Introduction

The purpose of this proposal is to create a synergistic intervention for introductory programming courses with the goals of increasing the attraction of students to computer science (CS) majors, improving retention of students in first-year programming, and encouraging non-CS majors to minor in computing for interdisciplinary career paths. We propose to develop a first-year programming curriculum that uses (1) Alice, which teaches introductory computing concepts in the context of animation and movie-making combined with (2) Media Computation to manipulate media and provide special effects. This blend of Alice with Media Computation will take advantage of the interest and motivation students find in video games and animated films. We believe that this approach will appeal to a wide range of students while covering all the same skills and concepts mandated by curriculum standards [2]. The primary goal is to provide a high impact, motivating context for introductory computing and programming courses at four-year and community colleges. Through our evaluation efforts, we will study the effectiveness of this approach for attracting and retaining majors and minors in CS.

Motivation: A primary motivation for this project is the decrease in the number of CS students. According to the Higher Education Research Institute at the University of California at Los Angeles (May 2005), "the percentage of incoming undergraduates indicating that they would major in CS declined by more than 60 percent between the fall of 2000 and 2004, and is now lower than … in the early 1980s" [38]. Recent data provided by the NSF CISE division further indicates a significant drop in the number of high-school students that complete advanced placement (AP) tests in CS [5]. While the number of AP tests taken overall has increased by 33%, there has been a drop of 20% in the number of AP tests completed in CS. More interestingly, the number of AP tests taken has gone up in every field except CS. As this data suggests, CS is no longer viewed as a "hot" career. Attracting today's students to computing has become an issue for CS departments across the nation and the situation shows little signs of improvement.

This precipitous drop in interest in CS stands in contrast with job demand data from the US Bureau of Labor Statistics, which predicts that, in the next ten years and after accounting for job losses due to outsourcing, there will be 1.2 million new jobs created in the field of information technology (IT). This is much greater than the number of computing-related degrees granted in the US. Thus, there is a crisis looming on the horizon. Under the current conditions, the US will be unable to satisfy the IT industry's labor demands.

This mismatch between interest and demand impacts the entire US population, not just computer scientists. In his bestselling book *The World is Flat* [16], Thomas Friedman argues for the need for "*Versatilists*." He points out that preparing students for careers in technology is not enough to compete with nations such as India and China. The greatest need in the global economy is for people who blend their understanding of technology with their knowledge of other domains, such as business or science. The United States is well poised to retain economic leadership by emphasizing an interdisciplinary focus and including technology as part of the mix. This, however, is difficult to achieve if students continue to display little interest in CS.

A variety of complex factors contribute to students' lack of interest and participation in CS. Many students are concerned with the CS culture. Some see computing as an asocial activity [31] that is best suited to those men drawn to computers from an early age [25]. Many female students in undergraduate CS programs reported feeling socially isolated from and less capable than their male peers, particularly if they did not have extensive computing experience prior to college [25]. In addition, many women and underrepresented minority students view the practice of CS as tedious, boring, and irrelevant [1, 25] with little room for creativity [31]. Beginning students have difficulty seeing the real-world relevance of topics such as byte representations and algorithmic efficiency [25]. Faced with a difficult curriculum, an unwelcoming culture, and course lectures and assignments that seem irrelevant to real-world problems, many women and underrepresented minorities choose not to pursue CS. We believe that to *get students in*

the door, we need to make the introductory CS curriculum more motivating and relevant. And in making CS more motivating, men, as well as women will become more attracted to the discipline.

A second, but equally important, motivation for this project is the problem of retaining the students in CS majors (or minors). On average, at least half of college students majoring in CS withdraw from the major [14], and the majority of these students withdraw during the first year [35]. Thus, there is a small window in which changes can make the greatest impact. Recent studies (such as [11, 23]) have shown that students who enter college without prior programming experience are at a disadvantage in successfully completing a CS degree. Two issues appear to be critical in determining success in the first year. First, many students enter college with a weak foundation in problem solving and logical reasoning [4, 28, 36]. Second, first-year programming courses rapidly introduce a broad range of programming concepts, and this overwhelms many students. And, we believe this second issue has become more of a problem by the adoption of object-oriented programming as the first programming paradigm. Third, there is the lack of curricular materials that address the previously described challenges. Instructional materials are needed that are designed to reach a broad range of majors, both CS and non-CS, in first-year programming courses. In an age where the demand for IT professionals is rising [15] and the need to think algorithmically [27] in many career occupations is critical, we need to increase the motivation of students to take a first course in programming and to consider combining a minor in CS with their major, in an interdisciplinary manner. CS departments are not currently successful at reaching the wide range of students who are taking introductory CS. The evidence for this statement includes international studies of programming performance [22], declining retention rates [13], and failure rates sometimes as high as 50% [31]. If students cannot successfully complete the first programming course, there is no chance for them to become or remain CS majors or minors.

Context: CS courses draw examples and assignments from a *context* or problem domain [21, 22]. The choice of context can influence both students' motivation [39] and the quality of their learning as it relates to transfer to other domains [12, 24]. A context that students relate to and find relevant can lead to deeper learning, and material that is learned deeply is more likely to transfer to new contexts [7]. Currently, the typical contexts for introductory computing include business applications (e.g., a bank account example for simple classes in object-oriented programming), systems building (e.g., writing functions to format numbers appropriately into strings), and purely abstract problems. Based on studies of female students, we know that many students are not understanding what problems introductory computing is helping them to solve [1, 25], and students often do not see relevance in their CS classes until their sophomore and junior years.

