A Simple Windows Program for Coding Fast-Changing Events in Observational Situations Divided Into Variable Intervals

In the field of behavioral science, observational situations applying a timer, pen, and paper have been one of the most common ways to collect data. Because of several shortcomings of this method, IBM-compatible software packages for coding observational data have been developed (see, e.g., Claro, Del Abril, Segovia, & Guillamon, 1990; Guthertz & Field, 1989; Hetrick, Isenhardt, Taylor, & Sandman, 1991; Noldus, 1991; Tarapacki & Kristal, 1990, 1993). These programs run under DOS with widely known hardware and standard keyboards and have, therefore, a large number of potential users. The present article describes a simple behavior coding program for Microsoft Windows. It allows a real-time data acquisition when fast-changing events are observed during periods divided into variable intervals. The coded variables include durations and onset times of the events anchored to the onset of the interval at the given time and the preceding sequence of the intervals. The onset of each interval is defined by the user, and the software is able to identify events lasting from one interval to another. Programming was performed with Microsoft Visual Basic Version 3.0 because of the Windows interface provided for written applications and its event-oriented approach, which is suitable for designing event coding programs.

User interface. Use of the software does not require any programming. It is menu driven like any other Windows application. The program displays the running time of the current interval and all values of the last event written to the output file. The display of the running time can be also disabled. The user display is illustrated in Figure 1.

Usage. When the program is started, it automatically prompts the user to specify the numbers of the subject and experimental session. Then it displays a note about the keys used during coding. After these preparations, the program is ready for events to occur. Note that only exclusive events can be coded. Each press on the space bar starts a new interval by restarting the timer and incrementing the number of the interval by one. The temporal distance from the onset of the preceding interval is also defined. Arrow (up, down, left, right) and numeric keys (1–5) are used to code onsets and offsets of each class of events. When any of the keys is pressed down, the onset time of the corresponding event is defined in relation to the onset time of the current interval. When the same key is released for defining the offset time, the length of the event is calculated. For events lasting (i.e., the key remains pressed) to the next or later interval (i.e., space bar press), the event is identified separately if it represents a continuation of the event started during the earlier interval. In this case, the continuing part of the event gets a code value of its own that is multiplied by 10 for each consecutive interval (see Figure 2). The user can pause the program during coding and, if needed, change

Figure 1. A screen view of the program during coding (note that trial number corresponds to the interval number).

the subject, session, interval, or running time values before continuation. In the case of an erroneously coded event, by pressing the “e” key, the user can mark the data of the event to be removed or modified in later editing of the data file. The program can also be set to beep after a desired time elapsed since the start of the interval when the display of the running time is enabled.

After each event, output data are appended to an ASCII output file located in the same directory as the program. This prevents a massive data loss in the case of a fatal error during coding. The data are not written until the event offset in order to prevent an inaccurate timing of the event duration caused by the disk access. The data of each event consist of six values corresponding to the subject number, session number, interval number, onset time of the event, and duration of the event (Figure 2). Thus, the user can easily import the data to statistical programs in a uniform manner independently of the number of the events or periods. The name of the output file is automatically specified as a combination of a three-digit subject number and a two-digit session number with a .dat extension. If the file of the corresponding name already exists, the new data will be appended to the end of the file.

The timing accuracy of the program is about 0.1 sec (tested on a 386SX computer at the processor speed of 10 MHz). Because of disk access for appending the data to the file after the offset of each event, the timing accuracy for event onsets occurring simultaneously with the file write is slightly poorer than 0.1 sec. The time reso-

	Subject	Session	Interval	Code	Onset	Duration
A	1	2	1	0	0	0
B	1	2	1	1	2.3	5.2
C	1	2	2	0	21.1	0
D	1	2	2	2	6.8	3.0
E	1	2	3	20	9.8	2.0

Figure 2. A sample output file. Column and line labels are not included in the file. Lines A–E refer examples of events downloaded during coding: (A) onset of Interval 1 (space bar press); (B) left arrow key pressed 2.3 sec after the interval onset and released 5.2 sec later; (C) an onset of Interval 2, 21.1 sec after onset of Interval 1 (space bar pressed again); (D) right arrow key pressed 6.8 sec after interval onset; (E) release not until 2.0 sec after the onset of Interval 3, which, in turn, occurs 9.8 sec after onset of Interval 2. (Note that code is multiplied by 10 in these situations.)

lution achieved could, however, be considered as adequate for the level of accuracy achieved by the observer during coding. If needed, the overall time resolution of the program can be increased by deactivating the display of the running time.

System requirements. IBM-compatible computers running Windows Version 3.1 or later will do for the program. A mouse is not necessary. The installation program requires SHARE.EXE /L:500 to be loaded in the autoexec.bat file.

