

The MiniCog and MiniCogWriter software programs and the MiniCog Rapid Assessment Battery (MRAB) were developed for and in the Kosslyn Laboratory at Harvard University, and are protected by copyright and US Patent Pending; please do not distribute. Please reference the following citations if you publish results from studies utilizing this software (for the MRAB tasks, cite both articles; for MiniCog and MiniCogWriter, but not the tasks, cite the second article).

Shephard, J. M., and Kosslyn, S. M. (2005). The MiniCog Rapid Assessment Battery: Developing a "blood pressure cuff for the mind." *Aviation, Space, and Environmental Medicine*, 76(6, Suppl), B192-B197.

Shephard, J. M., Kho, S., Chen, J., and Kosslyn, S. M. (2006). MiniCog: A method for administering psychological tests and experiments on a handheld personal digital assistant. *Behavior Research Methods*. 38(4), 648-655.

This research was supported by NASA cooperative agreement NCC 9-58 with the National Space Biomedical Research Institute.

A note about PDA models, memory issues, and updates

Due to lack of funds, ***the software has not been updated since 2006***. People often ask whether MiniCog runs on Palms other than the Tungsten C recommended below. I know that other users are running it on the Tungsten E2, but, due to that model's limited memory, it can accommodate only a subset of the MRAB tasks at any given time, so if you want to administer all the tests in one session, you should put them all on a memory card installed in the PDA. The tests cannot be run off the card. With PDAs, the expansion cards provide extra storage, but not extra "operating space". So, you can't "run" applications on the expansion card. install the MiniCog application on the device (not on the expansion card). Then, it will "see" the tests on the expansion card and *copy* them to the device itself for running. This will obviously be a problem by the time you get to the last test (or two or three) because there won't be enough memory. So, what is probably the best workaround is to run approximately half the tests, and then "delete" them from the device (but the originals should still be on the expansion card) and then run the second half (which will copy those down to the device).

I *believe* it will run on the Palm T|X, which has 128 M of memory, and the key layout seems correct, too, but this has not been tested. When I ran the program on a Lifedrive, the tests displayed correctly (even though designed for a 320 x 320 screen, rather than 320 x 480 as the T| X and Lifedrive have). (The Lifedrive is discontinued, plus has an unusual key layout, which is why I don't recommend it.)

Maybe one of the phones would work -- I note that the Centro and Treo 680 have 64 M of memory, 320 x 320 screens. The Treo 700p and 755p also run the Palm OS and have a lot of memory and the correct screen dimensions. (The phone screens have as high resolution, but are smaller overall than the PDA screens, so are not ideal. Sometimes the phones can be very cheap, although that probably depends on whether or not you sign up for cellphone service.)

I doubt that it will run on new Palm models (but this had not been tested). You can still buy the older models (used, usually) on ebay.com or amazon.com.

For information about running on Windows-Mobile platforms, please see the Appendix; and please note that this version of the software may be too buggy to be useful.

DOCUMENTATION

Jennifer Shephard,* Sam Kho, and Jolly Chen

*Contact with any questions:

Jennifer M. Shephard
 Division of Social Science - FAS
 University Hall 3 South
 Harvard University
 Cambridge MA 02138

1.617.495.7906

<http://isites.harvard.edu/socialsciencedivision>

jennifer_shephard@post.harvard.edu

1.0	Introduction	4
2.0	Scripting tasks: HTML syntax for MiniCogWriter	5
2.1	Table features	5
2.2	HTML source code	7
2.2.1	Demos and Trials	8
2.2.2	ShowTypes	8
2.2.3	Distractor and ShownBefore	9
3.0	Converting HTML files to PRC tasks using MiniCogWriter	11
4.0	The MiniCog application	13
4.1	Installing via hotsyncing	13
4.2	Running MiniCog	16
5.0	Viewing datafiles	21
5.1	SCORE files	22
5.2	USER files	23
6.0	Importing a distribution	25
7.0	The MRAB tests	26
	Screenshots	30
	Appendix A: Notes About Windows-Mobile Version	33

Parts highlighted in gray are recent (2008-09) modifications.

1.0 Introduction

This document describes software designed by Stephen M. Kosslyn and Jennifer M. Shephard of Harvard University and programmed by Sam Kho and Jolly Chen of Bay Area Software and cognitive tests created by Jennifer M. Shephard and Stephen M. Kosslyn with said software.

Currently the MiniCog software works with the Palm OS® personal digital assistants (PDAs). The MRAB tests run most successfully on Palm Tungsten Cs (other devices may not have enough memory or may have a different key layout). Information about Palm OS® devices can be found on this website: <http://www.palmone.com/>.

The software package consists of a tool for *compiling* psychological tasks on a desktop computer called **MiniCogWriter**, and a tool for *administering* the tasks on a Palm OS® PDA called **MiniCog.prc**, as well as a test battery, **MRAB**, for assessing current functioning in a number of cognitive domains.

Tasks are authored using an HTML editing tool (for instance, *Macromedia Dreamweaver*, *Microsoft FrontPage*, or *Netscape Composer*).

The following is a summary of the toolchain process:

1. Script a task using an HTML tool, producing a .html file.
2. Use *MiniCogWriter* to compile the .html file, producing a .prc file.
3. Use *MiniCog.prc* to administer the PRC task on a Palm OS® PDA.
4. Transfer datafiles back to the desktop computer for detailed analyses.

2.0 Scripting tasks: HTML syntax for *MiniCogWriter*

2.1 Table features

Source files for *MiniCogWriter* are HTML files with certain conventions. The main idea is to use a table to describe a task.

The screenshot shows a web browser window titled "MotorPerceptualReactionTime.html". The browser's developer tools are open, displaying the HTML source code. The code defines a table with 6 columns and 1 row. The columns are labeled: Text, Picture, Picture-240x320, Answer, Duration, and Delay. The table caption is "09-PMRT". The rendered table shows the following content:

Text	Picture	Picture-240x320	Answer	Duration	Delay
<p>You will see an oval appear in one of four locations (far left, left, right, or far right) at the bottom of the screen. You must respond by pressing the corresponding key (FL, L, R, or FR) as quickly as possible.</p> <p>For instance, the correct response to the example below is the R key. Please press the R key to continue.</p>			R	59700	250

The table must have six columns [NOTE that there are two picture columns, since one picture column contains pictures formatted for the Palm, at 320 x 320 pixels, and one column contains the pictures formatted for Windows-Mobile, at 240 x 320 pixels.] The first row must have the headings as shown: **Text**, **Picture**, **Picture-240x320**, **Answer**, **Duration**, **Delay**. The table caption represents the output filename.

Each row of the table is one screen on the PDA.

The **Text** column specifies what will be printed. Each 2.22 x 2.22 inches screen* on the Palm PDA can accommodate about 14 lines of text, averaging about 34 characters per line. This varies depending on the lengths of the words and the size of the characters ('m's are fatter than 'i's); aim for no more than 475 characters, and then see how it looks. You can specify the text color using standard HTML. The first screen of instructions above looks like this in HTML:

```
<td>
  <p>You will see an oval appear above one of the following four keys: FL,
    L, R, FR. You must respond by pressing the corresponding key as quickly
    as possible. </p>
  <p>&nbsp;</p>
  <p>For instance, the correct response to the example below is the R key.
    Please press the R key to continue.</p>
</td>
```

As of this release, the background is white.

Pictures are specified in the **Picture** [and **Picture-240x320**] column and will be displayed on the PDA with the Text drawn on top. (It is possible to display only text or only a picture, instead of both.)

Supported picture file formats are JPG, PNG, and GIF. (Note that some HTML scripting programs—for instance, *Frontpage* for the Mac—do not import PNG files.)

Programs like *Adobe Photoshop* or *Illustrator* or *Microsoft PowerPoint* can be used to create and edit picture files. (Note that *Illustrator* cannot save files in JPG, PNG, or GIF formats, so a graphic converter program—or even *Photoshop*—must be used to translate *Illustrator* or PDF files to the proper format.) If you use *Photoshop*, note that you must "flatten" the images or they will not look right on the PDA. You should also use "indexed color" for PNG files, and RGB color for JPG files. Pictures must be 320x320 pixels.

