Introduction

This proposal addresses three issues in computing technology education in community colleges: (1) the need to reduce first-year attrition rates in computer science and information technology, (2) the need to increase the number of computer-related majors, and (3) the need for an effective programming and problem-solving component in computer literacy courses, especially those where technology-related majors are enrolled.

Attrition from computer studies majors has been a longstanding problem in higher education. The attrition rate is a particularly dramatic problem in community colleges where the practice of open enrollment is conducive to a higher number of "at-risk" students. Over the last 30 years, researchers have investigated causes for attrition in the first year of a computer science curriculum [6-10]. There appear to be two major issues: the set of skills (problem solving, logical reasoning, etc.) needed to succeed, and the lack of previous programming experience.

The need for computer-related majors continues to be strong. The popular press regularly shows that the "top" jobs for the next several years will be related to computers and computing. However, computers continue to be stigmatized as hard and "nerdy", and many students' first formal exposure to the computer as a problem-solving tool is negative.

Over the last decade, computer literacy courses have evolved to being "personal-productivity tools" courses where students learn to use word processors, spreadsheets, a simple database tool, and perhaps a brief introduction to building a web-page. Programming and problem-solving components have been dropped, or at least de-emphasized. This may be due to the difficulty of teaching programming to novices. For technology-related majors, this is not justifiable because the problem-solving aspect of the literacy course provides necessary skills for success in the technology major. Indeed, the National Research Council's (NRC) Committee on Information Technology Literacy has made a recent call for Fluency with Information

Technology (FITness) [5]. The NRC argues that FITness is a set of minimum qualifications every college student should have. FITness demands a basic understanding of the fundamental concepts of problem solving and skill in algorithmic thinking. The members of the NRC committee define the "…general concepts of algorithmic thinking [as] functional decomposition, repetition (iteration and/or recursion), basic data organizations (record, array, list), generalization and parameterization, algorithm vs. program, top-down design, and refinement." [5]

We propose an ATE Project designed to decrease attrition in introductory programming courses, to increase the attractiveness of computing, and to improve computer literacy courses by reintroducing a module in programming as a form of problem solving. The project is titled "**J**ava-based **A**nimation: **B**uilding vi**R**tual **W**orlds for **O**bject-oriented programming in **C**ommunity colleges", **JABRWOC**. This project will involve the adaptation of existing materials for the community college level. These materials, supported by an NSF CCLI grant, were originally developed for an introduction to programming in a pre-CS1 course in a 4-year college setting.

The proposed JABRWOC project will study the effects of teaching fundamental programming concepts (in computer literacy and introductory programming courses in community colleges, as well as in a gentle introductory setting to poorly prepared students) using simulation and visualization in a 3-dimensional (3D), interactive, object-oriented (OO), animation environment. This approach will take advantage of a high-level of interest in graphics, animation and storytelling -- commonly found among students who have grown up in a multimedia world. However, the major emphasis is the use of visualization to teach and learn a strong core of fundamental programming concepts and problem-solving techniques.

*Visualization***.** Part of the difficulty in teaching the necessary problem solving and logical reasoning skills in a programming class is that students are simultaneously learning the syntax of a programming language, the design of algorithms, and the connection between the two. Soloway [11] notes that the real difficulty for novice programmers lies in “putting the pieces together”, i.e. figuring out what constructs to use and how to coordinate those constructs. Novice programmers not only need to learn how to design an algorithm for solving a problem but also how to translate the steps of the algorithm into specific programming constructs. But, making the connection between an algorithmic step and a specific programming construct requires some concept of how the computer executes that programming construct at runtime. We have observed many students have difficulty visualizing the steps of the execution of a program, or the current state of the program at any given time [20]. As a result, students have trouble figuring out what went wrong when things do not work. The use of graphics and visualization is recognized as an effective teaching tool in computer science to help students put the pieces together [12].

Goals and Objectives

The overall goal of this project is to decrease attrition in introductory programming, to increase the attractiveness of computing and computing-related majors, and to improve computer literacy courses in community colleges. Introductory programming courses serve as first level courses for computer science and information technology majors. Also, introductory programming and computer literacy courses often serve as pre-requisites or co-requisites for specialized technology courses for other technology majors.

We propose to achieve the goal of decreased attrition by teaching fundamental programming concepts using simulation and visualization in a 3D, interactive, OO, animation environment. Lowered attrition and an increased attractiveness will result in an increase in the number of students who are interested and able to continue their pursuit of a degree in computer and information technology, as well as other technology fields. The ultimate result will be an increase in the supply of technicians to meet our nation's workforce computing and information technology demands. To meet our goals, five objectives are planned. They are:

1. Improving computing fitness. We propose to improve computing fitness by using 3D animation to teach general concepts of problem solving and algorithmic thinking in computer literacy, CS0, using a programming module as part of the course. Whereas many computer literacy courses have no programming component, this programming module will provide fundamental programming concepts in problem solving tasks where students create 3D animations. The highly motivating environment will provide a high level of student motivation and meet the guidelines specified by the NRC in their Fluency report.

2. Decreasing attrition in introductory programming courses. Our project includes creating or significantly changing a first programming course that will provide experience in the fundamental concepts of programming and logical problem solving in an interactive, OO, 3D animation environment. Students will be tracked to determine the project's impact on attrition. See the evaluation section of this proposal for more details concerning assessment.

3. Improving achievement in introductory programming courses. In synchronization with objective 2, a follow-up study of students in the traditional first computing course will determine the effect on achievement in their computer studies major.

4. Increasing the attractiveness of computing and computing-related majors. Community college students who have not passed mathematics and/or English proficiency requirements have a limited selection of electives to take. The majority of their classes are remedial mathematics and/or English classes, but they do have the possibility of taking elective classes, particularly once they have passed the proficiency requirement in one area. Offering a gentle and appealing introduction to computing may enhance their opinion of computing, and encourage them to pursue technology-related degrees.

5. Professional development in the use of innovative technology. Participating faculty from three different types of community colleges will be given the opportunity to further their growth in using innovative technology to enhance instruction in their computer literacy and introductory programming courses. The software and curriculum materials are further described in the following project description section of this proposal. The Project Description section of this proposal provides a detailed depiction of project activities designed to meet our goals.

