On-line Homework/Quiz/Exam applet:
Freely available Java software
for evaluating performance on line

OAKLEY E. GORDON and THOMAS E. MALLOY
University of Utah, Salt Lake City, Utah

The Homework/Quiz/Exam applet is a freely available Java program that can be used to evaluate student performance on line for any content authored by a teacher. It has database connectivity so that student scores are automatically recorded. It allows several different types of questions. Each question can be linked to images and detailed story problems. Three levels of feedback are provided to student responses. It allows teachers to randomize the sequence of questions and to randomize which of several options is the correct answer in multiple-choice questions. The creation and editing of questions involves menu selections, buttonpresses, and the typing of content; no programming knowledge is required. The code is open source in order to encourage modifications that will meet individual pedagogical needs.

The Homework/Quiz/Exam software described in this paper grew out of a pedagogical need to provide numerous homework assignments in a statistics course and a pragmatic need to limit the amount of time spent grading those assignments. The project began in 1987 with the creation of a computer program to present, grade, and provide feedback on homework assignments for students enrolled in an introductory statistics course at Southern Utah University. At that time, the program could only be run locally in a psychology department computer lab, and the creation of new assignments required knowledge of computer programming. Over the years, the project has evolved in three significant ways: (1) The program has been rewritten in the Java programming language, allowing students access to the homework assignments over the Internet; (2) additional question formats have been added in order to expand the capabilities of the program to meet the needs of classes other than statistics; and (3) an editing function has been included to allow instructors to create and edit homework easily, without knowledge of computer programming. The software is currently being offered free for use to the public, and the code has been designated as *open source* so that other educators with access to computer programming resources may enhance its abilities to fit their needs.

The philosophy behind the development of the Homework/Quiz/Exam software is that it is better for software to meet the pedagogical needs of the instructor than it is for the instructor to have to shape his or her pedagogy to fit the limitations of the tool. To support this, the software has been designed to provide a variety of functions and to be open to a process of continual evolution. A related philosophical stand is that it is desirable that education (specifically, on-line education) should reflect a variety of instructor's viewpoints, rather than just the viewpoints of a relatively small number of individuals and companies that have the resources and knowledge to create on-line tools. To support this, we have chosen to make the tool both freely available and easy to use. For a more detailed discussion of this approach and its relevance to current trends in Internet software development, see Malloy, Jensen, Regan, and Reddick (2002).

This approach is congruent with the work of White and Hammer (2000) and their Quiz-o-Matic software. White and Hammer point out both the pedagogical advantages of providing self-scoring quizzes over the Internet and the challenge faced by an instructor who must expend the time and effort needed to learn and implement the technology that would make it possible. They offer a simple-to-use tool that creates HTML and JavaScript code that can display multiple-choice questions on the Internet. As compared with Quiz-o-Matic, the Homework/Quiz/Exam software greatly increases the pedagogical options available to an instructor, while maintaining an acceptable level of ease of use in the creation and editing of a set of questions. It also adds the feature of recording the students' grades onto a database. To make possible these additional features, we chose to develop the software with the Java programming language.

The selection of Java as a base for the project has important implications. White and Hammer's Quiz-o-Matic makes use of JavaScipt, but it should be noted that JavaScript is only marginally related to Java (Van der Linden, 1999). JavaScript enables browsers to perform a limited number of actions: to open and close windows, to show forms on the screen, and to gather responses. These functions are sufficient for the creation of multiple-choice

---

questions. Java, however, is a complete programming language. It opens up a vast territory of pedagogically sophisticated interactions between the student and the program, limited essentially only by the imagination of the designer and the time and resources it takes to implement new designs.

The exciting possibilities of using Java to develop educational tools has been examined by Briggs and Sheu (1998). They indicated that Java made it easy to create a graphic user interface (GUI), an aspect of design that is crucial to its ease of use. The capabilities of the Java GUI have continued to evolve, and unfortunately, this increase in capabilities has come at the expense of an increase in complexity of the language. Java has been developed by Sun Microsystems, and their current manual describing how to implement the GUI (Geary, 1999) is over 1,600 pages in length and assumes that you already have a familiarity with the Java programming language (for a good 775-page introduction to Java, see Van der Linden, 1999). Obviously, if the advantages of Java were to be realized by academia, an effort had to be made to develop a program that would allow instructors to take advantage of Java features without their having to learn how to program in that language. The Homework/Quiz/ Exam applet (an *applet* is a Java program that runs through a browser) makes the complexity of the Java language invisible to the instructors, who simply select what they want from a menu of choices and then fill in the text in the appropriate areas for a question. The choices available to the instructors are described below.