[*Note that not all PDAs have the same size screen. If the screens are *smaller*, e.g. on the Treo phone (about 1.75 inches) versus the Palm Tungsten C (2.2 inches), then the image *resolution* is higher (because the images are still 320 x 320 pixels). In general I think it is neither the size of the screen, nor the resolution, that is most important, but the visual angle occupied by the stimuli. We had our participants holding the PDAs fairly close to their eyes. A computer screen would be farther away, and lower resolution, but the images would appear bigger (higher and wider), and fill the same degree of visual angle. I think 320 x 320 pixels on a 72 or 98 ppi computer screen is fine if the User's eyes are about 12-18 inches from the screen. If your computer screen is capable of displaying at 144 ppi – or the User is sitting at quit a distance from the screen – then maybe it would be better to increase the image dimensions to 480 x 480 pixels or thereabouts.]

The **Answer** column specifies the correct answer for each trial (or, for the instructions, the correct key for proceeding); the valid choices are UP, DN, L, R, FL, FR, and NA. This coding was based on the classic key layout (where the keys progressed in a straight line across the bottom of the device from “far left” to “left” to “right” and “far right”, with the “up” and “down” keys in the middle). This key layout is recommended for administering the MRAB tests, but devices with other layouts can be used with MiniCog, although investigators may have to discover which coding maps to which key by trial and error.

The **Duration** column indicates how long to display the given screen. Since the MiniCog program will automatically proceed to the next trial after 60 seconds, time limits for each screen should not exceed 59720 milliseconds. (If two minutes pass without input from the user the MiniCog program will terminate altogether. Terminated runs will not be scored.) Note that this duration is maintained only in the absence of a response. Once a keypress is made, the screen clears. [These parameters were chosen due in part to limitations of a battery-powered device. On a regular computer you could give Users unlimited amounts of time to read instructions. However, you will have to decide how much time to allot for responses to stimuli, and also whether to score “runs” where long periods of time elapsed without User input. Although very long response times or periods without responses *could* be indicative of some type of cognitive problem, more usually it would be indicative of an interruption!]

The **Delay** column specifies the intertrial interval—in other words, the “blank space” between the stimulus (or instructions screen) presented in the current trial, and the presentation of the next stimulus (or instructions screen). If nothing is specified in this column, the default is 250 ms. [To a User 250 ms between trials can seem excruciatingly long. On many of our cognitive tests in the lab we have **no** intertrial intervals, or much shorter ones, 50 -100 ms -- unless waiting patiently is part of the task (as it is with attention and working memory tasks).] Responses (keypresses) made during this intertrial interval will be recorded (e.g., the timer starts running with the presentation of a given stimulus, and does not stop until either a response is made, or the next stimulus appears).

2.2 HTML source code

In addition to the table, the HTML source code is used to script aspects of a task. You need to "view source" or "view code", or in the case of Netscape Composer, specify "extra HTML" for "Table Properties".

You will see a line that looks something like this (for a task with straightforward responses and scoring):

```
<table border="1" width="600" cols="5" showtype="RANDOM" Demos="7" Trials="40">
```

or this (for an n-back task where users respond with one keypress for 'targets' and another for 'distractors'):

```
<table border="1" width="600" cols="5" ShowType="RANDOM" Demos="16"
Distractor="FL" ShownBefore="2" Trials="60" bgcolor="#FFFFFF">
```

Table border, width, and bgcolor are irrelevant (and may or may not appear automatically in your coding when you set up the table), and `cols="5"` will be automatic once the table is created.

2.2.1 Demos and Trials

Instructions and practice trials at the beginning of each script are classified as "Demos." Setting the Demos attribute, e.g.

```
Demos="7"
```

indicates that the first N rows (in this case, 7) in the table are instructions and/or practice trials, and are not to be scored. Note that for these demos (instructions and practice trials) the PDA will beep (if the sound is turned on) when a user presses the wrong key to proceed to the next screen. The PDA will not beep for wrong answers once the real trials have begun.

Setting the "Trials" attribute, e.g.

```
Trials="40"
```

tells the program how many non-Demo trials (e.g. "test" or "scored" trials) to present to the user and score (in this case, 40).

As of this release, a trial can consist of only one "screen," and therefore corresponds to a single row in the table.

However, the total number of trials does not have to correspond to the number of non-demo rows on the table. For instance, some MRAB tests (e.g., Perceptual-Reaction Time) have a limited number of stimuli (each occupying one row of the table) that are repeated over and over. Other MRAB tests (e.g., Three-Term Reasoning) have a large pool of stimuli, only some of which are used in any given administration.

2.2.2 ShowTypes

In order to have the flexibility of repeating stimuli without adding rows to the table, or using fewer stimuli in a given administration of a task than are represented in the table, we have devised the concept "ShowTypes".

By default, the trials will be shown sequentially, and the table rows are assumed to be in sequential order. (The "demos"—instructions and practice trials—will always be sequential.) However, a sequential order is not appropriate if the stimuli will be repeated

within a task—it will be too easy for the users to simply memorize the answers! And the point of including *more* stimuli (using more rows of the table) than are presented in a given administration of the task is that if stimuli are randomly selected each administration, there will be fewer repeats of a given stimulus *across* administrations (thus reducing the chances that users will memorize the stimuli themselves). For instance, if there are 64 rows of stimuli in the table, but the Trials attribute is set to 32, users will see only half of the stimuli each administration of the task, reducing the familiarity of any given stimulus and thus its memorizability.

ShowType="Random" will randomly select each trial from among all the non-demo rows of the table.

ShowType="RandomOnce" will randomly select each trial from among all the non-demo rows of the table, but will then remove that row from consideration (this allows for random presentation without any trials being repeated). Furthermore, if the "RandomOnce" feature is used, MiniCog will keep track of the trials shown in a given administration of the task, and remove those trials from consideration for the *next* administration to the same user, until all the trials have been shown once, at which point MiniCog begins again with the entire pool of stimuli.

If ShowType is omitted altogether, it's as if you typed ShowType="Sequential".

2.2.3 Distractor and ShownBefore

These are special attributes that allow MiniCog to administer and automatically score "n-back" tasks; for all other tasks these features may be omitted. In n-back tasks, there are target trials (those where the item is the same as the one displayed n trials previously) and distractors (those that are different from the n-back trials). Typically a user will be instructed to respond with one key to target trials and another key (or no key) to distractors.

For this type of task there is a correct response key (e.g., "FR") for target trials, and this is indicated in the "Answer" column on *every row* of the scored trials (because, depending on the order of presentation, any of these trials *could be* a target). The correct response to the distractor trials (whether a keypress or "NA") is indicated by the "Distractor" feature of the table properties, e.g.,

Distractor="FL"

In this case users must press the far left key if the item that appears is not the same as the one that appeared n-back (and the FR key if the item that appears *is* the same as the one that appeared n-back).

The value of ShownBefore is equal to 'n' (e.g., if it's a 2-back task, then ShownBefore="2").



The screenshot shows a web browser window titled "Spatial_2_back/SWM2.html*". The browser's code editor is open, showing the following HTML code:

```

1 <html>
2 <body>
3 <table BORDER COLS=5 WIDTH="800" showtype="RANDOM" Demos="16" Trials="60" Distractor="FL" ShownBefore="2">
4   <caption>
5   05-SWM
6   </caption>

```

The rendered page displays a table with the following content:

05-SWM				
Text	Picture	Answer	Duration	Delay
You will see the number 5 appearing in different locations on the screen. You must keep track of the locations and respond by pressing the FR key each time the number appears in the SAME LOCATION as it did TWO TRIALS before; otherwise press the FL key.				
Please press the FL key to continue.	 different  same as two back	FL	59000	250
For instance, each box below represents the				

The status bar at the bottom of the browser window shows "804 x 432 151K / 22 sec".

[Note this figure was taken from an older version of the program, for Palms only, thus there is only a single picture column.]

