

2005-2007 LITRE Grant Proposal

1. Project Title:

Integrating Algorithm Visualization into CSC 505

2. Project Coordinator:

First Name: Steffen

Last Name: Heber

Campus Address: 1519 Partners II (Centennial Campus)

Email Address: sheber@ncsu.edu

Campus Phone: (919) 513-2726

Unity ID: sheber

3. Other Participants:

NA

4. College or Unit:

College of Engineering

5. Department:

Computer Science

6. Project Description:

We propose to create a library of interactive visualizations for use in teaching of CSC 505: Design and Analysis of Algorithms. Currently, this class teaches students the use of data structures, fundamental algorithms, algorithm design, and algorithm analysis techniques mostly using conventional educational tools such as black board and overhead projector. Our goal is NOT to replace these tools, but to complement them with dynamic and interactive visual representations of abstract algorithm design and analysis concepts (e.g. growth of functions, recurrences), and animations illustrating how algorithms and data structures work. We believe that providing students with multiple views and teaching techniques will facilitate a better understanding of abstract concepts like the analysis of algorithms.

The suggested library will be online accessible for students and instructors. The library will be used as a complementary illustration tool to explore how algorithms work, and how they are analyzed. The idea is similar in spirit to Gloor's algorithm animation CD-ROM (Gloor et al. 1993) which accompanied the first edition of Corman, Leiserson and Reivest's algorithm textbook. We will start from this library and tailor the visualizations towards the curriculum of CSC 505, focusing on algorithm design and analysis concepts. Instead of HyperCard (Goodmann 1987) - which was discontinued in March 2004 - our library will use the highly portable Flash (Macromedia <http://www.macromedia.com/>).

To promote active engagement of students, we will complement the library by accompanying homework assignments which will make essential use of the library. For example students will be asked to explore pathological input values, or to reverse engineer an algorithm implementation.

Our library will cover the following subject areas: running time analysis (worst, best, and average case), asymptotic growth classes of functions, recurrences, data structures (stacks, queues, linked lists, priority queues), sorting algorithms (insertion sort, heap sort, merge sort, quick sort, counting sort, bucket sort), graph algorithms (DFS, BFS, topological sorting, shortest path algorithms), minimum spanning tree algorithms, dynamic programming, and greedy algorithms (fractional knapsack problem, Huffman codes). Starting from Gloor's library, we will develop visualization templates for representing the problem graphically and the step-by-step solution as an event driven sequence of snapshots. Separate windows will show pseudocode of the algorithm under investigation, a visual representation of the current state of the algorithm, a set of controls allowing the user to interact with the algorithm. At crucial events (function calls and returns, variable assignments, etc.) the user may be presented with interactive questions and additional explanations. This provides the user with the possibility to gauge and improve their understanding.

We suggest building the graphical application in Flash Professional 8 (Macromedia <http://www.macromedia.com/>) using Action Script. This guarantees problem free use virtually independent of operating system and client hardware specifications -- Flash Player is installed on 98% of desktops globally. ActionScript, the programming language for Flash Player, is a full-featured scripting language similar to JavaScript. It includes a visual user interface for editing scripts that includes automatic syntax completion as well as descriptions for the parameters of any given action. Flash also provides users with plentiful tools for dynamic user interactivity and graphical effects, and it allows us to mix audio and video effects.

References

* Gloor, Dynes, and Lee. Animated Algorithms. MIT Press, 1993, CD-ROM.

* Gloor. AACE -- Algorithm Animation for Computer Science Education. In Proc.

1992 IEEE Workshop on Visual Languages, pages 25--31, October 1992.

* Goodman. The Complete HyperCard Handbook. Bantam Books, 1987, ISBN 0966551427.

7. Project Objectives:

This project has three goals:

1. To develop and make online available a library of interactive visualizations which will be used by students and the instructor to demonstrate and explore computer algorithms and data structures taught in CSC 505.
2. To develop an accompanying set of homework assignments using the library.
3. To determine the effectiveness of this library as an educational tool.

The potential impact of this project on student learning as well as department, college, and university are detailed in 9 and 10.

8. Estimated number of students affected:

In the short term, the proposed project will impact only the student's who take CSC 505 class (enrollment: about 60 students per semester).

In the long term, if successful, the suggested algorithm visualizations could be used in the following classes: CSC 314 Data structures, CSC 316 Data Structures for Computer Scientists, CSC 541 Advanced Data Structures (total enrollment: about 200 students per semester).

In addition, the PI plans to develop similar visualizations tailored towards algorithms used in Bioinformatics. The BI/GS 501 Bioinformatics I, BI 502 Bioinformatics II, CSC 530 Computational Methods for Molecular Biology (total enrollment: about 75 students per semester).

9. Outcomes of the Project:

Algorithm visualization is widely considered as one of the most important tools in computer science education. Although it has been shown that the educational effect of simply observing visualizations is unexpectedly low, interactive visualizations which manage to engage students, and which are mixed with other teaching methods like black board or overhead projector, have proven extremely successful (Brusilovsky, and Spring, 2004). We believe that the proposed library will:

- * Enhance student learning of complex computer science concepts taught in CSC 505 by providing additional visual representations (empirical inquiry).

Many students do not fully comprehend key concepts of design and analysis of algorithms. We believe that this is often due to the fact that these concepts are communicated on a textual basis, invoking sophisticated mathematical arguments. We strongly believe that the learning outcomes of students could be improved by adding a visual representation. Most students are able to grasp information in graphical form better than in textual form (Felder and Silverman, 1988).

- * Help students to study and learn independently about fundamental algorithms (research from sources).

