

## Introduction

This proposal is a full development project that builds on our successful proof-of-concept CCLI grant. In this project, we plan to revise and fully develop the materials that were prototyped (in the proof-of-concept pilot project) and tested at Saint Joseph's University and Ithaca College. Equally important, this project is designed to extend usage and testing to at least 10 other schools, encompassing different types of institutions (private and public, large and small, HBCU, and research university) and serving a much wider variety of students. In this document, we describe our earlier grant, summarize our results, and detail our proposed expansion.

This proposal is to use *program visualization* in a powerful, high-level animation programming environment as a methodology for introducing programming to beginners. (*Program visualization* is the use of graphic visualization to directly relate lines of program code with actions that objects perform on-screen. Early examples of program visualization are Logo [31] and Karel the Robot [32, 3].) The target population for this project consists of two primary groups: (1) computing *majors* who have little or no previous experience in programming and poor mathematics backgrounds and are thus at *high-risk* of dropout or failure in the first course in programming for computer science majors (CS1), and (2) non-CS majors who are taking a course in programming (computer literacy with a programming component or a specialized computing course for non-CS majors). These target populations have been selected as test groups to determine the effectiveness of our approach on (1) reducing the attrition level of high-risk majors, (2) improving attitudes toward computing for those not typically attracted to courses in computing, and (3) attracting students into computing as a major or minor.

**Motivation:** A primary motivation for this project is evidence of the need to more successfully retain students in the computing major. On average, at least one half of college students majoring in computer science withdraw from the field [39]. While attrition occurs throughout the 4 years, the majority of students drop out during the first year [36]. Thus, there is only a small window in which changes can be made that will be effective in reducing attrition. Recent studies [12, 17] have shown students without prior programming experience are at a decided disadvantage in being able to complete a CS degree. Two major issues appear to be critical in determining success in the first year of a computer science program. The first is the set of skills (problem solving, logical reasoning, etc.) [5, 37, 30] and the second is the quantity of programming concepts that students must learn – a factor that we believe has been made less manageable by the adoption of object-oriented programming as the first programming paradigm.

A second, but equally important, motivation for this project is evidence of the need to develop more successful teaching strategies and materials for reaching a wider range of students who are taking first courses in programming. In an age where the demand for Information Technology (IT) professionals is rising [15] and the need to think algorithmically [29] in many career occupations is critical, we have a need to increase the motivation of students to take a first course in programming. But, non-CS majors are well known to have a dislike of programming courses. In addition, enrollment of women in computer science courses is declining. The American Association of University Women 2000 report [2] claimed that computer science courses are “overly technical” and are downright boring. Margolis and Fisher, in *Unlocking the Clubhouse*,

[25] point out that women prefer to work with real applications for computing and are not attracted to computing just for the sake of computing.

**Focus:** The focus of this proposal is to use a high-impact, high-interest program visualization environment, Alice, to introduce a strong core of fundamental programming concepts (objects, classes, methods, functions, decisions, variables, parameters, recursion, arrays, and events) and problem-solving design techniques. Animation programming exercises and projects will allow students to author on-screen movies and games, where the concept of an "object" is made tangible via on-screen characters that populate a three-dimensional virtual world.

**What is Alice?** The software to be used in this project is Alice [6, 33]. Alice is a freely available open source 3D interactive graphics programming environment, developed at Carnegie Mellon University (CMU) under the direction of Randy Pausch [33]. A new Java-based version runs on Windows. As part of this project, we plan to port the software to Macintosh, and Linux, and to provide mechanisms to run and interact with Alice programs via WWW browsers.



**Figure 1. Animation of figure skating**

In Alice, 3D models of objects (e.g., people, animals and vehicles) populate a virtual world. Alice programs, which have a strong object-oriented flavor, allow students to control the appearance and behavior of objects, have objects respond to mouse and keyboard input, or do any sort of computation that would normally be done in an introductory programming class. Students are immediately able to see how their animated programs run, affording an easy relationship of the program construct to the animation action. Alice is supportive of collaborative programming and encourages the development of algorithmic thinking and problem solving skills. Figure 1 displays scenes from the running of a typical virtual world in Alice. In this scene, a skater performs a traditional figure skating action.

Alice is a fairly large jump in the state-of-the-art for environments that support novice programmers. As such, its user interface is very difficult to describe adequately in static prose. Therefore, we strongly urge interested reviewers to download Alice from <http://www.alice.org> and try it out, or at least to refer to Appendix A to view screen captures of the interface in action.

Students use Alice drag-and-drop words (e.g. “if”, “while”) to form programs that are equivalent in expressive power to Java/C++/Pascal class languages. However, students are protected from making syntax errors by the programming environment, which only allows students to drag and drop tiles of program components to syntactically correct locations. Alice provides full support for all common control structures (e.g. if/then/else, while, for all), and provides data types including 3D-object, Number, Boolean, String, etc.). Alice provides both array and list aggregate data structures, and allows students to write arbitrarily large methods and questions (pure methods/functions

that do not have side effects) that take parameters of any type. Alice supports recursive method calls, limited polymorphism, and a rich interactive event structure for creating interactive worlds (programs) and characters (classes of objects that can be reused in other worlds/programs).

The design of Alice is driven by several key concepts: (1) *Make as much state visible to the student as possible.* Most changes are immediately visible (e.g. “set the color of the frog from green to red”). (2) *Animate all changes of state.* Students are able to see the change of state, whenever possible, through an animation: objects that are told to change their position move through space; objects that are told to change their color animate through color space, etc. (3) *Do not allow ill-formed programs at any time.* Alice’s sophisticated user interface allows students to build extremely complex programs that are always syntactically valid. Student’s errors are thus semantic rather than syntactic. (4) *Reify the notion of an “object.”* For many students, objects are very ethereal, abstract concepts. The “objects” in Alice are clearly visible on the screen: the Ice Skater object shown in Figure 1 is clearly an object in the 3D world, and students find that creating and associating methods with such visible objects (such as Spin, Jump, and Bow) is more intuitively obvious. (5) *Use 3D graphics to engage the students.* Please see Appendix A for a further description of Alice.

