bmoskal@mines.edu

Deborah Lurie
Mathematics Department
Saint Joseph's University
5600 City Avenue
Philadelphia, PA 19131
lurie@sju.edu

Stephen Cooper
Computer Science Dept.
Saint Joseph's University
5600 City Avenue
Philadelphia, PA 19131
scooper@sju.edu

ABSTRACT

This paper describes the evaluation of an NSF-sponsored educational research project. The primary focus of this project was to develop and evaluate a course curriculum designed to improve retention and performance for "at risk" introductory computer science majors. The results of this research suggest that the newly developed course and curriculum materials did improve students' performance and retention in computer science and their attitudes towards computer science.

Categories and Subject Descriptors

K.3 [**Computers & Education**]: Computer & Information Science Education – Computer Science Education.

General Terms

Documentation, Design, Experimentation, Human Factors.

Keywords

Visualization, Animation, Pedagogy, CS1, Study, Evaluation.

1. INTRODUCTION

This article reports the results of a study completed as part of a NSF-CCLI proof-of-concept grant (DUE-0126833). The primary purpose of this investigation was to determine whether fundamental concepts of programming, object-oriented programming in particular, could effectively be taught to students who had limited or no programming background using an innovative instructional approach developed by Drs. Wanda Dann and Stephen Cooper. In their classrooms, Drs. Dann and Cooper had observed that many computer science majors who had little or no previous programming experience, and perhaps also were poorly prepared in mathematics, were "at risk" of not succeeding in their first programming course, a rigorous CS1. Current research suggests that this observation is a broader concern in computer science education [11, 13, 15]. Drs. Dann and Cooper decided to seek to overcome this difficulty by creating a new course that utilized a three dimensional (3D) animation software, Alice, developed at Carnegie Mellon University (CMU). Drs. Dann and Cooper selected this software as the foundation for their course, believing that the programming visualization environment offered through Alice would be highly motivating to college students.

Curricula materials, including a textbook, were designed for a new course that would immediately precede the traditional CS1. Hereafter, the new course will be referred to as "the Alice course." The purpose of the Alice course is to provide students with the conceptual underpinnings of fundamental programming principles. The objective of this investigation was to examine the effectiveness of the Alice course for improving retention and performance of incoming "at risk" computer science majors. The study was conducted over two years at Saint Joseph's University (SJU) and Ithaca College (IC).

2. PRIOR RESEARCH

The seriousness of attrition in computer science programs has been described in several studies. More than half of college students that initially declare a major in computer science change their majors prior to graduation [12]. While attrition occurs throughout the four years, the majority of students leave computer science by the end of their freshman year [16]. Prior research suggests that major factors contributing to attrition include lack of experience with computers prior to entering college [17], limited or poor preparation in math [2], poor self-efficacy [1], socialization [16], and the university environment [3]. Recent investigations [11, 13, 15] have additionally found that students with no prior programming experience are at a disadvantage in successfully completing a computer science degree. This result stands in contrast to earlier work that found little or no correlation between prior programming experience and success in computer science.

We believe that the current result is related to the shift of most computer science departments from teaching introductory computer science courses using imperative languages (such as C and Pascal) to using object-oriented (OO) languages (such as C++ and Java). OO languages require that students not only learn the material for an imperative language core (e.g., assignment, decisions, functions, procedures, repetition, arrays) but also learn the additional concepts of class, object, information hiding, inheritance, and polymorphism. Many CS1 courses also include event-driven programming — yet another paradigm students must master. As the amount of material has increased, the time in class has not. Students with no prior programming experience are likely to be overwhelmed by the breadth and depth of material, contributing to student attrition.

The program visualization capabilities of the Alice software have been presented elsewhere [4, 5, 6, 7, 8]. Alice was developed by Randy Pausch and the Stage3 research team at CMU and is freely available at http://www.alice.org.

A pilot project was conducted in the first year of this study. Results of the pilot, reported in Cooper [5], used a preliminary data organization scheme. In this paper, we report an evolved data collection scheme, expanded assessment methods, and a summation of data collected over two years.

3. CURRICULUM DEVELOPMENT

As part of this study, a set of curricular materials and a textbook were developed. The curriculum materials support an innovative approach to teaching the fundamental concepts of OO programming to beginners. The goal in the development of these materials was to improve retention and performance of incoming at risk computer science majors in CS1. The Alice software was selected for the following reasons:

- Working with an easy-to-use 3D graphics environment is attractive and highly motivating to today's generation of media-conscious students.
- The visual nature and immediate feedback of program visualizations makes it easy for students to see the impact of a statement or group of statements. Further, it makes debugging easier.
- The drag-and-drop editor prevents students from making syntax errors that are prevalent for beginners.
- The 3D modeled classes and instantiated objects in Alice provide a very concrete notion of the concept of an object and support an "object-first" approach [6, 9].