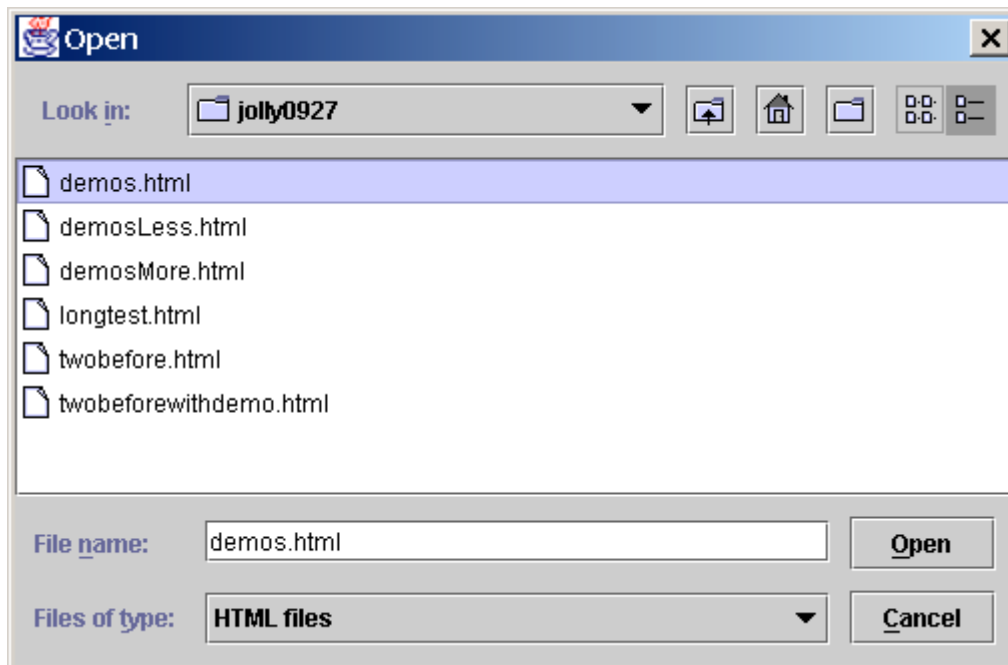
(The Demos—instruction and practice trials—for n-back tasks should be created in the same fashion as they would be for all other tasks. The ShownBefore and Distractor features apply only to scored trials, not Demos.)

3.0 Converting HTML scripts to PRC tasks using *MiniCogWriter*

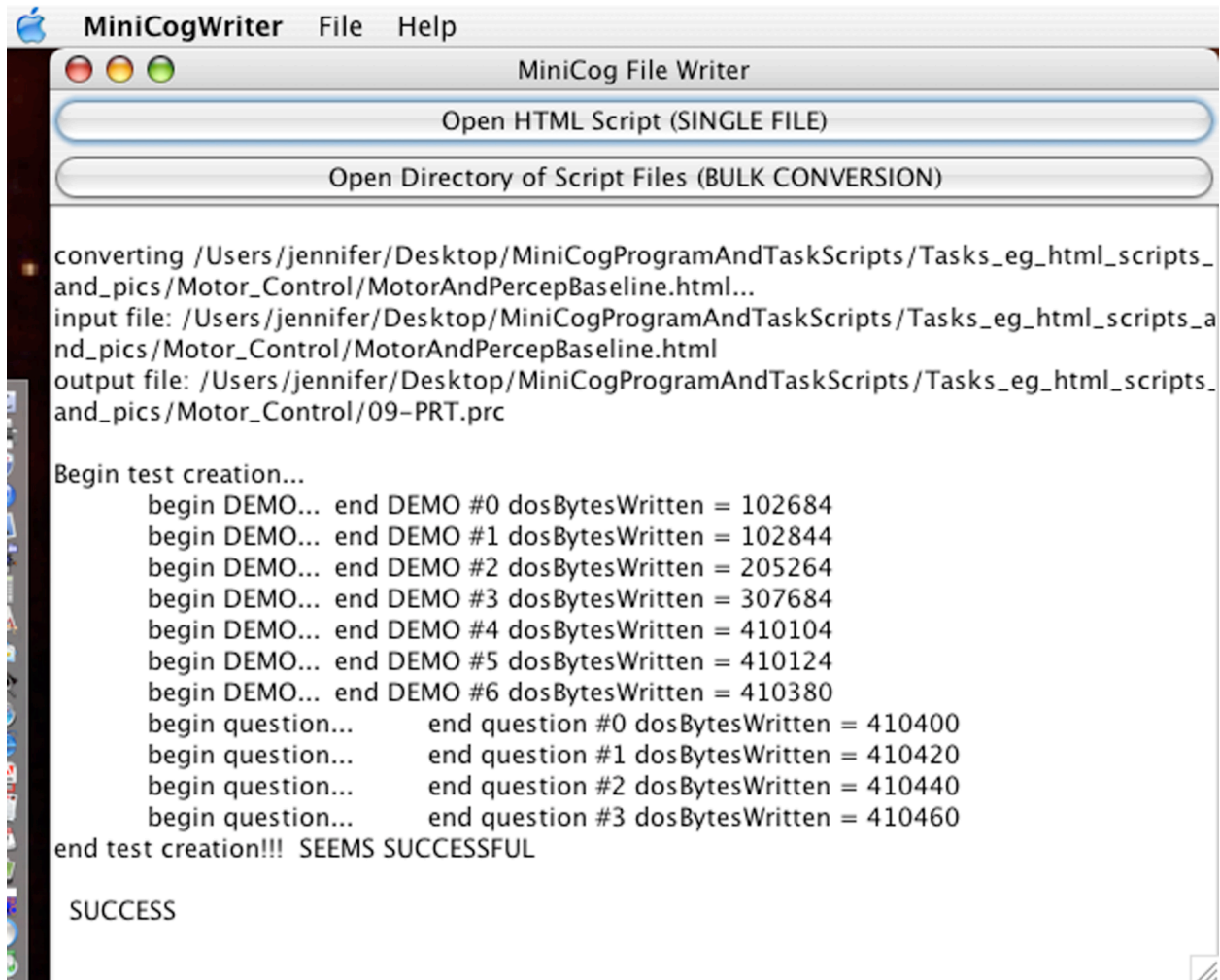


The tool for creating tasks is called **MiniCogWriter**, which takes as input an HTML file, and produces as output a PRC file. MiniCogWriter runs on any operating system featuring java, including Linux.

To convert HTML files to PRC format, simply open the file in MiniCogWriter; bulk processing is possible, but only if all the html files to be converted are in the same directory. [Note that MiniCogWriter also converts the HTML files to MCT (minicogtest) file compatible with the Windows-Mobile version of MiniCog.]



Selecting a file results in that source file being converted into a task file, ready for administration on a PDA. Status messages regarding the conversion are placed in the main window of *MiniCogWriter*.



The screenshot shows the MiniCogWriter application window with a menu bar (MiniCogWriter, File, Help) and a title bar (MiniCog File Writer). Below the title bar are two buttons: "Open HTML Script (SINGLE FILE)" and "Open Directory of Script Files (BULK CONVERSION)". The main window area displays a log of conversion activity:

```
converting /Users/jennifer/Desktop/MiniCogProgramAndTaskScripts/Tasks_eg_html_scripts_
and_pics/Motor_Control/MotorAndPercepBaseline.html...
input file: /Users/jennifer/Desktop/MiniCogProgramAndTaskScripts/Tasks_eg_html_scripts_a
nd_pics/Motor_Control/MotorAndPercepBaseline.html
output file: /Users/jennifer/Desktop/MiniCogProgramAndTaskScripts/Tasks_eg_html_scripts_
and_pics/Motor_Control/09-PRT.prc

Begin test creation...
  begin DEMO... end DEMO #0 dosBytesWritten = 102684
  begin DEMO... end DEMO #1 dosBytesWritten = 102844
  begin DEMO... end DEMO #2 dosBytesWritten = 205264
  begin DEMO... end DEMO #3 dosBytesWritten = 307684
  begin DEMO... end DEMO #4 dosBytesWritten = 410104
  begin DEMO... end DEMO #5 dosBytesWritten = 410124
  begin DEMO... end DEMO #6 dosBytesWritten = 410380
  begin question... end question #0 dosBytesWritten = 410400
  begin question... end question #1 dosBytesWritten = 410420
  begin question... end question #2 dosBytesWritten = 410440
  begin question... end question #3 dosBytesWritten = 410460
end test creation!!! SEEMS SUCCESSFUL

SUCCESS
```

4.0 The MiniCog application

The tool for administering tasks on a Palm OS PDA is *MiniCog*. It is a PRC file that must be hotsynced to the PDA or installed on an expansion card. It shows up as an icon in the Launcher screen.



