Easy first steps and their implication to the use of a mouse-based and a script-based strategy

Eldad Yechiam

Department of Psychology, Indiana University

Ido Erev, Avi Parush

Faculty of Industrial Engineering and Management,

Technion – Israel Institute of Technology

Mail should be addressed to the first author:

Eldad Yechiam, Department of Psychology, Indiana University, 1101 East 10th St., Bloomington, IN 47405, Phone: 812-856-4678, Fax: 812-855-4691

The authors would like to thanks Daniel Gopher, Brent Cohen, and three anonymous reviewers for their helpful comments.

Appeared in Journal of Experimental Psychology: Applied, 10 (2), 89-96.

Journal Homepage: http://www.apa.org/journals/xap.html

Abstract:

The present study evaluates the convention to design training environments by giving access to easy strategies first and progressing to more difficult and efficient strategies thereafter. An experiment was conducted focusing on training in a simplified editing task. This task could be performed by an easy (mouse-based) strategy as well as using a more efficient (script-based) strategy. Two learning environments were compared, based on the order of the introduction of the two strategies to performers: an “easy first” program and a “difficult first” program. The results highlight two interesting patterns. First, initial training in an easy strategy impaired the acquisition of a more efficient strategy. Secondly, learning the easy strategy first reduced between-performer variability. It helped poor performers but resulted in a lower proportion of high-level performers.

Key words:

Training, exploration, motivation, complex skills, programming, immediate reinforcement.

A basic assumption in modern training methods is that the training environment should initially encourage the trainer to use the easiest strategy, and progress to the more efficient and difficult strategies (see e.g., Carrol & Carrithers, 1984; Frederiksen & White, 1989; Gong & Salvendy, 1995). For example, the classroom lessons of beginning fencers typically include reduced-difficulty maneuvers characterized, for instance, by lower speed (Weyer & Shebilske, 1995). Likewise, Carrol and Carrithers (1984) developed a system called “training wheels” for a stand-alone word processor, which directs users to specific easy-to-use options. More complex alternatives are presented once the user has mastered the initial “novice friendly” options. It appears that using this system was more efficient than letting novices engage in the more difficult options in the initial stages of learning (Bannert, 2000; Carrol and Carrithers, 1984)

The logic behind this intuitive and effective approach can be divided into two major assertions. First, in many cases the use of the more difficult and advanced strategies depends on knowledge gained from the use of the simpler strategies (e.g., Frederiksen, Weaver, Warren et al., 1983; Frederiksen & White, 1989; Gong & Salvendy, 1995). Thus, presenting the simpler strategies first contributes to the learning process of the more advanced alternatives.

A second set of contributors to the value of initially teaching the easier strategies involves the reduction of stress and attention demands (Kramer, Larish & Strayer, 1995), and the risk of learned helplessness (Beck, 1977; Dweck & Licht, 1980). Reduction of attention demands is one of the basic premises of the part-task approach to training (e.g., Ash & Holding, 1990; Goettl, 1995; Schneider, 1985). This approach is based on the assertion that much of the hardship of a complex task is related to its initial high cognitive load (see e.g., Brown & Carr, 1989; Nissen & Bullemer, 1987). Learned helplessness is relevant because an early attempt to teach strategies that are too difficult can lead trainee to feel that mastering the task is impossible.

Evaluation of the contributors to the advantage of easy-first training methods highlights the robustness of this advantage. Yet it also suggests one possible exception to this advantage when the more difficult strategies are also the most efficient. If the performance in these difficult and efficient strategies is not improved by the use of simpler strategies, and learned helplessness is not a major problem, then starting with easy strategies might be counter productive.

A problem might arise because there are qualitative differences between the different stages of the training process: It seems that controlled processes dominate in the early phase of learning, and that automatic processes dominate in a later phase (e.g., Ackerman, 1987; 1990; Anderson, 1983; Shiffrin and Schneider, 1977). Accordingly, the more time a trainee spends on the easy task, the more the person is likely to advance towards automation of this task. However, the process of automation is not effortless. According to Ackerman (1987) and Anderson (1984) it requires going through a “declarative knowledge” phase where knowledge is gathered, and a “compilation of knowledge phase”, where it is processed. Therefore, at the point where automation is gained, learners have already invested efforts in extending their knowledge and automating the easy strategy. Consequently it would be difficult to move trainees to invest effort in learning a more difficult strategy. In the context of investment, this is known as “escalation of commitment” (Staw, 1981).

Moreover, in this setting, easy-first training creates a problematic incentive structure in which initial attempts to master the difficult-but-efficient strategies will be associated with a drop in performance. Thus, the natural tendencies to response to immediate reinforcements (see Herrnstein, Loewenstein, Prelec & Vaughan, 1993) and to focus exploration on strategies that are similar to strategies that have yielded the best outcomes (see Busemeyer & Myung, 1987; Erev & Gopher, 1999; Yechiam, Erev & Gopher, 2001) are expected to slow the adaptation process. The training might consequently lead to a choice of a locally optimal alternative over other more closely optimal alternatives.

Along the same line, in many cases performers tend to rely on proxies to the performance level that are based on how well a particular strategy has been learned (Bjork, 1999). For example, performers’ confidence in their answers to a question is high if their response time to these questions is relatively short (Costermans, Lories & Ansays, 1992). In the present context, this implies that if performers use an easy strategy and master it, they may use their feelings of fluency as a proxy to actual performance level. Once an easy strategy is available, the performance in a difficult strategy may be perceived is less fluent, thereby reducing the motivation to use it. Note that as performers have difficulty in estimating changes in fluency that occur over time (Benjamin, Bjork & Schwartz, 1998; Bjork, 1999), they are not likely to predict changes in the relative fluency of using different strategies.

Given these potential problems, it is suggested that the relative value of easy-first and difficult-first training programs is likely to depend on two main factors: The possibility of successful transfer from easier to more efficient strategies, and performance variability in the difficult strategy.

The role of cognitive fidelity

When it is possible to focus initial training on easy strategies that facilitate transfer to the most efficient strategies, easy-first training is likely to be most successful. This assumption is embedded in the literature that emphasizes cognitive fidelity (Brown, 1986) in training, or the attempt to induce transfer by developing simplified versions of tasks that promote transfer from simple to complex tasks (Goettl, Ashworth & Chaiken, in press; Gopher, Weil & Siegel, 1994; Hart & Battiste, 1992).