Project Description

We propose to develop additional curricular materials and modify existing materials for use in 1) teaching fundamental programming concepts using a 3D, interactive, OO, animated tool, and 2) using that same tool for teaching problem solving as part of a computer literacy course. The goal is to better prepare students for solving problems with a computer, either 1) enhancing their chances for success in a computer studies major, or 2) helping to meet the programming requirement as part of the NRC recommendation for fluency. We will use an animation environment to help students visualize program execution. As opposed to a traditional, text-oriented programming language, a 3D virtual world has the advantage that most of the program state (like object position and color) is intrinsic and visible in a "natural" way. Also, 3D worlds are more realistic than their 2-D counterparts and seem to encourage student exploration in creating variants of existing worlds and making novel worlds of their own.

Description of Alice. The software to be used in this project is Alice [4]. Alice is a freely available 3D Interactive Graphics Programming Environment, developed at Carnegie Mellon University under the direction of Randy Pausch. (See [3], [1], [2].) A new Java-based version, available at the end of 2002, will run on any Windows/Mac/Unix platform. Alice is primarily a scripting and prototyping environment for 3D object behavior, designed to make it easy for novices to develop interesting 3D animations and explore interactive 3D graphics.

3D models of objects (e.g., animals and vehicles) populate a virtual world in Alice. Alice has a strong object-oriented flavor. By writing simple programs, programmers can control object appearance and behavior via mouse and keyboard input. Alice serves as a good programming environment for the novice. Students are immediately able to see how their animated programs run, affording an easy relationship of the program construct to the animation action. Figure 1 displays scenes from the running of a typical virtual world in Alice. In this program, a frog hops over to a ladybug.

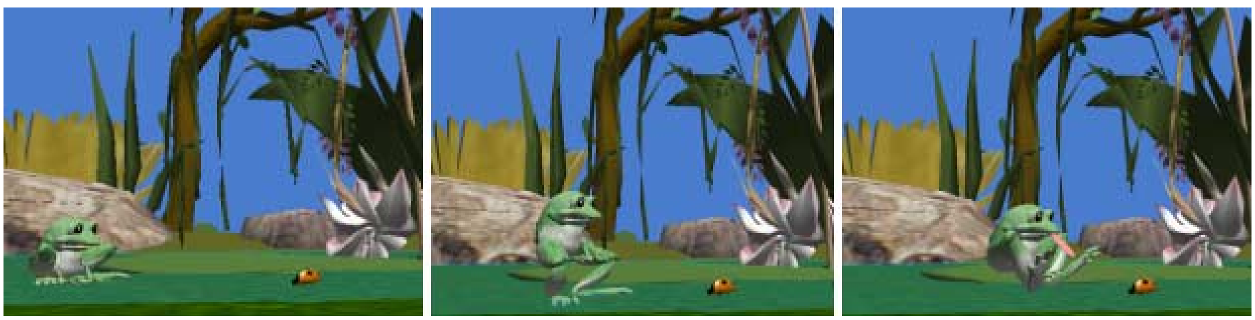

Figure 1. Animation of Frog Hop to Ladybug

Use of Alice in specific courses. We expect to use the Alice environment in three different types of courses, each of which is briefly described below.

1. A first course in programming to prepare students who have little or no previous programming experience to be able to succeed in a computer related discipline. Such a course is needed, as the typical first computer science course assumes that students have previous programming experience – or, the course moves at a sufficiently rapid pace so as to "lose" students who have never programmed before. In some community colleges, students who are interested in majoring in computer science or computer information systems but have little or no previous programming experience are given a class in some other language, e.g. BASIC. However, these courses experience similar high levels of attrition as the traditional CS1 course.

Alternative approaches are needed to help the novice learn about programming and programming constructs.

2. A class that students in need of remedial education can take prior to beginning the computer science or information technology curriculum. At the community college level, many students must take up to a year and a half of basic skills classes (demonstrating mastery of these basic skills) before they are permitted to begin an associates degree program. Many of these students, for financial aid reasons, need to maintain full-time status and thus more preliminary courses are needed in the curriculum. More importantly, to maintain motivation while completing their basic studies, these students want to take a course "in their major", or wish to try out different majors before having mastered mathematics and English skills.

3. A programming and problem-solving component as part of a computer literacy course, as prescribed by the NRC call for Fluency. When programming is taught as part of a computer literacy course (and often programming is not taught at all), students generally learn some Javascript or perhaps Visual Basic. (But, Javascript is interspersed with HTML and students may lose the concept of a program. VB helps students learn the widgets/event-driven aspect of programming but students may lose the concepts of sequence and program.)

Current Status. Two members of this JABRWOC project team, Cooper and Dann, have created materials for an introduction to programming with problem solving course. These materials have been successfully used prior-to/concurrent-with introductory computer science (CS1) at the university level. A NSF-CCLI grant is funding this work. Materials include lecture notes/slides, laboratory and project assignments and solutions, and sample exams and solutions. At the time of this writing, Dann, Cooper and Pausch have a contract with Prentice-Hall for publication of a textbook. Several other 4-year institutions (e.g., Bucknell and Duke University) are using these materials for teaching similar courses at their schools. Testing and evaluation are still in progress but early results of using the Alice materials are more than promising. (Appendix E contains the results of the first year of our study.) An article describing the study and early results has been accepted for publication in the refereed Computer Science Education Journal.

Modification of existing materials. The current instructional materials need to be adapted for use in a community-college setting. Most important is the need to prepare smaller self-contained instruction units. This would allow community college faculty to eliminate, at their discretion, some of the more advanced topics (e.g., generative recursion and iterators) and put greater emphasis on other topics. The pace of presentation of the materials needs adjustment – some topics will need more time, requiring more demos/lab exercises. Also, several new modules will be created for community college needs. For example, a unit on file management and file manipulation may be added for novice students working in a Windows environment.

Some of the pedagogical approaches may need to be modified. For example, the emphasis on pair programming/small team projects may not be possible in a community college, as the students, commuters, may not be able to complete projects with peers outside of class.