**The proof-of-concept grant:** To address the problem of attrition, Drs. Cooper and Dann wrote a proof-of-concept grant to develop a text and instructional materials for a pilot course, *Programming in Alice*, designed for incoming freshmen computing majors. The pilot project used program visualization (with the Alice software) to help students

develop an understanding of important programming concepts. The pilot course (and the associated text and instructional materials) has several key components:

- We used this strong visual environment to support an objects-first approach (described in the ACM and IEEE-CS Computing Curricula 2001 report [42]). The visual environment enables students to “see” their programs as they run. In a sense, we offload the mental effort from the student’s cognitive system to his or her perceptual system. It is much easier for a student to see that an object has moved backward rather than forward, as opposed to noticing that the “sum” variable has been decremented, rather than incremented. Today’s students are immersed in a world where interactive, three-dimensional graphics are commonplace; we attempted to leverage that fact without pandering to them.
- The instructional materials made heavy use of design. Design is a key element of programming that we believe cannot be taught too early. The specific design technique we used is storyboarding. The analogy of making a movie allows us to utilize the storyboard concept, which students know is an established movie-making process. Since most students cannot draw their storyboards, we encourage them to express their high level story in short prose snippets, which are progressively refined, essentially designing through pseudo-code and stepwise refinement.
- We used a large number of hands-on exercises. In addition to lectures and demonstrations, we conducted up to 50% of the class in a laboratory setting.
- We used open-ended assignments. Such assignments allow students to maximize their creativity while demonstrating mastery of the appropriate concepts.

- We emphasized teamwork. Many projects were done in pairs and some projects with teams of three or four.

This approach was specifically aimed at engaging students in learning computer science in a highly motivating and fun – but very serious – introduction to programming. A copy of a sample syllabus used in the proof-of-concept grant is available in Appendix B.

**An expanded, full study proposal:** This project is a full grant proposal that follows our successful proof-of-concept grant. In the prior proof-of-concept grant, we successfully demonstrated (as described below) that the approach worked to reduce attrition of high-risk CS-majors in two campus settings with two instructors. We also performed a peer review but have completed no formal testing of the materials in other settings. In this full project, testing will be extended to regional test sites, with many instructors, and in varying course curricula. Our hypothesis is that this approach will work in a larger setting, improving retention of high-risk CS-majors and improving attitudes toward computer science for non-CS majors as well as for CS-majors. We also hope the approach will act as a positive influence in helping undecided students to become computing majors or minors.

### **Results of 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." At-risk freshmen intending to major in computer science at Saint Joseph's University (SJU) and at Ithaca College (IC) were encouraged to take the Programming in Alice course (using the pilot materials developed in this grant) either prior to, or concurrent with CS1. Tables

1 and 2 below summarize data over two years of testing and data collection for students in CS1. The categories used in the tables are: *All*, *Test*, and *Control*.

The *All* category includes all students enrolled in CS1 during the test period. The *All* category includes majors and non-majors while the *Test* and *Control* groups include only subgroups of majors. The *Test* group category consists of majors who were deemed **high risk** and **took both CS1 and Alice** courses. The *Control* group category consists of majors who were deemed **high risk** and **took only CS1**. **High risk** is identified as having little or no previous programming experience and poor mathematical background (which we have defined as not yet being ready for Calculus). Historically, high-risk students have low retention rates in CS1.

Table 1 summarizes achievement of students in a rigorous CS1 course. The GPA figures are based on grades expressed in a scale of A = 4.0. Table 2 summarizes retention based on the percentage of students continuing on to CS2.

**Table 1. Achievement in CS1 – average GPA, based on 4.0 = A**

|                | 2001-2002  |      | 2002-2003  |      | Two-year Aggregate |      |
|----------------|------------|------|------------|------|--------------------|------|
|                | # students | GPA  | # students | GPA  | # students         | GPA  |
| <b>All</b>     | 49         | 2.49 | 84         | 3.08 | 133                | 2.86 |
| <b>Test</b>    | 11         | 2.8  | 14         | 3.14 | 25                 | 2.99 |
| <b>Control</b> | 10         | 1.3  | 4          | 1.83 | 14                 | 1.45 |

**Table 2. Retention**

|                | 2001-2002 | 2002-2003 | Two-year Aggregate |
|----------------|-----------|-----------|--------------------|
| <b>All</b>     | 65 %      | 48 %      | 54 %               |
| <b>Test</b>    | 91 %      | 86 %      | 88 %               |
| <b>Control</b> | 10 %      | 50 %      | 21 %               |

This study was conducted with strict adherence to factors in human subject research studies. It is important to note that the investigators in this study taught the Alice courses but *did not* teach the CS1 class and thus had no control over student grades in CS1.

**Peer Review:** Three peer reviewers reviewed the textbook and instructional materials. Also, each of the reviewers used the materials with their classes and provided information as to student reactions to the materials. Feedback from reviewers and their students was anecdotal, rather than a formal data collection for quantitative analysis.

**Anecdotal Findings:** Major anecdotal findings from our own observations and the peer review process were: (1) coverage of the first seven chapters of the book was easy to accomplish in a much faster than expected pace, (2) students found the book readable and the explanations clear, (3) students were “turned-on” by the 3D animation environment, and (4) students were highly-motivated – often spending extraordinary amounts of time on open-ended projects. We particularly note the positive reaction by women and African-Americans.

**Textbook and publications:** We have a contract with Prentice Hall to publish the textbook (see supplementary documentation for letter of support). See Appendix C for a copy of the table of contents for our text. Three papers have been published and presented: at the Visualization workshop [11], SIGCSE [7] and ITiCSE [8]. An article has been accepted and will appear in the Computer Science Education journal [9]. Also, a website, [www.sju.edu/~scooper/fall02csc1301/alice.html](http://www.sju.edu/~scooper/fall02csc1301/alice.html) has been created to maintain copies of lecture notes and support materials for the course.

**Outreach program for minority students:** Dr. Cooper is also supported by a private foundation and by NSF-DUE 0220499 (9/02-8/06) for the project PATHways to Careers in

MAthematics and Computer Science (PACMACS), an outreach program for city of Philadelphia minority high school students who have shown promise in mathematics. The program is designed to encourage minority youth to pursue careers in mathematics or computer science. A key component of the program is that students take the Programming in Alice course at SJU. Over half of the students who have succeeded in PACMACS have chosen to major in computer-related fields in college.