One elegant example of easy-first strategies with this property is incorporated in the part-task training approach of Frederiksen and White (Frederiksen & White, 1989; Frederiksen, White & Gutwill, 1999; White, 1984; White & Frederiksen, 1985). In the “Principled Decomposition” method trainees perform a sequence of sub-tasks, with the goal of enhancing the skills required for the performance of more complex tasks. The sub-tasks are constructed using experts’ analysis of the basic skills required to perform the complex task. It is argued that the conceptual knowledge required for performing the integrated task can be more easily gained in simpler and less stressful environments. In Frederiksen and White’s studies (e.g., Frederiksen & White, 1989; White & Frederiksen, 1985), a strong positive transfer was found in the transition from the sub-tasks to the complex task, and to other similar tasks. Another successful application is in computer interfaces where only relatively easy options are initially available. These interfaces are successful when the easy options enhance knowledge of the application and task environment (Bannert, 2000; Carrol & Carrithers, 1984; Carrol & Kay, 1988; Gong & Salvendy, 1995; Thimbleby, 2001).

In some cases, though, it is not possible to create cognitive fidelity. This may occur, for example, when the strategies are already designed (poorly) or when the strategies are designed by different agents or are a part of different products. For example, in statistics, menu-based statistical applications (especially the Statistical Package for the Social Sciences, SPSS), are often used to teach undergraduate psychology students (see Bartz & Sabolik, 2001). Other programs (such as older versions of the Statistical Analysis System, SAS) that do not facilitate easy first steps, are used to train more advanced students and professionals (Jolliffe & Rangecroft, 1997). The move from menu to script based applications entails a situation where there is little cognitive overlap between different alternatives for performing the task. Of course, knowledge in statistics is a skill developed by both applications; thus, there is some cognitive fidelity between these two application types. However, there are also unrelated components, such as following syntax rules, and these are the ones that commonly lead to difficulties in learning script based applications (Spector, 2001).

Another example is a situation where the strategies are dictated by the environment rather than by human design. One case in point is in the decision to speak a language given the languages one knows and those spoken in one’s environment. Using one’s native tongue is an easy strategy, inherently available, that may be hard not to draw on in a foreign country (Chiswick, 1997; Chiswick & Miller, 1995). Yet on the long run, not learning the local dialect is likely to be inefficient. For example, Chiswick (1997) reports that greater fluency in Hebrew enhanced earning among immigrants who moved to Israel. Thus, in this case, learning English first may have detrimental consequences.

When cognitive fidelity is low: The role of performer variability

In cases where successful transfer cannot be ensured, the benefit of easy first steps (e.g., a menu based application) may be countermanded by the fact that easy-first training environments can lead trainees to locally optimal strategies. In this case, easy-first training is expected to reduce the proportion of trainees that master the most effective but difficult strategy (e.g., the proportion of those who use a script based application). This possibility implies two interesting effects depending on the variability of trainees' performance in the more difficult strategy.

When difficult-first training programs ensure convergence to globally optimal strategies, easy-first training is predicted to impair performance. An example of an impaired performance is in Chiswick's (1997) finding concerning the (negative) effect of prior knowledge of English on mastering Hebrew, and the positive effect of mastering Hebrew on earnings.

A different effect is predicted when not all trainees are able (and/or motivated) to master the difficult strategy. In this case, training methods should try to train individuals with different skill level separately or differently (Snow, 1989). When this recommendation is not followed, difficult first training is assumed to create two groups: Those who acquire the difficult strategy and excel with it, and those who are overwhelmed by it and wash out. In contrast, easy-first training is predicted to lead most people to adopt the easy strategy. As a result, very few trainees are expected to "wash out" but also with few are expected to adopt the difficult and efficient strategy.

Thus, easy-first training is likely to help poor performers, as it decreases washing out. On the other hand, it may lead to sub-optimal performance in some high potential trainees, who would have been able to master the difficult strategy if they had not been exposed to the easy strategy. In other words, easy-first training is expected to reduce between-performer variability. The present investigation examines this eventuality.

Experiment

An experiment compared an “easy-first” and a “difficult-first” training program in a laboratory setting. The difficult but efficient strategy was the use of a script. The easier but less efficient strategy (the local optimum) involved the use of a computer mouse. The use of scripts is generally more difficult, as it has more memory and attention demands (Sein, Bostrom, Olfman & Davis, 1993; Sohn & Doane, 1997). On the other hand, scripts may also be more efficient, as they eliminate the necessity to perform single actions repeatedly in monotonous tasks. Furthermore, scripts ensure that the each repetition of the task conforms to the exact operations dictated by the script.

Two training environments, based either on scripts or menu selections, were compared in a simulated “editing” task. The task involved the replacement of numbers with letters in ten 30×30 spreadsheet matrices (replacing all of the numbers divisible by seven with the letter “Y”). The task was performed in two phases. In the *Difficult-first* training environment, performers were initially (in phase one) exposed to a difficult strategy: the use of a script. To correctly use this strategy, performers were required to find the correct script among ten available scripts and follow instructions to correctly type the script. The only alternative to using the script was to perform the task by manual replacement. In the second phase (the next 10 matrices), an easy strategy was made available: double-clicking a cell, which would lead to the replacement of 15×15 cells in the perimeter of the selected cell.

A second group of performers was presented with an *Easy-first* environment. These performers were exposed to the “double-click” strategy in the first phase of the task. In the second phase, use of the script was also made available.

It was predicted that the Difficult-first environment would encourage the use of the script-based strategy. Regarding the performance level on the other hand, it was assumed that the Difficult-first program would demonstrate high between-person variability. Specifically, some performers who are exposed to the difficult strategy first might realize the benefit of the script, while others might not use the script properly, and would have to use the inefficient strategy of manual replacement. This variability difference was predicted to decrease in the second phase of the tasks due to the availability of the “double-click” method to performers who failed to master the use of the script.

Method

Participants:

The participants were 42 students from the Technion - Israel Institute of Technology. They were all first or second year students in the faculty of Industrial Engineering who have passed a basic course in computer applications. The course covers Microsoft Excel, the application used in the present experiment. The sample included 14 women and 28 men. The participants were paid a sum of IS 40 (about $9) for their participation in the study. Participants were randomly assigned to two training conditions. Men and women were paired and randomly allocated to the two conditions.

