

Java experiments for introductory cognitive psychology courses

ADAM K. STEVENSON, GREGORY FRANCIS, and HYUNGJUN KIM
Purdue University, West Lafayette, Indiana

Interactive on-line experiments provide a unique and useful method for communicating material to students that is otherwise cumbersome and often confusing. The Java programming language is particularly suited for Internet-based programming applications of this sort because it bypasses many technical issues, including resource availability, security, and cross-platform compatibility. In most cases, topics appropriate to this medium of presentation should (1) not be easily demonstrated by other means, (2) represent an important finding in the field, and (3) be robust with respect to variations in both participants and equipment. The present paper outlines the integration of interactive experiments into an introductory cognitive psychology classroom, describing several experiments currently available on the World-Wide Web (WWW). Evaluation of the technical aspects of the technology as well as expansion of the format to other courses is discussed.

For undergraduate science students, some of the most important classes involve getting hands-on experience with the methods and materials of their discipline. Thus, chemistry classes include laboratory sections where students explore the chemical properties of substances, introductory physics classes include laboratory sections where students experimentally validate the fundamental laws of motion, and biology classes include laboratory sections where students explore animal and human physiology. The laboratory sections of these classes are so important that it is difficult to imagine learning these disciplines without them.

In contrast, psychology courses rarely involve hands-on experience. The few exceptions tend to be for psychobiology courses and courses that emphasize research methods. Of course, some topics in psychology do not easily lend themselves to a laboratory section, but many do. Cognitive psychology is the scientific study of mental processes (perception, memory, language, problem solving, decision making, etc.), and as a field it has gathered a number of key studies that demonstrate significant aspects of cognition in a laboratory setting.

An accompanying laboratory experience in cognitive psychology allows students to better appreciate the relationship between, and need for careful control of, variables. Furthermore, given that the majority of experiments in this field are carried out on computers, a computer-based laboratory provides not only an efficient method of presenting laboratory exercises but also experience inter-

acting with the environment and equipment used in psychological research. Many learning institutions require students in general introductory psychology courses to participate in faculty research projects to obtain this same sort of experience. However, these studies tend to change frequently and may or may not be related to cognitive psychology research.

Ideally, an accompanying laboratory section should not intrude on an already full lecture schedule. A computer-based laboratory can provide the means of avoiding conflicts with the lecture portion of the course if time can be arranged outside of class. Although commercial psychology laboratory materials are available, their use requires substantial effort from both the students and the instructors. Students must purchase software and have access to appropriate computer resources, or instructors must provide both software and resources at times and places that allow all students to complete the exercises. The effort required to coordinate such arrangements has likely precluded the use of such software.

In contrast, laboratory exercises located on the Internet offer a solution that is less costly and easier to maintain. The computer resources required to connect to the Internet are standard on many operating systems, and few students are likely to be unfamiliar with the appropriate software. For the instructor, installation is a one-time event on a single computer, so the time required to implement and coordinate lab exercises over the Internet is minimal. Laboratory exercises located on the Internet allow students to complete assignments at their leisure, in public computer laboratories or at home via a modem. There is also the potential for more widespread use (e.g., at regional campuses, distance learning).

The World-Wide Web (WWW) provides a framework for delivering highly accessible content through the Internet. Any computer with Internet access and Web browser software can access material on the WWW. However, until

This project was supported by a grant from the Multimedia Instructional Development Center, Purdue University. The authors thank the undergraduate students in Introduction to Cognitive Psychology at Purdue University for their participation during the ongoing development of this project. Correspondence should be addressed to G. Francis, Department of Psychological Sciences, Purdue University, West Lafayette, IN 47907-1364 (gfrancis@psych.purdue.edu).

recently, there have been restrictions on the types of content material that could be utilized by different computer platforms (e.g., Windows, Macintosh, UNIX, etc.). Static content such as text and images is platform independent; that is, they are accessible by any computer that supports a graphical Web browser (e.g., Mosaic, Netscape, Internet Explorer). Conversely, interactive content such as video, sound, and applications tends to be platform dependent. They often require specific hardware and/or software in order to use them. Given that the content we wish to use for laboratory exercises is often interactive, this leaves us with a dilemma. How do we provide distributable, platform-independent, interactive laboratory exercises over the Internet so that students will find them accessible and easy to use? There are a number of potential solutions, which we will outline briefly.