Prior to developing the curriculum materials, the researchers examined a wide variety of C++ and Java CS1 texts and compiled a list of fundamental concepts commonly taught in current CS1 courses. These concepts are: decisions, repetition (definite and indefinite, as well as recursion), functions/methods, collections (typically arrays, though sometimes lists), objects (including state and behavior), inheritance, encapsulation, polymorphism, and interactivity. Each of the identified fundamental concepts is introduced through the Alice curricula materials. The materials and additional details can be found at [18] and are described in [10].

4. RESEARCH QUESTIONS

The purpose of this investigation was to examine the effectiveness of the Alice course and curriculum materials for teaching fundamental programming concepts to at risk students. The specific research questions were:

1. Does exposure to this innovative approach improve student performance in CS1?
2. Does exposure to this innovative approach increase the retention of students into the next course (CS2)?
3. Does exposure to this innovative approach improve students' attitudes towards and confidence in their ability to succeed in a computer science program?

5. PARTICIPANTS

5.1 Participating Colleges

At SJU, there are approximately 25-30 new computer science majors each year. Over 5 years previous to this study, the retention rate was slightly less than 70% immediately following the freshman year. Students with minimal programming background and an insufficient math background had close to 0% retention (almost 100% attrition) by the end of the first year.

At IC, there are approximately 25-35 computer science majors among the entering freshman class each year. Over 5 years previous to this study, the retention rate for computer science majors during the first year was about 70%. This figure is clouded by the fact that the number of first year computer science majors reported at the end of each academic year includes not only retained students (initially entered as CS majors) but also those who declared a CS major during the year. If an average number of newly-declared majors are removed from the annual data, the first year retention rate adjusts to approximately 55%.

5.2 Student Population and Recruitment

In this investigation, the target population was first year computer science majors deemed "at risk" – having minimal previous programming experience and perhaps a weak mathematical background. Weak background in math was identified using a calculus readiness exam taken by all students entering SJU or IC the summer before their freshman year. Students who had no previous programming experience were identified using a self-report instrument.

To recruit students, an invitation to participate in the Alice course was sent to all incoming freshmen who had declared a computer science major. This letter described the potential benefits for improving performance in CS1 for students who had little or no previous programming experience.

6. METHODS

6.1 Sequencing of Alice Course

At IC, the Alice course was offered concurrently with CS1, as a 2-credit class running during the first half of the fall semester. At SJU, the course was offered prior to CS1, in a semester-long fashion. Both schools have a demanding liberal arts core and scheduling conflicts did arise. Thus, some students who expressed interest in the course were *not* able to take it.

6.2 Assessment Instruments

Data was collected from classic sources such as enrollment lists and grades reports. In addition, several instruments were selected or developed. A background survey was designed to acquire information from participating students concerning prior coursework in math and computer science. The survey can be found at [18].

To examine the first research question, grades and enrollment data were collected from CS1 class lists and grade reports. This data was collected over two years. During the second year, a content pre/post assessment instrument was developed and tested. The content assessment was created through a collaborative effort of the researchers and the evaluator and was designed to measure basic concepts in computer science common to CS1 and the Alice course. The original and the revised tests can be found at the above-mentioned web site. Data for performance (grades in CS1 and the pre/post assessment) is reported in section 7.2 below.

To examine the second research question, data was collected regarding whether or not students who had completed CS1 continued to the next course in the sequence, CS2. The retention data, collected over two years, is reported in section 7.3 below.

The third research question was examined (in the second year of the study) through the administration of an online attitudes survey. The specific instrument was developed and validated by Loyd and Gressard [14] and was designed to measure students' attitudes towards computers and computer science. This instrument consists of five subscales: anxiety, confidence, liking, usefulness and creativity. Based on the needs of the current investigation, the anxiety subscale was not administered. Additionally, three randomly selected students who took the Alice course at SJU participated in a focus group concerning their classroom experiences. At IC, five randomly selected students were individually interviewed. The focus group and individual interviews were based on the same set of questions which can be found at the above-mentioned website. The attitudes data, collected during the second year, is reported in section 7.4 below.

6.3 Data Collection

All appropriate human subjects procedures were followed in this investigation. The "treatment group" consisted of students from IC or SJU that completed the Alice course and that signed the informed consent form. Students at IC or SJU that only completed the CS1 course and signed the informed consent form comprised the "control group".

