A general interactive computer program for running signal detection experiments

GEORGE W. SEELEY, MARK BORGSTROM, and
JOHN MAZZEO

*Optical Sciences Center
University of Arizona, Tucson, Arizona 85721*

This paper describes a system that controls the presentation of stimuli and the taking of data for psychophysical signal detection experiments. It consists of two interactive computer programs written in FORTRAN for both the DEC VAX-11/780 and a 64-KB DEC 11/34. In general, the first program enables the experimenter to specify all the elements of his experiment, and these elements are used by the second program to run the experiment. The two main programs are DESIGNMOD and RUNEXP.

DESIGNMOD

As the name suggests, this module of the system allows the experimenter to specify the variables he wants to be manipulated in a particular experiment. It does this by asking the experimenter a series of questions. The answers are used by the RUNEXP program to control the actual running of the experiment. The first question requests the name of the experiment. This information is used to open a master file that is used by the next program and as part of the identification for all the files relating to that particular experiment. This is an important consideration, for it is useful not only for identifying the subject data files, but also for separating different experiments from each other, especially if they are being run simultaneously.

The next section of the program allows the experimenter to define the number and type of demographic and experimental questions he wants to ask of each subject. The demographic questions are entered first. The program then has the experimenter define: (1) the instructions to the subject, (2) the number of images to be displayed, (3) the form that the images will take (CRT display or hard copy), (4) the response required from the subject (e.g., signal present or not and, if present, the location), and (5) the number of points and the definition of the points in the certainty scale to be used by the subject. This information is written to the master file for later use.

Great care has been given to prompt the experimenter with as much information as necessary about the possible choices in all of the above-mentioned sections of the program. For example, when the experimenter is entering the instructions, he is given a display

that defines the number of characters and lines to which he must limit his instructions. This is done so that the limited on-line editing functions of most computer operating systems can be used by the experimenter without having to resort to the writing of special editing routines.

RUNEXP

This program consists of one master module and two main subroutines. The master module asks for the name of the experiment. Using this, it opens the data file for that experiment created with DESIGNMOD and reads in the information the experimenter supplied on how to run the study. It then calls, in sequence, subroutines DENTAK and TAKDAT.

DEMTAK. This subroutine deals with subject demographic information. It asks the subject for his identification number and then checks to see if a file using that name exists. If it does not, it opens a new file, asks the demographic questions defined by the experimenter, and writes the answers to the subject's file. It then generates and stores a random sequence for presenting the images during the study, using the last three digits of the subject's identification number as the seed. After completing this, the subroutine returns control to the master module.

If the subject's file already exists, this routine opens the file, reads in the demographic and random-number sequence data, sets a flag indicating that the observer has responded previously, and returns to the main module. This feature enables subjects to leave off at any time and to return to pick up where they left off previously. This can be a great advantage when the experiment is long, or when there are subject, experimenter, or equipment time constraints.

TAKDAT. This subroutine handles the running of the experiment. First, it checks to see if the subject has previous responses. If so, it reads in those responses from the subject's data file, finds out where the subject left off, and prepares to present the rest of the images from that point. If the subject has no previous responses, it starts from the beginning. If there are previous responses, the routine also checks to see if the number of responses read in matches the stored trial counter. If they do not match, it alerts the subject to call the experimenter. This feature is built in as an aid for the experimenter. It tells him that there is something wrong with the subject's data, and it displays the file information if requested. After studying the displayed information, the experimenter can edit the file using the system editing routine. With this check, he can catch any problems early and decrease the chances of losing data.

Once the data file has been corrected or, as in most cases, is determined to be correct, this routine displays the experimenter's instructions to the subject. It also

This work was supported by NIH Grant NCI P01 CA23417 and NIH Contract NO1-HV-12901.

tells the subject how to return to the instructions during the experiment (type H then <CR>), how to correct an error (B;<CR>), how to exit the program to return later (E;<CR>), and how to continue with the experiment (<CR>).

When the subject indicates that he is ready to continue with the experiment, the program checks to see whether or not the images are to be displayed on a CRT or as hard copy. If hard copy is to be used, the program assumes the experimenter is presenting the images to the subject, and it tells the experimenter which one to present at any one time based on the random order of presentation for that subject. If the images are to be displayed on the CRT, the program retrieves each image from memory and displays the images in the proper sequence to the observer. As each image is displayed, TAKDAT asks each of the experimental questions entered by the experimenter in the design module program and displays the appropriate certainty scale, or, in general, the range of choices the experimenter allows for each question.

Each time the subject responds to an experimental question, the program first checks to see whether he wants to view the instructions again, to correct his previous response, or to exit the program. If all of the responses are negative, it then checks to see if the response given is inappropriate for the range of choices allowed by the experimenter for that question. If this proves to be the case, then an error is indicated to the subject and the question is presented again with the appropriate scale. When a response is accepted, it is immediately written to the subject's data file.

The routine continues presenting the images and questions until the experiment is completed. If at any time the subject indicates that he wants to exit the program, the routine closes the file and tells the subject

to indicate to the experimenter when he can return for his next session. Whenever the subject completes the experiment, the program writes only the signal detection-related data to a master data file in a standard format. When all subjects have been run, this master file is accessed by data analysis programs and the results are printed out. Individual data files are also kept. These can be sorted on various combinations of demographic variables to search for other effects, once the main analysis is completed.

Depending upon the number of terminals available, the experimenter can run many different subjects at one time. Also, when using hard-copy stimuli, he is not restricted to one location to run the study but, instead, can run subjects almost anywhere. All that is needed is a hookup to a mainframe computer (telephone, modem, and terminal).

Transportability and Availability

The programs described in this paper can be obtained by sending to the authors an 8.5 x 11 in. self-addressed envelope stamped with sufficient return postage for 3 oz. The code and especially the logic should be easily transferable to other computer systems. For greatest ease of transfer, the computer system's FORTRAN should have BYTE or LOGICAL*1 declaration, "Q" FORMAT, and the capability of adding extensions to file names. There might also be problems with the OPEN statements and the RAN call, but these are the only limitations we know of, and they can be converted to the alternative codes of the user's system. Also, the CRT display routine is necessarily unique to the user's laboratory hardware. Thus, anyone requesting our system must write their own display subroutine.

(Accepted for publication October 15, 1982.)