Task:

The experimental task involved the replacement of numbers divisible by seven with the letter “Y” in a 30×30 Excel grid. The numbers were randomly determined and ranged from 10 to 30. Figure 1 is a screen capture of a part of the computer screen, showing the bottom right hand side of the matrix. The task was initialized and terminated by pressing the “Next Task!” button (see Figure 1). If more than thirty mistakes (incorrect replacements) were recorded at the end of a task, a message would then appear, indicating that more than thirty mistakes had been made, and directing the participant to continue in that task. Otherwise, the task number would be incremented by one, and a new random matrix would replace the old one. This indicated a repetition of the task.

There were three possible strategies for performing the task. These strategies were available to participants as described above, according to the condition to which they were assigned:

1. *Using scripts* - This strategy requires participants to go through a six-page manual of scripts. The first page of the script manual is a table of contents with the command names and appropriate pages. The next five pages include instructions of how to type each script. Each of the scripts in the manual involves a command name (e.g., “Replace macro“) and five parameters. The parameters are answers to beginner-level questions in Excel, such as “the name of the file with the ‘xls’ suffix (appears on the blue heading on the worksheet)”; “the number of cells in the matrix”. Of the ten

available scripts, only one randomly determined script is “correct”. In order to find the correct script, performers are required to enter the script in a text box (located below the matrix; see Figure 1) and press the “Execute” button by the text box. If the wrong script is correctly entered, then a “Try another command!” message appears at the center of the screen. If the correct script is entered with errors, then a “Syntax error” message appears. Finally, if the correct script is entered without errors, all of the numbers divisible by seven in the entire matrix are replaced with the letter “Y”.

Note that this strategy initially requires much trial and error to use properly. The performers must answer the questions correctly, as well as find the right script from the ten available options. Once the strategy is implemented correctly it requires typing a 35-character script, and one mouse click (pressing the “Execute” button) on the first use and. However, only one mouse click is required on each subsequent use (a total of 55 key presses for 20 trials)

2. *Using double-click* - In this strategy, double-clicking a cell leads to the correct replacement of 15×15 cells in the perimeter of the chosen cell. That is, the chosen cell becomes the center of a 225-cell square area in which all numbers divisible by seven are replaced with the letter “Y”. Finding the optimal cells for clicking is not easy: they are the exact centers of the four squares dissecting the larger matrix. Using this optimal strategy, ten mouse clicks are required per matrix (one double click for each center, a click on any cell to leave edit mode, and pressing the “Execute” button), which implies 200 clicks are required for 20 trials.

A simpler strategy is to fill the largest available space that is not filled already by placing the mouse in its middle. This strategy requires, on average, about fifteen mouse clicks per matrix (300 for the entire task). However, since participants do not know for sure when the task is finished, the actual number of clicks may be much higher. First, performers can overestimate the number of required clicks (for achieving less than 30 mistakes). This can result in much more clicks then necessary to cover the space. Also, performers may underestimate the number of required clicks, and press the “Next task” button when they have 30 or more errors, which results in an error message; and necessitates three clicks (removing the message, leaving edit mode, pressing the “Next task” button) as well as checking which cells should be replaced.

Note also that double clicking in Excel must be performed inside the cell rather than at the border of a cell. If the center of the cross pointer (seen in Figure 1) touches any of the borders of the cell, this selects the farthest cell of the matrix in the direction of that border. This implies that the center of the pointer must be correctly aimed repeatedly into the 3 mm by 4 mm (on average) center of the 5 mm by 6 mm Excel cells. Clearly, this is a potential source of errors and slower performance (see Proteau, 1992), especially for participants with low perceptual motor skills.

3. *Manual replacement* - This strategy involves the physical replacement of each and every number by the performer. This implies at least 260 mouse and keyboard clicks per matrix (5,200 clicks for the entire task).

The dependent variable was the average time of task completion, as there were no penalties for the number of errors over the limit described above. Unfortunately, we did not measure the number of mouse clicks and pointer aiming-errors. However, response time increased as a function of these factors as well as other task components. Therefore, it was collected as a general measure of performance. The time taken for measurement was the last half of each experimental phase, since our focus was to examine the outcome of learning (success or failure in the use of the difficult strategy) rather than the variability during learning. In addition, we examined the overall time it took to perform the task.

Apparatus:

The experimental task was constructed using Visual Basic for Applications in Microsoft Excel (Office 97 version). The experiment ran on eight Pentium-III computers with 17-inch screens (1024x768 pixels). A standard QWERTY keyboard and Microsoft mouse were used. Participants were seated in cubicles separated by dividers and were instructed not to speak to each other during the experiment.

Procedure:

Participants in both conditions were seated in front of a screen (seen partially in Figure 1) and were given the following written instructions: “You will immediately participate in a decision-making experiment. The payment for participating is IS40. During the experiment, you will be required to perform the following task: Replace all numbers divisible by seven with the letter “Y” in a matrix of 30×30 cells. You will perform the task in ten 30×30 matrices. To end a task and begin a new task, you need to press the “Next task!” button (the large button). Beginning and ending a task is performed individually at your own time. Please perform the task as quickly as possible”.

Participants in the Difficult-first condition were then given the following written and oral instructions: “You have two alternatives for performing the task: You can use manual replacement and, additionally, one of the items in this manual should operate”. At this stage, each participant was provided with the script manual described in the Task section above.

Participants in the Easy-first condition were given the following written and oral instructions: “You have two alternatives for performing the task: You can use manual replacement, and additionally, choosing any cell and double-clicking the mouse. This should replace any cell at a range of 15 by 15 cells”.

Participants in both conditions were not told that following the first ten trials there will be another set of ten trials, and how many sets of trials are expected. After completing ten trials, a message appeared, telling the participant to call the experimenter. At this stage, participants in both conditions were given the following instructions individually: “You will now perform the same task again. You may use the methods that you had used previously”. Participants in the Difficult-first condition were then presented with the mouse-based option, while participants in the Easy-first condition were presented with the script option. Upon completing the second run of the task, participants were given their fee and thanked.

Results and Discussion

Strategy choice:

Of the 21 participants in the Difficult-first condition, 13 (62%) succeeded in using the script-based strategy during the experiment. In contrast, only four participants in the Easy-first condition (19%) learned to use the script-based strategy. The proportion difference was significant ($Z = 2.83$, $p < .01$). In the Difficult-first condition, with the exception of two performers, the script strategy was successfully employed in the first trial of the task (these two performers began using scripts in trial 2 and trial 3). In the Easy-first condition, three of the four performers who used scripts did so successfully in the first trial on phase two. One performer used scripts only in trial 6.