One possibility is to include several copies of the lab exercise on a website, with one copy of each format required by each computer platform. Although this appears to be a simple solution, there are some drawbacks. First, students must have some knowledge of the computer platform they are using (but see Welch & Krantz, 1996, for a creative way around this using server programming). Second, creating multiple copies of the same material is an inefficient use of computer resources and the programmer's time.

A second possibility is to run the laboratory exercise on one central computer. Many implementations on the WWW utilize programming that is hosted and executed by the server (the computer that stores and provides the Web pages), after receiving the user's input over the Internet from the client (the computer where the Web page is viewed). However, in circumstances in which a large number of users are accessing the program, the resources of the server can become overwhelmed, leading to a halt (hang) or delay in program execution, or in more drastic cases, an error in the computer's operating system execution (crash). As a result, this approach is appropriate only for applications that will not overtax server resources. Furthermore, access to a website connected to the Internet is open to the public. Therefore, anyone in the world has the potential to run the Web-accessible program (typically known as a server-side include [SSI]), thereby leading to possible security issues (see Kieley, 1996; NCSA HTTPd Development Team, 1995).

A final alternative that avoids the resource use and potential security issues of SSIs is to download the interactive program and execute it on the client computer. In order to avoid problems with platform dependence, a programming language called Java (Sun Microsystems) may be used to write the programs. Java is particularly suited for Internet-based programming applications because unlike many computer languages, it is interpreted. A Java program consists of platform-independent instructions called *bytecodes*. A Java bytecode interpreter, known as a Java Virtual Machine (Java VM), can be installed on almost any computer platform. The Java VM then converts the bytecodes into machine code, which can be executed

by the computer. The advantage to writing programs in Java is that they can then be run on any machine with an interpreter. Many recent WWW browsers have a built-in Java VM. Thus a Java program on the WWW (called a Java applet), can be run via the Java VM in the Web browser and subsequently should run on any computer that can run the Web browser.

We decided to create interactive labs for cognitive psychology on the WWW by creating Java-based programs. Several other groups have similar goals (<http://kahuna.psych.uiuc.edu/ipl/>; <http://www.olemiss.edu/projects/PsychExps>). The remainder of this article describes the selection, design, and implementation of a series of short experiments written in Java and delivered over the WWW that illustrate important concepts and findings in cognitive psychology. These experiments were incorporated into a set of laboratory exercises for use with an introductory cognitive psychology course at Purdue University in the fall of 1997.

DEVELOPMENT OF THE PROJECT

Selecting Lab Topics

One needs to exercise care when considering what sorts of interactive programs to design for a cognitive psychology class. Some experiments and demonstrations are effectively presented during a lecture, and a complicated multimedia program is not needed when a color transparency would accomplish the same thing. On the other hand, many experiments require the subject to carefully attend to stimuli and to repeat a task many times. Students can learn by participating in such experiments, but are bored watching others participate. Such experiments would be the ideal venue for an interactive program in which the students can proceed on their own, outside of lecture.

Experiments appropriate for this format should not only be readily demonstrated by an interactive computer program, but should also represent seminal findings within the field. Furthermore, the demonstration should produce robust effects in a twofold manner. First, the effect should be easily obtainable by a single subject within a reasonable number of trials. Second, the effect should be robust not only with respect to the subject, but also with respect to the computer equipment on which it is executed. Although Java applets can be run on a wide variety of computers, there are minor differences in the execution depending on the hardware. For instance, speed of presentation will vary slightly depending on the processor, memory capabilities, and monitor specifications of the computer system. The preparation should be robust enough to yield the expected outcome, regardless of minor alterations resulting from hardware differences. To summarize, an appropriate topic for this medium of presentation should (1) not be easily demonstrated by other means, (2) represent an important finding in the field, and (3) be robust with respect to variations in both subjects and equipment. These criteria are pro-