**Related retention study:** Dr. Moskal, the evaluator for the proposed project, 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. Many of the undergraduates have reported that project participation has stimulated their interest in pursuing advanced degrees. What has been learned from this project concerning the retention and advancement of woman and minorities will be used to inform the efforts of the proposed project. Published results include [18, 20, 23, 26-28, 38]. An impressive result of this project is that, thus far, participating students have published 39 peer reviewed conference or journal papers (see [http://www.mines.edu/fs\\_home/bmoskal/scholprog/meetings/meeting03.html](http://www.mines.edu/fs_home/bmoskal/scholprog/meetings/meeting03.html) for a complete list). Dr. Moskal has also acted as an evaluator on a variety of different computer science education projects (see biographical sketch), providing her with the appropriate knowledge and background for leading the assessment effort on the proposed project.

**Related 3D interaction techniques study:** Dr. Pausch was previously supported by NSF-IIS 9812012 (9/98-8/03, \$498,028) for the project "Interaction techniques for high-dimensional spatial 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 [6, 10, 34, 35].

## **Goals and Objectives**

In our proof-of-concept study, as described above, we developed a textbook and instructional material to support teaching and learning fundamental programming concepts using simulation and visualization in a 3D, interactive animation environment. We pilot tested the text and instructional materials at two different institutions with two instructors. Our instructional materials were also examined by three peer reviewers, and used by their students, who provided written feedback regarding the quality of the text and instructional materials.

Highly positive results from our pilot testing provide a foundation for an expanded study, based on what we learned. Our overall goal in this full project is to revise and expand the instructional materials and to formally test their effectiveness with faculty and students at other types of schools in regional settings.

**Objectives:** Our goal is to build on the success of our NSF proof-of-concept grant by accomplishing three major objectives:

*(1) Fully develop and revise instructional materials based on our pilot project experience.* Certainly, suggestions for change (as a result of our experiences in the classroom and from student feedback) need to be implemented. Also, our publisher and

reviewers have recommended that our materials can also be used, with modifications, as part of a computing class for non-CS majors.

*(2) Actively work with faculty at regional tests sites in different types of institutions for a wider test group of students from diverse backgrounds and with various career goals.* We expect to share our instructional materials with others in a two-year approach. In the first year, a core faculty group at regional test sites will be selected as regional leaders and will pilot courses using the text and instructional materials in their own institution. The regional leaders will then work with us in the second year to recruit other faculty in nearby schools to pilot courses using these materials.

*(3) Develop, test, and implement improved evaluation tools to enable effective assessment of our materials with a wider test group.* The expansion of focus to include courses for non-CS majors (as well as courses for CS majors who have little or no previous programming experience) leads to a need to create assessment instruments for a different type of student (a non-CS major). Thus a revised instrument is needed.

**Anticipated results from this full project study:** A critical part of our objectives is to implement an evaluation plan to confirm our results in tracking the effectiveness of this approach in reducing attrition in CS1. The use of our materials with larger numbers of students at different universities will enable us to collect a larger set of data to see if our results are validated when we expand to other universities.

Additionally, we have seen that some non-computer science students in our courses have been attracted to major (or minor) in computing or a computer-related field. This was not part of our original study and so we have no reliable data on the effectiveness of this approach for attracting non-CS majors. This has become an

additional factor we wish to track as part of a full grant study, thus our expansion of focus to include courses for non-CS majors.

Another study that we wish to conduct in this full proposal is the ability of our Alice materials to attract and/or retain women and minorities. In our pilot study, we noted anecdotally that this approach is attractive to women. As stated in the introduction, evidence has been presented to show that women undergraduate CS students are interested in real applications for computing rather than computing for the sake of computing. Our approach to introducing programming has a use of computing in which “programming for a purpose” is built-in. One objective of this full project study is to add examples and exercises in our instructional materials that will improve our ability to demonstrate real applications of computing. Open-ended assignments can be used to give non-CS majors an opportunity to develop animations to simulate concepts in their domain of interest. For example, a geology student can simulate an earthquake and a chemistry student can simulate a molecular reaction. In this way, students will be introduced to the idea that computing is a way to work with examples and data that they care about. The hypothesis of this effort is that this approach used in courses for non-CS majors will result in improved student retention, improved attitudes toward computing and computer science, and will attract students to computing majors and minors.

Dr. Cooper has also observed the use of Alice as a possible cause for many of the minority high-school students in his PACMACS program choosing computer-related majors for college. By expanding the approach used in this grant to other settings, we hope to get sufficient statistical data to confirm or deny this observation.

## Detailed Project Plan

**Current status:** Based on the success of their Programming in Alice course, SJU and IC have added this course into their computer science curriculum. After attending presentations at SIGCSE and at ITiCSE, several faculty members from other colleges have also requested access to the Alice materials for their own courses. Four different schools have contacted the PI about the possibility of writing a CCLI A&I grant to modify the materials developed as part of the proof-of-concept grant.

**Modification of existing materials and development of additional instructional materials:** In response to our presentations at conferences and published papers, we have received much valuable feedback from faculty. Additionally, reviewers of our instructional materials have offered suggestions for revision of existing materials and development of additional materials. We summarize below the more significant modifications and additions that will be completed as part of this full grant project:

- 1) *Instructional materials to support transitioning from Alice to C++ or Java.* This is probably our most important, and most-needed, addition. For faculty members using Alice with students interested in becoming computer science majors or minors, we need to support a transition to Java or C++ (the most likely languages they will see in CS1). We are already adding a chapter to our textbook for such a transition. Corresponding lecture materials and laboratories will be needed!
- 2) *Developing materials that visualize an introduction to arrays and lists.* Several faculty reviewers believe we should take advantage of Alice's visual environment to help introduce collections of objects. This is a step beyond our prototyped instructional materials and involves a more abstract concept. We plan to explore

