The Use of Computer Animation in Teaching Discrete Structures Course

Chi-Cheng Lin **Mingrui Zhang**
Computer Science Department
Winona State University
Winona, Minnesota, MN 55987
{clin, mzhang}@winona.edu

Abstract

Abstract theoretical concepts of computer science are among the most complicated to teach. The instructors expect students to comprehend a number of computer algorithms in a short period of time, often within a week. Traditionally, the course "Discrete Structures" is taught to students without hands-on practice or programming. The instructor explains the algorithms by manually going through examples. Doing so may demand a large amount of time and cognitive effort on the part of the instructor. It also requires students to manually work through the algorithm for different examples to fully understand the behavior of the algorithm.

This paper presents a project that adopts a student-centered, learning by doing approach for students to learn and explore computer science theories by adopting computer animation and visualization techniques.

Introduction

Abstract theoretical concepts of computer science are among the most complicated to teach. Students who are encountering computer theories for the first time find that they must delve into logic, discrete mathematics, and programming. The instructors expect students to comprehend a number of computer algorithms in a short period of time, often within a week. They try hard to motivate students to learn, but students struggle and feel hampered and frustrated. However, understanding computer science theories is critical. Traditionally, the course "Discrete Structures" is taught to students without hands-on practice or programming. This course serves to introduce fundamental computer science theories to computer science majors. This course provides students with an introduction to the applications of discrete mathematical structures to computer science with emphasis on algorithms, analysis of algorithms, and structures such as graphs and trees. The instructor explains the algorithms by manually going through examples. Doing so may demand a large amount of time and cognitive effort on the part of the instructor. It also requires students to manually work through the algorithm for different examples to fully understand the behavior of the algorithm.

People tend to have a better response to pictorial information and extract more information from pictures than from text and numerical data. We can take the maximum advantage of pictorial information via animation, a dynamic, powerful visualization tool that assists humans to gain a better comprehension of abstract concepts as well as physical activities. The authors designed and implemented a project that adopted computer animation and visualization technologies to aid students to comprehend computer theories in teaching Discrete Structures. The primary goals of the project are 1) to establish a friendly and effective environment to teach computer theories by adopting advanced computer technologies, such as animation and visualization techniques, and 2) to motivate students to learn computer theories by creating animation products to visualize theoretical concepts. The project used a student-centered, learning by doing approach. The authors designed a framework suitable for developing hands-on labs for class use using the software 3D Studio MAX. Students in the class then developed computer animation products using the framework to visualize the computer algorithms and to demonstrate computer theories. Students worked in groups, which helps them learn and explore computer science theories collaboratively.

This paper discusses the approach and planning of the project, the animation framework, and the assessment of the effectiveness of this project. The results and snap shots of some of the student animation products are also included in the paper. We conclude the paper with the discussions on the experiences gained and lessons learned from this project.

Project Overview

Background

Various approaches have been adopted to use algorithm visualization in teaching computer algorithms. One approach is to provide animations for students to observe how computer algorithms work. A collection of such work can be found on-line at [5], as well as at many other web sites on the Internet. Although this kind of passive viewing is helpful for students to see the steps an algorithm performs, to make learning more effective, solutions are needed to make students take a more active role in algorithm visualization. Approaches such as inserting stop-and-think questions in the visualization [3], and the provision of dual modes, namely "show me" (i.e., passive) mode and "I'll try" (i.e., active) mode [2], have been proposed for this purpose. Another approach, such as Stasko's Samba [4], requires students themselves to create animations of computer algorithms.

Both of us teach the Discrete Structures course in our institution. We completed a project that employed the idea of algorithm visualization in the class. Our approach, similar to Stasko's, required students to be actively involved in creating animations to demonstrate how computer algorithms work. The details of our project are given in this section.

Plan and Approach

Our project was carried out over a period of two semesters in which the first semester served as the pilot study. The idea is to use the lessons learned and the feedback from students in the pilot study to improve the execution of the project in the second semester. Our approach is a student-centered approach and we would like students to be highly involved in the project. The project counted as 5% of a student's final grade to make students serious about it, while not to replace other important pedagogic activities of the class.

We used a professional, commercial software package, 3D Studio MAX, for developing animations. Students had to develop computer animations to visualize a variety of computer algorithms using 3D Studio MAXScript [1], the script language of 3D Studio MAX. This approach is unlike most of the algorithm visualization projects that use open software, such as JAVA, to develop their work. Although we want students to take an active role in this project, we don not want them to spend a significant amount of time figuring out how to use MAXScript – they should spend most of their time on studying algorithms. Therefore, a short introduction for using 3D Studio Max to develop an animation was given to the students. We also developed sample tutorial scripts showing students the fundamentals, such as how to create 3D objects, how to create vertices, edges, and labels of a graph, how to do recursion, and how to timestamp for animations.