Common Elements

Five different question types are currently available for selection within a homework assignment, they differ in terms of the type of response available for the student. All of the questions, however, have certain elements in common.

The question field. The text field within which the question is asked is essentially of unlimited length. It becomes a scrolling field if the text of the question is too large to fit in the area provided on the screen.

The view image button. This feature allows an instructor to associate an image with a question. Pressing the "View Image" button causes the image to be displayed on a separate window on the screen. If there is no image, the button is disabled (i.e., *ghosted*).

The story problem button. The amount of room on the screen available for presenting a question can become cramped if the screen includes other items, such as a place to type in an answer, and a place to receive feedback on its correctness. Although the question field will become a scroll field if needed, it is still undesirable to have to scroll through a very long question that is being displayed in a relatively small scroll field. To solve this problem, the applet provides a way to display a "story problem" whereby the question essentially takes up the entire screen. Any number of questions may be linked to any single story problem.

The evaluation button. After a student has entered an answer (e.g., typed in an answer on a fill-in-the-blank question or indicated the correct answers on a multiple-choice question), he or she presses the "Evaluation" button. This action causes the feedback to be displayed (see below), prevents the student from changing his or her answer, and enables a button that will move the student on to the next question.

Feedback. Students receive immediate feedback after each response, which allows them to correct errors in their thinking at the point at which they make them. The feedback consists of (1) an indication (literally thumbs up or thumbs down) of the correctness of their response, (2) the correct answer to the question, and (3) optional comments by the instructor. The comment area can be used to describe why one answer is correct and the others are incorrect or to make some point that pertains to the question. The comment area can be a useful tool for including teaching elements in the generally evaluative structure of the homework.

Response Formats

The question types available on the Homework/Quiz/ Exam applet differ in the type of response available to the student. The options are as follows.

Multiple-choice format. This response format is the traditional multiple-choice format in which only one option may be selected as the answer to the question. Any number of options (from one up) can be provided, limited only by what will fit in the available area.

Multiple-select format. In this format, the students are instructed to select all of the options that would be correct. There is no partial credit awarded; all of the correct answers must be checked, and all of the incorrect answers must not be checked, in order for the students to receive credit for answering correctly.

Multiple-choice graphic format. This is similar to the multiple-choice format, but here the student selects the correct answer from a number of images displayed on the screen. For example, in a statistics class, this question type is useful for displaying several curves and asking the student to select the curve with the correct rejection region for the story problem.

Fill-in-the-text format. This format calls upon the student to type in one or more words as a response to the question. There are major limitations to the applet's ability to evaluate such a response, for the user's response has to exactly match the correct answer supplied by the instructor when writing the question. There are many ways for a student to give an answer that the instructor would call "close enough" that the applet will evaluate as being incorrect. As a step toward alleviating some common mistakes, the applet deletes all blank spaces from both the student's answer and the correct answer before evaluating and ignores uppercase versus lowercase distinctions.

Fill-in-the-number format. This format is used to evaluate a student's ability to complete some calculation. One issue for this type of question involves differences be-

tween how the instructor and the student round off their answers; for example, the instructor needs to be able to tell the computer whether an answer of 4.33 and an answer of 4.3333 are both acceptable. The applet accomplishes this by allowing the instructor to input a "correct within ±" value that the program will use when evaluating an answer.

Randomization Features

At the end of a homework session, students have the option of turning in their grades or repeating the homework again in an attempt to improve their scores. The pedagogical advantages of repeating a homework assignment are lost if they write down all of the correct answers the first time through and then simply input those answers on a subsequent pass through the assignment. The applet currently provides two moves to inhibit this strategy. The instructor can indicate that a multiple-choice, multiple-select, or multiple-graphic question should have its options randomly sorted each time it is displayed; this at least stops the student from using the strategy of writing down that "a" is the correct answer to some specific question. The instructor can also indicate that any specific series of questions within a homework assignment should be presented in random order, causing the questions themselves to be in a different order every time a student goes through the homework

Homework and Quiz Mode

The applet was originally written to serve purely as a means by which to present homework assignments, but subsequently it has been enhanced to provide an additional evaluative context. In *homework mode,* the applet gives the student feedback and evaluation after each question, but they must, at that point, move on to the next question. With the feedback, the students learn from their mistakes, but they are not allowed to trivialize the process by then immediately changing their answers; they must go through the homework again if they wish to improve their scores. In *quiz mode*, the students are not given feedback on their answers, so there is no reason to limit their ability to move forward and backward through the list of questions. This allows them to go back and reconsider earlier answers before submitting their answers for a grade. The distinction we draw in our courses between a quiz and an exam is that a quiz can be taken over the Internet in an unsupervised setting, whereas an exam is taken in a proctored environment where a picture ID is required for admission. This distinction ties into some of the pedagogical issues involved in maintaining the integrity of education delivered over the Internet. For an examination of these issues, see Gordon (2000).