- possible visualizations of list and array structures and visual objects positioned within these structures.
- 3) *Greatly increased coverage of logical expressions.* The focus of our current materials is on *if* and *while* statements; using logical expressions only as needed to express the appropriate condition to solve the particular problem being presented. Feedback we have received indicates to us that others would like to specifically teach logical expressions using Alice.
  - 4) *Sharing of characters/worlds and placing student projects on a web page.* This has been our most-requested modification. While there is nothing pedagogically difficult about such sharing (and students can share characters/worlds by means of external disks), allowing virtual worlds to be placed on the WWW is extremely important for those who wish to incorporate Alice into courses for non-majors.
  - 5) *Significantly increasing the number and variety of high-impact demonstration worlds.* This is again primarily to create examples of interest to non-computer science majors. Instructors who are teaching specialized CS1 courses for non-CS majors need example worlds that are meaningful to the specific disciplines their students are studying. Suggestions include: example worlds that illustrate sine and cosine waves, simulated earthquakes, worlds illustrating Newtonian mechanics, interactive worlds that teach playing the piano, a flight simulator, etc.
  - 6) *Significantly increasing the number of exercises and projects as instructional resources.* Our prototyped materials have a minimal number of exercises and projects. From our work with students in our classes and with undergraduate students who worked in summer projects supported by our proof-of-concept

grant, we now have a development base that can be used to dramatically increase the number of exercises and projects for instructional resources.

7) *Developing and testing a model for open-ended project assignments.* We have found that open-ended assignments allow students much more room to express their creativity. Open-ended assignments have been cited as the “best part of the course” by many of our female students.

It is clear that the need exists for using the instructional materials in a wider range of courses than we had originally identified in our proof-of-concept grant. For programming components of computer literacy courses and for specialized CS1 courses for non-CS majors, we need to develop slower paced versions of our lecture materials. A slower pace allows the same material to be introduced without “watering down” the fundamental programming concepts.

**Development of tutorials:** A Ph.D. student at CMU, Caitlin Kelleher, has created a tool for building tutorials; these tutorials are presented as a graphical overlay on the running user interface as a "stencil" with holes cut in it. Holes in the graphical overlay allow students to interact with the running program, while constraining what they can do – thus, keeping them on track with the tutorial's lesson. The Alice interface is somewhat different from word processors and web browser software familiar to most students. Interactive tutorials can be used to ease the beginner into a first exposure to a programming environment. Also, we plan to create prototype tutorials designed for faculty – to guide faculty through some of the optional customizable parts of Alice and to help faculty adapt the tool to their needs.

**Development of demonstrations:** Another kind of instructional material we plan to add to our library of online resources is a short video that captures demonstrations of building and programming worlds with certain fundamental program structures. We have a tool that will allow us to capture the Alice interface and mouse motions during program construction. We plan to create prototype videos in QuickTime and RealPlayer formats.

**Establish regional test sites:** The use and testing of instructional materials will be done in two phases. The idea is to start with a core group of leaders in the first phase and then have the leaders serve as mentors for their regions during the second phase. To implement the first phase, three regional test sites will be established. We already have four faculty members (two of whom will serve as co-leaders for one region) who have agreed to participate as regional leaders: John Dougherty from Haverford College, Susan Rodger from Duke University, and Pamela Lawhead from the University of Mississippi and Constance Bland from Mississippi Valley State University -- the latter two will together serve as co-regional leaders. We have chosen these faculty members because they have previewed our Alice materials and have used the Alice software in classes or special summer projects, though no formal study has been conducted. Dougherty, Rodger, Lawhead, and Bland are, therefore, ideal candidates to serve as regional leaders for a formal study. Haverford, Duke University, the University of Mississippi, and Mississippi Valley State University represent a wide variety of institutions. Haverford College is an elite private liberal arts college, using Alice as part of a literacy course for non-majors. Duke University is a prestigious private national university, using Alice to teach computing to non-majors. The University of Mississippi is a state university, where students are not typically as strongly prepared as the students at Haverford or Duke.

While they have used Alice in short special programs, they are looking to use Alice in a similar way as was done by the PIs in our proof-of-concept grant. Mississippi Valley State University is a historically black college looking to use Alice in similar ways as the University of Mississippi. Drs. Bland and Lawhead have been working together on using innovative techniques, such as visualization, to help beginners learn to program (see for example their work with robots [21]), and have created courses for both of their universities. The regional leaders' classes will provide a variety of settings for testing our materials with students of diverse backgrounds and career goals. By using such widely varying universities, we will be able to discern how well our approach works with these different student populations.

At the start of the first year, a team-meeting session for the regional leaders will be conducted. (Later team meetings will be held every other month by teleconferencing.) Team meetings are designed to provide a cooperative working venue and will be used to organize and carry out the functioning of the overall project. Team meetings will be a critical component of the project, a way for co-PIs and regional leaders to exchange ideas and motivation. Initial team session goals include mastering the technology and engaging in the pedagogy of the current instructional methodology. We will offer demonstrations and share our experiences in using our materials, and help regional leaders prepare for their role as a mentor for other faculty in the second year of the project.

During the first academic year, the regional leaders will use the instructional materials in a course at their own school. We will provide assistance and model the mentor role. Also, we will work with the regional leaders to identify colleagues at neighboring colleges who may be interested in teaching with this approach.

**Recruitment of faculty members to attend the regional workshops:** The PIs will work with the regional leaders to recruit faculty from neighboring universities to attend the workshops. Specifically, we plan to recruit in the following manner: provide handouts at SIGCSE 2004 and ITiCSE 2004, possibly as part of a special session at SIGCSE 2004, by posting to the SIGCSE mailing list, from the Alice online communities, and from individual contacts.

**Regional workshops:** During the latter part of the project's first year, we will collaborate with the regional leaders to prepare and conduct a two-day workshop at the regional site. Each regional workshop will be offered for up to ten faculty colleagues from at least three neighboring schools (including the regional leader's own). The regional workshops will present the basics of using the Alice software, possible courses where the materials may be used, and strategies for how best to utilize the materials. Workshop attendees will also commit to using our evaluation materials to assess the effectiveness of this approach in reducing attrition of high-risk CS-majors, improving (non-CS major) student attitudes toward computing, and attracting new computing majors and minors.

Instructors, whom we know are knowledgeable programmers, can learn enough about Alice in a one-day workshop to be able to use Alice with their classes. The Alice development team has worked with IBM-Canada to create one-day workshops that have been successfully used with high school teachers. We expect to modify the Alice development team's one-day workshop and use it as the first day of the two-day regional workshops. The second day of the workshop will focus on the curricular and assessment materials, and strategies for using them.