Group projects were assigned to groups of two students. Toward the end of a semester, students presented their animations to the class with the explanations of the computer algorithms. A course evaluation was performed at the end of each semester. Based on the feedback and the results of students' projects from the pilot study, we found that most students did their projects by manually animating each step of execution of a computer algorithm. In order to let students focus more on the computer algorithm, we improve the demo scripts by separating data structures from the algorithm. We identified two types of data structures for sorting/search algorithms and graph algorithms, respectively. We expect students to put more efforts into debugging their algorithms than delving into the appearance of the animation. The revised scripts and demos were then used in the second semester in which this project was carried out, and proven a success.

To summarize, our project has the following characteristics:

- Student-centered approach. Our project adopted a student-centered, learning by doing approach. Students were required to create animation products in stead of point-and-clicking to view animations passively.
- Collaboration learning. Students worked in groups of two so that they can learn from their fellow students and explore computer theories collaboratively.
- Separation of data structure and algorithm. By providing pre-constructed data structures to students, students can spend more time on developing algorithm animation than creating data structure in the first place.
- Use of professional commercial software. 3D Studio MAX provides a complete, powerful set of 3D object modeling and animation functions. Its script language, MAXScript is easy to learn in spite of the complexity of 3D Studio MAX features. As the students can view fancy results immediately, they were highly-motivated to create their own animation products.
- Movies as final products. Using the commercial package makes generating an animation into a movie done easily. With a movie, the functions of rewind, fast forward, pause, and starting at any point in the animation comes along naturally.
- Availability of script files. Script files created by students are made available to others. They are free to examine, modify, and experiment on other groups' script files.

Project Assessment

It is important to assess and then improve the effectiveness of our project. We performed surveys at the end of both semesters. The evaluation performed at the end of the pilot study served as an assessment tool for us to improve the execution of our project in the second semester.

The results from the evaluation performed at the end of the pilot study shows that students enjoyed their projects, considered the projects excellent ideas. The animation projects enforced their learning of the computer algorithms. Most students selected the highest two ratings for each category of the survey. In general, the pilot study was a success. We did receive comments, such as "Force a general implementation of

algorithms …', "… to get assignment earlier …", "I didn't know what to expect …", and "… should teach more about MAXScript language, so we can make it more attractive …" from students, however.

Based on the results of the surveys, especially on the comments, we adopted the following approaches to improve.

1. Students will be given a month for their project.
2. A tutorial to train students to use MAXScript to make animation will be made available to the class.
3. Sample scripts need to be improved to strengthen on the implementation of the computer algorithms.
4. High quality animation demos need to be provided to help students to improve the quality of their animations.

At the beginning of the next semester, some senior students from the Computer Graphics course were assigned to create a tutorial for programming animations using MAXScript. The result was a high quality tutorial introduction to the MAXScript along with impressive demonstrations. Those scripts were made available to students taking discrete structures as early as possible. Over all, the project was a success. Students' suggestions include "Do more projects", "Make more of it, it is really fun", "Unexpectedly fun." They considered the projects are "relevant to the course" and "it is helpful" for their study. Besides the positive ratings and comments, certain problems remain, nevertheless. Some students were not motivated enough to study discrete structures. They thought the subject is difficult to grasp, boring to learn anyway. Some students regarded the animation project as an add-on to their work load.

Results and Examples

More than a dozen of algorithm animations have been produced by students taking discrete structures. In general, the students' projects can be classified into two groups. The first group consists of sorting/search algorithms, such as bubble sort, insertion sort, and binary search. The second group consists of graph/tree algorithms, such as Dijkstra's shortest path, Prim's minimal spanning tree, and binary search tree.

Sample snapshots of two of the animation products created by students are included in this paper. Figure 1 shows two snapshots taken from the insertion sort animation.

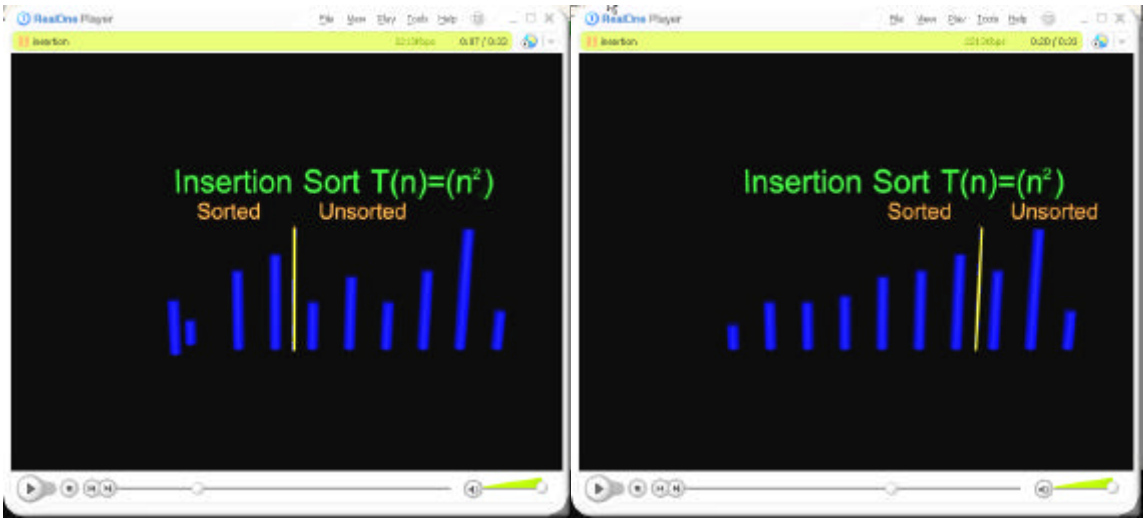

Figure 1: Snapshots from Insertion Sort Animation

Figure 2 shows two snapshots taken from Dijkstra's shortest path algorithm. The graph used in this animation served as a uniform framework for all graph algorithm animations. As mentioned earlier, the uniform framework allowed students to focus more on algorithms than on animation.