At IC, both the treatment and control groups completed the background survey, the content pre/post test and the attitudes survey at the beginning and end of CS1. At SJU, the treatment group completed the background survey, content pretest and the attitudes survey at the beginning of the Alice course and the control groups completed these instruments at the beginning of CS1. Both the treatment and the control groups completed the content posttest and the attitudes survey at the end of CS1 (with the treatment group also completing these instruments at the end of the Alice course).

6.4 Study Groups

All participating students were computer science majors. We used an "at risk" grouping mechanism, where "at risk" is defined as students with little or no previous programming experience and perhaps weak math preparation (not calculus-ready), as well. For analysis purposes, the following three groups of students were identified:

Treatment Group: *Students at risk and who enrolled in Alice course.*

Control Group1: *Students at risk and who did not enroll in the Alice course.*

Control Group2: *Students not at risk or low risk and who did not enroll in the Alice course.*

Only those students who completed both the pre and posttest are reported in the pre/post content data. Also, only students who completed the attitudes survey both before and after the course are reported in the attitudes data. Thus, the number of students (N) in each group varies depending on the assessment tool being used.

For the analysis of the two-year grades and retention data, subgroups based on risk level were defined as follows: **High Risk** is weak in both CS and math, **Medium Risk** is weak in CS only, **Low Risk** is not weak in CS but is weak in math, and the subgroup **Not at Risk** is not weak in CS and also not weak in math.

6.5 Statistical Analysis

Kruskal-Wallis (KS) one-way analysis of variance (ANOVA) was used to compare the performance and attitude data by study group. KS ANOVA is a non-parametric technique that makes no assumptions about the underlying distribution and requires a minimum of 5 subjects per group. Chi-square analysis was used to compare retention rate by study group. Since many of the risk groups had small sample sizes, only descriptive statistics were presented for this classification. All analyses were performed using SPSS Version 11.5 (SPSS for Windows, release 11.5.0 (2002) SPSS, Inc. Chicago Illinois).

7. RESULTS

7.1 Student Performance

Table 1 displays descriptive statistics for each groups' average performance on the pre/post content assessment. The difference scores did not differ between the three groups (KS ANOVA, p=.584).

Table 1. Comparison of Pre/Post Content (Mean ± SD)

Group	PreTest	PostTest	Difference
Treatment N=12	2.5 ± 1.8	6.1 ± 1.5	3.5 ± 1.5
Control Group1 N = 6	2.1 ± 2.2	6.1 ± 0.7	4.0 ± 1.8
Control Group2 N = 18	2.1 ± 1.6	4.8 ± 2.6	2.7 ± 2.9

Originally, this instrument was created using short answer questions. Multiple graders were trained at different institutions to score this exam. This resulted in poor interrater reliability and poor intrarater reliability from pre to post assessment. Therefore, our results, as presented in Table 1, must be interpreted with caution. Based on this experience, we have developed a multiple-choice pre-post assessment with the purpose of improving rater reliabilityThe revised instrument will be used in future studies, and is available at [18].

Table 2 displays the average grade for students in CS1 in the treatment and control groups as defined above, and also broken down into high risk, medium risk, and low risk subgroups. The Total column (far right of Table 2) shows the cumulative data for each group and subgroup over the two years of the study. A total of 25 students were in the Treatment group with an overall performance of 3.0 ± 0.8 GPA in CS1. A total of 30 students were in Control Group1 with an overall performance of 1.9 ± 1.3 GPA in CS1. A total of 52 students were in Control Group2 with an overall performance of 3.0 ± 1.2 GPA (P < .05, KS ANOVA). Students who were at risk and completed the Alice course were able to "hold their own" in CS1 whereas students who were at risk and did not complete the Alice course had lower performance levels in CS1. Most striking are the High Risk subgroups. The Treatment High Risk subgroup performed at a 2.98 GPA while the Control Group1 High Risk subgroup was considerably lower at 1.18 GPA.