**Mentoring.** Mentoring is a key factor in the success of the regional test sites. The idea for using 3D animation and visualization to teach beginning programming is novel, and faculty attending the regional workshops may be unfamiliar with the Alice software (except for possibly attending Cooper and Dann's conference presentations at SIGCSE or ITiCSE). A major motivation for setting up regional test sites and regional leaders is to provide a sub-community of support for other faculty as they implement the instructional materials into their courses. Faculty will be able to turn to the regional leaders as an immediate contact point. During the first year of this project, regional leaders will gain expertise from the team meeting in Ithaca and their own teaching with this approach in classes in their own setting. This will help prepare the regional leaders for situations other faculty may encounter (e.g. how to conduct open-ended project assignments).

Certainly, the co-PIs will also provide support to regional leaders during the first year and to all participating faculty in the second year. We also plan to have regular teleconferences with the regional leaders to discuss this project on at least a bi-monthly basis. In these teleconferences, we expect to share lessons learned as the project proceeds.

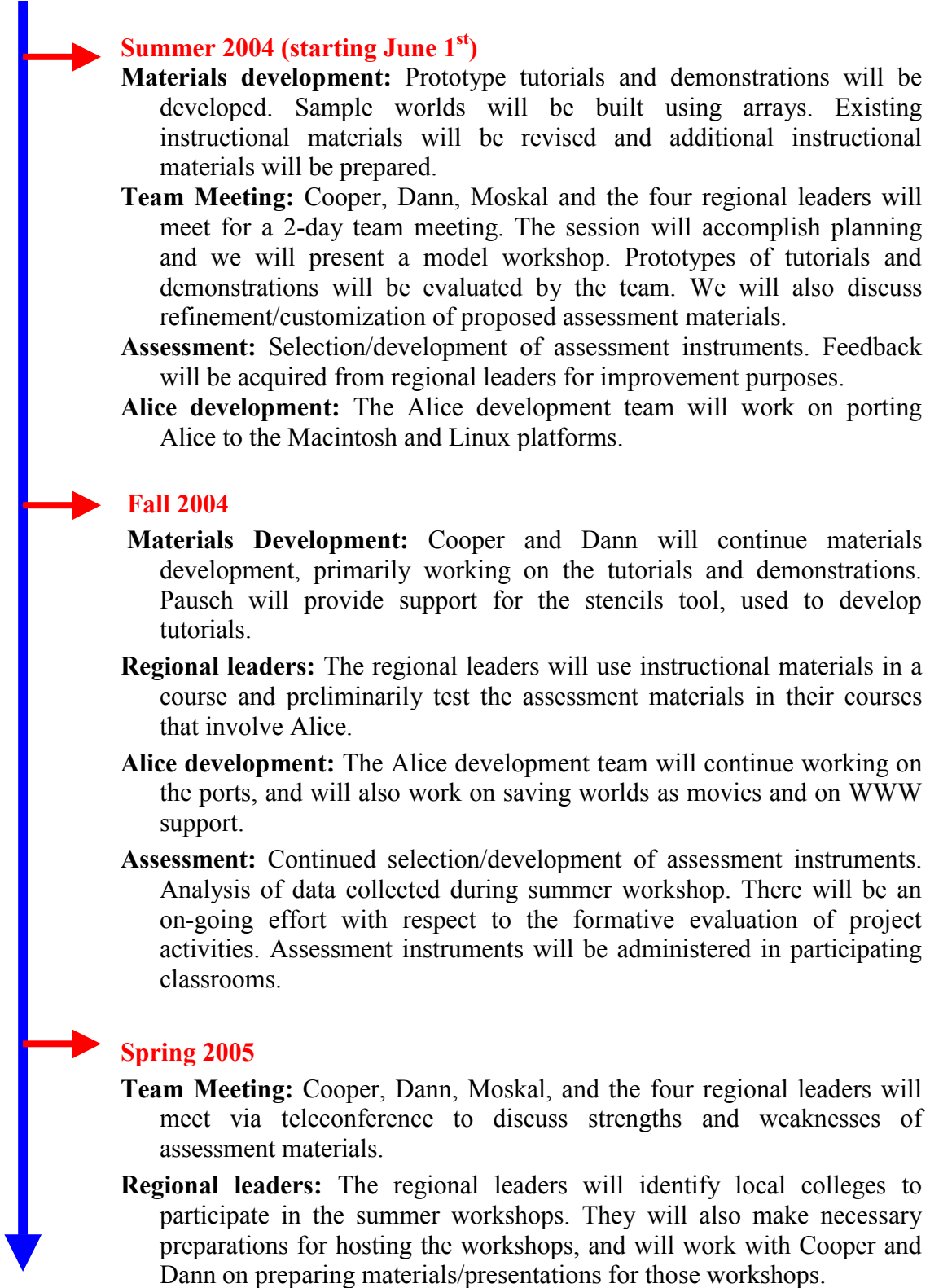
The Alice development team also plans to establish on-line communities to provide support for faculty as well. The website <http://www.alice.org> will contain a "community forum," which is a highly-organized electronic bulletin board with archival and search mechanisms for maintaining multiple threaded conversations on various topics. This will allow faculty members to share lesson plans and particularly engaging and effective exercises, to share solutions for any problems they encounter with the software, and to provide a venue for discussion of aspects of the system that are most pedagogically valuable, as well as which features they like added to the Alice system.


**Modifications to the Alice software:** At CMU, Dr. Pausch and his team 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, eventually numbering several thousand models, for use in instructional materials, all of which will be in the public domain and freely usable. It is important to develop models that reflect the diversity of our ethnic and racial heritages as well as multiple cultural backgrounds, and the library will do so. This team will also enhance the current initial “person builder,” that allows students to create highly stylized human beings, of any sort, to their own specification. The largest modifications to Alice will be porting the software to the Macintosh and Linux platforms, and creating support for students to share their creations. The porting of Alice to the Macintosh has already received an enthusiastic response from Apple Computer, which has agreed to provide detailed technical assistance in this effort, as Apple would like to use Alice in their current educational initiatives.