The Community Colleges

(Please note that different colors will be used throughout the remainder of this document to identify information specific to each of the three community colleges in this project.) Founded in 1967, Camden County College (CCC) is a public two-year community college in New Jersey. CCC has three main campuses, covering urban as well as suburban areas. CCC serves an enrollment of more than 12,000 credit students and roughly the same number of non-credit students. Programs cover technical fields like automotive technology and mechanical engineering; health professions like nursing and medical coding; and liberal arts and sciences like English and chemistry. CCC is a comprehensive community college dedicated to offering academically excellent and affordable public education.

Community College of Philadelphia (CCP) is the largest institution of higher education in Philadelphia County and is the city's only community college. Founded in 1964, CCP is a two-year, open-admission urban institution that provides transfer, career, development and continuing education to the 1.4 million residents of the city. CCP is the largest single point of entry into higher education for minorities in Pennsylvania. CCP offers associate degrees in Arts, Science, and Applied Science and more than 70 career and transfer programs in Business, Humanities, Allied Health, Science and Technology and the Social and Behavioral Sciences. CCP is accredited by the Middle States Association of Colleges and Schools and is also approved by the Council of Higher Education and the Department of Education of the Commonwealth of Pennsylvania. The College has served nearly 500,000 students over its 38-year history – and approximately 80% of graduates have remained in the greater Philadelphia area to earn their living. The average salary for Community College of Philadelphia graduates is more than $33,000. This data confirms that the College plays a tremendous role in educating the city’s workforce, as well as providing education that allows for employment with a self-sustaining wage. CCP’s curricula serve a variety of student needs. Career programs prepare students for immediate employment in their chosen field or for transfer and further study; almost 75% of graduates in transfer programs continue on to four-year institutions.

Programs/services in the Division of Community Services and Continuing Education bring credit and non-credit offerings to adults who do not choose to participate in, the College’s traditional degree programs. CCP offers Distance Learning, Customized Job Training, a General Educational Development (GED) Course, and no-cost Adult Basic Learning and Education (ABLE) to all qualifying Philadelphia residents. CCP’s Center for Business and Industry Training works closely with employers in the Greater Philadelphia area to create and deliver the programs they need to develop an educated and trained workforce.

Founded in 1967, Tompkins Cortland County College (TC3) is a public two-year community college, operating under the supervision of the State University of New York, offering associate degrees and certificates in the fields of allied health, business, hospitality, liberal arts, public service, and science and technology. Off-campus programs are offered in Ithaca, Cortland and Owego NY. TC3 is known throughout the SUNY system as an innovator. The League for Innovation in Community Colleges recently named TC3 as a "Learning College Champion." TC3 not only has a reputation within New York State but is also well-known through its international partners. TC3 recently received Middle States re-accreditation without qualification. TC3 intends to register its distance-learning program under the Distance Higher Education Initiative of the New York State Education Department's Office of College and University Evaluation. The major responsibility of this New York State Program is to provide quality assurance for the degree and certificate programs offered through distance education by public and private colleges and universities in the state.

The following tables describe the programs offered at the community colleges, as well as a profile of the students, in greater detail. Table 1 provides details on the college faculty, technology-related programs of study, courses taken by technology-related majors, technology requirements, deficiencies of current programs, and current attrition rates as related to this proposed study. Table 2 summarizes information on student demographics in the three community colleges. Importantly, the three community colleges have been selected to provide a spectrum of settings (rural, suburban, and city) and a diversity of populations (traditional college age and adult learners) and racial and ethnic groups. The information regarding deficiencies in current programs and current attrition rates will form the baseline data for this study.

Table 1. Community College Technology-Related Majors and Curricula

	Camden County College (CCC)	Community College of Philadelphia (CCP)	Tompkins Cortland Community College (TC3)
Faculty	The Computer Studies department consists of the following programs: Computer Science (CS), Computer Information Systems, Computer Systems Technology, and Computer Graphics. The Computer Studies program consists of 3 tenured and 1 non-tenured full-time faculty. The CS department has a total of 10 full-time faculty and over 50 adjuncts.	The 2001-2002 CIS Department faculty included 12 tenured full-time faculty, 4 non-tenured full-time faculty, and 56 part-time instructors. Members of ethnic minorities represent about a third of the faculty and about one in four are female.	The CIS program has 2 full time members and the CS program includes 1 full time faculty member. The department includes a full time Education Technology Associate.
Technology related majors	Computer related technology-related majors: Transfer Programs (Liberal Arts and Sciences) Computer Science A.A Information Systems A.S. Computer Graphics A.A. Career Programs Computer Information Systems A.A.S. Computer Information Systems: PC Track A.A.S. Computer Graphics A.A.S. Video Imaging A.A.S. CADD A.A.S. Computer Integrated Manufacturing/Engineering Technology A.A.S.	The Computer Information Systems Department at Community College of Philadelphia offers 5 degree programs: 4 A.A.S. degrees in Computer Information Systems (CIS), which are vocationally-oriented and 1 A.S. degree in Computer Science (CSCI) which is transfer-oriented. The CIS degrees include Local Area Network Administration, Internet Operations and Development, Computer Applications, and Computer Programming. Currently there are approximately 1,800 students majoring in CIS and another 160 majoring in CSCI. An additional group of roughly 400 students majors in “Culture, Science and Technology with an Interest in Computers”, a transfer-oriented	Computer Forensics A.A.S. (52 majors in Fall 2002), Computer Information Systems A.A.S (88 students), Computer Science A.S. (14 students), Electrical Technology: Electronics and Computer Systems A.A.S (28 students), Engineering Science A.S. (34 students).

