Tom Lianza
Intro to HCI

Homework #3 – Models of Human Performance

Question 1

In order to determine the execution time for performing the delete task, we have to break the task up into its component parts. For each of these parts, different processes will take place within the user determining how long they will take. One would expect a significant time difference between the speed of a new user and the speed of a veteran user, so I have separated the two in order to get a better understanding of each.

For first time users we will apply the Uncertainty Principle because unfamiliarity with the interface will cause them to process much more data in order to get to what they're looking for. We know that the time it will take for the cognitive process is $T=I_CH$ where I_C is a constant and $H=\log_2(n+1)$. Given this information, if we assume that time it takes to process one option given only one option is 70msec (as given in the Model Human Processor) we can solve for the constant I_C: $70=I_C\log_2(2) \Rightarrow I_C=70$.

In order to calculate the impact distances have on selection, we apply Fitts's Law. We assume that I_M = 63 msec/bit, as determined using the averages provided in the Model Human Processor. Given this, we can apply the rest of the formula to determine how long it takes for movement: $T_{pos} = I_M\log_2(2D/S)$ where D is the distance between buttons and S is the size of the destination button. In cases where the movement is directly or nearly directly up and down (i.e. we lift the stylus straight up off the screen and then move it back down), we will assume the movement takes 140 ms (70 ms each way).

The Existing Method	
Repeat User	**First Time User**
1. Touch the item you want to delete – this is the first action for both users and the point at which we will start measuring execution time	
2. Select "details" $\tau_p + \tau_c + \tau_m$ $100+70+63*\log_2(2(37)/4) = 435ms$	2. Select on "details" $3\tau_p + \tau_c + \tau_m$ $3(100)+ 70*\log_2(4)+63* \log_2(2(37)/4) = 705ms$
3. Select "delete" $\tau_p + \tau_c + \tau_m$ $100+70+140 = 310ms$	3. Select "delete" $8\tau_p + \tau_c + \tau_m$ $8(100)+ 70*\log_2(9)+140 = 1161ms$
4. Select "okay" $\tau_p + \tau_c + \tau_m$ $100+70+140 = 310ms$	4. Select "okay" $3\tau_p + \tau_c + \tau_m$ $3(100)+ 70*\log_2(4)+140 = 580ms$
Total Time: 1055ms	Total Time: 2446ms

The Proposed Method	
Repeat User	**First Time User**
1. Touch the item you want to delete – this is the first action for both users and the point at which we will start measuring execution time	
2. Select "delete"	2. Select "delete"

$\tau_p + \tau_c + \tau_m$ 100+70+63*log$_2$(2(49)/4) = 460ms	$4\tau_p + \tau_c + \tau_m$ 4(100)+ 70*log$_2$(5)+ 63*log$_2$(2(49)/4) = 853ms
3. Select “okay” $\tau_p + \tau_c + \tau_m$ 100+70+140 = 310ms	3. Select “okay” $3\tau_p + \tau_c + \tau_m$ 3(100)+ 70*log$_2$(4)+140 = 580ms
Total Time: 770ms	Total Time: 1433ms
Percent Gain: 27.01%	**Percent Gain: 41.41%**

Clearly from the results, new users would reap the majority of the benefits. This makes sense because in eliminating objects that appear on the screen, the new users have dramatically less information to process. The effects are not as prominent in the regular users however, primarily because they were able to bypass processing a great deal of that information in knowing where they were going beforehand. The designer should not suggest the design to her manager because neither of the results exceeds 50%. One important factor to consider is the delay between clicking each button and having a menu appear. In this example, we assumed the time to be zero only because of a lack of data. If this time was great enough, it might push the results over 50% because removing a menu step would save an even more significant amount of time.

<u>Question 2</u>

1. If the screen were made brighter, the execution time would decrease. The Variable Perceptual Processor Rate Principle states that perception time is inversely related to

stimulus intensity. Making the screen brighter is an example of increasing the intensity, which would shorten the execution time.

2. Both auditory and visual images pass into the sensory image stores. When the intensity of a stimulus is increased, the magnitude is reflected in the physical codes of the stores. This does not have a direct effect on working memory (as it is not affected by the physical attributes of stimuli) however it does extend processing time because the sensory stores are being used by the music, taking up space and perceptual processing that would otherwise be used for the delete task.
3. The Power Law of Practice states that as the number of times a task is performed increases, the amount of time it takes to perform that task decreases. In using the Palm Pilot for a year, the user would have had a great deal of practice with it and their execution time would have decreased significantly from the first time they had used it. With the new interface, some of the experiences that had helped the user develop speed over the course of the year would no longer be applicable. The user would have to relearn and re-practice the process in order to take full potential of the efficiency benefits the new design might offer.
4. Fitts's Law gives us a method by which we can calculate processing time based on the distances between buttons and the sizes of buttons. It makes sense that a larger button would be easier to hit, and therefore require less processing to aim for it. Fitts's Law supports this theory, as one of the coefficients is (D/S). As S increases, the fraction shrinks and the processing time decreases.

<u>Question 3</u>

A complete "command-N" sequence will take 210ms: 70 to hit "command", 70 to hit "N", and 70 to release both fingers at once. Bearing this in mind, the first thing the user will have to do is process where his fingers will be placed before he can execute the sequences. This processing time will not be subtracted from the time it takes for a window to open, as we are still in the stages of preparation. The first keystroke the user makes is “command” and the second is “N.” It is at this point that we can start counting down until the window appears – 800ms. The user will release his fingers and then proceed with another iteration of the sequence.

After three more sequences (3 x 210 = 630), there will be only 100ms left before the window appears and 4 windows will have been queued. The user then presses the “command” key, but before he hits the “N” key the window appears. The user still hasn’t processed the window and so continues the sequence; calling yet another new window. The processor times given in the problem extend the time before he recognizes the window by 240ms ($\tau_p + \tau_c + \tau_m$). The user also needs 70ms (another τ_c) in order to process the fact that the window is appearing in error and that he should stop. This extra time gives a total of 1110 seconds from the first time he hit “n” to when he realizes he should stop. A total of six windows will appear, and at the time he realizes that he’s made the mistake his fingers will both be off the keyboard.