Allowing students to share their creations takes two forms, both using the WWW. First, students will be able to create mpeg or QuickTime movies of their worlds, which they can place on their WWW pages. For interactive worlds, we need a technologically more advanced solution, and we are developing a WWW mechanism where Alice worlds can be transferred across the Internet and played interactively in a WWW browser.

## Timeline/schedule:





**Assessment:** Moskal will modify assessment instruments as necessary. Formative evaluation of project activities will be on going. Evaluations will be completed on data collected in prior semester.

**Materials Development:** Cooper and Dann will continue materials development. They will also assist the regional leaders in preparing for the summer workshops.

**Alice development:** Testing of the ports, and finishing development of solutions for sharing of worlds.

**Summer 2005**

**Regional team Meetings:** Cooper and Dann will travel to each of the three regions to collaboratively conduct workshops with the regional team leaders.

**Assessment:** Implementation of workshop assessment activities.

**Fall 2004 – Spring 2005**

**Regional use:** The individual faculty who attended the regional workshops will use instructional materials in their course. Faculty will also be responsible for collecting data for the overall impact of the Alice materials on their teaching.

**Regional leaders:** The regional leaders will assist the faculty who attended the workshop, as needed, as will Cooper and Dann. The regional leaders will also be responsible for working with Cooper and Dann to set up a follow-up workshop for Summer 2005.

**Materials Development:** Cooper and Dann will continue materials development. They will also assist the regional leaders in preparing for the summer workshops.

**Assessment:** Formative feedback acquired from faculty who use Alice in their classroom will be collected. Data will also be collected from students in participating classrooms.

**Alice development:** Bug fixes, gallery support, and maintaining the online communities.

**Summer 2005**

**Evaluation:** Collected data will be analyzed from all of the participating schools. A summary report will be prepared.

**Regional team Meetings:** Cooper and Dann will travel to each of the three regions to collaboratively meet with the regional team leaders and faculty from regional schools for overall project feedback. Faculty will share their experiences with using the materials, and will provide suggestions as to overall improvement.

## Evaluation Plan

As is suggested by the timeline above, assessment and evaluation will be on-going activities throughout the proposed project. Formative evaluations will be used to improve project-related activities. Summative evaluations will be used to determine whether project related objectives are being reached.

**Formative:** Qualitative research techniques will be used for formative evaluation purposes. Qualitative techniques have the advantage of providing detailed descriptive information and this type of information is useful for project improvement purposes.

Peer Review of Curriculum Materials: As part of this project, instructional materials that were developed during the pilot project will be revised. At least three external experts in the fields of computer science and computer science education will review all revised materials and further revisions will be made based on their recommendations. One of these experts is expected to be Randy Pausch, developer of the Alice tool. Peer review is a widely accepted technique for examining the content, construct and criterion validity of instructional materials [1].

Student Interviews: All the instructional materials have been pilot-tested as part of the original grant. Student feedback was used for revision purposes. Information has not, however, been collected from non-CS majors. As part of this proposal, 5 randomly selected non-CS majors at each site will be interviewed. A special effort will be made to ensure that woman and minority students are represented. The interview will focus upon the students' experiences with the instructional materials and how these experiences influenced their future educational goals and decisions. A specific emphasis of the interview will be to determine whether the course materials influenced students to

consider computer science as a major. The individual interviews will allow the PIs to acquire a detailed understanding of students' experiences with the instructional materials.

Workshop Feedback: At the conclusion of each day of the summer workshop, anonymous feedback will be acquired from workshop participants. Specifically, workshop participants will be asked what additional materials and support they require in order to implement Alice in their classrooms. A formal survey will also be administered at the end of the workshop to examine the participants perceived workshop benefits.

Telephone Conference Calls: In order to provide support to regional leaders and to acquire feedback from regional leaders, bi-monthly teleconferences will be held. The investigators, evaluator and regional leaders will all participate in these meetings. This will support the exchange of project information and on-going project improvement.

**Summative:** The summative evaluation will primarily use quantitative research techniques to support the statistical analysis of our efforts. At each of the locations, control classrooms will be selected. The control classrooms will consist of first year CS majors and non-CS majors who are completing the CS courses that share similar goals with the Alice courses. In pilot and control classrooms, a survey will be given to determine the students' background in mathematics, computers and computer science. The data collected by the readiness survey will be used to examine relationships between math and computer science preparedness, student performances and rates of attrition.

Attitudes Survey: At the beginning and end of the semester, students in both the pilot and control classrooms will complete a computer science attitude survey. This survey will help to determine whether students' participation in their first computer science course has stimulated their interest in pursuing a degree in computer science.

Two surveys have been identified as potentially being appropriate to our purposes ([14] and [22]). As part of this project, a more extensive search will be completed to locate the most current and appropriate instrument.

Statistical comparisons will be made between the pilot and control classroom, and among CS and non-CS majors, providing evidence as to the influence of the Alice materials on students' CS interests. Attention will be given to the relationship of the students' entering level of mathematics and computer science readiness and their interest in a CS degree. Comparisons will be made between female and minority students in the pilot and control classrooms. We do not expect these numbers to be large enough to make statistical comparisons within an institution. But these numbers will be large enough to statistically compare changes in female and minority attitudes across institutions.

Pretest/Posttest: The PIs have already developed a pre and posttest as part of the pilot project. This instrument was developed with the purpose of measuring fundamental concepts within beginning computer science classes. As part of the proposed investigation, this instrument will be further refined and validated using expert review. Statistical comparisons will be made between the pilot and the control classrooms' performances on this instrument. This instrument will also be modified in an appropriate manner to account for differences between the CS majors and non-majors.

Again, attention will be given to students' entering readiness and their performance on the pre and post instrument, and comparisons will be made between female and minority students in the pilot and control classrooms.

Retention Statistics: Statistical comparisons will also be made between the number of students in the pilot and control classrooms that select to pursue a computer

science degree. This data will suggest whether the newly developed course contributes to student retention in computer science. Readiness will be examined to determine whether it was a factor in outcomes of these statistics. Statistical comparisons will also be made between female and minority students that participate in this study.