As for the remaining participants, of the eight Difficult-first participants who did not master the script strategy, five used manual replacement for the entire first phase of the study, and went on to use the double-click strategy in the second phase. One of these participants was released from the experiment shortly after the beginning of the second phase (this was an extremely slow performer, who had been performing the experiment for over an hour and a half). Surprisingly, three participants did not even complete the first phase of the study, but chose to leave the experiment prematurely, even though they thought it would cost them their fee (they were nevertheless given their fee once they had left the room). In the Easy-first condition, all of the remaining 17 participants used the double-click strategy for the duration of the first and second phases.

Thus, it appears that the Difficult-first program increased the successful use of the difficult strategy, while it also demonstrated a larger proportion of participants who performed badly in the first phase. In this respect, the avoidance behavior displayed by three participants can be considered a very poor task performance strategy.

Completion times

Figure 2 demonstrates the outcome of the usage of different strategies on completion times. In the analysis of completion time averages we could not use the results of the participants who was released after the first phase. There were thus 21 participants in the Easy-first condition and 17 participants in the Difficult-first condition.

We compared the average completion times among all performers in the two conditions. The results show that overall, completion times in the Easy-first condition were shorter (93.7 seconds compared to 116.1 seconds). Repeated measures ANOVA was conducted with condition and time (two phases). The results (summarized in Table 1) show no main effect of condition, but rather, the expected interaction of time and condition (F (1,28) = 20.12, $p < .01$). Performers in the Easy-first condition had a significant advantage in the first phase (119 seconds compared to 216 seconds; t (26.9) = 2.62, $p < .05$; Cohen's d = 0.87). This is explained by the advantage of the mouse-based strategy in the learning phase. In contrast, Easy-first performers had, on average, a disadvantage in the second phase (69.6 seconds compared to 16.1; t (27.9) = 8.32, $p < .01$; Cohen's d = 2.61). Note, though, that the overall long-term advantage of Difficult-first performers in phase two is without including the performers who left (see the discussion section).

Examination of the predictions regarding variability:

To examine the effect of strategy choice on performer variability in the two conditions, we ranked all of the performers' completion times in the last half of the first and second phases. Thus, we could add the participants who left or were released as poor performers. This is justified, primarily, by the fact that all of the performers who left failed to use the difficult strategy, and their leaving was because the manual strategy was boring and effortful. Their leaving was therefore due to their inability to complete the experimental task. This was confirmed in short interviews with these participants that were conducted following the experiment, and also by their initial low performance levels. A performer who left after the fourth trial had the slowest completion time in the Difficult-first condition (1,012 seconds). Two other performers left before completing trial one after more than 1,500 seconds (25 minutes) have elapsed, well over the time it took for any of the performers that did not use scripts to complete the first trial. The performer who was released had the second slowest time in phase 1 (343 seconds average), and was discharged after having a very slow pace in the first trial of phase 2. The outcome of the analysis including these four performers is illustrated in Figure 3.

A comparison between participants in the first and last quartiles (ten performers) in the first phase shows that in the last such segment, seven out of ten performers (70%) were from the Difficult-first condition, compared to 18% in the second to fourth segments. The proportion difference was significant ($Z = 2.86$, $p < .01$). In the first (high performance) segment, all ten performers were from the Difficult-first condition, a significantly higher proportion than in the second to fourth segments

($Z = 4.32$, $p < .01$). Thus, as predicted, the Difficult-first condition led to increased between-performer variability.

This difference is also evident in the distribution of the ranked scores. The between-performer standard deviation in the Difficult-first condition was 15.84 compared to 6.52 in the Easy-first condition. Levene test for Homogeneity of Variance shows that the difference was highly significant (t (40) = 5.83, $p<.01$; cohen’s $d = 1.84$). The results of the test are presented in Table 2.

As expected, the differences in variability between the two conditions decreased in the second phase, since performers in the Difficult-first condition who were ranked last moved to being ranked in the second or third quartile (owing to the use of the mouse-based strategy). The between-performer standard deviation in the Difficult-first condition was 13.97 compared to 8.3 in the Easy-first condition. Levene test indicates, however, that the difference was still significant (t (40) = 2.65, $p < .05$; cohen's d = 0.83).

Note that examining the variability of average completion times in the two conditions without the performers who left (see Table 2) reveals that in the second phase of the task, the variability in the Easy-first condition (26.8) was higher than in the Difficult-first condition (9.0). This difference was significant (t (1,36) = 2.21, $p < .05$, Cohen's d = 0.74). Analyzing the data without treating these extreme cases thus seems in this case to produce misleading results.

Discussion

The present research highlights two interesting properties of training programs that begin by teaching the easiest strategy, and progress to the hardest and most efficient strategy. First, when difficult and efficient strategies are not similar to easier strategies for performing the same task, easy-first training programs can reduce the likelihood that the more efficient strategies will be adopted. The present study demonstrates this pattern in a laboratory study of a simulated editing task with a difficult strategy of using scripts and an easier strategy based on mouse clicks. In a training environment where the mouse-based strategy was introduced sooner (an Easy-first condition), fewer performers used scripts.

A second observation, and the main contribution of the current research, involves the effect of easy-first training on performance variability. It was argued that easy-first training tends to reduce between-person performance variability. It helps weaker performers, results in less performers achieving high-level performance. In the experimental study, this was demonstrated by the fact that in easy-first training, fewer performers used the manual strategy or displayed avoidance behavior and left the experiment. However, fewer performers advanced to using the script. This led to lower variability in the ranked completion time scores in phase one.

In the second phase of the task, a mouse-based strategy was presented to the participants who were initially introduced to the script-based strategy (the Difficult-first condition). Our original assumption was that this would reduce the variability in the Difficult-first training condition, as it would allow trainees who failed to use scripts to “advance” to the mouse based strategy. While the variability did decrease, unexpectedly, the difficult-first environment also had somewhat different effects. Three participants in this condition have chosen to leave the experiment during the first phase. These participants did not benefit from the late introduction of the easy strategy. This must serve as a warning for the use of difficult-first training in programs where the success of individual trainees is more important than group averages.

