Final Report: Adventures in Alice Programming

CRA - CDC DMP Duke University Summer Internship 2008

Gaetjens Lezin

Mentor: Professor Susan H. Rodger

Adventures in Alice Programming Project

Abstract

The Adventures in Alice Programming Project of 2008 at Duke University ran during the months of June and July. The project involved two main events. First, there was a two-week workshop to teach Alice and investigate the use of Alice in middle schools and high schools. The second was two third week options for teachers to develop lesson plans timed with two one –week student camps. The second major event involved two separate groups of students and teachers for the first and second week. The goal of the project is to obtain data about attitudes and how well kids and teachers are able to learn Alice. A pre and post test was given at the beginning and ending of each event. The results are in the process of being reviewed at an offsite in Colorado. The first set of tests evaluated knowledge on Alice material, the second was a survey about attitudes toward computing.

Related Prior Work

Previous studies have shown the number of students choosing computer science majors has been declining dramatically. In one article published by Professor Rodger of

Duke University, alarming statistics about this issue were reported. One, “According to the Computing Research Association’s fall 2000 report, 23,416 students selected Computer Science or computer engineering as their intended major”. [6] This is especially disturbing because of another statistic from the U.S. Bureau of Labor Statistics. They expect a 1.15 million rise in the number of IT jobs over the decade 2002-2012. It is apparent that something needs to be done to address this issue. The Adventures in Alice Programming Project is exploring one way of approaching a solution. The Alice approach aims to reach out to students in grades 5 – 12 with a wide spectrum of ethnicities both male and female. Ultimately the Adventures in Alice Programming Project would like to influence a diverse group of students. Groups such as women, minorities, and underrepresented groups are the focus [6]. Previous publications written about Alice have stated educational levels should be considered when attempting to apply the Alice approach. Problem solving and understanding algorithms are common hurdles that novice programmers have difficulty overcoming [2]. The authors’ stress, in order to develop progressive innovative instructional methods, researchers must dig below the surface. “It is easy to say that they do not know how to solve problems. But this is too simplistic.”[2] This quote makes it clear that a total solution must consider all levels of education and educational backgrounds. there needs to be an evaluation and serious consideration of what level of schooling and difficulty of classes the students have taken. This will benefit the child because they will start off at an appropriate level, not one that is too advanced for them or not advanced enough.

The Alice approach is not the first visual tool used in teaching novice programming. [3] “In the past decade educators have developed a myriad of tools to help novices learn to program.” [3] A few named in the article are JPie, Karel Universe, Raptor, Iconic Programmer, and Visual Logic. This statement leads to a few distinguishing questions. How is Alice different? Why is it better? Can it really influence the future of computing, so that students don’t shy away from computing majors because of a fear of programming? These questions are addressed in a publication by Cooper, Goldman, Carlisle, McNally, and Proulx. In summary it relays the idea of the main educational benefits of Alice consisting of the following major points; one, ability to create movies or computer games [3] (referred to as “programming with purpose” in the reading.) Two, Alice avoids the hassle of syntax errors and compiler messages that quickly confuse novice programmers [3]. The drag and drop ability of Alice makes this possible. Three, Alice is always in a state where it can run [3]. This allows students to see exactly how their code affects the objects in their program. The college professors who have used visual teaching tools such as Alice in their courses gave this statement: “With immediate feedback, we have seen students understand concepts more and our course can cover more concepts without losing valuable hands-on experience.” [3] The best part of the three features mentioned above is their abilities to motivate. Attitude is very influential when students are deciding to pursue careers not only in computing but all careers. The same article reports, “we have found that the ability to direct your own movie is extraordinarily attractive to a wide range of students”. [3] Alice facilitates directing 3 - D movies. Certainly this is a direct result of the features mentioned above. This demonstrates why

Alice is a great tool that is a better choice than other visual tools for accomplishing the ultimate goals of the Adventures in Alice Programming Project.