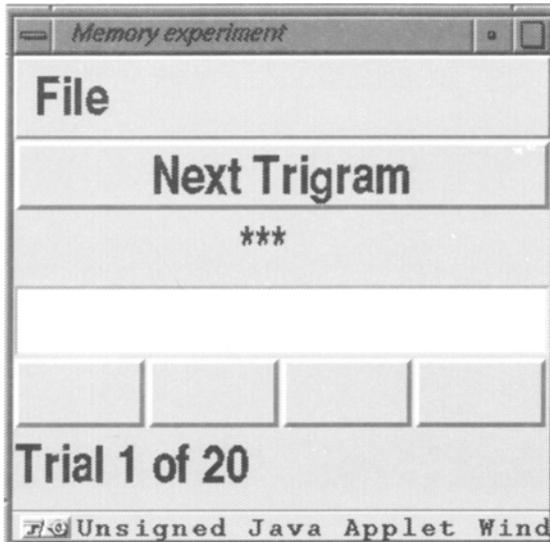


Figure 1. The interface of the Brown–Peterson applet.

posed as a guideline. Instructors may have various other objectives in mind for their classrooms (e.g., ensuring that all students attend to or participate in the demonstration). We have completed implementation of several demonstrations that generally satisfy these criteria and have several others in development that will be described below.

Brown–Peterson Task

The first demonstration replicates the experiments performed by Brown (1958) and Peterson and Peterson (1959) that contributed to a new interpretation of human memory. In the memory task, the subject views a trigram of consonants (e.g., GKT) and then performs a number of algebraic computations (e.g., counting backward by 3s) for less than 20 sec. The data show that recall of the trigram is less likely as the subject works on the algebraic computations for longer durations, suggesting that a short-term memory (STM) system holds information for several seconds. Without an active effort by the subject, information in STM fades away (e.g., while the subject is busy counting backward). The Java applet interface contains four components: the Next Trigram button, the presentation line, the response line, and the response buttons (Figure 1). An experimental trial begins when the subject presses the Next Trigram button. A trigram is then briefly displayed on the presentation line (just below the trigram button). The trigram disappears; multiplication problems are presented on the presentation line and possible answers are displayed on the faces of the response buttons. Additional multiplication problems are presented for 1, 11, or 22 sec (randomized across 20 trials). Following the series of multiplication problems, the subject is prompted to recall the trigram and enter it on the response line. Once 20 trials are completed, the individ-

ual’s performance is analyzed and a graph of the data is presented in a separate window (Figure 2).

Serial Position Effects on Immediate Memory

A second demonstration experiment explores effects of list position on free recall. In many instances, people are presented with a list of items and must remember each of the items (e.g., grocery lists). If the order of the items is not required for accurate recall, the task is said to be unordered or “free.” A general finding of free recall tasks is that recall of an item is strongly influenced by the item’s position in a list. Surprisingly, this property holds for many types of items and for a wide variety of durations (seconds to years). The effect of serial position has played a major role in the development of memory theories (see, e.g., Baddeley, 1992; Nairne, 1988).

The serial position applet uses an interface that is similar to that of the Brown–Peterson experiment just described. It utilizes a Next Series button to begin trials, a presentation line to present a series of letters, and a response line for serial recall of the letters presented. After the individual’s data are analyzed, a graph of their results is presented in a separate window.

Visual Search

A third demonstration investigates aspects of attention in a visual search task (Treisman & Gelade, 1980; Treisman, Sykes, & Gelade, 1977). It is a classic experiment that makes strikingly clear the time needed to bring attention to bear on different regions of visual space. The subject searches a visual image for a particular item and responds as quickly as possible after determining whether or not the item is present.

Searches are divided into two types—those that require selective use of attention and those that do not. In the latter, the target item seems to “pop out” of the display and the subject can respond quickly. Notably, this pop-out effect allows the subject to respond equally fast as the number of other (distractor) items increases. In the other type of display, it seems that the subject is forced to study many items individually until the target item is found. In these cases, the target item does not pop out, and search time increases with the number of distractor items.