We believe that changing the *context* of introductory courses without changing the skills or concepts introduced will have a dramatic impact on student motivation. We propose to use the context of animation. Today's students have grown up watching animated movies, such as *Toy Story* and *Shrek*, and playing video games. The interest these students had in these activities as young children helps to motivate them to want to understand the concepts that underlie their creation. Using Alice and Media Computation, we believe that introductory CS can be presented through animation, while maintaining all of the same skills and concepts mandated by curriculum standards [2]. Rather than manipulating strings and numbers in an abstract setting, students can learn iteration, conditionals, objects and classes, and methods and parameters by manipulating data types common in the contexts of digital storytelling, video game production, and movie-making: actors, scenes, pixels in pictures, sound samples.

The PIs of this proposal have been successful at introducing more motivating contexts in two different settings. Dann and Cooper's project investigated the use of 3-D animation as a program visualization tool (Alice), and Guzdial and Ericson's project investigated the use of Media Computation for teaching/learning introductory programming. We propose to combine Alice and Media Computation to create a synergistic approach to introducing computing. A textbook and supportive instructional materials will be created to provide a combined approach, oriented around the creation of animated movies and video games with special effects. The focus of this approach will be on creating a steppingstone from "computing as facilitated movie-making" to "computing as traditional, textual

programming." A program visualization tool will be used to create animated movies, and traditional textual programming will be used to create special effects for the animated movies. In this manner, students will learn both the algorithmic process of programming and the syntax of a text-based object-oriented programming language (Java). We propose a series of three-day workshops for undergraduate faculty along with web-based support. Results of our previous work lead us to hypothesize that this combined context is one that will attract and motivate a diverse student population. We propose to evaluate the effectiveness of this approach as a means of creating a context that attracts prospective students to "peek through the CS door" and encourage them to stay in the room.

Alice: *Alice* is a programming environment designed to enable novice programmers to create 3-D virtual worlds, including animations and games [30, 32]. In Alice, 3-D models of objects (e.g., people, animals and vehicles) populate a virtual world and students use a drag and drop editor to manipulate the movement and activities of these objects. Alice was designed through an iterative process of studying how novices try to describe the motions of objects in a 3-D world, and then modifying Alice so that the novices' expectations would be met [33]. Alice makes use of program visualization to allow students to immediately see how their animation programs run, enabling students to easily understand the relationship between the programming statements and constructs and the behavior of their animations. In Alice, students learn the basics of computing, but where the objects of concern are actors and scenes in a virtual world. Alice programs have a strong object-oriented flavor, allowing students to control the appearance and motion 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 learn about objects and aggregation by addressing the component features within objects—lifting an arm, turning a head. Students learn about sequencing and iteration by constructing a series of actions in a scene. Further details about Alice may be found in Appendix A.

Alice [29] has been used successfully in undergraduate classes that draw a diverse range of students into computing [8, 9, 10]. A set of curricular materials and a text [10] were developed and piloted tested in introductory computer programming courses (pre-CS1) offered at Saint Joseph's University (SJU) and Ithaca College (IC). At SJU, the materials were taught prior to CS1, while at IC the materials were taught concurrently with CS1. Details of the results of this study are provided below in the Results of Prior NSF Support section.

Media Computation: Starting with the insight that communication is an inherent human need that makes for a powerful context for teaching programming, the Media Computation approach to introductory computing has students programming in Python or Java to manipulate media [17, 22, 34]. The course was developed at Georgia Tech to fulfill a computing requirement for non-CS and non-Engineering majors. Students who take this course are majoring in Liberal Arts, Management, or Architecture. The course has been a majority female course since its inception in Spring, 2003 with a typical class size of about 300.

Students in Georgia Tech's Media Computation courses manipulate images (scaling, cropping, mirroring, manipulating colors, converting to grayscale) and sounds (reversing, splicing, shifting frequencies, slowing down or speeding up) by writing programs that create these effects. In particular, we teach the basics of digital image and video effects, such as background subtraction, chromakey, and compositing images into the frames of a video sequence. Textbooks have been written for the Python and Java versions of the course [18, 19]. It has been adopted successfully by other schools, such as Gainesville College and University of Illinois-Chicago. See the Appendix for a complete list of schools. Details of the results are provided below in the Results of Prior NSF Support section.

A synergetic intervention, Alice and Media Computation: The Alice and Media Computation approaches have a common strength: strong motivation and appeal for students who have experience with video games and multimedia in their daily lives. Alice uses 3-D animation and program visualization to introduce fundamental programming concepts such as sequencing, decision making, and repetition, as well as an intuitive understanding of object-oriented concepts such as classes (including behavior and

state), objects, methods and inheritance. Media Computation uses 2-D graphics and sound computations that correspond to real-world applications for moviemaking. This project will create a synergetic intervention that uses Alice to introduce computing in an animated movie-making context, and then uses Media Computation in Java as a "special effects studio" for Alice animations.

Alice allows students to assemble programs using a drag-and-drop editor. An advantage of the drag-and-drop editor is that students can focus their attention on understanding programming constructs without the initial distraction and frustration of syntax details. A drawback is that students must then enter traditional CS courses without experience in dealing with syntax errors. Since instructors may see syntax as a crucial aspect of learning to program, Alice is sometimes characterized as "a toy…not real programming." Some schools have attempted to teach a single introductory programming course that begins with Alice and then transitions to more traditional programming during the semester. This proposal is a result of their request for the development of instructional materials that transition students from Alice to traditional programming. We believe that Media Computation will provide the appropriate transition.