	Camden County College (CCC)	Community College of Philadelphia (CCP)	Tompkins Cortland Community College (TC3)
	Computer Systems Technology A.A.S. Engineering Technology A.A.S. Photonics A.A.S.	program (developed under NSF grant NSF-DUE-9652153, 1996-2000) that includes 15 to 21 credits of computer courses, depending on electives.	
Computer classes taken by technology majors	CS: Students take a two semester programming sequence in C++ (incorporating the material covered in CS1), CSC121 and CSC122. They then take a class in data structures in C++ (CSC223). Business Administration: Students take CSC121, CSC122, or CSC161, or they can take a programming sequence in Java (CSC161, CSC262). Computer Information Systems: Students take either one of the above mentioned programming sequences in C++ or Java, or a programming sequence in QBASIC/VB (CSC111, CSC112, and CSC213) Computer Information Systems: PC Track: Students take the above mentioned programming sequence in QBASIC/VB. Computer Integrated Manufacturing/Engineering Technology: Students take a two semester programming sequence in PLC (CIM211, CIM212). Computer Systems Technology:	The CSCI curriculum mirrors the first two years of the four-year Computer Science degree guidelines issued by the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronic Engineers (IEEE). It emphasizes algorithm development, data structures, and computer architecture, and includes a strong mathematics component with 18 credits in Calculus, Discrete Mathematics, and Linear Algebra. All CIS majors must complete a core of CIUS courses, with CIS 106 – *Introduction to Programming* and CIS 103 – *PC Applications*, usually being completed in the first semester. All CSCI majors usually complete CSCI 111 – *Programming and Algorithm Development I* in their first semester.	Computer Science: Students complete computer applications and database design courses in addition to a heavy emphasis on Mathematics. The programming courses required include an introduction to programming using Visual Basic (CSCI 160) and two semesters of Java (CSCI 165 and CSCI 201); assembly language is required as well. Computer Information Systems: Students complete computer applications, networking, operating systems, and database design courses. In the programming area, students study Visual Basic in CIS 108, take two semesters of C++ programming (CIS 213 and 223), take one semester of Java (CIS 225) and complete a Web Programming course (CIS 227). Computer Forensics: In addition to courses in networking, operating systems, security system design and intrusion detection software,

	Camden County College (CCC)	Community College of Philadelphia (CCP)	Tompkins Cortland Community College (TC3)
	Students take one of the above mentioned programming sequences in C++, Java, or QBASIC. Engineering Technology: Students take a semester of programming in QBASIC. Photonics: Students take a computer programming elective.		students are required to take CIS 108 as well as CIS 213. Engineering Science: Students in this major are required to take CSCI 160. Electrical Technology: Students in this major are required to take CSCI 160.
Technology requirements by all majors	There is no college-wide technology requirement, per se. However, many programs require that their students take a class in computer literacy.	There is no College-wide requirement for students to complete a computer course, though all degree programs address this issue within their individual curricula. Most programs include at least one computer course, but there are a few, such as Early Childhood Education and Photographic Imaging, which use an infusion model, including computer-related topics in discipline specific courses. Of the programs that require a computer course, most require CIS 103 – *PC Applications*, while Mathematics, Engineering, and most physical science majors take CSCI 111 – *Programming and Algorithm Development I.* CIS 103, comprises one third of instruction in the Department. During the 2001 calendar year, 4,004 students enrolled in CIS 103. This course currently includes instruction in the Windows operating system, Microsoft Word, Microsoft Excel, and the use of the Internet and e-mail. The	All degree programs at TC3 must satisfy a list of 11 General Education Goals established by the faculty in Spring 2000. Goal #4 requires that all students will be able to "Use computer technology effectively for communication, research, and data management." Most programs require that students take at least 2 credits of Computer Applications courses, one in Word Processing, and the other either in Spreadsheets or Databases. Some programs meet this goal by infusing the use of these computer applications into the degree program itself. In either case, all students at TC3 are exposed to these two applications (word processing/data management) at a minimum. In addition, almost all students are required to take Academic Writing

	Camden County College (CCC)	Community College of Philadelphia (CCP)	Tompkins Cortland Community College (TC3)
		instruction in Word and Excel mirrors the core level in each of these areas in the Microsoft Office User Specialist (MOUS) certification program. Other introductory computer courses include CIS 101 – *Computer Literacy*, a lighter version of CIS 103 that spends more time on Windows, the Internet and e-mail and less time on Word and Excel, and CIS 100 – *Introduction to Computers,* a one credit introduction to using a PC for Word Processing, the Internet, and e-mail.	(ENGL101) which includes a 3-day information literacy workshop run by our Instructional Librarian. This workshop includes instruction in the use of computers and databases in information literacy. Also, many of our students enroll our distance learning classes, or, have traditional courses that include online components.
Deficiencies with existing programs	Many students taking CSC111 and some students taking CSC121 have little or no preparation in terms of basic operating system fundamentals such as file management, installation of programs, and computer configuration. We have difficulty getting students to think like problem solvers and programmers. Students tend to memorize rather than understand. Many students suffer a high level of frustration when left on their own to write programs. Students do not seem to recognize the need for, and application of, what they are learning in the introductory course.	The high attrition rate among students who study computer programming. The difficulty in introducing object-oriented concepts to people who have developed procedural approaches to programming. The absence of a look at programming (object-oriented programming in particular) in any computer literacy courses at the College.	Many students enter TC3 with a desire to complete a computer-related degree, but lack the necessary preparation. Many students have had little exposure to computer programming at the high school level. In addition, the math/logic preparation of students can be weak, especially in the CIS degree program. The high rates of attrition in introductory programming courses, as demonstrated in the previous section, suggest that students have difficulty understanding the concept of formalizing logic in the form of an algorithm or a computer program. We have found that students often

	Camden County College (CCC)	Community College of Philadelphia (CCP)	Tompkins Cortland Community College (TC3)
			have more difficulty expressing an algorithm as a generic solution (in the form of pseudocode or a flowchart) than they do actually coding a solution, where the results can be verified by a compiler. A tool that will allow students to visualize program logic could greatly enhance the students' ability to express solutions to problems in the form of an algorithm, creating the opportunity for increased student success in the introductory programming courses.
Attrition from programming classes	The attrition rate in CSC111 (defined as those students who do not earn a grade of C- or better) was 53% in the 2001-2002 academic year. In CSC121, the attrition rate was approximately 50%. In CSC122, the attrition rate was 50%. Over the past 5 years the average attrition rates were 47% in CSC111, 51% in CSC121, and 56% in CSC122.	The attrition rates in CIS 106 and CSCI 111 are both around 30 % (do not complete the course with a grade of "C" or above) In both courses, 15 to 20 % of the students withdraw before completion and another 10 to 15 % fail the course. The student retention rate, as measured by the number of students enrolling in a second programming course, is also low. During the 2001-2002 academic year, for example, only 93 of 421 students in CIS 106 continued on to CIS 116 and only 88 of 238 students in CSCI 111 continued on to CSCI 112.	Approximately 56% of students do not pass (earn at least a D) the introductory CIS class, and 42% do not pass the introductory CS class. Please see Appendix A for more details.