Other publications describe, in further detail, specific programming concepts that Alice would be a great tool to use when teaching. These include looping and recursion, algorithms, methods, functions, and object oriented design strategies. A publication written by Dann, and Pausch, does a great job of contrasting Alice and other tools that are used to teach recursion. It also gives specific Alice examples of Tail, Multiple, and Structural recursion [4]. This article shows the flexibility of Alice to approach teaching solutions from different directions with different examples. A different publication by Dann, Cooper, and Pausch demonstrates how the Alice approach can introduce many fundamental programming concepts. These are basic ideas that need to be learned so a student can have a solid foundation in computer science. They introduce teaching approaches for introducing object level methods as well as static methods called from main in a java program. In the writing they are described as "character-level," and "World level." [5] Further reading reveals teaching strategies for functions, parameters, and variables.

Introduction

Alice is a 3-D virtual World environment. It allows novice programmers to create 3 – D animations, movies, and games. The Adventures in Alice Programming Project aims to explore teaching methods and developing Alice material that can be adapted into school curricula. The goal of injecting Alice into curricula across the country is to increase the number of students who choose computer science majors. It has been accepted across the members of this field the number of students choosing computing as a path for higher learning has been decreasing. The Adventures in Alice Programming Project is pioneering one possible solution to this problem. This research project will collect data across the country. The following is a list of participating sites;

- Durham, North Carolina
- Virginia Beach, Virginia
- San Francisco, California
- Denver, Colorado
- Charleston, South Carolina
- Oxford, Mississippi

Funding for the Adventures in Alice Programming Project is provided by the National Science Foundation through grant number Grant ESI-0624642. Additional funds were provided by IBM.

Teacher Training

The first major event of the Alice research project was the teacher training workshop. There were two sets of high school and middle school teachers. Each set had to attend 3 weeks of Alice Training. The first week of the training workshop ran from June 16 until June 20. Professor Rodger of Duke University and Professor Don Slater of Carnegie Mellon University conducted the event for the first week. Each day the teachers received training on Alice strategy and technique. Lessons were presented either in a workshop format, with short lecture explanations followed by hands on experimentation, or with teachers and students building a world along with the presenter. The same format was used for the student camp and extended teacher workshop. The research teaching assistants consisted of myself and four Duke undergraduate students; Deborah Nelson, Jenna Hayes, Henry Qin, and Ruthie Tucker. Each one of us also gave several presentations over the four weeks. During the weeks of training we assisted whoever was giving a presentation by walking around and answering any questions the teachers had. It was important to make sure every lesson ran from start to finish with as little interruption as possible. This would allow the presenters to cover as much Alice material as possible. At the same time, we had to make sure the number of teachers left behind was kept to a minimum. On Friday Professor Rodger organized an animation show in which all the teachers demonstrated the worlds they had been working on. This was a very useful observation tool, it became apparent what programming concepts they were actually using and we could see if they were applying what we taught them throughout the week. Through the course of the first week topics covered included; [6]

- Introduction to using Alice
- Motion (translation and rotation) during scene set-up
- Problem Solving and Storyboard Design (Participants work on building a simple Alice storyboard)
- Storyboard -> Code
- Revisiting translational and rotational motion
- Functions (built-in) and parameters

- Interactivity and Events
- Camera Control
- Participants work on building a world for their storyboard

- Repetition
- 3D Animation Tips
- Sound
- Group reviews, discussions of storyboards (Storyboards on the wall)
- Work Session: Build a world from storyboard (second one)
- Animation Fair: Demo of Day 3-4 worlds

The second week of the teacher training workshop ran from June 23 to the 27th. The teachers made noticeable progress in the first week so this week involved more involved Alice concepts. Topics covered during the second week were the following; [6]

- Lists
- Work session: pairs make topic lists

- Mutable variables
- Presentation of topic lists from Day 6
- Sample unit plan - presentation

- Advanced Functions
- Presentations of unit plans from Day 7
- Sample lesson plan - presentation

