Rich Content Plug-ins for the Teaching Machine

Theodore S. Norvell
Faculty of Engineering
Memorial University
St. John's NL Canada
(709) 737-8962
theo@mun.ca

Michael P. Bruce-Lockhart
Faculty of Engineering
Memorial University
St. John's NL Canada
(709) 737-8937
mpbl@mun.ca

ABSTRACT

In this paper, we describe a pedagogical tool for providing rich content in early programming courses.

Categories and Subject Descriptors

K.3.1 [**Computer Assisted Instruction**]: CS1, object-oriented programming, C++, Java.

General Terms

Design, Experimentation, Languages.

Keywords

introductory programming, rich content, plug-ins, Teaching Machine.

1. DESCRIPTION

The Teaching Machine (TM) [1] is a pedagogical tool for demonstrating how computer programs work. It runs standard C++ or Java code on a model machine which blends elements of computer, compiler and memory manager into a coherent abstraction of a language processor. It is used in conjunction with WebWriter++, a small JavaScript based authoring language, to create interactive notes for most of our programming courses [2].

In common with many others [3], we are finding that modern students, particularly in CS1 courses, accustomed to manipulating rich content with their own computers, are disappointed by traditional programming examples and assignments. Recent advances in the TM [4], notably the development of (a) a plug-in system and (b) internal scripting, have allowed us to build rich content plug-ins that can be activated from within code being run on the TM.

2. GOAL

Our goals are twofold:

1. to develop plug-ins to abstract rich environments of interest such as image or audio processing or small-world simulations.
2. to allow students to use those plug-ins in developing their own code.

3. Application

A rich image plug-in has been developed. The extended version of the TM that resulted has been integrated into interactive, on-line notes for our course in Introductory Programming.

The plug-in displays a pair of images which may be loaded or saved as standard jpeg files. Student written and example code may access and manipulate images via calls to the plug-in. We use it to demonstrate processing of existing images (for example by reading a colour photo into the first image then creating a grey-scale version on the second image), as well as simple image generation.

By plugging the TM itself into the standard Eclipse environment students use for their assignments, they are able to use the TM's rich plug-ins to develop their own code.

4. Results

Interactive notes on image processing for a course in Introductory Programming, featuring the image plug-in, will be demonstrated.

5. REFERENCES

[1] Bruce-Lockhart, M., Norvell T.S. and Cotronis Y., Programming and Algorithm Visualization in Engineering and Physics. Proceedings of 4th Progr. Vis. Work. (Florence, Italy, June 29-30, 2006). Electronic Notes in Theoretical Computer Science, 178:111-119, 2007.

[2] Bruce-Lockhart, M. and Norvell, T.S. 2007. Developing Mental Models of Computer Programming Interactively Via the Web. In Proceedings of the 37th Frontiers in Education Conference (Milwaukee, WI, Oct. 10-13, 2007).

[3] Matzko, S. and Davis, T. 2006. Teaching CS1 with Graphics and C. In Proceedings of Innovation and Technology in Computer Science Education (Bologna, Italy, June 26 - 28, 2006). ACM Press, New York, NY, 526-531. DOI= http://doi.acm.org/10.1145/1140124.1140170

[4] Bruce-Lockhart, M., Crescenzi, P., and Norvell, T. 2009. Adding test generation to the Teaching Machine. ACM Trans. Comp. Ed. (June 2009).

ITiCSE '09, July 3–8, 2009, Paris, France.
Copyright 2009 ACM 1-58113-000-0/00/0004…$5.00.