Potential limitations

The present experiment has some methodological limitations, which bear mentioning. First of all, participants who moved to using the script strategy had slower overall completion times. This may suggest that participants chose the mouse-based strategy not as a result of a failure to realize which was the ultimately better strategy, but because of their ability to predict which strategy would be better for the experiment.

While this argument cannot be rejected entirely, it must be pointed out that the participants did not know how many task repetitions were going to take place, especially as they already had one surprise repetition. They only knew that the current repetition had ten trials. It is likely that if more repetitions have taken place, the advantage of those who had used the script (among the Easy-first group), which was already evident in the last five trials, would have led to shorter total completion times.

A similar limitation is the relative short time of the experiment (both actual and predicted), which may have contributed to the tendency to pick out the less efficient, mouse-based, strategy in the Easy-first condition. Performers in the Easy-first condition may have decided not to invest additional effort to explore a solution that "might" be more efficient in the event that the task will be continued for an extended period (see Ackerman & Woltz, 1994). The issue of predicting the repeatability of a certain experiences is interesting for future research. It appears that performers may have inherent biases in this domain (Bar-Hillel, 1973; Gneezy, 1995).

Finally, the fact that three participants in the Difficult-first condition left and one was released raises the question of how representative were the remaining participants. We used ranked completion times to partially overcome the problem, including the participants who left or were released in the last quartile. In addition, in examining completion times we focused out analysis on the differences between performers in the Easy first condition who used different strategies, and between script users in the two conditions. We did analyze the overall differences between the two conditions, but the results are considered to be effected by the fact that four performers were not included.

Potential generality

We have focused on a situation where there is virtually no overlap between the demands of alternative strategies for performing the task. It is clear, though, that this represents an extremity in the lack of cognitive fidelity. Studies should investigate real-world situations with different levels of fidelity in an attempt to examine the generality of the present results.

It is our belief that the present findings may be generalized to tasks where cognitive fidelity of two alternative strategies is low. For example, Gong and Salvendy (1995) examined a strategy involving menu selection and a strategy involving a command-line interface in the task of performing system-management operations. It was found that the initial use of menu selection improved the use of the command-line interface, and this led to superior performance levels in the long run. However, it appears that in a condition where trainees were permitted to use both strategies, they tended to employ menu selections and rather than commands, and this led to lower performance levels. These results are similar to those of the present experiment, showing that an earlier presentation of a simpler strategy coupled with the ability to use this strategy later on in training, did not enhance the use of a difficult and efficient strategy.

Another examination of potential generality should pertain to the fact that in the present study participants were not told that the script strategy is a more efficient way to complete the task. It is our belief, based on anecdotal findings, that the results may be robust to situations where performers are more informed.

In the course of teaching statistical analysis applications to graduate behavioral science students, a similar phenomenon became evident to the authors. Graduate students who have experience in menu-based applications find it difficult to move to code-based applications even though the efficiency of such applications is explained to them. Typically, students do not learn code-based applications (such as SAS) unless prompted by formal and mandatory requirements. A related difficulty may characterize the use of scripts in macros. The macro utility allows a series of user actions to be performed conjointly and repeatedly, thereby reducing repeated and monotonous work. Nevertheless, it appears that few users employ macros (see e.g., Lau & Weld, 1999; Mackay, 1990; Rosson, 1984; See review in Yechiam, 2003), even if they recognize that using macros is an efficient strategy (Lambrecht, 1998). One possibility is that macros are not used because performers are attracted to alternative procedures that do not require script writing (e.g., “copy and paste”, “search and replace”). This was reported by Lau and Weld (1999) as one of the common reasons stated by performers for not using Elisp macros.

In general, it appears that the motivation to learn script-based strategies is sensitive to the availability of strategies that do not involve scripts. This is understandable given that scripts require knowledge that goes beyond what is needed to simply perform a task. Acquiring this knowledge may seem like a waste of time since it does not immediately improve task performance and adds nothing to the task-domain knowledge. The only advantage of this knowledge is in the long-term use of what appears to be a difficult strategy. Learning scripts can therefore be considered as a pure “investment for the future”, an act, which under a noisy reinforcement structure (see Rachlin, 1995), performers do not easily make.

Conclusions

The goal of designers of training programs is to help performers master efficient strategies, such as the use of scripts, even if they are difficult to learn. This can be achieved by cognitive fidelity in training modules (Goettl et al., In press), which implies that performers go through a transition process from one or more simpler strategies to the more complex strategy. In reality however, there are cases where due to task complexity or a difficulty of controlling the task environment, cognitive fidelity cannot be ensured.

In these cases, the benefit of using easy first steps is expected to depend on the goal of the training. To demonstrate the importance of the training goal, consider the difference between a highly competitive environment in which only a small proportion of the trainees are expected to succeed, and an environment in which all trainees are expected to acquire the trained skill. An extreme example of the former is competitive sports. In this setting it is understood that only a small proportion of the trainees are likely to embark on a successful career. Thus, easy-first training programs that are sub-optimal for high potential trainees should be avoided. For an extreme example of an environment in which all trainees are expected to succeed, consider potty training. It is easy to see that easy-first training programs are optimal in this setting.

Most software training environments fall somewhere between these two extreme examples. The current analysis suggests that a focus on a high percentage of successful users should favor training environments that use mouse selections and menus. A focus on a small proportion of highly productive users should favor the more difficult code-based training.

References

Ackerman, P.L. (1987). Individual differences in skill learning: An integration of psychometric and information processing perspectives. *Psychological Bulletin*, *102*, 3-27.

Ackerman, P.L. (1990). A correlations analysis of skill specificity: Learning, abilities, and individual differences. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *16*, 883-901.

Ackerman, P.L., & Woltz, D.J. (1994). Determinants of learning and performance in an associative memory/substitution task: Task constraints, individual differences, volition, and motivation. *Journal of Educational Psychology*, *86*, 487-515.

Anderson, J.R. (1985). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Ash, D.W., & Holding, D.H. (1990). Backward versus forward chaining in the acquisition of a keyboard skill. *Human Factors*, *32*, 139-146.

Bannert, M. (2000). The effects of training wheels and self-learning materials in software training, *Journal of Computer Assisted Learning*, *16*, 336-346.

Bartz, A.E., & Sabolik, M.A. (2001). Computer and software use in teaching the beginning statistics course. *Teaching of Psychology*, *28*, 147-149.

Bar-Hillel, M. (1973). On the subjective probability of compound events. *Organizational Behavior and Human Performance*, *9*, 396-406.