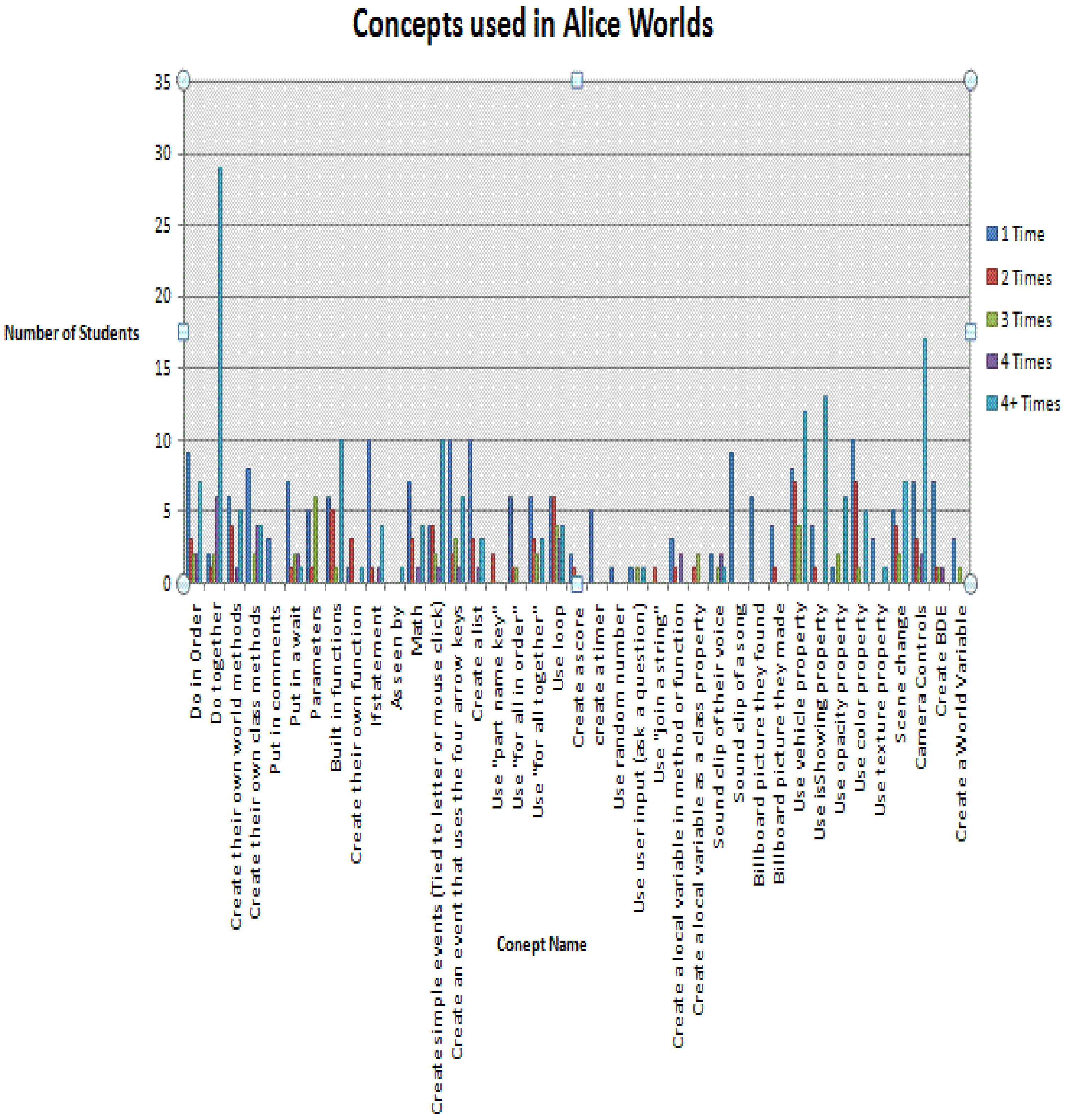
- Array/Lists

- Presentation of lesson plans and demo worlds
- Work session: in pairs, build a second lesson plan, including a project assignment
- Presentation of second lesson plan and project assignment