Table 2. Grades (GPA) in CS1 in Each Group

Group	2001-2002		2002-2003		2-Year Total	
Treatment	N	GPA	N	GPA	N	GPA
High Risk	7	2.86	12	3.05	19	2.98
Med. Risk	2	3.65	4	2.93	6	3.17
Total	9	3.04	16	3.02	25	3.03
Control Group1	N	GPA	N	GPA	N	GPA
High Risk	10	1.32	2	0.50	12	1.18
Med. Risk	14	2.36	4	2.75	18	2.45
Total	24	1.93	6	2.00	30	1.94
Control Group2	N	GPA	N	GPA	N	GPA
Low Risk	19	2.28	3	3.33	22	2.43
Not At Risk	7	3.34	23	3.51	30	3.47
Total	26	2.57	26	3.49	52	3.03
Total of all students	N	GPA	N	GPA	N	GPA
	59	2.38	48	3.15	107	2.73

7.2 Retention

Table 3 displays the retention statistics over two years for students in CS1 by study group and risk level. Treatment group students experienced dramatically higher retention rates in CS1 (88%) than the control groups (47% for control group 1 and 75% for control group 2, $P < .05$, chi-square). As with GPA performance, the most striking contrast can be seen between the High Risk subgroups. The Treatment High Risk subgroup had a two-year retention rate of 88% while the Control Group1 High Risk subgroup retention rate was 15%. Thus, the Alice Treatment group outpaced other groups in the study in terms of retention.

7.3 Attitudes

Table 4 displays descriptive statistics for each of the subscales for the pre and post attitudes survey. A higher difference score suggests an improved attitude from pre to post assessment. The Treatment group had higher increases in attitude scores, except for usefulness, than other groups. Statistical analysis indicates attitude differences were not great enough to be statistically significant for confidence, liking, or usefulness (Kruskal-Wallis ANOVA, p=NS). But, a significant *decrease* was found in attitudes towards creativity (Kruskal-Wallis ANOVA, $p < .05$) in Control Group1. This suggests that at risk students who did not take Alice had more negative attitudes with respect to creativity in computer science after CS1.

Table 3. Percentage Retention in CS1 in Each Group

Group	% retained 2001-2002	% retained 2002-2003	2-Year Total
Treatment			
High Risk	86 (6/7)	83 (10/12)	84 (16/19)
Med. Risk	100 (2/2)	100 (4/4)	100 (6/6)
Total	89 (8/9)	87 (14/16)	88 (22/25)
Control Group1			
High Risk	10 (1/10)	50 (1/2)	15 (2/12)
Med. Risk	57 (8/14)	100 (4/4)	67 (12/18)
Total	37 (9/24)	83 (5/6)	47 (14/30)
Control Group2			
Low Risk	63 (12/19)	100 (3/3)	68 (15/22)
Not At Risk	86 (6/7)	78 (18/23)	80 (24/30)
Total	69 (18/26)	81 (21/26)	75 (39/52)
Total of all students	59 (35/59)	83 (40/48)	70 (75/107)

Table 4. Difference Scores for Attitudes Survey (Mean ± SD)

Group		Pre	Post	
		Confidence	*Confidence*	*Diff*
Treatment	N=7	14.6 ± 2.5	16.9 ± 4.2	2.3±3.4
Control Group1	N=5	17.6 ± 4.5	16.2 ± 3.1	-1.4±2.7
Control Group2	N=12	14.4 ± 4.2	15.8 ± 2.2	1.4±3.4
		Liking	*Liking*	*Diff*
Treatment	N=7	18.0 ± 3.6	20.9 ± 4.7	2.9±3.8
Control Group1	N=5	22.6 ± 6.0	22.2 ± 6.1	-0.4±1.9
Control Group2	N=12	18.5 ± 6.3	19.6 ± 3.5	1.1±6.8
		Usefulness	*Usefulness*	*Diff*
Treatment	N=7	12.7 ± 2.1	13.9 ± 1.2	1.2±2.1
Control Group1	N=5	15.4 ± 3.4	14.8 ± 1.8	-0.6±2.8
Control Group2	N=12	12.8 ± 4.0	14.1 ± 1.6	1.3±4.1
		Creativity	*Creativity*	*Diff*
Treatment	N=7	8.1 ± 1.7	10.4 ± 2.1	2.3±2.7
Control Group1	N=5	12.2 ± 3.1	9.8 ± 2.4	-2.4±2.5
Control Group2	N=12	8.7 ± 1.6	9.5 ± 2.3	0.8±2.7

7.4 Focus Groups and Interviews

Focus groups and individual interviews were also used, as a descriptive measure, to obtain attitude information from the treatment group. Students were queried regarding attitude and also for feedback on improving the course and instructional materials. It is clear from these interviews that students had a highly positive experience in the Alice course and that this experience had further stimulated their interest in computer science. Another encouraging outcome of these interviews is that all of the participating students indicated they would recommend this course to other students.