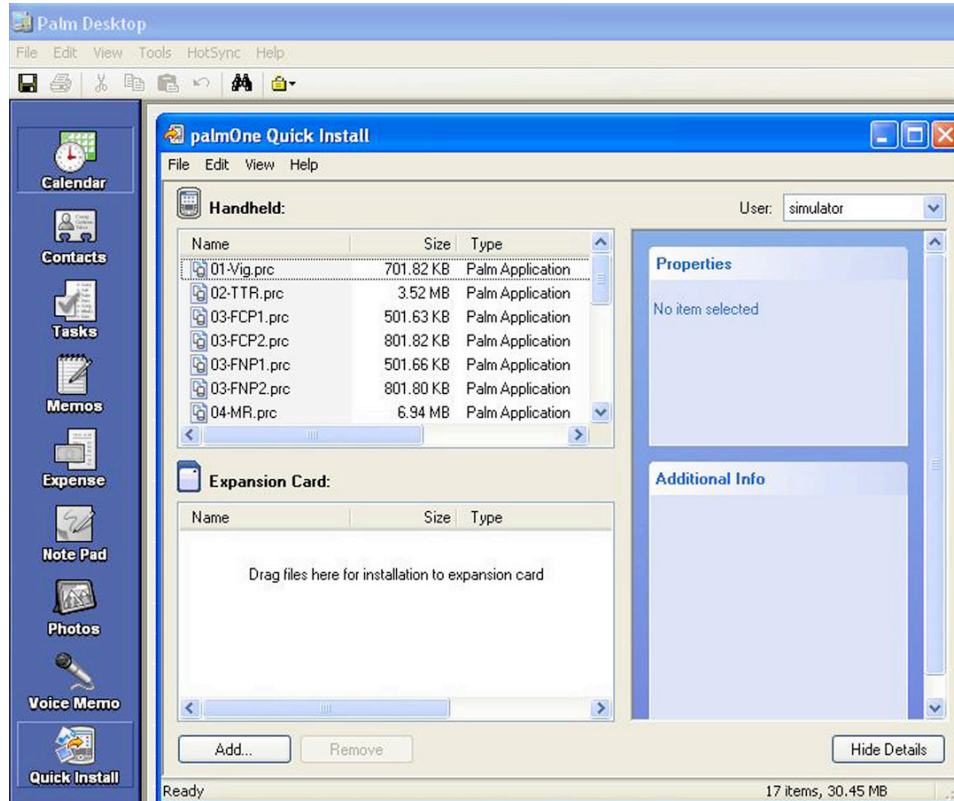
4.1 Installing via hotsyncing

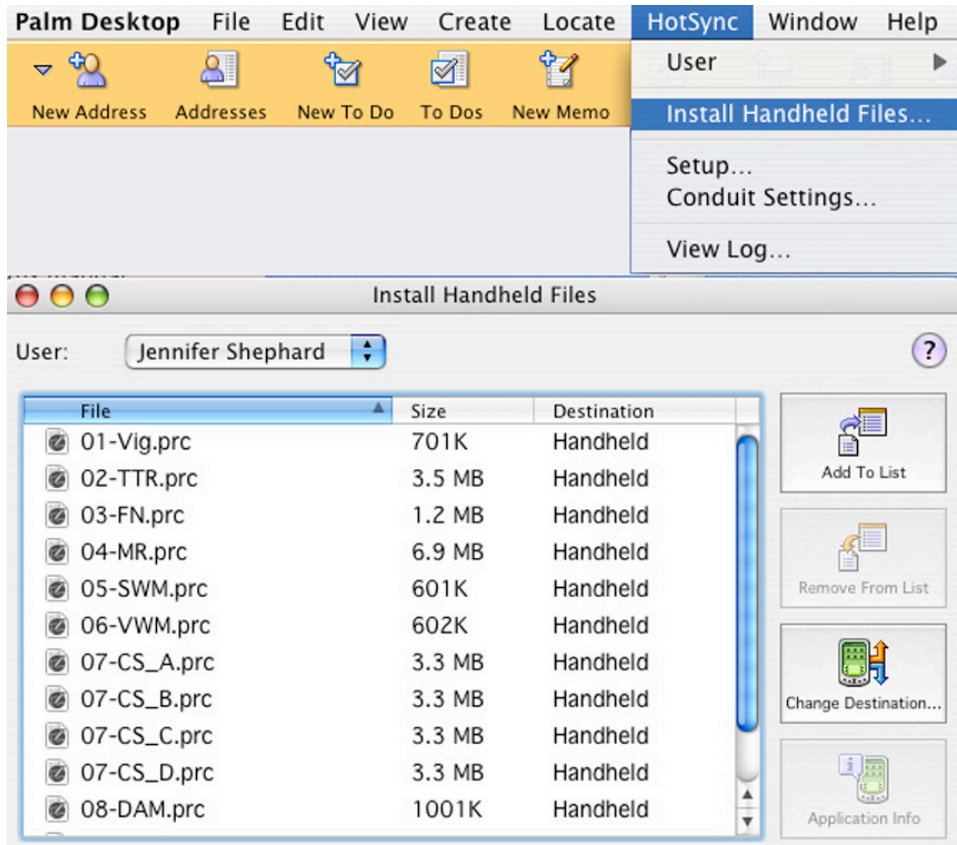
Along with the MiniCog PRC, the MRAB test PRCs, or your own task PRCs, must also be hotsynced (or installed on the expansion card). The task PRCs do not show up on the PDA as separate files, but the size of MiniCog in the Launcher info display will increase dramatically. MiniCog and the accompanying MRAB tests require about 30 Mb of memory on the PDA. [For notes on installation on Windows-Mobile PDAs, see Appendix.]

Name	Size
MiniCog_files_to_install_on_Palm_Oct2004	--
MiniCog.prc	44 KB
09-PRT.prc	404 KB
08-DAM.prc	1,004 KB
07-CS_D.prc	3.3 MB
07-CS_C.prc	3.3 MB
07-CS_B.prc	3.3 MB
07-CS_A.prc	3.3 MB
06-VWM.prc	604 KB
05-SWM.prc	604 KB
04-MR.prc	6.9 MB
03-FN.prc	1.1 MB
02-TTR.prc	3.5 MB
01-Vig.prc	704 KB

To hotsync *MiniCog* and *MRAB* to the PDA, use the *Quick Install* button in the *Palm Desktop* application on a Windows computer, or the "Install Handheld Files..." command under the "HotSync" menu in the *Palm Desktop* application on a Macintosh. One can usually shortcut this process by simply double-clicking on the PRC files (if you want to install many at once, select them all and then double-click or "open" them); for more information about installing files and hotsyncing, see the documentation that comes with your Palm OS® device.

Please note that some PDAs will lose their installed programs if the batteries are allowed to die (which can happen if the devices are not stored on their chargers or if they are used in very cold temperatures). To prevent data loss, MiniCog automatically stores the data on an expansion card if one is present. Moreover, the program and accompanying test files can be stored on the expansion card, and thus will not be lost if the battery dies. Please see the documentation that accompanies the expansion card for installation instructions (typically you will need a “card reader” also, and will use that to copy the relevant files into the “Launcher” folder on the card from your desktop computer, just as if the card were an ordinary disk).





4.2 Running MiniCog

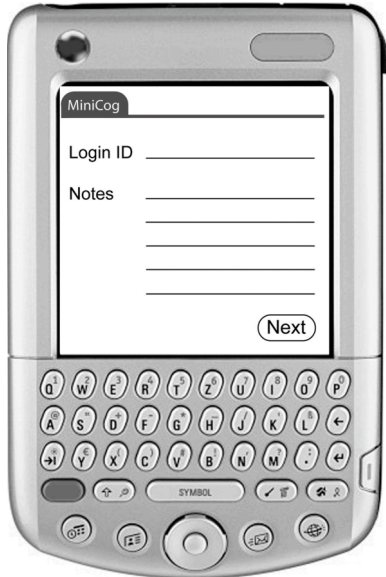
Users will need to know a very few basics to get started with MRAB or other tasks. First, if the PDA doesn't have a keyboard, they must learn enough graffiti (or how to use the on-screen keyboard) in order to enter their login IDs, notes, and some basic demographic information. They also need to memorize the key naming convention used in the instructions for each task.

For the MRAB tests, the key on the far left is referred to as FL; the key immediately left of the scroll keys is referred to as L; the "scroll up" and "scroll down" keys in the middle are referred to as UP and DN, respectively. The key to the immediate right of the scroll keys is referred to as R; and the far right-hand key is referred to as FR. Because of the nature of the instructions screens with these tasks, we recommend using a device (e.g., the Palm Tungsten C) that has the classic layout.

For scripting your own tasks, other layouts may be used; you must simply instruct the subject about which keys to press (another common layout is illustrated below).