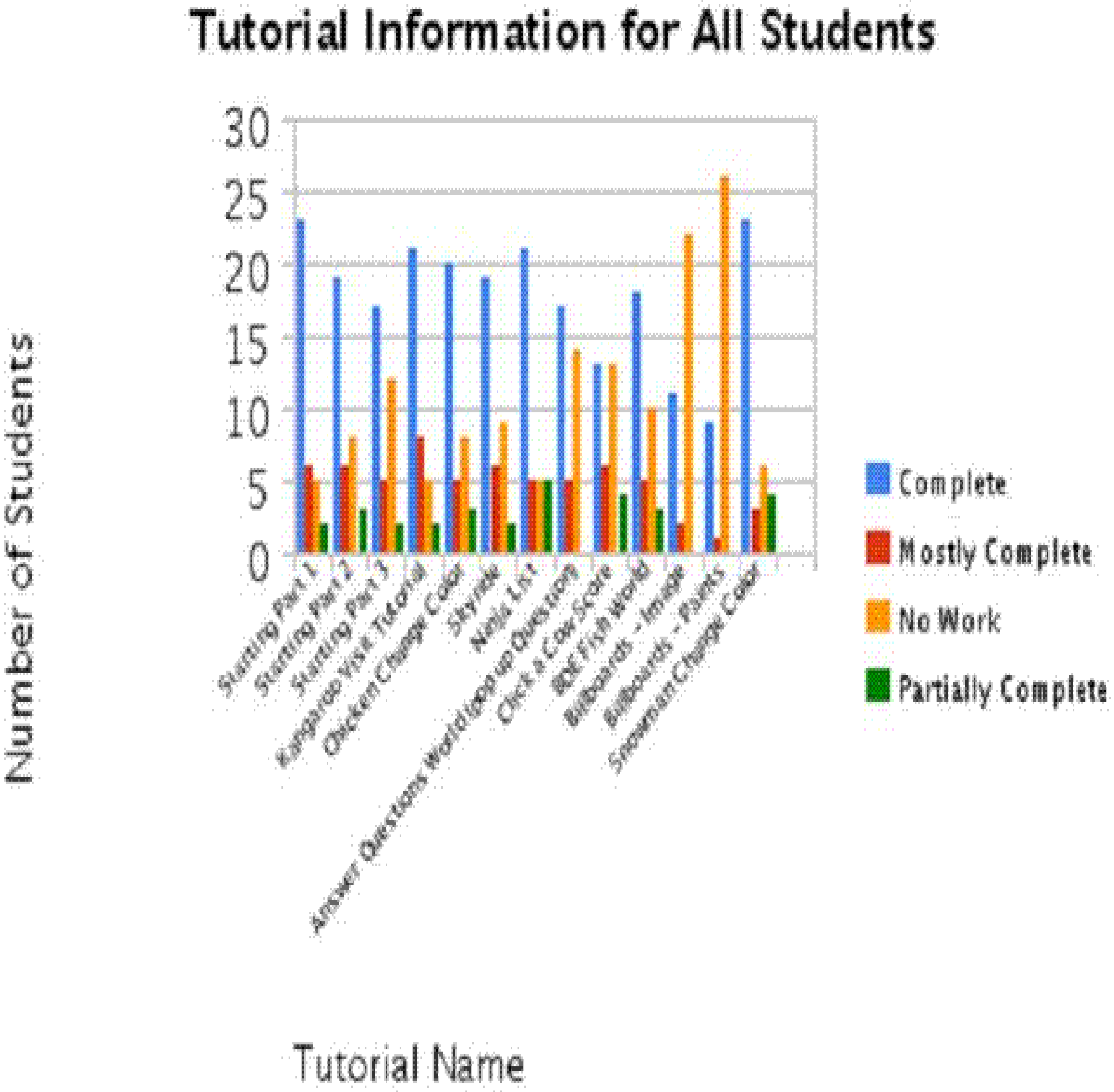
After the two weeks of teacher training, the Adventures in Alice Programming Project conducted a one week student camp and extend teacher workshop. The event began on July 7 and ran until July 11th. During this most of the teachers focused on creating multiple lesson plans to meet North Carolina Public School standards. The lesson plans they created can be found on the Alice Workshop web page. The students who attended this first camp session were taught much of the same lessons the teachers were. The format was the same, (lecture style with students following along). They were given free time to build their own unique worlds which they demonstrated during the middle of the week and at the end. The final week of the Adventures in Alice Programming Project training camp ran from July 14 to the 18th. A different set of kids and teachers attended this event. Everything else remained identical to the previous week. The reason for this was to remain consistent for the research study.

Data Collection and Organization

The second portion of the Adventures in Alice Programming Project involved a closer examination of the Alice worlds that the students created. I spent a few days organizing and building a data base using Microsoft access to hold the information. The other research assistants collected all the data. They combed through every Alice world that each student created to amass a wide variety of information. We all collaborated with Professor Rodger to determine the type of information we needed to extract from what we collected. A few prominent points we looked for were Alice structure statements. These are specific terms used in the Alice programming language.   For example, do together, for all in order, for all together, BDE (beginning during end) events to name a few. The research team also looked for programming related concepts such as loops, class variables, method variables and general coding style. The graph below shows different concepts we searched for and how many students used it either once, twice, three, four, or four plus times. This chart makes it easy to recognize what the students learned, and what they actually were interested in using in their own animations.