The visual search experiment on the Web is divided into three portions—a practice session, a feature-based search task, and a conjunctive search task. There are separate buttons at the bottom of this Web page to begin each portion of the experiment. The subject begins a trial by pressing the space bar on the computer keyboard. A fixation cross is briefly presented in the center of the window, followed by a visual array of letters (Os and Ns) in various colors (Figure 3). The task is to determine whether the target (a red letter O) is present as quickly and accurately as possible. If the target is present, the subject is instructed to press the “z” key, and otherwise to press the “/” key. Feedback on accuracy and reaction time is pro-

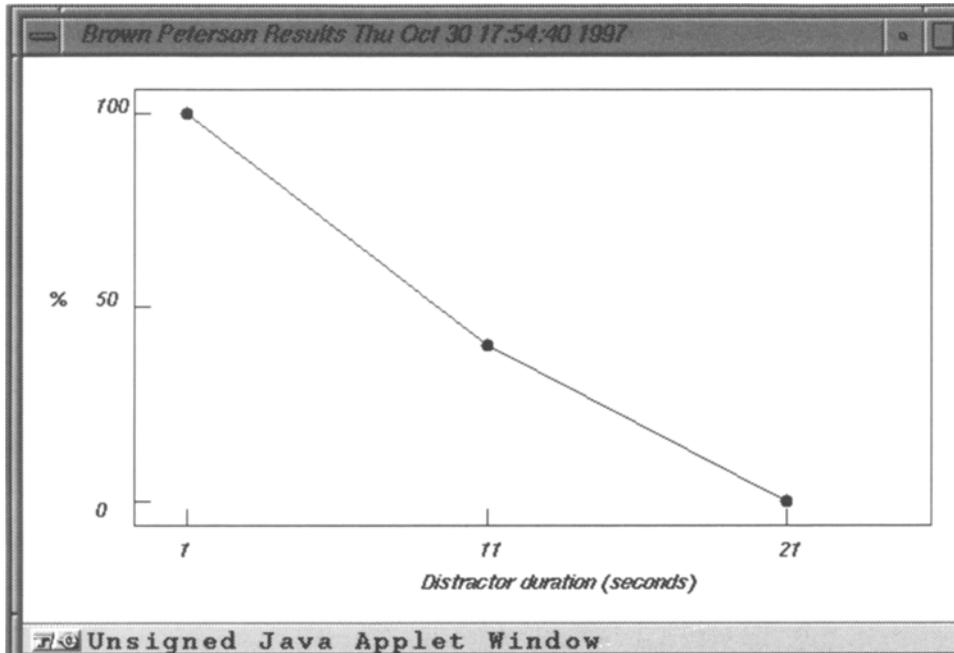


Figure 2. Results displayed for the Brown–Peterson experiment. The Java class for plotting results is identical for all the experiments. The reusability of programming code is illustrated.

vided on the computer screen following each response. The practice session is to familiarize the student with the task, and the practice block may be repeated as often as desired. The feature session is an example of a pop-out situation in which the target differs from the distractors along one or both feature dimensions. As a result, the target is easy to find (e.g., a red O in a field of blue Ns). In the conjunction block, the target differs only along one dimension; thus both dimensions must be searched and attention is required. These searches tend to be more difficult (e.g., a red O in a field of blue Os and Ns as well as red Ns). Following the completion of all three portions of the experiment, the subject's results for performance on both the feature and the conjunctive trials are presented in a separate window.

Other Demonstrations

Recently we have completed several other demonstrations, including the following:

Attentional blink. Subjects see a rapidly presented sequence of letters and respond whenever they see a target (J or K). Typically, the second target letter is not detected if it quickly follows the first. The results suggest that processing the first letter occupies the available resources so that the second letter is ignored.

Sternberg memory task. Subjects see a small (1–4) set of numbers and then are asked to determine, as quickly as possible, whether a target number was in the set. Reaction time is measured in relation to set size. This lab demonstrates characteristics of searching STM.

Mental rotation. Subjects decide as quickly as possible whether or not two shapes are rotated versions of each

other. Reaction time increases with the amount of rotation, thereby suggesting important properties of mental imagery.

Receptive fields. The subject's task is to find a receptive field, partly identify its excitatory and inhibitory regions, and categorize the neuron on the basis of its receptive field properties.

In addition to these demonstrations, we have several other laboratory exercises in various stages of development. A demonstration of apparent motion is in the final stages of programming. This experiment in visual perception (Wertheimer, 1912) involves the successive presentation of two separate light sources in a dark room (in this case, filled white circles on a black background). When the time between the offset of the first light and the onset of the second is short enough, a single light is perceived moving from the location of the first light to that of the second. Other planned laboratory exercises include a series of choices demonstrating properties of decision making and neural network simulations.