The Media Computation approach *is* traditional text-based programming, but in a context that is very much like Alice's storytelling and movie-making context. Rather than making an abrupt switch between programming in Alice and programming in Java, we plan to interweave usage of Alice and Media Computation, presenting programming concepts in one and then reinforcing the concept in the other. This will allow us to maintain a movie-making focus throughout the course, while allowing students to tackle both the programming concepts and programming language syntax separately. Typical student projects would involve learning traditional CS concepts, but in motivating contexts that utilize both Alice and Java programming. Examples of such projects include the following:

- To teach students about indexing arrays in multiple ways, we might have students create echoed sounds in Java, which involves adding values from samples earlier in a sound to those later in a sound, but scaled down by a magnifier. This echoed sound might then be imported as a wav file for use in an Alice story about a character calling out in a cave.
- To teach students about recursion, we might use fractal generation code written in Java to generate realistic 2-D images of clouds or trees, and then import those images as jpg files to create backgrounds for Alice scenes.
- To teach students about replacing elements in an array or matrix, we might have students create a scene in Alice with an all-green or all-blue background, then save it out as a JEPG image. Using the chromakey algorithm (the same process by which meteorologists appear to stand in front of weather map), we can then replace that background with an actual picture, or even place a picture of the student into the scene.

Results of Prior NSF Support

The work in this proposal builds on the successes of two prior innovative research efforts in attracting and retaining students into computing: Alice and Media Computation.

Alice (NSF-0126833, 0302542, 0339734): Alice has been used successfully as an intervention to draw at-risk students (who are disproportionately female or underrepresented minorities) into computing [8, 9, 10]. At-risk students were defined as those students who had demonstrated less success in math and/or those who had little previous programming experience. A textbook [10] was developed and pilot tested in introductory computer programming courses (pre-CS1) offered at SJU and IC. Additionally, a detailed set of curricular materials [6], including several different curricular models (with complete lecture notes), laboratory exercises, solutions, exams, assignments, sample student projects, and other material, was created. At SJU, the materials were taught prior to CS1, while at IC the materials were taught concurrently with CS1. The primary results of this investigation were:

1) The average grade for at-risk students exposed to Alice was a 3.0 GPA in CS1, which is comparable to the grades of students who were at no risk or low risk. The average grade for at-risk students not exposed to Alice was a 1.2 GPA in CS1.
2) 88% of at-risk students exposed to Alice enrolled in CS2 after CS1. Only 47% who were not exposed to Alice enrolled in CS2. ($p < .05$, chi-squared)

Further details concerning this investigation are available in [26].

Based on the results of our proof-of-concept study, we were awarded two follow-on grants. The first grant involved the use of Alice with three diverse community colleges to attract and retain majors, and to improve the computer literacy course. The second grant involved the running of regional summer workshops to train faculty on using and teaching with Alice. This past summer, 95 faculty members from 57 schools attended one of our three workshops, and many are teaching with Alice this year. Preliminary results for both of these studies suggest that using Alice has had a statistically significant positive impact on students' understanding of basic programming concepts in both the community college and four-year college setting. A list of the schools using Alice is available in the Appendix.

Media Computation: An evaluation of the Media Computation class at Georgia Tech found that approximately 90% of the students who take *Introduction to Media Computation* earn an A, B, or C [37]. The success rate is the same for women and members of minority groups as it is for white males. Unlike many students taking introductory CS, Media Computation students reported finding the class relevant to their chosen careers and their everyday lives in a technological society, even a year later [20]. Some of these students are now pursuing CS minors or CS majors.

Building on the success of Introduction to Media Computation, Georgia Tech is now offering a follow-on course which covers data structures in a Media Computation context. In its first semester, the Media Computation-based data structures course had 32 students, 75% of whom were female. The success rate in the second course was over 90%. Georgia Tech has just created a new *BS in Computational Media* that has over 100 majors in its first year, 25% of whom are female.

Goals and Objectives

Goals: The overall goals of this project are to:

- *Provide a high impact, motivating context for introductory computing and programming courses.*
- *Increase the flow of students into computing studies at the undergraduate level, and eventually, into computing and computing-related careers.*
- *Increase the retention rate of students in computing, especially during the crucial first year.*

Objectives and Outcomes: To meet these goals, the objectives and outcomes of the project are to:

Objective 1: *Provide a high impact, motivating context for introductory computing and programming courses by creating a text and instructional materials.* We propose to create a textbook and supplementary instructional material that use a combination of Alice (program visualization) and Media Computation. Although an Alice text and a Media Computation text each separately exist, we believe that combining the two into one textbook is needed to assist the instructor in transitioning between Alice and Media Computation concepts and techniques. We also propose to provide online access to the supportive instructional materials. Use of these materials is expected to be highly motivating to students and result in an increase of students participating in computer-related degrees at the college level. It is further expected to improve the retention rate of first-year computing majors.

Measurable Outcomes for Objective 1:

i. Completion of textbook and instructional materials that combine Alice and Media Computation.
ii. Existence of online support for instructional materials.

iii. The online instructional materials will be accessed at least thirty times within a given week as measured by an electronic counter.
iv. The majority of participating students will evaluate the courses that use these materials positively as measured by a closed response survey.
v. Retention rates of first-year CS degrees at the participating schools will increase by 5%.
vi. Courses that utilize the above described materials will have a greater retention rate in terms of declared CS majors and minors than comparable courses at the same level.