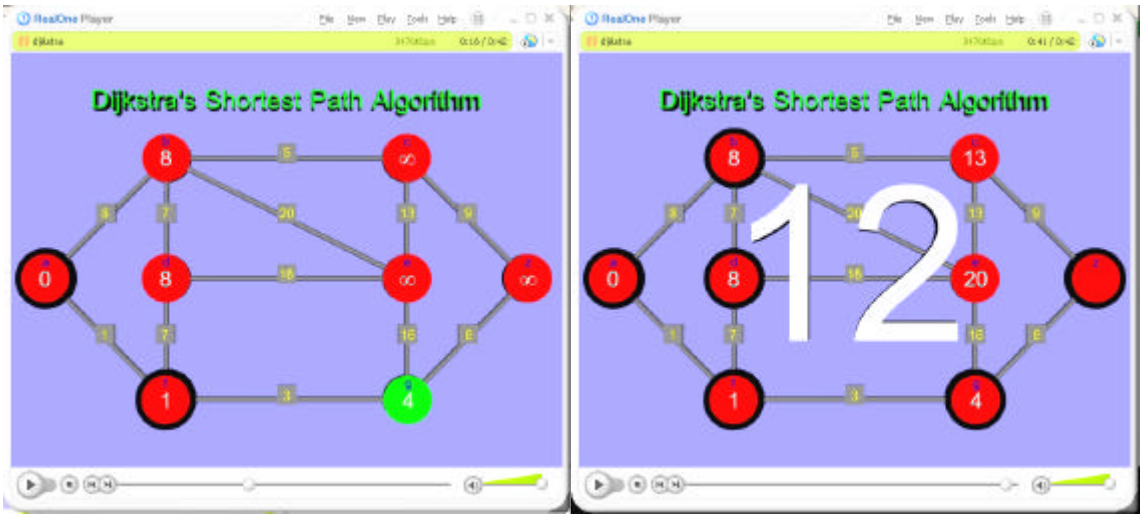

Figure 2: Snapshots from Dijkstra's Shortest Path Algorithm Animation

Conclusion

Algorithm visualization is a valuable teaching and learning tool in computer science, and was adopted in our Discrete Structure class. We employed a student-centered approach which required students to create 3D animation products instead of watching animations passively. Students worked in groups so that they can benefit from collaborating learning. 3D Studio MAX was used to develop animation products. The major advantage of using commercial software is the complete, rich set of modeling and animating tools available for creating "cool" animation products, and it really motivated

students to study computer algorithms. Students also benefited from exposure to professional software package. Cost is a concern, however. In order to let students focus more on the computer algorithm, we created a framework that separated data structures from the algorithm so that students can put more efforts into debugging their algorithms than delving into the appearance of the animation. The result of our project is promising. Most students considered learning algorithms by making animation is helpful and enjoyable, and would like to do more animations. Only a few students considered it as a burden extra to their study.

Based on the experiences obtained from this project, we are currently working on projects for improving teaching and learning computer science courses besides the Discrete Structure course. One of such projects is the use of animation in teaching the Introduction to Assembly Language and Architecture course. Another is the research on using projects in the upper division courses to support teaching lower division courses.

References

1. Bicalho, A. and Feltman S. (2000). *Mastering MAXScript & the SDK for 3D Studio MAX*. Alameda, California: SYBEX.
2. Jarc, D. J., Feldman, M. B., and Heller R. S. (2000). Accessing the Benefits of Interactive Prediction Using Web-based Algorithm Animation Courseware. *In Proceedings of the thirty-first SIGCSE technical symposium on Computer science education*, Austin, Texas, 32(1):377-381.
3. Naps, T. Eagan, J., and Norton L. (2000). JHAVÉ -- An Environment to Actively Engage Students in Web-based Algorithm Visualizations. *In Proceedings of the thirty-first SIGCSE technical symposium on Computer science education*, Austin, Texas, 32(1):109-113.
4. Stasko, J. (1997). Using Student-Built Algorithm Animations as Learning Aids. *In Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education*, San Jose, California, 29(1):25-29.
5. Zhang, M. A small collection of algorithm animations on the Internet, from http://csweb.winona.edu/zhang/others.htm.

Acknowledgements

This project was supported in part by the Minnesota State Colleges and Universities Center for Teaching and Learning and the Bush Foundation.