Programming

Java is an object-oriented language. Code is written in "classes," which are somewhat like generic templates (e.g., there is a class for a button). Objects are instances of classes (e.g., an OK button or a Cancel button). In contrast to procedural languages like C, FORTRAN, and BASIC, the operation of the program depends on the behavior of objects and their actions on one another, not the top-to-bottom execution of the code. For instance, in a procedural language, if we wanted to retrieve responses from the user, we could construct a loop conditional on the

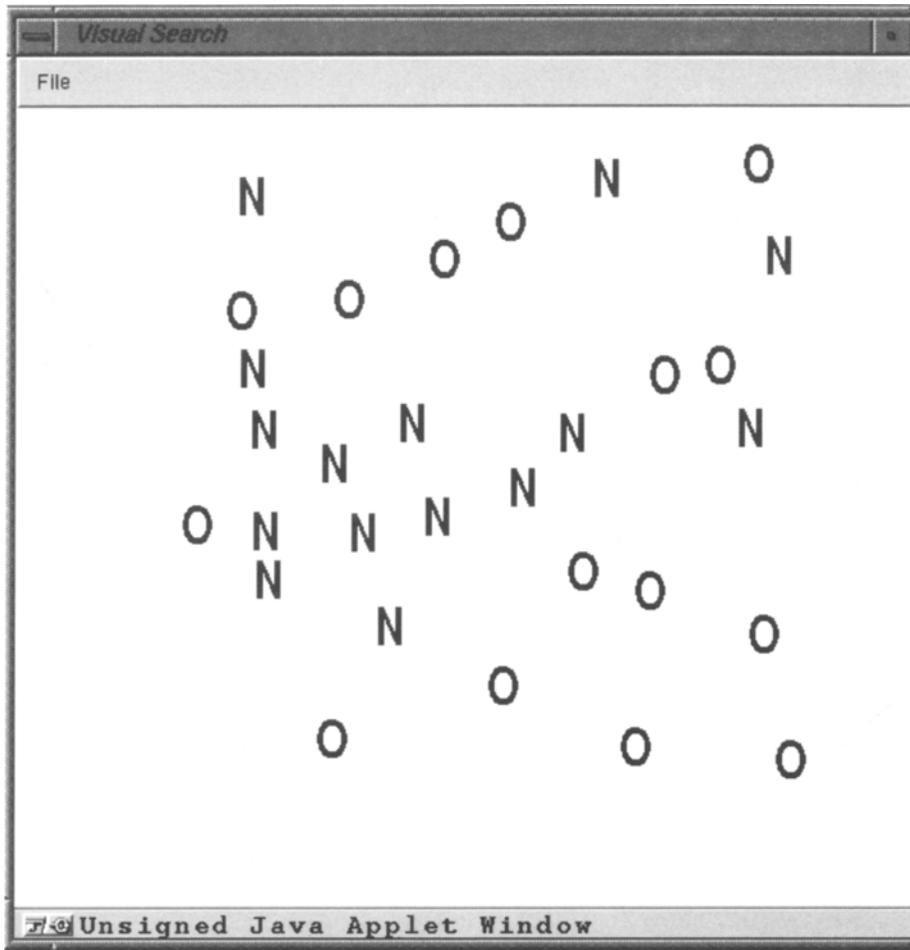


Figure 3. The interface of the visual search applet, conjunctive search condition. Note—The background color has been changed from that in the experiment to provide contrast for printing.

execution of the desired response. In Java, an event handler class retrieves any response and coordinates it with the appropriate code.

One of the important advantages of Java is the ability to reuse computer code. This dramatically speeds up development time, especially for graphic user interfaces. Java comes with a large number of generic classes that can be used or easily modified to fit the needs of the programmer (e.g., text fields, buttons, image handlers for graphics, etc.). In addition, programmers can develop their own classes that may vary in reusability. For example, the class that is used to give feedback to students who perform one of the experiments outlined here is identical for every experiment. It constructs a simple graph of the student's performance (Figure 2). Conversely, the class that comprises the "Brown-Peterson task" mentioned earlier is very specific to the task and not easily reused for different tasks. With proper planning, source code can be reused. For example, the mental rotation and Sternberg task experiments both measure reaction times to visual stimuli. Each of the programs for these labs extends a

more general reaction time lab, which includes code to measure reaction times, store data, open windows, present stimuli, and perform other various bookkeeping tasks. The two experiments simply define different stimulus types and different axes for the data plot. This simplicity, coupled with the cross-platform compatibility of Java, offers a powerful set of tools for developing experiments.