Objective 2: *Attract prospective students into an Alice/Media Computation class by enabling them to "peek" into the introductory CS curriculum.* We propose to create interventions that allow current students to show off their work to prospective students in showcases for open audiences such as a college fair or visiting high school groups. These interventions will include support for showcases of student work and a website featuring student-created movies. Our strategy is to present computing in a more compelling context. In addition to offering a context for learning object-oriented programming concepts, this approach offers the benefit of providing a medium for students to express themselves in a way that has real-world meaning. This strategy is particularly appealing to women and students in subcultures with strong traditions of storytelling, such as the African-American, Native American, and Hawaiian populations.

Measurable Outcomes for Objective 2:

i. Completion of an online showcase of student work.
ii. The online showcase will be accessed at least thirty times within a given week.
iii. Students who view the online showcase will evaluate it positively as reflected by an online survey.
iv. Number of declared entering CS majors and minors at participating schools will increase by 5% when compared to baseline date collected prior to intervention.

Objective 3: *Provide faculty development and support for instructors in using this new integrated approach*. We propose to make a broad effort to provide faculty development workshops for CS instructors in community colleges and traditional undergraduate programs, to introduce programming as storytelling and movie-making. We plan to hold regional three-day workshops in Atlanta (Georgia Tech), Pittsburgh (Carnegie Mellon), Providence (Roger Williams), and Philadelphia (SJU). Support will be provided to instructors as they adopt/adapt this approach to their own courses and student needs.

Measurable Outcomes for Objective 3:

i. Development of materials and implementation of summer workshop.
ii. The majority of participating teachers will positively evaluate the workshops as reflected through a survey instrument.
iii. There will be a statistically significant change in the teachers' knowledge base as measured by a pre and post workshop assessment.

In this way, we expect to attract more students, particularly women and underrepresented minorities, into computing by reaching their instructors and providing them with a new approach to teaching introductory CS. We plan to retain these students in computing by offering them an opportunity for creativity in introductory programming and an audience for their creations.

Impact: While we do plan to target students directly (through a showcase website, our textbook, and end-of-term on-campus presentations), these efforts are mostly meant to encourage students to "peek in the door" of computing, to see what it offers. Our most significant approach to meeting this goal is teacher workshops where we have the potential of a *multiplier effect*—for every teacher attending our workshops, we have the opportunity to reach hundreds of students. We believe that we can inform and inspire teachers in our workshops to make changes in what and how they teach, and in this way, change the connotation of CS without changing the denotation of CS—and bring more students in the door.

As is described in the next section, we plan to offer two summer three-day workshops the first year and four the second year to teachers in community college and undergraduate settings. The first year's workshops will be offered in Atlanta (Georgia Tech) and Pittsburgh (Carnegie Mellon). The second year, the workshops will be offered again in Atlanta and Pittsburgh, and also in Philadelphia (SJU) and Providence (Roger Williams). Based on workshop participation in our previous NSF grants, interest in this proposed project from many of the workshop attendees (see the letters of support), and SIGCSE workshops, we expect to host 25-35 teachers per workshop. We plan to directly impact 150-210 teachers. Based on feedback from Alice faculty workshops and ICE@GT summer camps for high school teachers, we believe that 80% of the teachers will change what and how they teach in response to attendance at the workshops. If each teacher impacts 60 students per year (30 per semester), *we will indirectly reach approximately 10,000 students over the course of this proposal.* In the Media Computation classes at Georgia Tech, approximately 10% of the students in the first course take the second course—that is, to take more computing courses beyond their requirement. We believe that the Alice/Media Computation combination will be even more powerful. *If only 15% of the students continue, we will have had a positive impact, having significantly impacted approximately 1500 students in two years.* These projected numbers are just the students who are being impacted by the teachers who are physically attending our workshops. Through our dissemination of our materials and results (in book form, on the Web, and at ACM SIGCSE conference workshops), we expect to reach even more teachers and have an even greater impact on the numbers of students in computing studies and careers.

While the previous discussion has focused on the objective of attracting students to CS, we also expect to meet the objective of increasing the retention of students, and the objective of increasing the number of non-CS majors that minor in computing. Our Alice work, as described above, has resulted in an increase in retention in CS1-CS2 by at-risk students from approximately 1 in 2 to better than 8 in 9. We are confident that we will see a significant rise in retention of students from CS1 through CS2 (where the majority of attrition in CS typically takes place).

Detailed Project Plan

To have the maximum impact on the CS pipeline, we believe that it is important to generate students' interest in CS early and to support the development of their interest through the first year of CS. We will focus on developing instructional materials and on interventions that attract students into introductory classes, keeping these students involved in CS, and supporting faculty teaching these classes. Details of the project plan are described here in terms of (1) materials development, (2) target groups and (3) focal points, based on our goals and objectives.

Materials development: We plan to create a text that combines Alice and Media Computation. Key to the success of this new integrated approach is the interweaving of fundamental, object-oriented programming concepts (creating animated movies using Alice) and learning the details of syntax and textual programming skills a real-world language (creating special effects for animated movies using Java). Based on our previous work, we believe that such an interweaving can be accomplished. A potential topic outline is shown in Figure 1.

Please see the supplementary materials for a letter of support from Prentice Hall, publisher of the previous Alice [10] as well as Media Computation texts [18, 19]), and likely publisher of the proposed combined Alice and Media Computation textbook.

In addition to a textbook, we plan to develop instructional support materials including lecture notes (slides), sample quizzes and exams, sample labs, and sample projects. We plan to create variations for use in several scenarios: for use in summer outreach programs and other short duration activities, as a 1-2 credit course, and as a traditional 3-4 credit course (with variations for pre-CS1 and CS1).