Table 2. Community College Student Demographics

	Camden Community College (CCC)	Community College of Philadelphia (CCP)	Tompkins Cortland Community College (TC3)
Student Population Profile	Currently, there are 6000 full-time students, and another 6600 part-time students. There are also 12,000 non-credit enrollments. CCC is the largest county college in New Jersey.	A current profile of the College's student body shows that of the 41,000 students, In terms of financial need, approximately 65 % of full-time students receive some type of financial aid.	For 2000-01, enrollment was 2378 FTE. The enrollment in the current year is expected to be approximately 2640 Full-Time Equivalent students (FTE).
Students' Age	The mean age is 27.	Approximately, 54 % of students are non-traditional college age, that is, more than 25 years of age.	Students age 21 and under make up 27% of the CIS majors, and 58% of the CSCI majors.
Students' Sex	58% of the college enrollment is women.	63% of the college enrollment is comprised of women	46% of CIS and 25% of CSCI students are female.
Students' Ethnicity	64% of CCC's students are Caucasian, 19% are African-American, 6% are Hispanic, and 5% are Asian.	Approximately 48 % of CCP's students are African-American, 28 % are white, 11 % are Latino, and 6 % are Asian-American.	75% and 68% of CSCI majors of CIS majors are white. See Appendix B for detailed ethnic breakdown.
After Graduation	In 2000, 931 students graduated.	More than 95 % of the computer science graduates transfer to a 4-year institution, with about 60 % transferring to Drexel, and others to La Salle, University of Pennsylvania, Penn State, Rutgers, St. Joseph's University, Temple, and Villanova. About 15 % of the Computer Science graduates transfer to a variety of colleges and universities other than these eight, including West Chester University, Arcadia University, and Massachusetts Institute of Technology.	92% of graduates were either employed or continued their education after graduation. Approximately 65% of graduating CIS students obtained jobs in a computer-related field (e.g., computer support, computer or network technician, database programmer) The companies who hire these graduates include Borg-Warner, Cornell, and CBORD Group. About 1/2 the graduating CS students transfer immediately, to schools such as RIT and SUNY Binghamton.

Specific Implementation Strategies

CCC proposes to develop a course in programming concepts aimed at students who have been identified as having a weak mathematics background and little or no programming experience. This course would also be appropriate for (and will likely be heavily populated by) students who are still at the basic skills level and need to take additional courses to complete their schedule. Students who take this course will be tracked through subsequent programming courses to determine the effect of the new course on their achievement in those courses relative to students who have not taken this course.

Specifically CCC hopes to:

- Decrease attrition in subsequent programming courses
- Improve performance of students in subsequent programming courses
- Reduce need for instructors in the various programming courses to spend time teaching basic concepts such as operating system and file management
- Give students hands-on experiences with an instructor present to help get them over any initial anxieties and unfamiliarity related to problem solving and programming.
- Attract students to Computer Science and Information Technology careers

CCP proposes to use Alice early in CIS 106 and CSCI 111 (introductory programming classes taken by the CS and CIS majors, as well as by several other technology majors) to introduce the object-oriented programming paradigm. CCP also proposes to use Alice to teach instructors to adopt an “object-oriented first” approach to programming as opposed to introducing the object-oriented paradigm from the viewpoint of one who first understands procedural programming. It is expected that units in Alice will comprise approximately 5 weeks of material. (Please see Appendix C for a description/syllabus of these courses as they currently exist, and with the proposed modifications.)

CCP proposes to use Alice in CIS 101 and CIS 103 to expose students to a modern object-oriented understanding of what computer programming is. It is expected that the 4-5 week unit on Alice will replace a 5-week unit on the Internet and e-mail. (Recent student surveys show students are least satisfied with this section of the course, as most already have sufficient background in these topics. To accommodate students who do not have sufficient background, a non-credit unit may be offered.)

The CIS Department believes that, based on available information on the current success with Alice elsewhere, Alice could be used to introduce students to object oriented programming concepts in a manner that both reduces dropout rates and increases students' understanding of object-oriented concepts. It expects to change the way instructors and students understand computer programming, and consequently, to change the nature of computer programming in a revolutionary way based on the object-oriented paradigm in and of itself, rather than on an understanding of it as viewed from a procedural programming foundation.

At TC3, Alice will be used during the first five weeks of the Introductory Programming courses in the CIS and CS curricula. These courses are CIS 108, Introduction to Computer Information Systems, and CSCI 160, Introduction to Computer Programming. (Please see Appendix D for a comparison of the courses as they currently exist as opposed to how they will be with the Alice units included.) The software will be used to introduce the students to concepts of algorithm and program logic, and terminology used in object-oriented design and programming environments.

TC3 would like to improve student success and retention in the Computer Information Systems, Computer Forensics, and Computer Science programs at TC3 in the following ways:

- increase the passing rate in CIS 108 (from 44% to 70%)
- increase the passing rate in CSCI 160 (from 58% to 75%)

- increase the rates of retention from CIS 108 to CIS 213 to have more students elect to stay in the CIS program and not be "turned off" by the introductory course
- increase the rates of retention from CSCI 160 to CSCI 165
- improve student performance in CIS 213 (indicated by higher pass rate)
- improve student performance in CSCI 165 (indicated by higher pass rate)

TC3 faculty have much experience in developing and implementing online courses. TC3 expects to develop online versions of CIS 108 and CSCI 160 that make use of the Alice instructional materials. TC3 believes this to be a unique opportunity to test the use of Alice tutorials and instructional materials in a distance learning setting.