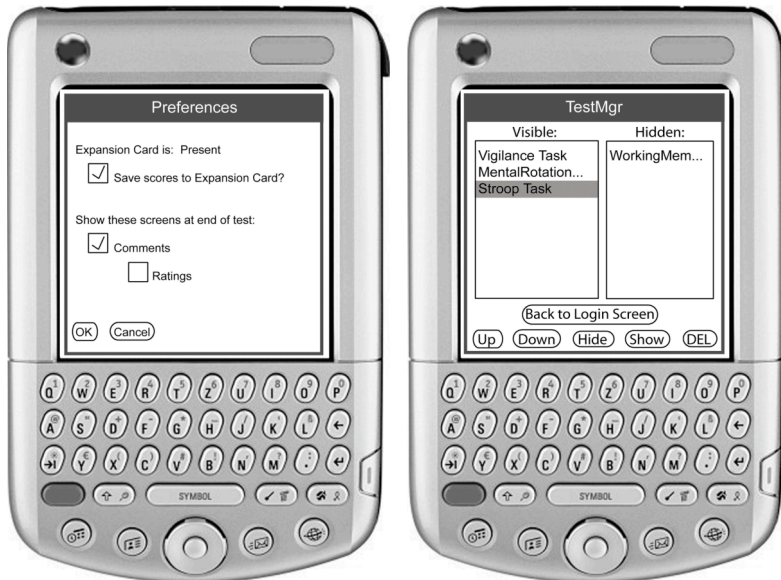


MiniCog starts with a **Login** screen:



The **Login ID** can contain up to ten characters. The **Notes** can contain up to 199 characters.

By clicking the “MiniCog” tab in the upper left of the Login screen, users with the passcode can set **Preferences** specifying whether data should be saved to an expansion card (if one is present) and whether test-takers will be asked to rate and comment on their own performance at the end of a task; they can also use the **Test Manager** to select specific tasks for administration and specify their order (if multiple tasks are installed). To delete tasks, you must first move them to the “Hidden” column.



If the Login ID is *new*, a **license agreement** that reads as follows will appear. The user must agree to the terms in order to be able to use the software. (If the user selects "decline," the program will quit.)

Agreement

By using MiniCog (the "Program"), you are bound to the terms of this Agreement. This software is licensed by the President and Fellows of Harvard College on behalf of Stephen Kosslyn and Jennifer Shephard and is protected by US copyright and a US patent pending. You may not copy, distribute, modify, or reverse engineer this Program. The Program is provided "as is" without any warranties of any kind. This software is NOT a diagnostic tool, does not constitute medical advice and should not be used for medical purposes or relied on as a guide for undertaking physical activities.

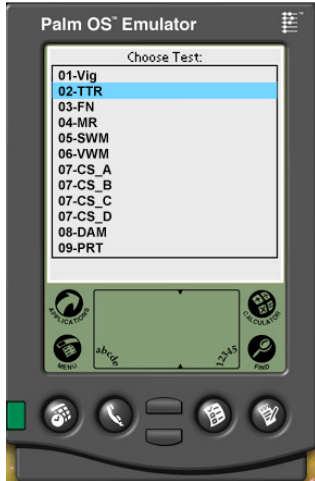
[Accept] [Decline]

Again, if the login ID is *new*, a **demographics** screen will follow the license agreement:



The **Year of Birth** and **Education** lines can contain up to ten characters (or as few as one character). **Handedness** and **Sex** are specified by tapping one of the boxes. (Ambidextrous individuals must pick one or the other.)

For a user to be considered the "same" (e.g. for purposes of saving to a specific scorefile), the user must use the same Login ID each time. If the same Login ID is used, the agreement and demographics screens are bypassed after the first login, and users proceed straight to the **list of tests**.



The list of tests created using *MiniCogWriter* and hotsynced to the PDA is displayed here (in numerical and alphabetical order unless the Test Manager was used to select a specific subset or display them in a different order). If you install more tasks than fit in the window, a scroll bar will appear on the right side of the list. For now, the stylus is used to make a selection.

Depending on the Preferences settings, users may be asked to rate and/or comment on their performance at the end of each test:

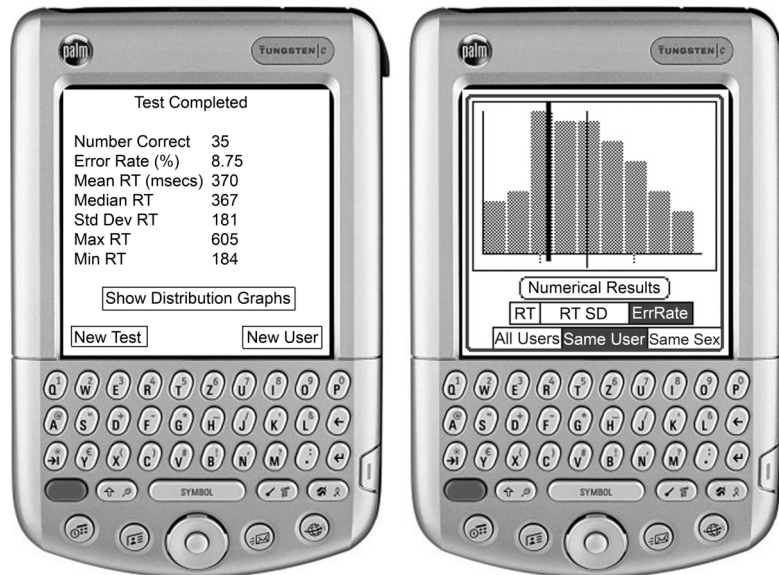
<p>Congratulations--you have completed this test! Please rate how you believe you performed according to the scale below:</p> <p>1 – Extremely poorly</p> <p>2 – Not very well</p> <p>3 – Average</p> <p>4 – Quite well</p> <p>5 – Excellently</p>	<p>If you have any comments about your performance, please note them here:</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>Next Test More Options</p>	<p>Back to List of Tests</p> <p>Logout</p> <p>Enter passcode: _____</p> <p>and View Results</p>
--	--	--

Users selecting “Next Test” on the Comments screen will proceed directly to the next task on the list (unless they are at the last test, in which case they receive a warning that they are at the end of the list and are asked whether they want to start over). Users selecting “More Options” on the Comments screen will proceed to the **Results Passcode** screen.

[NOTE: MiniCog saves the data from each task at the end of the task, and then again after the ratings/comments are entered. This leads to duplication in the –USER and –SCORE files; it’s easy to recognize the duplication and ignore the second set of data, but investigators should be aware of this bug (should be fixed in a later version).]

The passcode is hardwired in this release. It is 83202138. However, for now **we recommend that users not be given the passcode**, for two reasons. First, most users are participating in an experiment; knowing their results after each administration of a task may affect their performance (not to mention that it could cause anxiety, which might endanger their missions). Second, the histograms will be skewed and completely meaningless without normative data.

If the correct passcode is entered and the user selects, "View Results," the user may view results, both as summary text and as histograms.

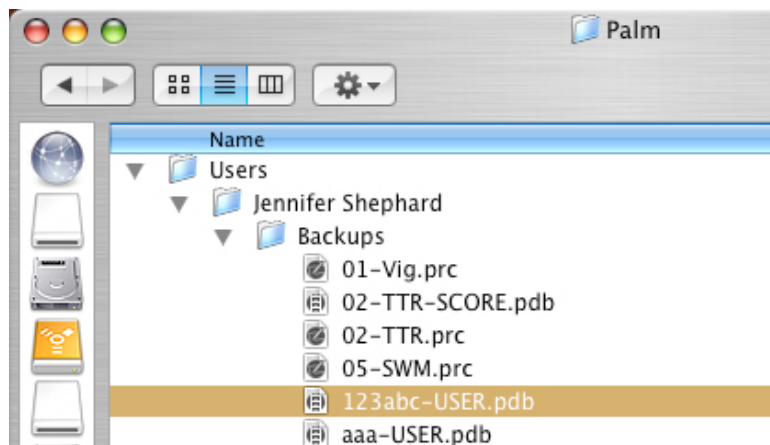


The two tick marks on the X-axis indicate a standard deviation (SD) from the mean, and the mean is represented by the tall thin line. The user's score is indicated by the tall thick black line.

"Normal" histograms will require many administrations of the task, and as no normative data are installed, the histograms generated may not appear "normal," and will probably not be useful at this time.

5.0 Viewing datafiles

MiniCog saves two types of datafiles: -USER files and -SCORE files. You will find these in your "Backups" folder after hotsyncing.