Beck, J. (1977). Locus of control, task experiences and children's performance following failure. *Journal of Educational Psychology*, *71*, 207-210.

Benjamin, A.S., Bjork, R.A. & Schwartz, B.L. (1998). The mismeasure of memory: When retrieval fluency is misleading as a metamnemonic index. *Journal of Experimental Psychology: General*, *127*, 55-68.

Bjork, R.A. (1999). Assessing our own competence: Heuristics and illusions. In D. Gopher & A. Koriat (Eds.), *Attention and Performance, XVIII: Cognitive regulation of performance, Theory & Applications* (pp. 435-459). Cambridge: MIT Press.

Brown, J.S. (1986). From Cognitive to Social Ergonomics. In D.A. Norman & S.W. Draper (Eds.), *User centered systems design new perspectives on human - computer interaction*. N.J.: Lawrence Erlbaum Associates.

Brown, T.L., & Carr, T.H. (1989). Automaticity in skill acquisition: Mechanisms for reducing interference in concurrent performance. *Journal of Experimental Psychology: Human Perception & Performance, 15*, 686-700.

Busemeyer, J.R., & Myung, I.J. (1987). Resource allocation decision making in an uncertain environment. *Acta Psychologica*, *66,* 1-19.

Carrol, J.M., & Carrithers, C. (1984). Blocking learner error states in a training-wheels system, *Human Factors*, *26*, 377-389.

Carrol, J.M., & Kay, D.S. (1988). Prompting, feedback and error correction in the design of a scenario machine. *International Journal of Man-Machine Studies*, *28*, 11-27.

Chiswick, B.R. (1997). Hebrew Language usage: Determinants and effects on earnings among immigrants in Israel. *Journal of Population Economics, 11*, 253-271.

Chiswick, B.R., & Miller, P.W. (1995). The endogeneity between language and earnings: international analyses. *Journal of Labor Economics, 13,* 245–287.

Costermans, J., Lories, G., & Ansays, C. (1992). Confidence level and feeling of knowing in question answering: The weight of inferential processes. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18*, 142-150.

Dweck, C., & Licht, B. (1980). Learned helplessness and intellectual achievement. In J. Garber & M. E. P. Seligman (Eds.), *Human helplessness: Theory and applications* (pp. 197-221). New York: Academic Press.

Erev, I., & Gopher, D. (1999). Cognitive Game-Theoretic Analysis of Attention Strategies, Ability and Incentives. In D. Gopher & A. Koriat (Eds.), *Attention and Performance, XVIII: Cognitive regulation of performance, Theory & Applications*. Cambridge: MIT Press.

Frederiksen, J.R., Weaver, P.A., Warren, B.M., Gillotte, H.P., Rosebery, A.S., Freeman, B., & Goodman, L. (1983). *A componential approach to training reading skills (Final Report 5295)*. Cambridge, MA: Bolt Beranek and Newman.

Frederiksen, J.R., & White, B.Y. (1989). An approach to training based upon principle-task decomposition. *Acta Psychologica*, *71*, 89-146.

Frederiksen, J.R., White, B.Y., & Gutwill, J. (1999). Dynamic mental models in learning science: The importance of constructing derivational linkages among models. *Journal of Research in Science Teaching*, *36,* 806-836.

Gneezy, U. (1995). Probability Judgments in Multi-Stage Problems: Experimental Evidence of Systematic Biases. Discussion paper. Tilburg - Center for Economic Research (RePEc:fth:tilbur:9601).

Goettl, B.P. (1995). Part-task training of complex tasks: Utility of backward transfer. *Proceedings of the Human Factors and Ergonomics Society*, *39,* 1345-1349.

Goettl, B.P., Ashworth, A.R.S. III, & Chaiken, S. (in press). Advanced distributed learning for team training in command and control applications. In S. Fiore & E. Salas (Eds.), *Where is the learning in advanced distributed learning*. NJ: Erlbaum.

Gopher, D., Weil, M., & Bareket, T. (1994). The transfer of skill from a computer game trainer to actual flight. *Proceeding of the Human Factors Society 36th Annual Meeting,* 1285-1290.

Gong, Q., & Salvendy, G. (1995). An approach to the design of a skill adaptive interface. *International Journal of Human-Computer Interaction, 7,* 365-383.

Hart, S.G., & Battiste, V. (1992). Field test of video game trainer. *Proceeding of the Human Factors Society 36th Annual Meeting,* 1291-1295.

Herrnstein, R.J., Loewenstein, G.F., Prelec, D., & Vaughan,W., Jr. (1993). Utility maximization and melioration: Internalities in individual choice. *Journal of Behavioral Decision Making, 6*, 149-185.

Jolliffe, F., & Rangecroft, M. (1997). Statistical software usage by student and consultant statisticians in the UK - A Comparative Study. *Maths & Stats, 8*, Issue 3.

Kramer, A.F., Larish, F.J., & Strayer, D.L. (1995). Training for attentional control in dual task settings: A comparison of young and old adults. *Journal of Experimental Psychology: Applied*, *1*, 50-76.

Lambrecht, J.J. (1998). Job task-analysis approaches for microcomputer application curriculum development. *Proceeding of the National Delta Pi Epsilon Research Conference*. University of Minnesota, Twin Cities.

Lau, T., & Weld, D. (1999). Intelligent macros: programming by demonstration for text editing. An unpublished manuscript. Washington University.

Mackay, W.E. (1990). Patterns of sharing customizable software. *Proceedings of ACM CSCW '90 Conference on Computer-Supported Cooperative Work.* Portland, Oregon: ACM Press.

Nissen, M.J. & Bullemer, P. (1987). Attentional requirements of learning: Evidence from performance measures. *Cognitive Psychology*, *19*, 1-32.

Proteau, L. (1992). On the specificity of learning and the role of visual information for movement control. In L. Proteau & D. Elliott (Eds.), Vision and motor control. *Advances in psychology*, No. 85. (pp. 67-103). Oxford, England: North-Holland.

Rachlin, H. (1995). The value of temporal patterns of behavior. *Current Directions in Psychological Science, 4*, 188-192.

Rosson, M.B. (1984). The effects of experience on learning, using and evaluating a text-editor. *Human Factors, 26,* 463-475.

Schneider, W. (1985). Training high performance skills: fallacies and guidelines. *Human Factors, 27,* 285-301.