As described earlier, it is expected that all three community colleges will start with the successful materials developed by Cooper and Dann. (See Appendix E for initial results at the 4-year school level.) These materials will be modified, and new materials will be created as well.

The Team Partnership

The JABRWOC project embodies a partnership between three community colleges, two 4-year colleges, and two universities. The project team membership includes a project leader at each of three community colleges: Eric Howd at TC3, a rural community college, William Taylor at CCC, a suburban school (that also has an urban campus), and Charles Herbert, chair of the computing department at CCP, an urban school. Dr. Wanda Dann at Ithaca College, Dr. Stephen Cooper at Saint Joseph's University, Dr. Barbara Moskal at Colorado School of Mines, and Dr. Randy Pausch at Carnegie Mellon University round out the team. Drs. Dann, Cooper and Pausch are the authors of the 3D animation text and instructional materials. Dr. Moskal, an assessment expert, will be developing/selecting the assessment instruments, evaluating the effectiveness and appropriateness of these instruments, administering these instruments and interpreting results.

Dr. Pausch, at CMU, will support the software. As part of this grant, CMU will be responsible for providing periodic releases and updates for software maintenance. CMU will provide a mechanism for people to report bugs in the software and will track the bugs for software updates. This is necessary to provide a community of support for instructors and students by responding to email and providing a common ground for sharing ideas, 3D models, and examples. In addition, CMU will create and maintain an online gallery of 3D models for use in instructional materials. It is important to develop models that reflect the diversity of our ethnic and racial heritages as well as multiple cultural backgrounds.

The budget for this grant is high. It is important to note that the budget is in support of an unusually large partnership of 7 schools. Additionally, this grant expects to impact more than 6000 students and 60 faculty members. Thus, the project scope is expansive.

Deliverables

The deliverables will consist of instructional materials, and a report of the results of testing and implementing the materials. The instructional materials will consist of lecture notes, lab exercises covering all of the topics (with solutions), and sample programming project assignments. They will be provided online, with searching capabilities. Prentice-Hall has expressed an interest in packaging Dann, Cooper and Pausch's text [14] with computer literacy texts, for use in a computer literacy course. It is not expected that a new text will be necessary as part of this project. The Alice software is freely available from the web [4]. Pausch and the Stage 3 development team at Carnegie Mellon University are actively involved in modifying the software for use in an educational setting as a pedagogic tool. They will continue to work with this project team to modify the software and to make such modifications freely downloadable.

Timeline/schedule

Summer 2003 (starting July 1st)

Team Meeting: All Co-PIs and other representative community college faculty will meet at Carnegie Mellon with the development/support team. The purpose of this meeting will be to discuss the different approaches to be used by each of the community colleges, have a representative from CMU demonstrate some of the subtle features of the Alice system, and for Cooper and Dann to present their current curricular materials, and discuss their experiences in using those materials.

CCC: Taylor will work with Cooper to master the software, and to modify existing materials to create a new course for students who have not yet demonstrated mastery of basic mathematics and/or English skills. Taylor will also be responsible for training a teaching assistant.

TC3: All TC3 faculty will attend the team meeting at CMU. All faculty will then work with Dann in developing instructional materials for use in one section of CSCI 160 and one section of CIS108.

CCP: Herbert will work with Cooper to master the software, and to modify existing materials, and to create new materials as appropriate. Work will begin on both the development of units for the literacy course, as well as for the incorporation into the introductory computer classes. Atkins will assist Herbert with curricular issues, and in preparing for approval by the college-wide curriculum committee. Other CCP faculty will receive training on the use of Alice.

Fall 2003

Materials Development: CC faculty will prepare instructional materials for community college student needs. Dann and Cooper will cooperatively work with Taylor, Howd, and Herbert in a support role. Moskal will cooperatively work with Taylor, Howd, and Herbert to prepare procedures, pre-post tests, surveys and other materials that will be used to collect data for evaluating the project effectiveness. Taylor, Howd, and Herbert will provide feedback to Pausch to enhance the list of 3D models needed for the 3D model gallery. Pausch will provide support for the stencils tool, used to develop tutorials.

CCC: Taylor will finish developing materials, and CCC will offer 2 sections (to be capped at 12 students per section) of the new course. Taylor will work with Cooper and Moskal in preparing pre and post-tests, and in setting up surveys and other materials to be used to determine the course's effectiveness. The assessment materials will not be used this semester. Taylor and Cooper will provide Pausch with initial feedback (especially in regards to 3D models). Cooper will spend time at CCC helping out as necessary. One TA will assist in class, and be available 4-6 hours per week in an open lab.

TC3: CSCI 160 and CIS 108 sections will be run. Howd will work with Moskal in determining the evaluation materials to be used for the Spring semester, and in set up of necessary environments. Howd will attend the winter meeting.

CCP: Herbert will continue work on developing curricular materials, and revising curricula. Herbert and other faculty will attend winter team meeting.

Team Meeting: During the winter break, the team of Co-PIs will meet. It is expected that each community college will share their experiences to date, and will work as a group to deal with individual problems. Also, preparations for the first pilot test will be made.

Spring 2004

Pilot Testing: Instructional materials will be tested in one or two sections of a course on each community college campus.

Support: 3D modelers at CMU will generate additional models for the software gallery. Dann and Cooper will provide support for instructional materials, CMU team for software support, Moskal for evaluation support. Pausch will work with CC faculty to track software bug reports and prepare maintenance updates.

CCC: 2-3 sections of the new course will be offered. A pilot test will be run, and feedback will be collected on the instructional materials developed.

TC3: Sections of each course will be run again, with initial evaluations to determine the effectiveness of the pilot test.

CCP: Herbert will run a pilot test of the curricular materials in a computer literacy section. Other faculty will pilot 1 section each of programming incorporating Alice.

Summer 2004

Evaluation: The data collected during the pilot testing will be evaluated.

Revision: Based on evaluation of the pilot test, instructional materials will be updated and revised.

Team Meeting: The first semester's data, where appropriate, will be discussed and evaluated. Prepare for full-year testing.