You can open these files in *Microsoft Excel*, but not by clicking on them (usually). You need to open *Excel* first, then use the "Open..." command under the "File" menu (making sure that the 'file...open' window shows "All Documents"). The text is "delimited," and the delimiter is a comma. We also distribute a program called *pdb2csv* (and other similar programs can be downloaded from the web) to convert the PDB files into comma-delimited text (CSV); this will remove the messy text that appears at the beginning of the tables below (e.g., 02-TTR-SCORE dÄ BÀ° B÷È X`TEXTREAD @ê† @-@ê† Æ02-TTR-SCOREæÒQùæÒQ).

The -SCORE files give you the summary scores for each user for each administration of that task, and the -USER files give the detailed trial-by-trial data for a given user for each administration of each task. The files will look like the tables below. (The ellipses [...] are there to indicate rows that have been deleted to keep the table a manageable size; there are no ellipses in the actual datafiles.)

5.1 SCORE files

This first table, below, is a –SCORE file and gives the summary results for each administration (regardless of user) of a hypothetical reasoning task.

LOGIN ID	YEAR OF BIRTH	EDUCATION	SEX	HANDEDNESS	TIME OF TEST	NUMBER CORRECT	AVERAGE RESPONSE TIME (MSECS)	ERROR RATE (%)	STANDARD DEV	TPUT (/sec)	RAT
aaa	1960	14	M	L	3/23/05 9:15	7	6331	12.5	1185	0.144	4
123abc	1967	17	M	L	3/23/05 13:05	6	5847	25.0	1099	0.130	3
123abc	1967	17	M	L	3/24/05 12:41	8	4674	0	2215	0.214	5

Note that **Average Response Time** is calculated based on *correct trials only*, and **Standard Deviation** is calculated (for correct trials) using the following formula:

$$\sqrt{\frac{n \sum x^2 - (\sum x)^2}{n^2}}$$

Throughput is calculated by dividing the number of trials answered correctly by the sum of the response times for all trials (correctly and incorrectly answered), and then multiplying by 1000 (to get the result in items/second). **[NOTE: There is a BUG in the throughput calculation, and this calculation CANNOT be trusted.]**

5.2 USER files

The second table, below, is a -USER file, and gives the trial-by-trial data for each administration of each task by a particular user ("123abc"), as well as specifying the session notes for each login, and the demographic information entered at the time of the first login.

The TEST NAME, TRIAL #, and TIMESTAMP are self-explanatory. The TRIAL INDEX refers to the row in the html table used to create the task. This could be important if you are interested in differences between individual stimuli and need to know which trial index refers to which stimulus. Please email minicog@wjh.harvard.edu if you need the html scripts for the MRAB tests. The INPUT is the user's response, and the EXPECTED ANSWER is the correct answer. ELAPSED TIME is the user's response time (accurate approximately to 10 ms).

123abc-USER	≤=‡?Ä	□□@aá@	aöü@aöü	ÊTEXTREAD	□p@	
ê	□	Ä				
ê	□	Ä	@			
ê	□	Ä	@			
ê	□	□	□	LOGIN ID	YEAR OF BIRTH	EDUCATION
123abc		1967	17	M	L	
SESSION NOTES	TIMESTAMP					
	3/23/05 13:04					
TEST NAME	TRIAL #	TIMESTAMP	TRIAL INDEX	INPUT	EXPECTED	ELAPSED TIME
02-TTR	0	3/23/05 13:04	25	FR	FL	3400
02-TTR	1	3/23/05 13:04	7	FL	FL	5390
02-TTR	2	3/23/05 13:05	3	FL	FL	4750
02-TTR	3	3/23/05 13:05	10	FR	FR	6950
02-TTR	4	3/23/05 13:05	13	FL	FL	6560
02-TTR	5	3/23/05 13:05	12	FL	FR	7750
02-TTR	6	3/23/05 13:05	20	FR	FR	6430
02-TTR	7	3/23/05 13:05	17	FL	FL	5000
TEST NAME	RATING	COMMENT				
02-TTR	3	I know I made a couple of mistakes				
TEST NAME	TRIAL #	TIMESTAMP	TRIAL INDEX	INPUT	EXPECTED	ELAPSED TIME
05-SWM	0	1/1/05 0:00	1	FL	FR	0
05-SWM	1	1/1/05 0:00	0	FL	FR	0
05-SWM	2	3/23/05 13:06	2	FL	FL	670
05-SWM	3	3/23/05 13:06	0	FR	FR	490
05-SWM	4	3/23/05 13:07	3	FL	FL	450
...
SESSION NOTES	TIMESTAMP					
2 cups of espresso!	3/24/05 12:40					
TEST NAME	TRIAL #	TIMESTAMP	TRIAL INDEX	INPUT	EXPECTED	ELAPSED TIME
02-TTR	0	3/24/05 12:40	8	FL	FL	4652
...
TEST NAME	RATING	COMMENT				
02-TTR	5	I felt I could concentrate better this time				

You can see by comparing the *input* with the *expected answer* for the first administration of the 02-TTR task that user 123abc made two mistakes (on Trial #0 and

#5). In addition, you can calculate that the average elapsed time for the correct trials is 5847 msec, and that the throughput is .130 (the number correct, e.g., 6, divided by the sum of all the response times, multiplied by 1000); these figures match the summary statistics reported for this user in the –SCORE file, above, for this task.

The next task (05-SWM) appears immediately below the first one (02-TTR) because the user selected "next test" at the end of the first task. If the user had quit and then logged in again, there'd be a new set of (or at least, space for) SESSION NOTES and a new TIMESTAMP for the next login. (And, indeed, this happens after the user finishes the 05-SWM task; see end of table.)

Note that the first two TIMESTAMPS for the 05-SWM task, as well as the first two ELAPSED TIMES for the same task, are incorrect. That is because these first two trials don't "count". The task is a two-back task, and since no stimuli appeared two back from the first and second stimuli, there is no basis for a comparison, and those first responses are discarded from the scoring algorithm. (This would be true for any n-back task.)

6.0 Importing a histogram from CSV

MiniCog has the capability to display histograms on a PDA. These can show an individual's performance relative to that of an aggregate population, or relative to his/her own past performance. Since it's not very useful to display histograms with just a few scores, you can create your own file of aggregate scores, taken from different users tested over time on different PDAs. Essentially you are creating a new -SCORE file for each task, containing the data not just from users of that particular PDA but from as many other users as you wish to include.

To set the aggregate statistics on a PDA, one would create the aggregate -SCORE data using *MS Excel*, outputting a CSV file, for instance 3TermReasoning-SCORE.csv (making sure to use *exactly the same* data format and 'task-SCORE' name as the MiniCog program generates automatically). You then must use a program that converts CSV files to PDB format, for instance *pdb2csv*. Upon the next *HotSync*, the aggregate statistics will have been set on the PDA.

Note: We are not releasing any normative data at this time.

7.0 The MRAB tests

Based on the literature in cognitive psychology and cognitive neuroscience, we devised an initial set of tests that tap key information processing. The tests themselves are modeled after corresponding tasks in the literature. For more information about the selection of the tests and current validation studies, please see *Shephard JM, Kosslyn SM. 2005. The MiniCog Rapid Assessment Battery: Developing a "blood pressure cuff for the mind". Aviation, Space, & Environmental Medicine, 76: B192-197.*

Unfortunately, our funding was cut several years early, and we were not able to complete most of the studies we had planned. We found some promising results as far as our fatigue study was concerned; Error Rates on the four MRAB tasks sensitive to variations in attention -- vigilance, filtering, divided attention, and perceptual motor reaction time -- were negatively correlated (in a range of $-.36$ to $-.50$, $p \leq .01$) with the amount of sleep participants received the previous night, and Response Times on the same four tasks were positively correlated (in a range of $.35$ to $.41$, $p \leq .01$) with participants' self-reported "stress level". As far as practice is concerned, we found that usually 8-9 practices with the entire task battery were enough to flatten out the performance curves (at least among Harvard students and other high IQ, high functioning, adult populations). We usually had subjects come in to practice the tests on 3 or 4 different days within a 2-week period. Usually they could get through the first two practices in about an hour, and then do three practices in each succeeding hour-long session. We didn't want them practicing more than one hour a day, or, say, two days in a row, if possible, because we didn't want them to get bored and stop trying.