Sein, M.K., Bostrom, R.P., Olfman, L., & Davis, S.A. (1993). Visualization ability as a predictor of user learning success. *International Journal of Man-Machine Studies*, *39*, 599-620.

Sohn, Y.W., & Doane, S.M. (1997). Cognitive constraints on computer problem-solving skills. *Journal of Experimental Psychology: Applied*, *3*, 288-312.

Shiffrin, R.M., & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. *Psychological Review*, *84*, 127-190.

Snow, R.E. (1989). Aptitude-treatment interaction as a framework for research on individual differences in learning. In P.L. Ackerman, R.J. Sternberg, & R. Glaser (Eds.), *Learning and individual differences: Advances in theory and research. A series of books in psychology*. (pp. 13-59). New York: W.H. Freeman.

Spector, P.E. (2001). *SAS® programming for researchers and social scientists*. Second Addition. Sage Publications.

Staw, B.M. (1981). The escalation of commitment to a course of action. *Academy of Management Review*, *6*, 577-587.

Thimbleby, H. (2001). Permissive user interfaces. *International Journal of Human-Computer Studies*, *54*, 333-350.

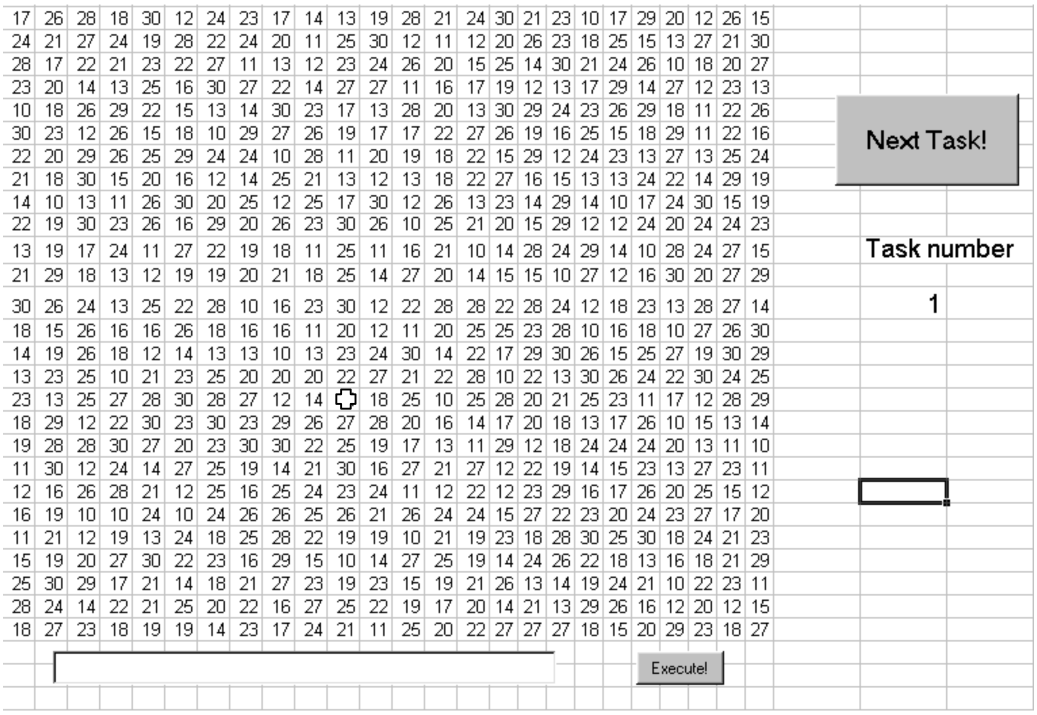
Weyer, A.M., & Shebilske, W.L. (1995). A case history of complex skill acquisition. Theoretical and pedagogical implications of fencing experiences. *Applied Research in Coaching and Athletics Annual*, *95*, 93-111.

White, B.Y. (1984). Designing computer activities to help physics students understand Newton's laws of motion. *Cognition and Instruction*, *1*, 69-108.

White, B.Y., & Frederiksen, J.R. (1985). QUEST: Qualitative understanding of electrical system troubleshooting. *ACM SIGART Newsletter*, *93*, 34.

Yechiam, E. (2003). Why are macros not used? A brief review and suggestions for improving training. Manuscript Submitted for Publication.

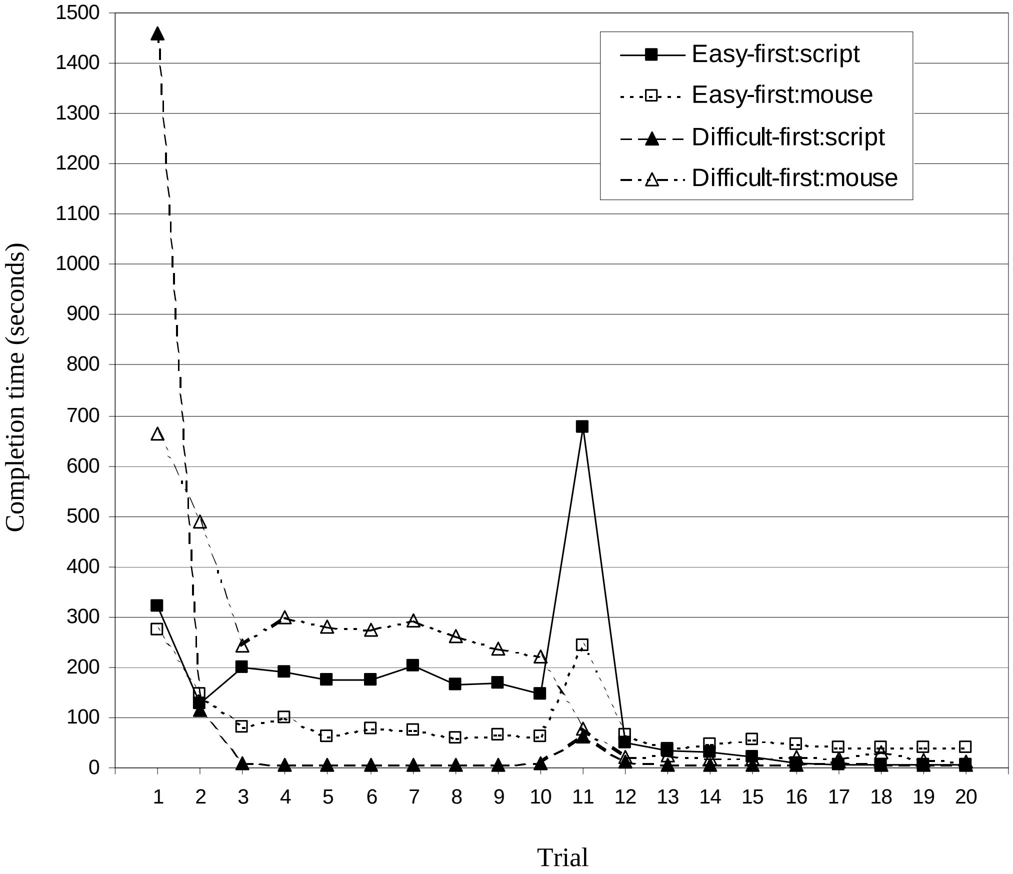
Yechiam, E., Erev, I., & Gopher, D. (2001). On the potential value and limitations of emphasis change and other exploration-enhancing training methods. *Journal of Experimental Psychology: Applied*, *7*, 277-285.