CCC: Taylor and Cooper will revise instructional materials. Additional faculty will be trained on Alice, and on how to use the materials that have been developed for the new course. Faculty will have the opportunity to customize the materials as appropriate. Two additional teaching assistants will be identified and trained. A dedicated computer lab will be built.

TC3: All faculty will work at analyzing and assessing the data collected during the spring semester. They will revise curriculum materials. Howd and faculty will attend the summer team meeting.

CCP: Herbert will work with Cooper and Moskal on incorporating appropriate assessment materials for future testing of the material's effectiveness. Herbert and other faculty will make curricular revisions based on experiences in using Alice in the past Spring semester.

Fall 2004 – Spring 2005

Two-Semester Testing: Individual faculty will teach all their sections of a course using the revised instructional materials. This test will be larger than the pilot test in that more sections will be involved and the testing will be carried out over two semesters. But, not all sections of a course will be using this approach.

CCC: 6-8 sections of the new class will be taught. Expansion to the Camden (urban) campus will occur. All sections will be evaluated.

TC3: Alice will be implemented in all campus-based sections of CSCI160 of CIS108, with continued assessment.

CCP: First full test of materials, with full assessment, to be conducted by Herbert and other faculty in literacy and programming classes.

Summer 2005

Evaluation and Revision: Collected data will be analyzed and instructional materials will undergo a second revision.

Training Session: Training sessions will occur on individual community college campuses. Experienced faculty on each campus will conduct a training session for other faculty in their department or a nearby community college. Participants may include adjunct faculty who will be teaching classes that use the instructional materials.

CCC: Workshops will be offered for faculty at neighboring community colleges who are interested in creating a similar course at their campuses. (A sample letter of interest is included from Gloucester County College.) Part-time faculty at CCC will also attend this workshop. Instructional materials will continue to be revised as appropriate.

TC3: All faculty will work on the design and delivery of an online course in Alice to accompany the online sections of CSCI160 and CIS108. Faculty and Howd will assess the data collected during evaluations and make curricular changes as appropriate.

CCP: Continued work on revision of curricular materials based on assessments. Herbert and Cooper will coordinate and run a training session for 48 adjunct faculty for use of Alice in all computer literacy sections for the next year. Cooper will select and train graduate students from SJU to serve as TAs.

Fall 2005 – Spring 2006

Full Adoption: All instructors of all sections (for courses involved in this project) will adopt and use the instructional materials in this approach.

Follow-up: A follow-up check on the progress of students who participated in the 2004-2005 two-semester testing will be completed.

CCC: 10-12 sections of the class will be offered. Students who took the class in earlier years will be tracked to determine the course's effectiveness in recruiting technology majors.

TC3: Continued use of Alice in all sections of CSCI 160 and CIS108. Implementation of online course including Alice. Presentation of results at such conferences as CCSCNE and SUNY CIT.

CCP: Continued use by faculty of Alice in programming sections. Widespread trial of Alice as part of computer literacy course. Evaluation and assessment of the success of Alice in both environments.

Summer 2006 (Ending June 30th)

Summative Evaluation: Final analysis of data will be conducted. A summary report will be prepared.

For National Dissemination

Summer 2006 (Beginning July 1st)

Repository: A searchable repository of lecture notes, lab exercises, tutorials, and related instructional materials will have been created.

Fall 2006 – Spring 2007 (requires additional support)

Workshops: Conducted on-site at Community College locations around the US.

Team meetings occur twice a year (during summer and winter breaks). Team meetings are designed to provide a cooperative working venue for the Co-PIs, faculty, and the development team. The meetings will be used to organize and carry out the functioning of the overall project. It is expected team meetings will be a critical component of the project – a way to exchange ideas and motivation, face-to-face. Initial training session objectives include mastering the technology, and engaging in the pedagogy of the current instructional methodology. Preparation of materials will entail both adaptation of current instructional materials and development of new materials. For example, community college faculty will develop tutorials specifically designed for the community college student. The pilot test is intended to be a trial run to test the integration of the instructional materials into course structure and to get feedback from students as to what works and what doesn't. The goal of enhancing the gallery of 3D models is to provide diversity in racial and ethnic themes.

Innovative Aspects

There are several innovative aspects about this proposal, some of which are listed here. The software tool is a 3D animation tool specifically designed for the novice programmer. Course materials are being developed specifically for community college students that include traditional, adult, and minority students. CCP is planning to simultaneously run 50 sections of a computer literacy course, all of which use Alice, and the majority of which are taught by adjunct faculty. TC3, a leader in distance learning, is planning to create distance-learning modules of Alice instruction. CCC is planning to involve several retired senior citizens as teaching assistants -- thus building a partnership between its oldest community members and its youngest.

Experience/capability of PIs

We believe that we are uniquely qualified to run this investigation. Dr. Dann has twelve years experience in teaching introductory computer science. Her expertise spans applied research in program visualization, curriculum development, and workshop presentations for event-driven programming. Dr. Cooper has been teaching introductory computer science for ten years. He also has 10 years of industry experience, thus having good knowledge about what industry "needs" in the way of computer science graduates. They have been working with Alice for 5 years, and along with Dr. Pausch have authored a text on learning to program with Alice that will be published by Prentice-Hall in Fall 2003. Our experiences with using and teaching computing concepts with Alice enable us to complete this project. The software developers of Alice at CMU work as a team with the textbook authors and will be a dynamic part of this grant. Participating faculty in this grant will make regular visits to meet with the development team to discuss the design of Alice and how to better tailor it to the needs of novice programmers. The CS education community is quite interested in our Alice work, as they have accepted several of our papers for conferences. (See the PIs' biographical sketches for a complete list.)

Dr. Moskal, an expert in educational assessment, will oversee the implementation of the assessment process throughout this project. Dr. Moskal has extensive experience in assessment and evaluation and has worked as an assessment consultant on two other NSF-funded computer science curriculum development projects.

Mr. Taylor has more than 15 years experience teaching at the community college level. And with more than 10 years at CCC, has the respect and support of his colleagues and administration to implement this project. Mr. Howd is experienced in supporting college faculty in the use of technology and has an extremely close relationship with the computer studies departments at TC3. Mr. Howd also has experience in preparing distance learning materials, which will be invaluable to the team. Mr. Herbert also has significant teaching and curricular development experience at the community college level.