Finally, the tools required to develop applications in Java can be easily obtained over the Internet at no cost. Sun Microsystems makes the Java Development Kit (1997) readily available from its website (<http://www.javasoft.com/products/>). In addition, numerous books, tutorials, and documentation can be found on the WWW and at your local bookstore or university library. More information on where to find these resources can be viewed at our Java Resources Web page (<http://www.psych.purdue.edu/~coglab/java.html>).

Website Design

Each demonstration experiment is presented on a separate Web page, along with background information and

instructions. The background information consists of historical information and an explanation of the importance of the finding to the field of cognitive psychology. This is followed by detailed instructions regarding the subject's performance on the task—similar to what would be found in an actual empirical study. A button or set of buttons is located near the bottom of the Web page whose function is to start the experiment. Initiating an experiment results in a separate pop-up window in which the experiment is executed. This window may be moved by clicking and holding the left mouse button and dragging the window. This allows the student to review the instructions on the Web page underneath if necessary. At completion, another pop-up window containing a graph of the student's results appears, when appropriate.

The website is stored on the Department of Psychological Sciences server at <http://www.psych.purdue.edu/~coglab/index.html>. The design is intended to focus on ease of navigation, with multiple means of accessing any aspect of the site. The browser window is split into two parts using the HTML frame tag. The left-most 25% (hereafter referred to as the *control panel*) contains an HTML document that has controls for navigating the site. The remaining 75% displays the experiment Web pages and supporting pages (see Figure 4).

The control panel contains a pull-down menu with HTML hyperlinks to the experiment pages. In addition, a table of hyperlinks is available that allows access to the supporting pages, which contain the following: general information about the project, on-line help, e-mail access to the project developers for questions and feedback, a list of hyperlinks to related information, information on Java programming resources, and a hyperlink to return to the home page of the site. Any hyperlink clicked on in the control panel leads to the display of the selected Web page in the right portion of the browser window. Alternatively, within the support and experiment pages there is a navigation bar with links to all the support pages, as well as a separate experiment index page that can be used instead of the pull-down menu on the control panel. The on-line help page provides answers to commonly asked questions about the software.

INTEGRATION INTO THE COURSE

The on-line laboratory experiments were incorporated into a section of Introduction to Cognitive Psychology in the fall of 1997 at Purdue University. The laboratory exercises were assigned as related topics were covered in lecture. Students were required to complete the laboratory experiment and answer questions about the lab exercises. Some research suggests that the inclusion of a follow-up assignment to computerized laboratory exercises may be useful in increasing the retention of the material (Welsh, 1993).

Evaluation

The last question in the lab assignment always asks for comments, criticisms, and concerns about the lab. Student

responses have been overwhelmingly favorable. Many students commented that a complicated experiment made sense after they completed the lab. Many students were also excited to discover that their data agreed with the expected results. We are currently considering other methods for quantifying student reactions.

For faculty, the laboratory programs provide a nice complement to lecture. The instructor can discuss the theoretical significance of a particular experiment and know that everyone understands the data, because each student has actually generated data. Because the labs are carried out on students' schedules outside of lecture, the instructor does not lose any lecture time to a laboratory section. There are several technical issues pertaining to the integration of this media into the classroom, which are discussed below.

One important concern is the differences in students' ability to use computers. The interface for both the Web pages and the experiments was designed to avoid intimidation. In addition, on-line help was provided and e-mail access to the developers was encouraged. Several students took advantage of this and were able to resolve software problems and complete the assignments. Finally, an evening session was organized to assist any students who had a great deal of trouble or who had little experience with computers and browser software. Most students had no trouble completing the experiments successfully. Those students who did have problems were usually able to solve them after discussion with the authors via e-mail.