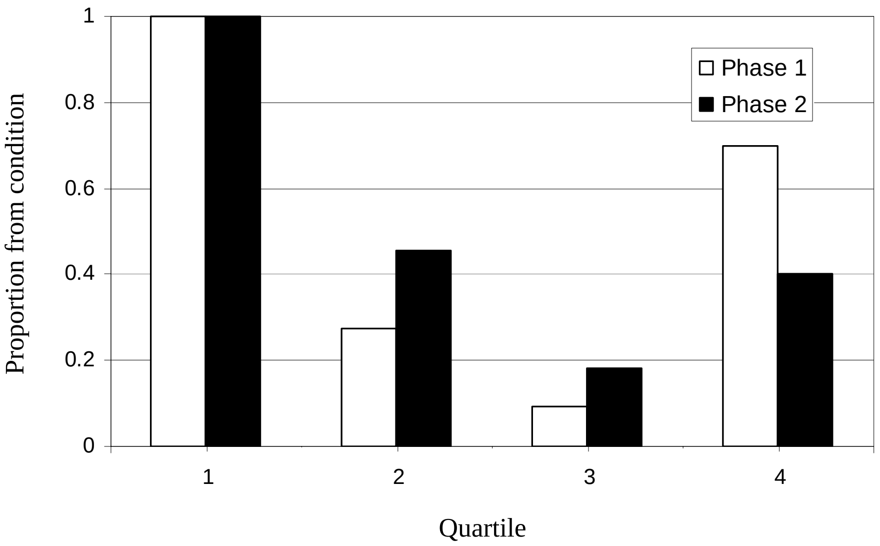


17	26	28	18	30	12	24	23	17	14	13	19	28	21	24	30	21	23	10	17	29	20	12	26	15
24	21	27	24	19	28	22	24	20	11	25	30	12	11	12	20	26	23	18	25	15	13	27	21	30
28	17	22	21	23	22	27	11	13	12	23	24	26	20	15	25	14	30	21	24	26	10	18	20	27
23	20	14	13	25	16	30	27	22	14	27	27	11	16	17	19	12	13	17	29	14	27	12	23	13
10	18	26	29	22	15	13	14	30	23	17	13	28	20	13	30	29	24	23	26	29	18	11	22	26
30	23	12	26	15	18	10	29	27	26	19	17	17	22	27	26	19	16	25	15	18	29	11	22	16
22	20	29	26	25	29	24	24	10	28	11	20	19	18	22	15	29	12	24	23	13	27	13	25	24
21	18	30	15	20	16	12	14	25	21	13	12	13	18	22	27	16	15	13	13	24	22	14	29	19
14	10	13	11	26	30	20	25	12	25	17	30	12	26	13	23	14	29	14	10	17	24	30	15	19
22	19	30	23	26	16	29	20	26	23	30	26	10	25	21	20	15	29	12	12	24	20	24	24	23
13	19	17	24	11	27	22	19	18	11	25	11	16	21	10	14	28	24	29	14	10	28	24	27	15
21	29	18	13	12	19	19	20	21	18	25	14	27	20	14	15	15	10	27	12	16	30	20	27	29
30	26	24	13	25	22	28	10	16	23	30	12	22	28	28	22	28	24	12	18	23	13	28	27	14
18	15	26	16	16	26	18	16	16	11	20	12	11	20	25	25	23	28	10	16	18	10	27	26	30
14	19	26	18	12	14	13	13	10	13	23	24	30	14	22	17	29	30	26	15	25	27	19	30	29
13	23	25	10	21	23	25	20	20	20	22	27	21	22	28	10	22	13	30	26	24	22	30	24	25
23	13	25	27	28	30	28	27	12	14		18	25	10	25	28	20	21	25	23	11	17	12	28	29
18	29	12	22	30	23	30	23	29	26	27	28	20	16	14	17	20	18	13	17	26	10	15	13	14
19	28	28	30	27	20	23	30	30	22	25	19	17	13	11	29	12	18	24	24	24	20	13	11	10
11	30	12	24	14	27	25	19	14	21	30	16	27	21	27	12	22	19	14	15	23	13	27	23	11
12	16	26	28	21	12	25	16	25	24	23	24	11	12	22	12	23	29	16	17	26	20	25	15	12
16	19	10	10	24	10	24	26	26	25	26	21	26	24	24	15	27	22	23	20	24	23	27	17	20
11	21	12	19	13	24	18	25	28	22	19	19	10	21	19	23	18	28	30	25	30	18	24	21	23
15	19	20	27	30	22	23	16	29	15	10	14	27	25	19	14	24	26	22	18	13	16	18	21	29
25	30	29	17	21	14	18	21	27	23	19	23	15	19	21	26	13	14	19	24	21	10	22	23	11
28	24	14	22	21	25	20	22	16	27	25	22	19	17	20	14	21	13	29	26	16	12	20	12	15
18	27	23	18	19	19	14	23	17	24	21	11	25	20	22	27	27	27	18	15	20	29	23	18	27

Task number

1


Completion time (seconds)
1500
1400
1300
1200
1100
1000
900
800
700
600
500
400
300
200
100
0
Easy-first:script
Easy-first:mouse
Difficult-first:script
Difficult-first:mouse
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Trial


1
0.8
0.6
0.4
0.2
0
Proportion from condition
☐ Phase 1
■ Phase 2
1
2
3
4
Quartile