Dr. Pausch and his Stage 3 research and development team have created Alice and have significantly modified the software over the past 10 years to work with novice programmers (supported by numerous federal grants). The Stage 3 team is part of the prestigious Human-Computer Interaction Institute and Computer Science Department at Carnegie Mellon.

Evaluation/Assessment

Formative evaluations will be qualitative in nature and will be used to improve project-related activities. Qualitative techniques have the advantage of providing detailed descriptive information. This type of information is useful for project improvement purposes. The formative evaluation techniques used here include: background survey, peer review of curriculum materials, student interviews, student focus groups, and workshop surveys

Quantitative research techniques will be used to support summative evaluations. Summative evaluations examine whether project related objectives are being reached. For purposes of comparison, two types of control classrooms will be used. The first will be computer studies students without previous programming experience taking the existing introductory computer science class. The second control will be freshmen computer science majors completing their first college programming course who have not taken the Alice course. Summative evaluations will include attitude surveys, pre/posts, and retention statistics.

Background Survey. In pilot and control classrooms, a survey will be given to determine students' background in mathematics and computing. Data collected by the survey will be used to examine relationships between computer preparedness, performance and attrition.

Peer Review of Curriculum Materials. This project will result in the development of curricular materials. At least three external computing studies experts will review all materials and revisions will be made based on recommendations. Peer review is a widely accepted technique for examining the content, construct and criterion validity of instructional materials [13].

Student Focus Groups. Students participating in this study will be asked to participate in focus group activities. Questions in these groups will focus on their experience in using Alice and their perspective on how it affected their understanding of programming concepts. The focus group will ensure students have a chance to express their viewpoint.

Workshop Surveys. At the conclusion of the summer and winter workshops, a reflective survey will be distributed to participants. This survey will evaluate, through self-report, the effectiveness of the workshop for providing participants with the information they need to implement the materials in their classrooms. A second survey will be mailed to participants a year after the workshop. The purpose of this survey is twofold: to determine how many of the participants have used the materials in their classrooms, and for those that did use the materials, to evaluate effectiveness.

Attitudes Survey. Attitudes surveys will be used for both the programming groups of students as well as for the literacy students. (For the computer literacy students, the control group will be sections of the computer literacy course that do not include modules using the project approach.) At the beginning and end of the semester, students in both pilot and control classrooms will complete a computer attitude survey. For the computer studies students, this survey will help determine whether students' participation in their first computing course has further stimulated their interest in pursuing a degree in computing. As part of this project, an appropriate instrument to measure student confidence in problem solving will be composed and tested.

Statistical comparisons will be made between pilot and control classrooms, providing evidence on the impact of the new course on students' computer science interests. Comparisons will also be made between female/minority students in pilot and control classrooms.

Pretest/Posttest. In a collaborative effort between the investigators and assessment consultant, a pretest/posttest assessment instrument will be developed for use in programming classes. The researchers will identify fundamental concepts in programming and problem solving as typically taught in introductory computer science courses. The resultant instrument will be included in the expert review, described in the Formative section of this proposal. Statistical comparisons will be made between pilot and control classrooms' performances on the instrument. Once again, comparisons will be made between female/minority students in pilot and control classrooms.

Retention Statistics. Statistical comparisons will be made of the students in the pilot and control classrooms that select to continue to pursue a computing degree. This information will suggest if the newly developed course contributes to student retention in computer science. Descriptive comparisons of female/minority students in the pilot/control classrooms will be made.

Impact

Demand for programmers, technicians, operators, software engineers, etc. in the US is expected to increase over the next decade. An aim is to increase the attraction and retention rate among students who are interested in computer technology-related majors. Another goal is to lower the attrition rate, especially among the most vulnerable students (those with poor programming background). It is also hoped that, using animation, students will become more immersed in their computer studies and will increase their overall effort. One other benefit is the attraction of non-majors into the computer studies majors.

Future plans

Future plans include more widespread use (both in community colleges and 4-year schools) of materials that have been and will be developed. It is expected other colleges will use the materials in other ways and are interested in studying their effectiveness. Another planned avenue is high schools teachers' involvement to modify materials for use in their classes. Other plans include using the materials to encourage women/African-Americans to pursue careers in computer science. There are strong connections between female learning and arts and graphics [15, 16] and between storytelling and African-American learning/culture [17, 18, 19]. It will certainly be possible to focus on arts/graphics aspects of the 3D animation software or on storytelling aspects.

Dissemination of results

Dissemination activities will begin immediately through a project web site. Presentations will be made at local and national conferences (e.g., SIGCSE, ISECON, CCSCNE, the Community

College Computer Consortium of New Jersey). Papers will be submitted to computer science educational journals, such as Computer Science Education and JERIC. Instructional materials including laboratory exercises will be made available on the web site.

Results from prior NSF support

Drs. Cooper and Dann are currently supported by NSF-DUE 0126833 (1/02-6/03, $75,000) for their project "Decreasing attrition using animated virtual worlds." Results of the first semester of work are available in Appendix E. They show that the use of Alice significantly lowers attrition. A much larger and detailed study is currently underway. Publications include an accepted article to be published in the Computer Science Education Journal, a contract with Prentice Hall to publish our textbook (see supplementary documentation for letter of support), and submitted articles to SIGCSE 2003 and ITiCSE 2003.

Dr. Moskal is currently supported by NSF-DUE 9987037 (8/00-7/04, $496,995) for the project "Improving the retention of women and minorities through research experience, mentoring and financial assistance." Early results of this project suggest that project participation is having a positive impact upon the retention and advancement of female and minority students. Additionally, many of the undergraduates have reported that project participation has stimulated their interest in pursuing advanced degrees. Published results include [20-26].

Dr. Pausch is currently supported by NSF-IIS 9812012 (9/98-8/03, $498,028) for the project "Interaction techniques for high-dimensional spacial data." This research has led to two major contributions: A set of 3d interaction techniques, many of which were embodied in Jeff Pierce's dissertation, and continued development and distribution of the Alice authoring toolkit. Published results include [2, 27-29].