A second concern pertains to the software itself. Although the Java VMs in several of the more popular browsers claim to be compatible with the Java language, we have experienced different results for different platforms and browser versions. In some instances, the experiments will not execute or do so incorrectly (see <http://www.psych.purdue.edu/~coglab/help.html> for details on known problems). In the majority of these cases, simply obtaining a newer version of the Web browser software has solved the problem. In other cases, a browser could successfully interpret the demonstrations on one operating system, but not another. The only explanation is incompatibility of the VM shipped with the browser. As known problems and their solutions arise, they are included on the on-line help page.

Finally, because Java is a new language, programming development faces some risks. During the design of our experiments, the Java language has undergone substantial changes. As a result, a number of the language calls in the present experiments will no longer be supported in future versions of the Java compiler and run-time systems. As a result, it will be necessary to rewrite those portions of the experiments in the near future. Unfortunately, these rewrites cannot be done soon because the current Java VMs in popular Web browsers do not incorporate the replacement language calls and will not run Java programs that include those calls. Thus, we must wait for Web browsers to incorporate the latest features of Java before writing code that includes the newest capabilities of the language. This rather annoying characteristic should dis-

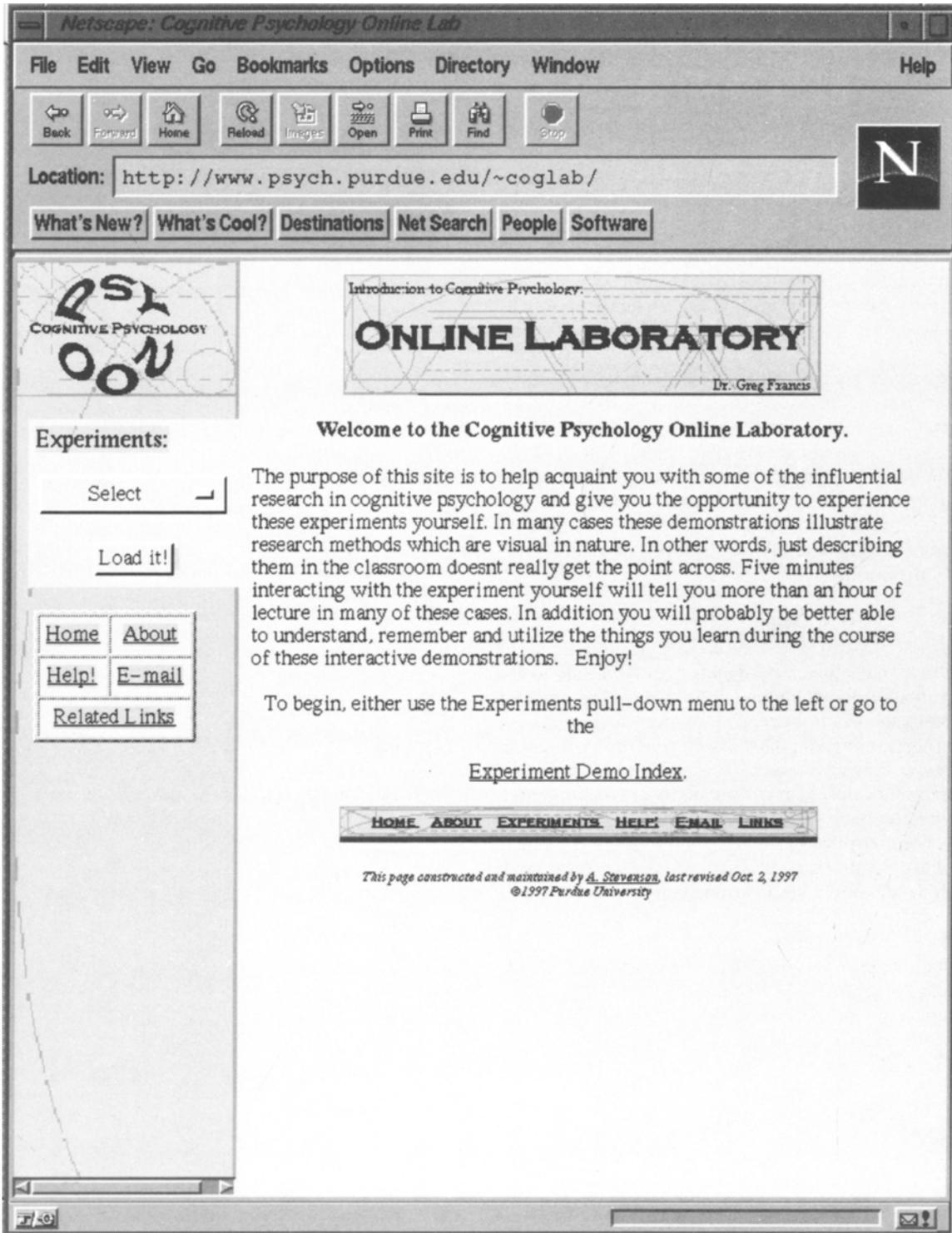


Figure 4. The interface of the on-line laboratory website. The left portion of the screen shows the control panel; the right portion displays experiment Web pages and supporting pages.