Due to time constraints, students learn about many important data structures and algorithms only in reading and homework assignments. Here, in addition to providing a visual representation, visualization has an important motivational character. It can make studying algorithms more fun by making algorithms less intimidating and more accessible. This will improve students' attitude against studying algorithms and result in better outcomes.

- * Help instructors to illustrate algorithms in class.

References

* Brusilovsky and Spring, (2004). Adaptive, Engaging, and Explanatory Visualization in a C Programming Course. In: L. Cantoni and C. McLoughlin (eds.) Proceedings of ED-MEDIA'2004 - World Conference on Educational Multimedia, Hypermedia and Telecommunications, Lugano, Switzerland, June 21-26, 2004, AACE, pp. 1264-1271.

* Felder and Silverman, (1988). Learning styles and teaching styles in engineering education. Engineering Education, 78 (7), 674-681.

10. Project impact on NCSU:

We believe that

1. Student learning and understanding of complex data structures and algorithms will be improved.
 2. Students will have a more positive attitude towards the classes in which visualizations are used.
 3. Design and Analysis of Algorithms is one of the most commonly taught computer science classes, and therefore the suggested library has the potential to become a resource for other instructors in- and outside NC State University.
 4. The use of visualization in computer science teaching will serve as a positive example of LITRE and NCSU as being on the cutting edge in innovation in teaching.
 5. This project can be used as a model and starting point to attract external grant funding for similar projects, for example the visualization of algorithms used in Bioinformatics.
-

11. Project Assessment Plan:

The PI plans to test the effectiveness of the proposed visualization library in three different ways:

1. Assessment of students' learning outcomes given additional visual representations.
2. Assessment of students' attitude towards the class
3. Assessment of effort in creating visualizations as well as 'teacher outcomes'.

(1) There have been many studies about the effectiveness of algorithm visualizations in class, see (Hundhausen et al. 2002) for an overview. Often students failed to understand what was happening inside a program or an algorithm in the presence of a well-developed non-interactive visualization. It has been shown, that in order to be successful, it is essential to generate interactive visualizations which are integrated with other teaching aids (Hundhausen et al. 2002). To test the effectiveness of our library in class, we will perform post-teaching tests monitoring students understanding with or without visualization. However, our main focus is to test if the presence of an interactive algorithm animation helps students to learn independently about fundamental algorithms. We will use a test procedure adapted from (Lawrence et al., 1994). The students will be divided into two equal groups, and both groups will learn about Prim's Minimum Spanning Tree Algorithm in a lecture. Subsequently, both groups will use a textual description of Kruskal's Minimum

Spanning Tree Algorithm compiled from Corman, Leiserson and Rivest's Introduction to Algorithms textbook (Corman et al., 2001). Group 1 will have access to an interactive animation, while group 2 will get a static visualization like Figure 23.4, p 568 of Introduction to Algorithms (Cormen et al., 2001). Both groups will complete a multiple-choice test and answer free-response questions designed to articulate concepts relating to understanding the algorithm.

(2) The PI will develop an end-of-course questionnaire eliciting student feedback and gauging their satisfaction.

(3) Although there is a general belief that algorithm visualizations are helpful educational tools, teachers have been hesitant to use them because they are afraid that they might be too complicated to handle, and that they use up too much class time (Hundhausen et al. 2002). In order to address these concerns, the PI will keep journals with observations concerning:

- * Effort generating visualizations and teaching material (lecture and homework assignments) that utilize the library.
- * Impact on classroom time and student/teacher interactions when visualizations are used.
- * Impact on teaching strategies.

References

* Hundhausen, Christopher, Douglass, Sarah, and Stasko. A Meta-Study of Algorithm Visualization Effectiveness. Journal of Visual Languages and Computing, Vol. 13, No. 3, June 2002, pp. 259-290.

* Lawrence, Badre, and Stasko. Empirically Evaluating the Use of Animations to Teach Algorithms. Proceedings of the 1994 IEEE Symposium on Visual Languages, St. Louis, MO.

* Cormen, Leiserson, Rivest, and Stein. Introduction to Algorithms (Second Edition) published by MIT Press and McGraw-Hill, 2001.

12. Staffing and Support:

The PI will coordinate and supervise software code development, a programmer will be hired to write the code. The PI will develop the accompanying homework and perform the project assessment.

13. Financial Support Requested:

EPA salary total:

SPA salary total:

Other salary: \$6000

Equipment: \$300

Cost associated with assessment:

Other financial support requested:

Total Funds requested:

Additional Explanation of how funds will be used:

We request \$6000 for 200 hours of programming at \$30/hour.

We request \$300 for Studio 8, Macromedia's suite of tools to design, develop and maintain websites, applications, and interactive

media, includes Dreamweaver, Flash Professional, Fireworks, Contribute, and FlashPaper.

14. Funding Breakdown:

Total funding requested for fiscal year 2005-2006: 6300

Total funding requested for fiscal year 2006-2007: 0

15. Staff Support and/or Technical Support Requested:

None requested.

16. Timetable for Implementation:

Our goal is to complete the visualization library by fall 2006 and test its efficacy by spring 2007.

17. Human Subjects Protection:

If your proposal project involves research using human subjects, you will need approval from the Institutional Review Board for the Protection of Human Subjects in Research (IRB) prior to final approval. IRB information is available at <http://www.ncsu.edu/sparcs/irb>



18. Proposal Release:

By submitting this proposal the applicant grants the LITRE Advisory Board permission to make this proposal available as an example for future grant applicants. All personal information will be removed if this proposal is used as an example.