1. Introduction to Alice
    - Create your first world
    - Using methods: Making characters speak and move
    - An introduction to classes and objects
2. Introduction to Java
    - Create turtle objects and name them (declare an object variable)
    - Telling a turtle object what to do (invoke a method)
    - Teach the turtle how to do new things (create a method)
    - Add a parameter to a method
    - Mapping classes and objects from Alice to Java
3. Program Design with Storyboards
4. Introduction to Programming in Alice
    - Creating a method, with parameters, using numbers and strings.
    - Simple looping
    - Using While
5. Doing Sound Manipulations in Java
    - Sound encoding as sample values in an array
    - Simple indexing: Changing volume of a sound
    - Indexing with scaling: Creating a fading sound for an object moving away from the camera in Alice
    - Complex indexing with scaling: Creating an echo
    - Methods that return objects vs. side effects
6. Doing Picture Manipulations in Java
    - Picture encoding as a matrix of RGB values
    - Indexing a whole matrix: Negating images, grayscale, reducing red
    - Indexing part of a matrix: Changing only part of a picture, scaling pictures
    - Recursion in fractal generators, to create backgrounds for Alice
7. Manipulating Pictures from Alice
    - Replacing in a matrix: Using chromakey to put the user in the scene
    - Creating mirrors of images
    - Modifying Alice texture maps
8. Classes and advanced object-oriented programming
    - Inheritance in Alice
    - Inheritance: Implementing confused turtle
    - Creating cartoon strips: Scenes from Alice, turned into a cartoon strip with word/thought bubbles in Java
9. Movies
    - Lists of pictures
    - Modeling with aggregation
10. Web into Alice
    - String processing on Web content: Placing headlines and images in Alice scenes
    - Exceptions for networking

Figure 1. Potential topic outline for our new text

A note on combining 2-D and 3-D: Media Computation works on jpg images that are 2-D and on wav files, which are sounds. The sounds can be used in 3-D Alice worlds with no problem. The jpg images can be used wherever Alice uses 2-D images: for mirrors, paintings, billboards, or even for the textures that get wrapped around 3-D images. The other way to use Media Computation as "special effects" is by

producing an image out of Alice (e.g., saving a scene as a 2-D image) and processing it in Media Computation (such as in the production of "comic strips").

Target groups: We plan to offer Alice/Media Computation classes for two levels of students: four-year and community college.

Four-year college faculty and students: Two of our institutions (SJU and IC) will pilot at least one version of the new course. Instructors and students in these pilot courses will provide formative feedback for revision purposes, as is described in the evaluation plan of this proposal. We will impact faculty and students through our summer teacher training workshops. Students in our pilot courses will be selected to act as student assistants for the summer workshops.

Community college faculty and students: The first-year results from our current NSF ATE-supported project indicate that using Alice prior to or as part of CS1 helps to improve student performance and increase retention in CS courses at the community college level. However, community college faculty members are often concerned about transitioning students from Alice to a more traditional text-based object-oriented language (e.g. Java, C++, or VB). Accordingly, several community colleges have expressed interest in participating in this project by using the Alice/Media Computation class developed in this proposal. (See the supplementary materials for letters of support.) We plan to offer workshops specifically targeting community college faculty (and, thereby, community college students). We expect at least two community colleges will participate formatively in this study.

Focal point 1: *Attracting prospective students into an Alice/Media Computation class by enabling them to "peek" into the introductory CS curriculum.* By placing introductory programming in an animated movie-making context, we hope to make CS more attractive to students, particularly women and underrepresented minorities. However, unless we provide opportunities for students to appreciate what is going on inside the CS classroom, we are unlikely to realize the full benefits of the context change. We propose to provide two types of opportunities for prospective students to "peek" into the classroom: 1) opportunities to see on-campus student work and 2) highlighting student work on a showcase website.

One advantage of introducing programming through creating animated movies is that the artifacts students create can be readily understood and appreciated by prospective students. We have noticed that students completing an introductory programming class based on the Alice system often invest many hours perfecting their Alice projects and frequently show their completed Alice projects to friends. The practice of students showing their work to friends is, in essence, an informal recruiting program for the class. Further, the fact that students with no programming experience can readily appreciate programming artifacts of other students creates the opportunity to use these artifacts (with written permission from the student creators) as tools for recruiting new students. We will take advantage of this recruiting tool in two ways: showcase events and a website of student work.

To attract students to our course, we will work with the teachers who participate in the workshops (see below) to create local showcase events. A showcase event is similar to an "opening night" of a movie, except that instead of a movie, students will be viewing the virtual worlds created by their peers. Students will have the opportunity to see what their peers are doing, and thus, what they could be doing. Only programs for which written permission from the student developer was acquired will be viewed at these events.

We will also make available to participating classrooms a website where students can share their work broadly. A warning will be posted at this website that students should not post their programs unless they are willing to share them with the world via the internet. This website will be publicized through papers, professional talks and electronic correspondence as a place where students can creatively display their work and where other students can view these efforts. Through this website, prospective CS students will see that CS can be a creative endeavor.