appear in a year or so as Web browsers and the Java language synchronize.

Future Plans

In the long term, we wish to develop a number of additional experiments to be included in this project. Full incorporation of the on-line laboratory into the course will involve a series of 5 to 10 experiments covering a variety of topics. Labs would be assigned every other week with follow-up assignments to assess the student's involvement with and learning of the subject matter. Furthermore, a more quantitative analysis of the effectiveness of the laboratory exercises will be developed.

The on-line laboratory approach could also be extended to higher level courses. Perception as well as memory courses could use this approach to demonstrate important research paradigms and results that might be too specific or involved for introductory-level courses (see, e.g., <http://www.psych.purdue.edu/~coglab/VisLab/initial.html>). Statistics and research methods courses could use this approach to demonstrate experimental design and analysis. This approach could be adapted to any instance in which the subject matter lends itself to computer presentation and is robust.

CONCLUSIONS

Interactive on-line experiments provide a unique and useful method for communicating material to students that is otherwise cumbersome and often confusing. The Java programming language is well adapted to programs delivered over the Internet. It provides alternatives for taxed server resources and client–software incompatibility. This approach has great promise as a pedagogical tool in both introductory and higher level courses. In an area of exponential growth and development, Java stands out. Its development has been and continues to be rapid.

As the Java language continues to develop and software problems with Java interpreters are resolved, the incorporation of this technology into the psychology classroom will provide the instructor with a useful and powerful tool.

REFERENCES

- BADDELEY, A. (1992). Is working memory working? The Fifteenth Bartlett Lecture. *Quarterly Journal of Experimental Psychology: Human Experimental Psychology*, **44A**, 1-31.
- BROWN, J. (1958). Some tests of the decay theory of immediate memory. *Quarterly Journal of Experimental Psychology*, **10**, 12-21.
- THE JAVA DEVELOPMENT KIT 1.1.3 [Computer software]. (1997). Palo Alto, CA: Sun Microsystems. Available URL: <http://www.javasoft.com/products/>
- KIELEY, J. (1996). CGI scripts: Gateways to World-Wide Web power. *Behavior Research Methods, Instruments, & Computers*, **28**, 165-169.
- NAIRNE, J. S. (1988). A framework for interpreting recency effects in immediate serial recall. *Memory & Cognition*, **16**, 343-352.
- NCSA HTTPD DEVELOPMENT TEAM (1995). *Security concerns on the Web* [On line]. Available URL: <http://hoo.hoo.ncsa.uiuc.edu/security>
- PETERSON, L. R., & PETERSON, M. J. (1959). Short-term retention of individual verbal items. *Journal of Experimental Psychology*, **58**, 193-198.
- TREISMAN, A., & GELADE, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, **12**, 97-136.
- TREISMAN, A., SYKES, M., & GELADE, G. (1977). Selective attention and stimulus integration. In S. Dornic (Ed.), *Attention and performance VI* (pp. 333-361). Hillsdale, NJ: Erlbaum.
- WELCH, N., & KRANTZ, J. H. (1996). The World-Wide Web as a medium for psychoacoustical demonstrations and experiments: Experience and results. *Behavior Research Methods, Instruments, & Computers*, **28**, 192-196.
- WELSH, J. A. (1993). The effectiveness of computerized instruction at the college level: Five suggestions for successful implementation. *Behavior Research Methods, Instruments, & Computers*, **25**, 220-222.
- WERTHEIMER, M. (1912). Experimentelle Studien über das Sehen von Bewegung. *Zeitschrift für Psychologie*, **61**, 161-265.

(Manuscript received November 3, 1997;
revision accepted for publication June 12, 1998.)