Pre and Post Workshop Assessments: A pre and post assessment will be developed through a collaborative effort of the PI and the evaluator of workshop activities. This instrument will be designed to measure key aspects of the Alice software and of teaching with the Alice software. Statistical comparisons will be made between the pre and posttest performances of the faculty that participate in the workshops. This will suggest the effectiveness of the proposed workshops for the stated purposes.

**Testing with human subjects:** As students will be used to help evaluate the materials, all appropriate human subject procedures will be followed throughout the investigation. This includes receiving approval from internal Human Subject Research Review Boards at all institutions, obtaining all necessary written consents, and the maintenance of strict confidentiality throughout the study. Dr. Moskal is well versed in the process of acquiring human subject approval and has been certified by the National Institute of Health for completing the on-line faculty course, *Human Participants Protection Education for Research Teams*.

### **Capabilities of PIs**

We believe that we are uniquely qualified to run this project. Dr. Dann has 12 years experience teaching introductory computer science. Her expertise spans applied research in program visualization, curriculum development, and workshop presentations for event-driven programming. Dr. Dann is a member of the Visualization Working Group,

studying the effectiveness of visualization in computer science education. Dr. Cooper has been teaching introductory computer science for 10 years. He also has 10 years of industry experience, knowing what industry “needs” in the way of CS graduates.

Drs. Cooper and Dann have been working with Alice for 5 years. Our experiences with using and teaching computing concepts with Alice enable us to complete this project. We are working synergistically with the developers of Alice at CMU and make biannual 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 team has been quite responsive to implementing our requests and suggestions.

We already have a draft of a book to be published by Prentice Hall in early 2004. Our Alice work is receiving strong support from the CS community. Several of our papers have been accepted for conferences. (See the PIs’ biographical sketches for a complete list.) We have been contacted by numerous faculty members at various colleges and universities asking for access to our text and to our materials. We have also received valuable feedback from many faculty members who have been examining our materials.

Our results from our proof-of-concept grant have been extremely encouraging. We have demonstrated on our college campuses that teaching Alice really does make a difference for our at-risk computer science majors.

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. Dr. Moskal will be responsible for all data analysis. (See biographical sketches.)

Dr. Pausch has been the director of the Alice project for the last eight years, during which time he has led its development both during its origin at the University of Virginia and later at CMU. Dr. Pausch and his team are in a unique position to make the substantial, necessary changes to the Alice system for this work, and have also shown a willingness to modify the software to meet the educational and pedagogic needs of this effort. Dr. Pausch's team consists of several computer science Ph.D. students, a full-time staff programmer, and six undergraduates, including artists capable of creating the custom 3D models and texture maps required for the gallery of 3D objects.

### **Dissemination**

Dissemination activities will begin immediately through a project web site. This web site currently contains all the curricular materials we have developed. It will be augmented with the tutorials, demonstrations, and modified instructional materials including exercises. We will also include sample syllabi for various possible uses of Alice in different kinds of courses. We intend to make this material available to the CITIDEL collection of the NSDL. Prentice Hall also will maintain a web site for the text. There we will keep solutions to exercises and projects, and sample exam questions.

Presentations will be made at local, national, and international conferences (e.g., CCSCNE, SIGCSE, FIE, ITiCSE, ASEE). We expect several conference papers, on the ways in which our materials will be used at the colleges in this study, and their effectiveness. Papers will be submitted to computer science educational journals (JERIC and Computer Science Education). We expect at least one journal article, arguing for more pre-CS1 preparatory material and examining different ways of using our materials.

And, the <http://www.alice.org> website will have a lot of traffic, especially the online communities. This is a particularly effective mechanism for reaching many faculty members (such as those who teach in community colleges or high school teachers) who are typically not involved with SIGCSE or the other main computer science education channels. We already are receiving several requests per week for electronic copies of our textbook that have originated from an early version of the online communities.

### **Future plans**

Upon completion of this project, we expect to use the best practices developed in the formulation of a CCLI national dissemination grant. Successful development of regional test sites in this project will lay the groundwork for wider dissemination. Another possible avenue is involvement with high schools teachers to modify materials for use in high school introductory programming classes. We are currently working with several high school teachers in examining ways to incorporate our materials into their classes. And we are working with several community college colleagues for use of our materials at the community college level.

Another hope is that the data we collect across several campuses will be sufficient to demonstrate our anecdotal observations about the impact Alice has on women and minorities. We hope to be able to encourage women, African-Americans, and other minorities to pursue careers in computer science. There are strong connections between female learning and arts and graphics [24, 13] and between storytelling and African-American learning/culture [16, 19, 4]. It will certainly be possible to expand our focus on the arts/graphics aspects of the 3-D animation software or on the storytelling aspects.

## References

- [1] American Educational Research Association, American Psychological Association & National Council on Measurement in Education, *Standards for Educational and Psychological Testing*, American Educational Research Association, Washington, DC, 1999.
- [2] AAUW. *Tech-Savvy: Educating Girls in the New Computer Age*. New York: American Association of University Women Education Foundation, 2000.
- [3] Bergin, J., Stehlik, M., Roberts, J., and Pattis, R., *Karel++*, *A Gentle Introduction to the Art of Object-Oriented Programming*. New York: Wiley, 1997.
- [4] Burk, N. (2000) Empowering at-risk students: storytelling as a pedagogic tool. *Annual Meeting of the National Communication Association*. ERIC#: ED447497.
- [5] Butcher, D. and Muth, W. (1985) Predicting the success of freshman in a computer science major. *CACM*, 28(3), 263-268.
- [6] Cooper, S., Dann, W., & Pausch, R. (2000). Alice: A 3-D tool for introductory programming concepts. *Proceedings of the 5th Annual CCSC Northeastern Conference 2000*, Ramapo, NJ, April, 28-29, 2000.
- [7] Cooper, S., Dann, W., & Pausch, R. (2003). Teaching objects-first in introductory computer science. *Proceedings of the 34<sup>th</sup> SIGCSE Technical Symposium on Computer Science Education*, Reno, NV, February, 2003.
- [8] Cooper, S., Dann, W., Dragon, T., Dietzler, K., 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, June 30- July 2, 2003.
- [9] Cooper, S., Dann, W., & Pausch, R. (2003). Using animated 3D graphics to prepare novices for CS1. *Computer Science Education - Special issue on graphics and visualization*, to appear.