Focal point 2: *Provide faculty development and support for instructors in using this new integrated approach*. As part of the dissemination efforts of the Alice course and the Media Computation course, we

have found that faculty development workshops, accompanied by ongoing year-long support, are crucial for successfully implementing new curricula in other institutions. Our approaches are non-traditional in that they require use of multimedia and graphics (including animation). These are topics with which many faculty members are neither familiar nor comfortable. Further, we have found that many college faculty members (particularly at community colleges) are not comfortable with the concepts of objects and object-oriented programming, concepts that are necessary to effectively teach the Alice/Media Computation course. To enable faculty to develop the skills and expertise necessary to teach the proposed course, we will conduct three-day, short, intensive summer workshops.

In addition to facilitating skill development, summer workshops will provide an opportunity for the faculty at the different participating schools to develop a sense of community. To provide continuing support to instructors after the training workshops, we will create an Alice/Media Computation instructors forum and website. The forum will allow instructors to communicate with each other and the project PIs throughout the year. A website will also be created that contains the instructional materials.

Implementation: To support the interventions we have proposed, it will be necessary to do some system development and merging of the Alice and Media Computation curricular materials.

Modifications are required to support a single curriculum that enable students to create projects that utilize both Alice and Media Computation, and allow students to share their work with others. One of the important elements of our approach is interweaving programming in Java with programming in Alice by encouraging students to create projects that combine elements created using both Alice and Media Computation. To facilitate the usage of the products of Media Computation in Alice and vice versa, some technical changes will be needed.

- ***Using the Products of Media Computation in Alice:*** Students will likely want to export jpg images and wav files already associated with characters in Alice as the basis for their Media Computation work. Further, we would like to support students creating animated special effects such as fire or rain. To support this, we plan to enable students to export the texture maps associated with Alice characters and objects as jpg images, to be able to export sound files associated with Alice characters and objects as wav files, and to enable students to easily import jpg images and create animated special effects.
- ***Helping Students Share their Work:*** Currently, a student who wants to share an animation built with a friend or family member must convince that person to download a potentially large file and possibly install new software. For many friends and family members, particularly those who are less comfortable with computers, this is too much to ask. To facilitate the effort to allow prospective CS students to "peek into the classroom," it is important to make it easy for students to quickly share animated worlds they create in Alice. To support this, we propose to develop a tool that allows students to create small videos of their Alice worlds that can be emailed to a friend or posted on a website, and to create tools that allow students to save "lower resolution" versions of their Alice worlds to send to fellow Alice users. Alternatively, we may use third-party software for performing the recordings.

Project management: The project co-PIs will act as a team to manage the three branches of this project: 1) development of a text and instructional materials, 2) workshops and support for college level instructors/students, and 3) project evaluation. To coordinate these efforts, the team will schedule monthly teleconferences and face to face meetings twice a year.

In this collaborative project, Cooper will be the lead PI for the project, and, in this role, will oversee the implementation of all project activities. The following individuals will participate in the development of the text, the creation of the instructional materials and the implementation of the proposed workshops: Cooper, Dann, Ericson and Guzdial. Guzdial will additionally develop the website to showcase student animations and provide technical support for Media Computation. Moskal will be responsible for evaluation of the project.

Timeline:

Semester	Activities
Fall 2006	1) Modify each of the software tools to enable them to support the merging of the curricula. 2) Merge the Alice and Media Computation courses and prepare first draft of a textbook.
Spring 2007	1) Pilot a new course as CS1 at SJU 2) Pilot a new course as CS0 at IC 3) Advertise for the Summer 2007 workshops 4) Evaluation: Formative data collection
Summer 2007	1) Review formative data, revise text and instructional materials 2) Submit text to a publisher 3) Create online repositories of materials, and set up other online support 4) First round of three-day workshops for faculty development in Atlanta and Pittsburgh 5) Evaluation: Workshop feedback
Fall 2007	1) Second round of course offerings 2) Pilot new courses at participating schools (participants from the summer 2007 workshops) 3) Propose workshop at SIGCSE 4) Presentation and workshop at CIT (national community college conference) 5) Evaluation: Obtain baseline data from these participating schools 6) Evaluation: Data collection in pilot courses
Spring 2008	1) Second round of course offerings at participating schools 2) Birds-of-a-feather meeting at SIGCSE for early adopters to discuss initial results 3) Generate a wider number of participating schools (who will attend summer 2008 workshops) 4) If accepted, run workshop at SIGCSE 5) Evaluation: Data collection in participant courses
Summer 2008	1) Second round: Faculty development workshops in Atlanta, Philadelphia, Providence, and Pittsburgh. 2) Evaluation: Additional data collection and analysis
Fall 2008	1) Evaluation: Summative data collection and analysis

Evaluation Plan

As is suggested by the timeline above, assessment and evaluation activities will be ongoing throughout the proposed project. Attention will be given to both the formative and summative aspects of evaluation. The formative assessment plan described in the section that follows is designed to acquire information that may be used to improve project-related activities as they are being implemented. In other words, this project uses the cyclical model of assessment that was described in the RFP. The summative portion of this assessment plan is designed to measure the attainment of the project outcomes. These outcomes were discussed early and were constructed based on the project goals and objectives.

An external evaluator, Dr. Barbara Moskal, will provide an unbiased interpretation of the project results. Dr. Moskal has a great deal of experience in the assessment of educational interventions (see biographical sketch). She will work with a graduate student in the selection or development of appropriate assessment instruments, and in the analysis and interpretation of the collected data. She will also work

closely with the project investigators to ensure consistency of the assessment instruments with the project goals and objectives. Although the project investigators will be responsible for the administration of assessment instruments, she will assist in this process, to ensuring uniformity across administrations. She will further provide ongoing feedback to the investigators concerning the formative assessment results.

