ABSTRACT

USING COMPUTER GRAPHICS AND ANIMATION TO VISUALIZE COMPLEX PROGRAMMING CONCEPTS ON THE WEB

* Issac Herskowitz
* Anna Raynes

This presentation describes the **C-book** and **C++ book** software development projects. C-book is a computer aided teaching tool (CAT) for instructors of the C programming language. C++ book is a CAT tool for instructors of the C++ programming. The C++ book project has been developed to run directly on the WEB.

The tools provides graphics and animation allowing the instructor to visualize selected complex lecture concepts. Related literature has shown that visualization is a highly effective pedagogical way of transmitting information, and both tools reflect in this area.
The rational of the C-book development project design is to respond to the following needs:

1) To present complex programming problems in order for them to be easily learned.
2) To provide materials with alternative representations as a response to different learning styles.
3) To present, in a pedagogically sound way, a programming language whose design parameters don't include teachability.

The presentation will contain a detailed description of the software, a discussion on development tools used to create the software and thoughts why it is important to develop educational software on the WEB.

PROBLEMS WITH TEACHING PROGRAMMING

Three general problems related to teaching programming in C are addressed by C-book. First, complexity is characteristic of even relatively straightforward programming problems. Second, the differences in learning styles among those who would learn programming rule out a single approach as suitable for all. Third, the absence of teachability in the C language from conception through its many versions makes it particularly hard for a beginner to grasp.

* Touro College

Complexity characterizes even relatively straight- forward programming problems. According to Sanders & Gopal (1991) programming is difficult to learn. Many techniques and approaches have been undertaken to help new students of programming grasp complex programming concepts. Some programming languages such as Basic, Pascal, First Programming Language

(FPL) and LOGO are designed for instruction. Other approaches use graphical representations of programming languages such as Flowcharts, Nassi-Schneiderman diagrams and other graphic representations. C-book is an additional teaching tool for instructors of the C programming language. It was designed as a computer-aided teaching tool and not as a tutorial. C-book provides graphics and animations, allowing the instructor to visualize selected complex programming concepts. Thus, C-book helps instructors explain difficult programming constructs and render them visually concrete to their students.

The differences in learning styles among those who would learn programming rule out a single approach as suitable for all. C-book provides materials using alternative representations that respond to different learning styles. Gardner (1983) points out that students have different learning styles and require expanded modes of learning. Constructivist theorists (Spiro, 1991) provide a model of teaching that is facilitated by allowing the learner to view multiple perspectives, analogies, representations, explanations and dimensions. There are many studies that show that complex content material requires multiple representations, analogies, explanations and dimensions (Spiro et al, 1987). Research has shown that visualization is pedagogically a highly effective way of presenting information (Bergin et al, 1996). According to Arnheim (1969), the literature suggests that learning is facilitated when abstract concepts are visually broken down into smaller, concrete, digestible units. Students of the television generation are naturally visual learners (Healy, 1990). Mayer (1975) did an empirical study showing that low-ability students benefit more than higher-ability students by using "models" or visualization techniques. Many students do not understand programming concepts because programming languages are not representing conceptional features of natural languages. DuBoulay (1981) states that "visibility" is an important aspect for novices to understand programming processes by being able to view selected parts and processes of the program. C-book uses graphics and animation that help students visualize exactly what is taking place behind the scenes in the program. This helps students digest abstract concepts that can be visually broken down into simpler units. C-book lets students comprehend difficult concepts that would remain unclear in the absence of visual dissection.

The absence of teachability in the C language, from conception through its many versions makes it particularly hard for a beginner to grasp. This programming language design parameters did not include teachability but was designed to provide professional programmers a language that can easily be used to control a computer's hardware and peripherals. This makes the C programming language difficult to learn (Obrien, 1995). This language has many

obscurities of syntax that inhibit a learner from understanding basic logic constructs and problem-solving skills needed to learn a first programming language.

CONCLUSION

Although C and C++ is not pedagogically suited for an introductory course, it is an industry standard. It is also widely adapted as a standard for computer-science curricula at colleges and universities in the United States and in many other countries. Therefore, the C-book project was developed to address the need for improved modalities of instruction for a language not designed for teachability.

REFERENCES

Arnheim, R. (1969). *Visual thinking*. Berkeley: University of California Press.

Bergin, J. (1996). An overview of visualization: its use and design - A Report of the Working Group on Visualization. *ACM SIGCSE Bulletin*, 28, 192-200.

duBoulay, B., O'Shea, T., & Monk, J. (1981). The black box inside the glass box: Presenting computer concepts to novices. *International Journal of Man-Machine Studies*, 14,

Gardner, H. (1983). *Frames of Mind*. New York: BasicBooks.

Healy, J. (1990). *Endangered Minds*. NY: Simon & Schuster.

Mayer, R. E. (1975). Different problem-solving competencies established in learning computer programming with and without meaningful models. *Journal of Educational Psychology*, 67, 725-734.

O'Brien, F. (1995). Near Distance Software Education. *IEEE Proceedings Software Education Conference,* 20-23.

Sanders, I., & Gopal, H. (1991). AAPT: Algorithm Animator and Programming Toolbox. *ACM SIGCSE Bulletin, 23, 41-47.*

Spiro, R. J., Vispoel, W., Schmitz, J., Samarapungavan, A., & Boerger, A. (1987). Knowledge acquisition for application: Cognitive flexibility and transfer in complex content domains. In B. C. Britton & S. Glynn (Eds.), Executive Control Processes in Reading (pp. 177-199). Hillsdale, NJ: Lawerence Erlbaum Associates.

Spiro, R. J., Feltovich, P. J., Jacobson, M. J., & Coulson, R. L. (1991). Cognitive flexibility, constructivism, and hypertext. *Educational Technology*, 24-33.