- [10] 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.
- [11] Dann, W., Cooper, S. & Pausch, R. (2002). Using 3d interactive animation to provide program visualization as a gentle introduction to programming (extended abstract). *Proceedings of the 2<sup>nd</sup> Program Visualization Workshop*, Aarhus, Denmark, June, 2002.
- [12] Davy, J.D., Audin, K., Barkham, M. and Joyner, C. Student well-being in a computing department. *Proceedings of the 5<sup>th</sup> Annual Conference on Innovation and Technology in Computer Science Education*, Helsinki, Finland, (2000), 136-139.
- [13] Ettenheim, S., Furger, R., Siegman, L., and McLester, S. (2000) Tips for getting girls involved. *Technology and Learning*, 20(8), 34-36.
- [14] Francis, L. (1993) Attitude toward computers scale. *Computers and Education*, 20(3), 251-255.
- [15] Freeman, P., and Aspray, W. *The Supply of Information Technology Workers in the United States*. New York: Computing Research Association, 1999.
- [16] Goss, L. and Barnes, M., Eds. *Talk That Talk*. New York: Simon and Schuster, 1989.
- [17] Hagan, D., and Markham, S. Does it help to have some programming experience before beginning a computing degree program? *Proceedings of the 5<sup>th</sup> Annual Conference on Innovation and Technology in Computer Science Education*, Helsinki, Finland, (2000), 25-28.
- [18] Harvey, B. (2000). Inhibition of sulfate-reducing bacteria (SRB) activity by pesticides in tidal-marsh sediments. Poster presented at the *Annual Meeting of Environmental Sciences*, Nashville, Tennessee.
- [19] Heath, S. B. *Ways with Words*. Cambridge, UK: Cambridge University Press, 1983.

- [20] Huntzinger, D, McCray, J.E., Van Cuyk, S., Siegrist, R. (In Press). Numerical modeling of unsaturated flow and transport in on-site wastewater treatment systems as affected by soil clogging. Paper to be presented at 9<sup>th</sup> *National Symposium on Individual and Small Community Sewage Systems, American Society of Agricultural Engineers.*
- [21] Lawhead, P., Bland, C., et al. (2003). A roadmap for teaching introductory programming using Lego mindstorms. *SIGCSE Bulletin*, 35(2), 191-201.
- [22] Loyd, B. H. and Gressard, C. P. (1996) Computer attitude scale. *Journal of Computing Research*, 15(3), 241-259.
- [23] MacDonell, M., Moskal, B., Knecht, R. & Lasich, D. (In Press). The engineering process: Examining male and female contributions. Paper to be presented at the *Frontiers in Education Conference*, Reno, Nevada.
- [24] Mahoney, K. and Van Toen, B. (1990) Mathematical formalism as a means of occupational closure in computing - why 'hard' computing tends to exclude women. *Gender and Education*, 2(3), 319-330.
- [25] Margolis, J. & Fisher, A. (2002). *Unlocking the clubhouse, women in computing*. Cambridge, MA: MIT Press.
- [26] McCray, J.E., J.E. Huntzinger, D., Van Cuyk, S. & Siegrist, R. (2000). Mathematical modeling of unsaturated flow and transport in soil based wastewater systems. Paper presented at *Water Environment Federation WEFTEC*, Anaheim, CA.
- [27] Moskal, B., Lasich, D. & Middleton, N. (2001). Science related degrees: Improving the retention of women and minorities through research experience, mentoring and financial assistance." Paper presented at the *American Society for Engineering Education*, Albuquerque, New Mexico.

- [28] Moskal, B. (2000). Looking to the future: Women in science and engineering. *Proceedings of Frontiers in Education Conference*, Kansas City, Missouri. (Paper resulted in receiving the *New Faculty Fellowship* award for demonstrating a dedication to Excellence in the field of Engineering Education).
- [29] National Research Council. *Fluent With Information Technology*. Washington, D.C.: National Academy Press. 1999. Available at: <http://www.nap.edu/html/beingfluent/>
- [30] Oman, P. (1986) Identifying student characteristics influencing success in introductory computer science courses. *AEDS*, 19(2-3), 226-233.
- [31] Papert, S., *MindStorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, 1980.
- [32] Pattis, R., *Karel the Robot*. New York: Wiley, 1981.
- [33] Pausch, R. (head), Burnette, T, Capeheart, A.C., Conway, M., Cosgrove, D. DeLine, R., Durbin,J., Gossweiler,R., Koga,S., and White, J. (1995). Alice: Rapid Prototyping System for Virtual Reality , *IEEE Computer Graphics and Applications*, 15(3), 8-11.
- [34] Pierce, J., Stearns, B. & Pausch, R. (1999). Two handed manipulation of voodoo dolls in virtual environments. *ACM Symposium on Interactive 3D Graphics*, April 1999.
- [35] Pierce, J., Pausch, R., Sturgill C. & Christiansen, K. (1999). Designing a successful HMD-based experience, *Presence*, 8(4), August 1999, pp. 469-473.
- [36] Seymour, E. and Hewitt, N. *Talking about leaving*. Boulder : Westview Press, 1997.
- [37] Stephens, L., Konvalina, J., and Teodoro, E. (1985) Procedures for improving student placement in computer science. *Journal of Computers in Mathematics and Science Teaching*, 4(3), 46-49.

- [38] Van Cuyk, S. & Siegrist, R. (2000). Pathogen fate in wastewater soil absorption systems as affected by pretreatment and soil clogging genesis. Paper presented at *Water Environment Federation WEFTEC*, Anaheim, CA.
- [39] Division of Research, Evaluation and Communication, Directorate for Education and Human Resources. *The Learning Curve: What We Are Discovering about U.S. Science and Mathematics Education*. L. E. Suter (ed). Washington, DC: National Science Foundation, 1996 (NSF 96-53).
- [40] Software Visualization – bi-annual conference (associated with FCRC) information available at: <http://www.softvis.org/softvis03.html>
- [41] Program Visualization – bi-annual conference (associated with ITiCSE) information available at: <http://stwww.weizmann.ac.il/g-cs/benari/pvw/pvw.htm>
- [42] IEEE/ACM's Computing Curricula 2001 is available at: <http://www.acm.org/sigcse/cc2001/>