8. SUMMARY

Overall, the data presented in this paper supports the effectiveness of the Alice course for improving students' performance in CS1, retention within computer science, and attitudes toward computer science. Although the difference scores from the pre to post test for the content assessment did not yield significantly different results, we believe that this may be due to a flawed instrument that displayed low rater reliability. Future investigations will focus upon improving this instrument for measuring students' content knowledge. We did find that at risk students that participated in Alice, on average, received significantly higher grades than at risk students that did not participate in Alice. The overall result of two years of data shows that at risk students who completed the Alice course performed as well in CS1 as students who were not at risk. The Alice course seems to "level the playing field."

Furthermore, high risk students (those with both weak CS and weak math backgrounds) who participated in the Alice course displayed high retention rates while high risk students who did not participate in the Alice course exhibited low retention rates. At risk students that did not participate in Alice were the only group to display a consistent *decrease* in attitudes from pre to post assessment. This was statistically significant with respect to the creativity subscale. At risk students that participated in Alice (based on descriptive statistics) displayed the greatest increase in attitude subscales except usefulness from pre to post assessment.

As a proof-of-concept study, the data presented here is for small sample sizes, which makes statistical analysis more difficult. It is our intention to continue this work with larger groups of students to provide further support for the interpretations drawn above.

9. ACKNOWLEDGMENTS

This work was partially supported by NSF grant DUE-0126833. The authors would also like to thank Wanda Dann for her contributions to this paper as well as to the overall project.

10. REFERENCES

[1] Busch, T. (1995) Gender differences in self-efficacy and attitudes towards computers. *Journal of Educational Computing Research*, 12(2), 147-158.

[2] Clark, V., and Chambers, S. (1989) Gender based factors in computing enrollments and achievement: evidence from a study of tertiary students. *Journal of Educational Computing Research*, 5(4), 409-429.

[3] Cohoon, J. (1999) Departmental differences can point the way to improving female retention in computer science. *SIGCSE*, 198-202.

[4] Cooper, S., Dann, W., & Pausch, R. (2000) Alice: A 3-D tool for introductory programming concepts. In Proceedings of the 5th Annual CCSC Northeastern Conference 2000, Ramapo, NJ, April, 28-29, 2000.

[5] Cooper, S., Dann, W., & Pausch, R. (2003) Using animated 3d graphics to prepare novices for CS1. *Computer Science Education*, 13(1).

[6] Cooper, S., Dann, W., & Pausch, R. (2003) Teaching objects first in introductory computer science. *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, Reno, NV, February, 191-195.

[7] Dann, W., Cooper, S. & Pausch, R. (2000) Making the connection: Programming with animated small worlds. *Proceedings of the 5th Annual Conference on Innovation and Technology in Computer Science Education*, Helsinki, Finland, July 11-13, 2000.

[8] Dann, W., Cooper, S., & Pausch, R. (2000) Using visualization to teach novices recursion. *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education, Canterbury, England*,109-112.

[9] Dann, W., Cooper, S., Dietzler, K, Dragon, T., Ryan, K., & Pausch, R. (2003) Objects: Visualization of behavior and state. *Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education*, Thessaloniki, Greece.

[10] Dann, W., Cooper, S., & Pausch, R. (2004) *Learning to Program with Alice*. Prentice-Hall, to appear.

[11] Davy, J.D., Audin, K., Barkham, M. and Joyner, C. (2000) Student well-being in a computing department. *Proceedings of the 5th Annual Conference on Innovation and Technology in Computer Science Education*, Helsinki, Finland, 136-139.

[12] Division of Research, Evaluation and Communication, Directorate for Education and Human Resources. (1996) *The Learning Curve: What We Are Discovering about U.S. Science and Mathematics Education*. L. E. Suter (ed). Washington, DC:National Science Foundation,(NSF 96-53).

[13] Hagan, D., and Markham, S. (2000) Does it help to have some programming experience before beginning a computing degree program? *Proceedings of the 5th Annual Conference on Innovation and Technology in Computer Science Education*, Helsinki, Finland, 25-28.

[14] Loyd, B. H. and Gressard, C. P. (1996) Computer attitude scale. *Journal of Computing Research*, 15(3), 241-259.

[15] Morrison, M. and Newman, T. (2001) A study of the impact of background and preparedness on outcomes in cs1. *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education*, Charlotte, NC, 179-183.

[16] Seymour, E. and Hewitt, N. (1997) *Talking about leaving*. Boulder: Westview Press.

[17] Shashani, L. (1997) Gender differences in computer attitudes and use among college students. *Journal of Educational Computing Research*, 16(1), 37-51.

[18] http://www.sju.edu/~scooper/alicelec/alice.html