Current tests assess attention (vigilance, divided attention, and filtering), perceptual reaction time, verbal and spatial working memory, verbal and spatial problem-solving (three-term reasoning and mental rotation), and cognitive set switching. The tests have been ordered within MiniCog so that there is a fairly easy test at the beginning, so that the hardest tests are not entirely clumped together, and so that the tests generally get easier toward the end of the order. We recommend that users practice the tests a number of times (8-12) before taking them under stressful conditions. First time administration (including the time to learn how to use the PDA and reading the instructions carefully) may take up to half an hour [for Harvard students, who are a young, highly intelligent, very computer literate, and highly educated population; results may vary with other populations]. Once the user is familiar with the device and tests, completion of the battery may require only about 15 minutes. A description follows.

Vigilance (01-Vig.prc)

Vigilance is the ability to concentrate and wait for a specific event over a sustained period of time. A series of small geometric shapes is presented one at a time on the screen. The set of shapes contains one target (a particular parallelogram) and four distractors. Each stimulus stays on the screen for 500ms, followed by an intertrial interval of 1, 2, or 3 seconds (responses can be made during the intertrial interval). There are 90 trials presented over approximately 3 minutes and 45 seconds; with instructions this test may take about four and a half minutes to complete. [MiniCog

presents these trials randomly, so in a typical run the User will see the target on about 1/5th of the trials.]

Three-Term Reasoning (02-TTR.prc)

The classic three-term series reasoning problems, such as "Kate is not as young as Sam; Kate is less old than Thomas—True or false: Sam is not oldest," are administered; test-takers indicate their responses by pressing one button for "true" and another for "false". There are only eight trials, since these problems are fairly hard. Users are allotted up to 45 seconds to respond to each trial [the average RT is typically quite a bit less than this, maybe 15-20 seconds; however, this is very variable depending on the subject population. Thirty-two different stimuli are included in the html script. MiniCog uses the "RandomOnce" feature to randomly select each of the eight trials for a given User from among all the non-demo rows of the table, and then removes that row from consideration; the program keeps track of the trials shown in a given administration of the task, and removes those trials from consideration for the *next* administration to the same User, until all the trials have been shown once, at which point MiniCog begins again with the entire pool of stimuli.]

Divided Attention (03-DA.prc)

Divided attention requires one to focus on two unrelated types of stimuli or stimulus features. Users see a series of geometric shapes (circles, triangles, squares, and stars) in different colors (white, black, light-grey, and dark-grey) presented on the screen, one at a time. They must press one button if the shape is a triangle *or* is black and another button if the shape is a circle *or* is white (no triangles are white and no circles are black, to prevent response competition). There are 40 trials in this test. Users are allotted up to 10 seconds to respond to each trial (average RT is usually much less).

Mental Rotation (04-MR.prc)

Pairs of Shepard-Metzler-like stimuli are presented, with one member of the pair rotated relative to the other. Test-takers are required to try to "mentally align" one form so that it is at the same orientation as the other, and then to decide whether the two are identical or mirror-images. There are 32 trials in this test. [As with the TTR task, the "RandomOnce" feature is used to select from a pool of 64 trials for each administration.] Users are allotted up to 20 seconds to respond to each trial (average RT is much less).

Spatial Working Memory (05-SWM.prc)

Information often must be "held in mind" as one manipulates it in some fashion, which requires working memory; a classic working memory task requires participants to see a series of stimuli and respond when a stimulus appears that had also appeared two (or, in more challenging versions, three) trials earlier in the sequence (this is called the "n-back" test). We have implemented two working memory tests: a spatial 2-back test that requires the test-taker to keep track of a sequence of locations, and a verbal 2-back test that requires the test-taker to keep track of a sequence of digits.

In the spatial test users see a digit ('5') appear in one of four locations on the screen on each trial; they must indicate whether the location is the same as the location of the digit two trials earlier in the sequence. There are 60 trials in this task; each

stimulus stays on the screen for up to one second (it goes away if the user responds within that one second period), with an intertrial interval of one second.

Verbal Working Memory (06-VWM.prc)

In this test users see one of four digits ('1', '2', '3', or '4') in the center of the screen in each trial; they must indicate whether the digit is the same as the one presented two trials earlier in the sequence. There are 60 trials in this task; each stimulus stays on the screen for up to one second (it goes away if the user responds within that one second period), with an intertrial interval of one second.

Cognitive Set Switching (07-CS_A.prc, 07-CS_B.prc, 07-CS_C.prc, 07-CS_D.prc)

We have implemented a simplified version of the Odd-Man-Out task, which on each trial presents a series of four letters (e.g., T H t H), and users must decide which letter does not belong in the series (in this case, the lowercase 't'). There are two stimulus criteria (letter identity and case), only one of which applies to any given trial. Two, three, four or five (randomly determined) trials in a row have the same criterion, and then the criterion switches. Users will form a mental set based on each series of trials they receive, and thus one should find increases in RT, and possibly decline in accuracy, when the criterion is shifted. [This test was never implemented for Windows-Mobile, so the html scripts are in the old Palm-only format (a 5-column html table, rather than 6-column). This differs from the classic cognitive set switching tasks like the true Odd-Man-Out Test and Wisconsin Card Sort in that it does *not* have a learning component to it. That is, there is always a clear "answer" as to which character is the odd one out, so Users don't have to learn the sorting criterion based on feedback. However, they do implicitly learn to search for the feature that marks the odd character, either its identity (*H* or *T*) or its case (capital or lower). Having the right criteria in mind leads to faster identification, and throws one off when the criteria shifts.]

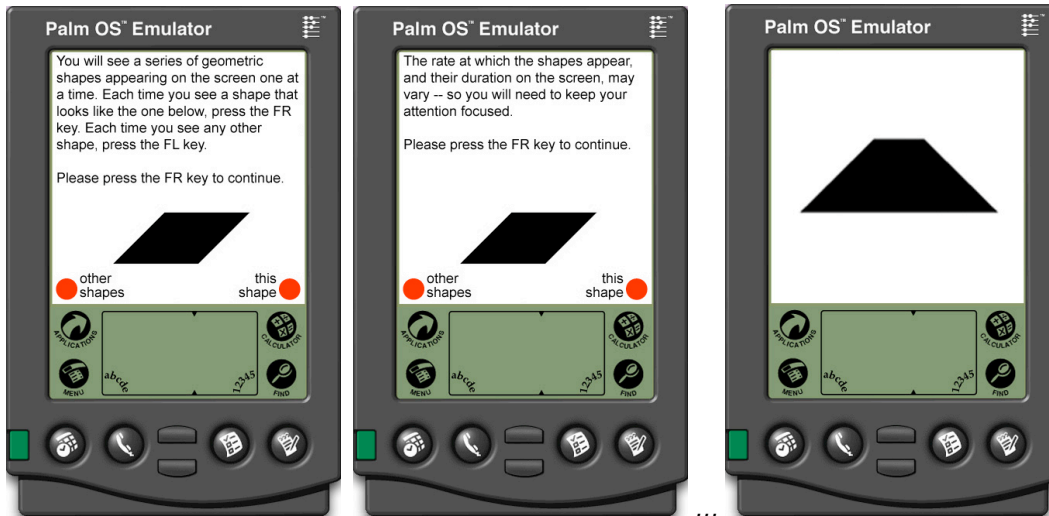
This is the only test currently implemented that requires a sequential order of presentation. This is necessary in order to manipulate the number of occurrences of a certain trial type in a row. Because of the sequential order to the presentation, it is theoretically possible that a user could memorize the stimuli and correct answers. Therefore, we have implemented four versions of the tests, and request that users alternate among the different versions each time they take the test. This means that the investigator should make sure that the versions the user is not supposed to be completing in any given administration are "hidden" – otherwise, when the user does one version, say "A", and presses "next test" at the end, he will proceed to version "B" and so on, which is no good. Each versions consists of 24 trials. Users are allotted up to 20 seconds to respond to each trial (average RT is much less). [If your software (for example, E-Prime, is capable, I recommend programming a "true" cognitive set switching task, with multiple sorting options per stimulus and feedback to assess learning (e.g., how many trials with feedback does it take for the User to learn the sort criteria, and when the criteria is changed, how long--how many trials and how much time per trial--does it take for them to realize this).]