All appropriate human subject procedures will be followed throughout the investigation. This includes receiving approval from internal Human Subject Review Boards at all institutions, obtaining all necessary written consents, and the maintenance of strict confidentiality throughout the study.

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 Alice and Media Computation Approach: As part of this project, Alice and Media Computation instructional approaches will be interwoven to create a text and instructional materials that provide a blend of program visualization, 3-D animation, and multimedia approaches for introducing fundamental computing and object-oriented programming concepts as is described in Objective 1. The interwoven techniques and materials will be the focal point of faculty development workshops at two levels: (1) community college; and (2) four-year college and university. To provide formative feedback on the blended Alice and Media Computation approach and thus, with respect to Objective 1, at least four CS faculty will review the workshop techniques and materials. The peer review feedback will be used to guide revisions. Peer review is a widely accepted technique for examining the content, structure, and criterion validity of instructional techniques and materials [3].

Online Survey: A survey will be developed and administered on-line with respect to the website that is created to showcase student work. This survey will acquire anonymous feedback from users for improvement purposes. Adjustments made based on this information will assist in the attainment of Objective 2.

Workshop Feedback: At the conclusion of each day of all workshops, anonymous feedback will be acquired from workshop participants. They will be asked what additional materials/support they require in order to implement this new approach in their classrooms. A formal survey will also be administered at the end of the workshop to examine the participants' perceived workshop benefits. This formative evaluation method will be used to improve the implementation process with respect to Objective 3.

Telephone Conference Calls: To provide support to and acquire feedback from participating faculty, bi-monthly teleconferences will be held. The investigators, evaluator and faculty will all participate in these meetings. This will support the exchange of project information and ongoing project improvement with respect to all objectives.

Summative: The summative evaluation will primarily use quantitative research techniques to support the statistical analysis of our efforts. At each location, pilot and control classrooms will be selected, if possible. The pilot class will use the materials developed here as part of their first programming course; the control classes will participate in a traditional first-year programming course. If a control classroom is not available, historical data will be collected before the intervention is implemented for comparison purposes. In pilot and control classrooms, a survey will be given to determine student background in mathematics, computers, and CS. The data collected by the readiness survey will be used to examine relationships between mathematics and CS preparedness, student performances and rates of attrition.

Development of Materials: Several of the project outcomes that were previously discussed require the development of materials. By the conclusion of the grant, the following materials will be available for the evaluator's review: a completed textbook and set of instructional materials (Objective 1, Outcome i),

online support for instructional materials (Objective 1, Outcome ii), online showcase of student work (Objective 2, Outcome i) and materials for the summer workshop (Objective 3, Outcome i).

Online Support Community Statistics: Two online support communities will be provided: (1) an Alice and Media Computation support network for instructors (Objective 1, Outcome ii), where the PIs will provide online instructional materials and respond to email requests; and (2) a community forum website, where students will be able to share completed work and work in progress with others (Objective 2, Outcome i). Statistics will be collected on usage of these two sites for the evaluation of Objective 1, Outcome iii and Objective 2, Outcome ii. As stated in these outcomes, evidence of success will be evaluated as these sites being accessed at least thirty times each week.

Surveys: Several closed response surveys will be developed and administered to participants. To measure Objective 1, Outcome iv, students that complete the new program will be asked to respond to a closed response survey at the conclusion of the course. A survey will also be created that will be administered on-line to students that visit the on-line showcase. This survey will be brief and will ask users to evaluate the appropriateness of the site, Objective 2, Outcome iii. A survey will also be developed and administered to the teachers that participate in the summer workshops. This survey will be used to evaluate Objective 3, Outcome ii.

Retention and Attraction Statistics: At the collegiate level, statistical comparisons will be made between the number of students in the pilot and control classrooms that are retained in a CS degree, Objective 1, Outcome v and vi. Where control classrooms are not possible, historical data will be used for comparison purposes. This data will suggest whether the Alice and Media Computation approach 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. Attraction statistics will also be examined as a measure of Objective 2, Outcome iv. Within the participating schools, the number of declared computer science majors and minors is expected to increase by 5% when compared to historical data.

Pre and Post Workshop Assessments: The investigators and evaluator will jointly construct a pre and post content assessment for participating teachers to complete. The pre-assessment instrument will be administered at the beginning of the workshop and the post-assessment will be completed at the end of the workshop. Data collected through this instrument will be examined using a paired t-test. A statistically significant change in teachers' knowledge is anticipated as is reflected through Objective 3, Outcome iii.

Capabilities of the PIs

We believe that we are uniquely qualified to carry out this project. Dr. Cooper has been teaching introductory CS for 11 years. He also has 10 years of industry experience and is familiar with what industry requires from CS graduates. Dr. Dann has 15 years experience teaching introductory CS. Her expertise spans applied research in program visualization, curriculum development, and workshop presentations. Dr. Dann has served as a member of the Visualization Working Group, studying the effectiveness of visualization in CS education. Drs. Cooper and Dann have been working with Alice for 7 years. Their 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. In the past, this team has always been responsive to implementing our requests and suggestions. We have written a book published by Prentice Hall [10] concerning the use of Alice in the CS classroom. 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 partial list.) As

part of our current CCLI grant, we have offered workshops to assist faculty in learning to teach with Alice, and have had more than 100 faculty requests to participate. We have demonstrated on our college campuses that teaching Alice really does make a difference for our at-risk CS majors.