Conclusions. A specific feature of the present software is its applicability to situations divided into variable, user-defined intervals. Tarapacki and Kristal (1990, 1993) have described a program that similarly divides the test period into several intervals. In contrast to the present program, those intervals are of a fixed length and must be defined before the coding. Also, the present software allows not only frequencies and durations but also onsets of the events within each interval to be registered. Another feature is that the events representing a continuation of the event started during the earlier interval can be separated from the data. This is especially important when an onset of each interval is defined, for example, by an experimental stimulus that may contribute to the continuity of the event started before the stimulus.

Unlike earlier software (Claro et al., 1990; Hetrick et al., 1991; Noldus, 1991), the present program is able to code only exclusive events—that is, one event at a time. However, one should take into account that use of the time-locking property of the program permits numerous nonexclusive events to be caught. There are two ways to do this: (1) by using video recordings, one observer can code each class of events in separate runs, or (2) several observers can code simultaneously different classes of events even in a natural situation. The synchrony between the event classes is achieved by starting the tape from the same point in all runs or starting all programs at the same moment in a field situation. For nonexclusive and fast-changing event types, the methods described above are useful since the accuracy of the observer does not limit the amount or type of the coded events. Guthertz and Field (1989) have described a program that has been used in these ways in coding mother-child interactions. However, it has been designed to code changes of states (latencies from onsets of events to onsets of the next ones) within a single time period.

In the present program, the event lengths are coded by keeping a corresponding key pressed as long as the events last. For recording fast-changing exclusive events, this way seemed to provide the best reaction time from the observer. In many other programs (see, e.g., Claro et al., 1990; Hetrick et al., 1991; Tarapacki & Kristal, 1990, 1993), onsets and offsets of events are identified by separate keypresses. This method is useful especially if the events to be coded are nonexclusive and they last a long time.

In contrast to the previous software (Claro et al., 1990; Guthertz & Field, 1989; Hetrick et al., 1991; Noldus, 1991; Tarapacki & Kristal, 1990, 1993), no preset configurations defining keys, event categories, or other observational variables are provided by the program. Neither descriptive summaries nor printouts of the gathered data are supplied. For data reduction purposes, statistical packages especially for Windows have become very easy to use and are usually needed for final analyses. In fact, most users are likely already familiar with such software and often not so enthusiastic about learning to use a new program. Since a minimum of effort is needed to use the program, no extensive training of personnel is required. Modern portable computers are nowadays capable of running Windows programs allowing data acquisition not only in laboratories but also in natural situations. We have successfully applied the program in coding spontaneous behavioral responses in cats during intertrial intervals of classical conditioning (Korhonen, Ruusuvirta, & Arikoski, 1995).

Availability. The program is free of charge and available from Timo Ruusuvirta, University of Jyväskylä, Department of Psychology, P.O. Box 35, FIN-40351 Jyväskylä, Finland. Requests should be accompanied by a formatted 1.44-Mb 3.5-in. diskette.

REFERENCES

- CLARO, F., DEL ABRIL, A., SEGOVIA, S., & GUILLAMON, A. (1990). SBR: A computer program to record and analyze sexual behavior in rodents. *Physiology & Behavior*, **48**, 489-493.
- GUTHERTZ, M., & FIELD, T. (1989). Lap computer or on-line coding and data analysis for laboratory and field observations. *Infant Behavior & Development*, **12**, 305-319.
- HETRICK, W. P., ISENHART, R. C., TAYLOR, D. V., & SANDMAN, C. A. (1991). ODAP: A stand-alone program for observational data acquisition. *Behavior Research Methods, Instruments, & Computers*, **23**, 66-71.
- KORHONEN, T., RUUSUVIRTA, T., & ARIKOSKI, J. (1995). *Spontaneous directed head movements during intertrial intervals of differential conditioning: A link between elicited and emitted behavior?* Manuscript submitted for publication.
- NOLDUS, L. P. J. J. (1991). The Observer: A software system for collection and analysis of observational data. *Behavior Research Methods, Instruments, & Computers*, **23**, 415-429.
- TARAPACKI, J. A., & KRISTAL, M. B. (1990). An adaptable microcomputer program for recording behavior durations. *Physiology & Behavior*, **47**, 381-384.
- TARAPACKI, J. A., & KRISTAL, M. B. (1993). An adaptable microcomputer program for monitoring behavior. *Physiology & Behavior*, **53**, 621-622.

Note—This research was supported by the Finnish Academy. We wish to thank the following individuals: Professor Jüri Allik for his contribution to the development of the predecessor of the described program; William Hetrick, Sarah Ransdell, and an anonymous reviewer for their constructive criticism.

Timo Ruusuvirta, Tapani Korhonen, and Juha Arikoski
University of Jyväskylä, Finland

(Manuscript received May 12, 1995;
revision accepted for publication August 29, 1995.)