Authoring

The process of implementing the options described above in a Java applet is highly complex and should be made invisible to the instructor using the tool. Through the use of a GUI, the authoring applet makes the creation and editing of homework and quizzes a relatively simple task. The instructor begins by selecting from a list of existing courses or presses a button to create a new course; the necessary directory structure and files are created automatically. The instructor then selects a homework item to edit from a list of the existing homework from that course or presses a button to create a new homework item. Once in a homework item, menus and buttons provide control over the type of question and its configuration. Questions appear on the screen as they would appear to the students, within a larger frame containing the controls the instructor needs to edit the questions. Questions, answers, and comments are typed directly into the relevant areas of the screen.

The Recording of Grades

The applet provides a means for storing and retrieving students' grades for the homework and quizzes. This ability comes at the expense of having to establish a flow of communication between the applet and a server, and this in turn requires the assistance of someone who has the knowledge and authorization to work with a server. All of the required technical information can be obtained from our staff, who will communicate directly with the appropriate staff at other institutions to help them implement the necessary processes. We are also offering to let instructors from other institutions use our server and work directly with our staff.

Discussion

The Homework/Quiz/Exam software allows the instructor to implement many of the suggestions made by Romero, Berger, Healy, and Aberson (2000) for applying cognitive learning theory in the design of on-line materials. The homework questions provide a context for elaborative processing, the feedback the students receive helps them to self-monitor their progress accurately, and their ability to succeed or fail at the assignments serves as an operational definition to the student of whether they actually understand the material or not. The use of deadlines for completing the homework serves to help the students pace their progress through the material.

The use of Java to create the Homework/Quiz/Exam software allows us to provide the resource over the Internet, while also allowing us to create any feature that is normally available within a full computer-programming language. This provides us with the tools to shape the product to meet pedagogical needs, rather than shaping our pedagogy to meet the limitations of the product. A long-term commitment to continue to develop the tool and the decision to designate the code as open source ensure an evolution of the tool over time. Currently being developed are two new question forms: an essay question that is graded by the instructor and various forms of story problems for which new data are randomly generated each time the homework is opened (for similar approaches, see Bradley, 1978; Malloy & Jensen, 2001). Undoubtedly, additional features will be envisioned and implemented over time. In terms of the everyday usability of technology, using an ar-

chitectural metaphor, this leads to habitable software, which is software that shapes itself to fit changing needs and remains comfortable to dwell within for an extended time (Gabriel, 1996). A working copy of the applet may be accessed at http://www.psych.utah.edu/learn/homework.

REFERENCES

BRADLEY, D. R. (1978). An interactive data-generating and answer-correcting system for problems in statistics. *Behavior Research Methods & Instrumentation*, **10**, 218-227.

BRIGGS, N. E., & SHEU, C.-F. (1998). Using Java in introductory statistics. *Behavior Research Methods, Instruments, & Computers*, **30**, 246-249.

GABRIEL, R. P. (1996). *Patterns of software: Tales from the software community*. New York: Oxford University Press.

GEARY, D. M. (1999). *Graphic Java: Mastering the JFC* (3rd ed.). Palo Alto, CA: Sun Microsystems Press.

GORDON, O. E. (2000, Fall). Pedagogical issues in Internet education. *AABSS Journal*, **1**, 66-74.

MALLOY, T. E., & JENSEN, G. C. (2001). Utah Virtual Lab: JAVA interactivity for teaching science and statistics on line. *Behavior Research Methods, Instruments, & Computers*, **33**, 282-286.

MALLOY, T. E., JENSEN, G. C., REGAN, A., & REDDICK, M. (2002). Open courseware and shared knowledge in higher education. *Behavior Research Methods, Instruments, & Computers*, **34**, 200-203.

ROMERO, V. L., BERGER, D. E., HEALY, M. R., & ABERSON, C. L. (2000). Using cognitive learning theory to design effective on-line statistics tutorials. *Behavior Research Methods, Instruments, & Computers*, **32**, 246-249.

VAN DER LINDEN, P. (1999). *Just Java 2* (4th ed.). Palo Alto, CA: Sun Microsystems Press.

WHITE, R. J., & HAMMER, C. A. (2000). Quiz-o-Matic: A free Web-based tool for construction of self-scoring on-line quizzes. *Behavior Research Methods, Instruments, & Computers*, **32**, 250-253.