Concepts used in Alice Worlds
Number of Students
35
30
25
20
15
10
5
0
1 Time
2 Times
3 Times
4 Times
4+ Times
Do in Order
Do together
Create their own world methods
Create their own class methods
Put in comments
Put in a wait
Parameters
Built in functions
Create their own function
If statement
As seen by
Math
Create simple events (Tied to letter or mouse click)
Create an event that uses the four arrow keys
Create a list
Use "part name key"
Use "for all in order"
Use "for all together"
Use loop
Create a score
create a timer
Use random number
Use user input (ask a question)
Use "join a string"
Create a local variable in method or function
Create a local variable as a class property
Sound clip of their voice
Sound clip of a song
Billboard picture they found
Billboard picture they made
Use vehicle property
Use isShowing property
Use opacity property
Use color property
Use texture property
Scene change
Camera Controls
Create BDE
Create a World Variable
Conept Name

One interesting fact that is clearly apparent is “do together” is something that the kids pick up well. Over 25 of them used it more than 4 times in the worlds they created. This was a very simple concept and most of them understood how to use it correctly. On the other hand a concept such as a loop was not seen as much. According to our chart less than 5 students used it more than four times. Another chart generated from our information shows how well the children followed along according to the different tutorials we presented to them. Since all the tutorials involved building Alice worlds we were able to go and look at each tutorial world the students built and examine them to see exactly how much of the world was completed. The chart below demonstrates this.


Tutorial Information for All Students
Number of Students
30
25
20
15
10
5
0
Starting Part 1
Starting Part 2
Starting Part 3
Kangaroo Visit Tutorial
Chicken Change Color
Skyride
Ninja List
Answer Questions World (pop up Question)
Click a Cow Score
IDE Fish World
Billboards - Image
Billboards - Plants
Snowman Change Color
Complete
Mostly Complete
No Work
Partially Complete
Tutorial Name

The Adventures in Alice Programming Project has shown us that our students for the most part paid attention and completed the tutorials we taught them. Although there were a few who did not, an overwhelming percentage did follow along. I believe this is because they enjoy learning to create their own animations to fulfill story board ideas that they come up with. Alice provides an easy way for them to do that and as we can see in our chart they are willing to learn how to use the Alice features. Only time and more research will show if the Alice approach will in fact change the statistics and result in more students entering computing majors and pursuing computer related careers. The Alice Research Project 2008 at Duke University was a small part of a big effort to work towards a total solution for solving the growing problem.

Appendix

[3] Cooper, Stephen, Kenneth J. Goldman, Martin Carlisle, Myles McNally, and Viera Proulx. Tools for Teaching Introductory Programming: What Works? 2006. <http://dsys.cse.wustl.edu/resources/papers/gross-sigcse-2006.pdf>.

[2] Cooper, Stephen, Wanda Dann, and Randy Pausch. ALICE: A 3-D TOOL FOR INTRODUCTORY PROGRAMMING CONCEPTS. Apr. 20. July 2008 <http://www.sju.edu/~scooper/alice/ccscne00.pdf>.

[4] Dann, Wanda, and Randy Pausch. Using Visualization To Teach Novices Recursion. July 2001. <http://www.sju.edu/~scooper/alice/p109-dann.pdf>.

[5] Dann, Wanda, Stephen Cooper, and Randy Pausch. Objects: Visualization of Behavior and State. June 2003. <http://www.sju.edu/~scooper/alice/p109-dann.pdf>.

[6] Rodger, Susan H. "Adventures in Alice Programming." July-Aug. 2008. July 25 <http://www.cs.duke.edu/csed/alice/aliceinschools/workshop08/schedule.php>.

[1] Rodger, Susan H. An Innovative Approach with Alice for Attracting K-12 Students to Computing. Recent Paper Publications for Susan H. Rodger. 7 May 2007. 14 July 2008 <http://www.cs.duke.edu/csed/rodger/papers/ibmmay07.pdf>.