We have significant project management experience with NSF grants. Dr. Cooper is PI and Dr. Dann co-PI of a million dollar ATE grant (concluding next summer) involving 7 schools, approximately 80 faculty, and more than 3000 students. While we do not yet have final results, our initial results, in terms of student attitudes, retention within, and attraction to the major, are promising.

Dr. Mark Guzdial and Ms. Barbara Ericson have a long history of innovation in computing education. Dr. Guzdial has been teaching introductory computing to adults and children since 1980. One of his first academic publications was on his work using music as a context to teach composition of functions in Logo to third through sixth graders in 1986. He won an NSF Career award in 1995 for his work in supporting modeling and collaboration in computing and engineering learning. Ms. Ericson is the Director of CS Outreach for the College of Computing at Georgia Tech. She is the developer of the ICE@GT workshops for high school teachers in the state of Georgia and coauthor on the Java Media Computation textbook. Dr. Guzdial developed the Media Computation approach for teaching introductory computing, which has since been adopted by several institutions (from two year colleges to research universities). Dr. Guzdial has written two textbooks to support others in adopting this approach.

Dr. Moskal, an educational assessment expert, will oversee the implementation of the assessment and evaluation process throughout this project. Dr. Moskal has extensive experience in assessment and evaluation and has worked as an assessment consultant on three other NSF-funded CS curriculum development projects. Dr. Moskal will be responsible for all data analysis. (See biographical sketches.)

Our experience in working with female and minority students: Dr. Guzdial has significant experience in working with female students, as his Media Computation courses at Georgia Tech have been very popular among women. Dr. Cooper has been running PAthways to Careers in MAthematics and Computer Science (PACMACS) for the past 5 years. PACMACS enables 10 African-American students from (non-magnet) Philadelphia public high schools per semester to take a programming class with Alice (as well as a mathematics class) at SJU, as well as receiving mentoring, and other preparations for college. PACMACS has been quite successful, with many of its graduates either majoring in, or graduated with college degrees in CS and/or engineering. Dr. Dann has been working with the CS and Hawaiian Studies faculty at the University of Hawaii to create a combined Alice and Hawaiian Studies course for native Hawaiian college students.

Our experience in working with community college faculty: Drs. Cooper and Dann are just completing a successful three-year grant to work with the Community College of Philadelphia (an urban HBCU), Camden County College (a suburban campus), and Tompkins Cortland Community College (a rural school). We have learned the importance of providing detailed course and supporting materials, as community college faculty typically have much higher teaching loads than four-year school faculty. Additionally, we have learned the value of community college faulty getting together to share experiences. We intend to split up some of the sessions of our three-day summer teacher workshops into "community-college" and "four-year college" sessions to take advantage of this dynamic.

Dissemination

Over the course of this grant, we expect to generate knowledge in two main areas: 1) research results concerning students' decisions to continue with or leave CS and computing-related fields, and 2) practical experience in how to use the Alice/Media Computation approach to increase enrollments of students in undergraduate CS courses. As part of our work on this grant, we are planning Alice/Media Computation specific events to facilitate the dissemination of our results to researchers and teachers who are interested

in the use of Alice/Media Computation to increase retention and the diversity of students in CS related courses at their home institutions.

Summer 2007 and Summer 2008 Workshops: We will host two three-day workshops the first summer and four the second summer on Alice/Media Computation. Our goal is to provide hands-on experience in using the software and to share hints and techniques for successful adoption of this approach.

Reaching the Larger CS Education Community: We plan to present our research findings at professional conferences to help inform the design of CS courses that address the needs of a wide range of students. In addition to submitting journal articles, we expect to present the results of our work at SIGCSE, ITiCSE, FIE, CHI, HCC, and CIT. To reach community college faculty, we plan to give talks and run workshops at the Conference for Information Technology and Innovations conferences. Additionally, we expect to present at several Prentice Hall Information Technology conferences.

To enable instructors at other academic institutions to begin using the Alice/Media Computation approach, we plan to produce a textbook and create an online community to provide support and materials for educators teaching Alice/Media Computation courses. We will also send announcements about all workshops for prospective Alice/Media Computation teachers at the community college, and four-year school levels to the SIGCSE members list.

Future Plans

The Institute for Computing Education at the College of Computing at Georgia Tech has been working to improve high school computing courses. We will adapt the materials that we create for community colleges and four-year colleges so that they can be successfully used by high school teachers. In particular we will adapt the materials for use in Advanced Placement (AP) CS courses. Exposure to this combined approach in high school should increase the number of students with an interest in CS.

As we continue to pursue Alice/Media Computation, we plan to begin work that focuses on Hispanic students and Pacific Islanders. Because the focus of this work is on teaching programming, rather than teaching the English language, we plan to produce versions of our systems and materials in Spanish and other relevant languages. Other possible groups that may well benefit from our Alice/Media Computation approach include Native Americans who have a strong storytelling culture and the deaf community where the use of ASL as a means of communication. Alice/Media Computation will allow this group to benefit from the strong visual nature of our approach.

Summary

The purpose of this proposal is to merge two successful curricular approaches to teaching introductory programming at the college level, Alice and Media Computation. Alice is a software environment that supports 3-D programming using a drag-and-drop editor. Through the use of Alice, students can learn the basics of algorithmic programming without the burden of syntax errors. Media Computation teaches students to use text-based programming to edit 2-D graphics and sounds. A major benefit of this combined approach is the use of animation and storytelling. Students who have grown-up in a multimedia world are likely to find these programming environments to be both challenging and motivating. A primary outcome of this investigation is expected to be an increase in the attraction and retention of CS majors and minors.