Filtering (08-FN.prc)

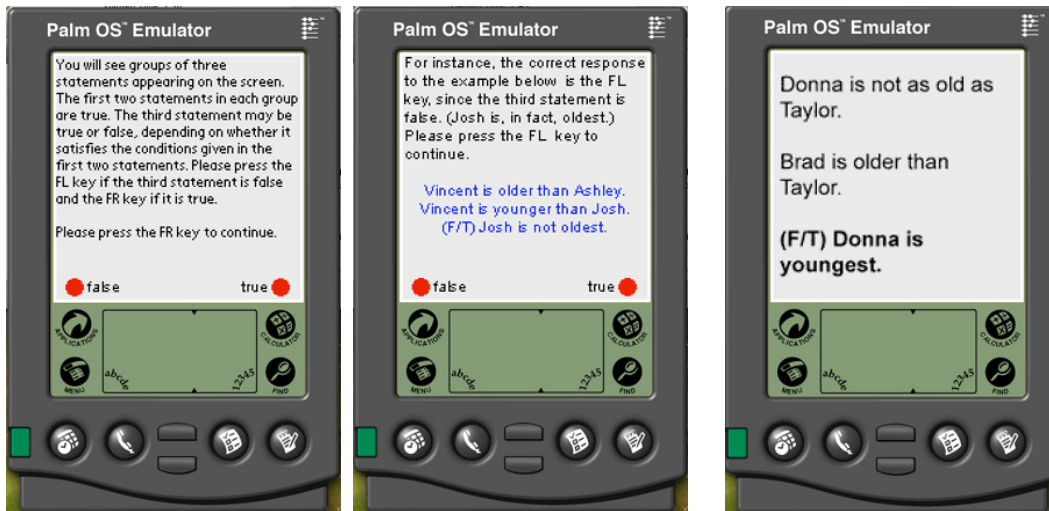
Filtering is the ability to focus on what is important and ignore what is irrelevant to the task at hand. Groups of digits are presented on the screen, and users must indicate the number of digits present, regardless of what the digits are. In some trials the number and meaning of the digits are congruent (e.g., "4 4 4 4"); in other trials they are not congruent (e.g., "4 4 4 4"). There are 84 trials in this test. Users are allotted up to 10 seconds to respond to each trial (average RT is much less). [MiniCog presents these trials randomly, so on average half the trials will be congruent and half will be incongruent. Given a more flexible program I would design this so that *exactly* half the trials were congruent and half non-congruent. We do have a classic "Stroop" color filtering task, too, but abandoned it due to colorblindness and color interference (in the environment) issues. I can send the files, if you want them.]

Perceptual-Motor Reaction Time (09-PMRT.prc)

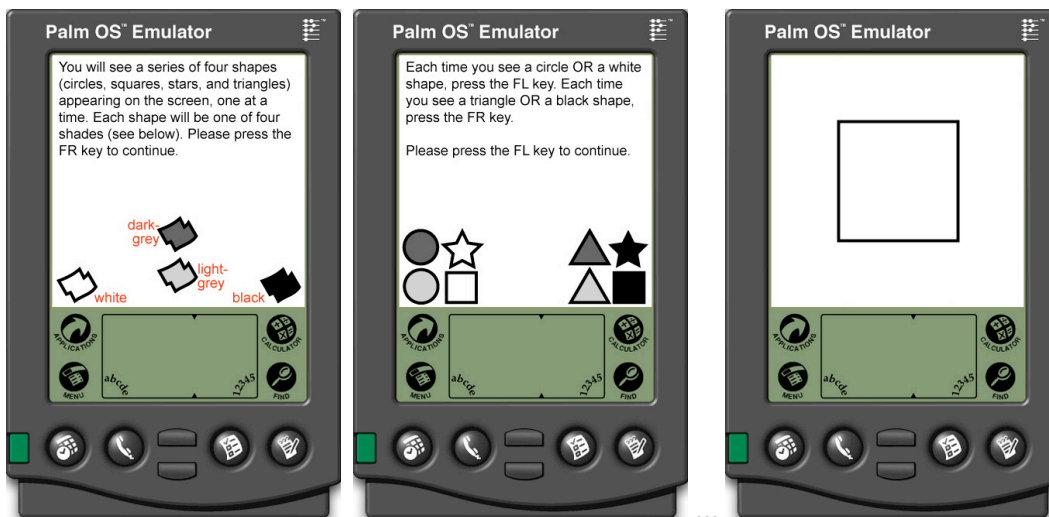
It's important to discover whether changes in response times or errors on the tests may be due simply to a slower motor response rather than cognitive deficits. For instance, hunger, cold, or gloves could affect the ability of a user to move his/her fingers quickly and accurately across the rather small and closely spaced keys on the PDA. Therefore we have developed a test where a stimulus (a small oval) appears above randomly selected keys on the PDA. Users must respond by pressing the corresponding key as quickly as possible. There are 40 trials in this test. Users are allotted up to 5 seconds to respond to each trial (average RT is much less).



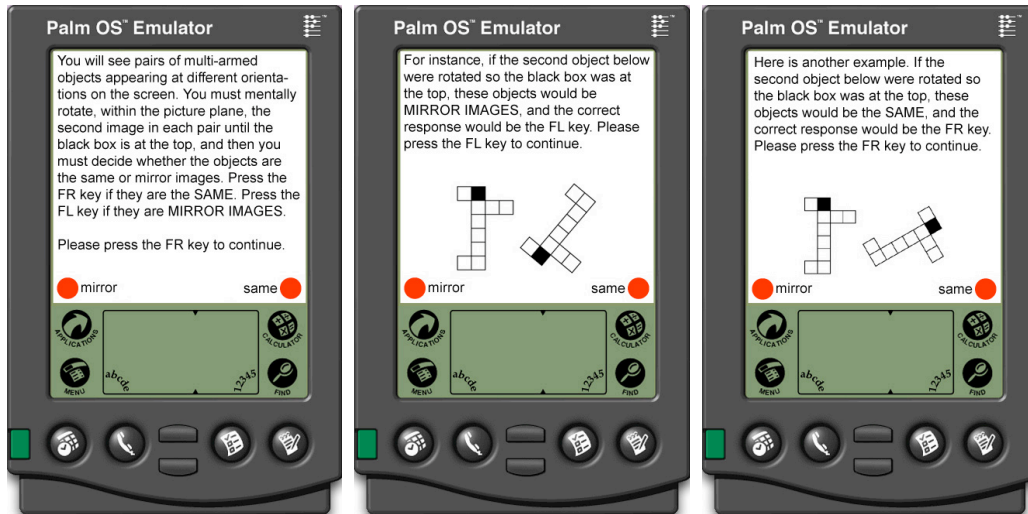
Screenshots of Vigilance test (01-Vig)



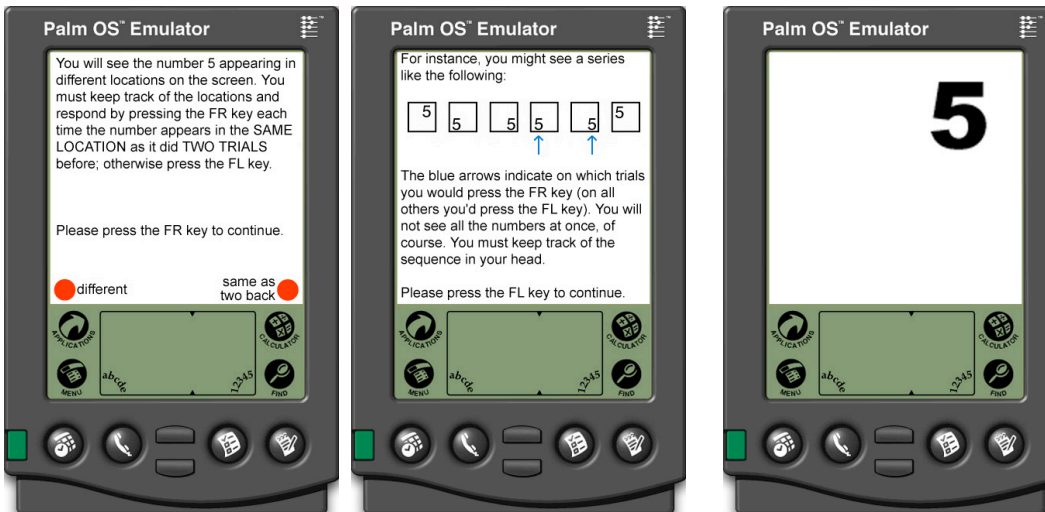
Screenshots of Three-Term Reasoning test (02-TTR)



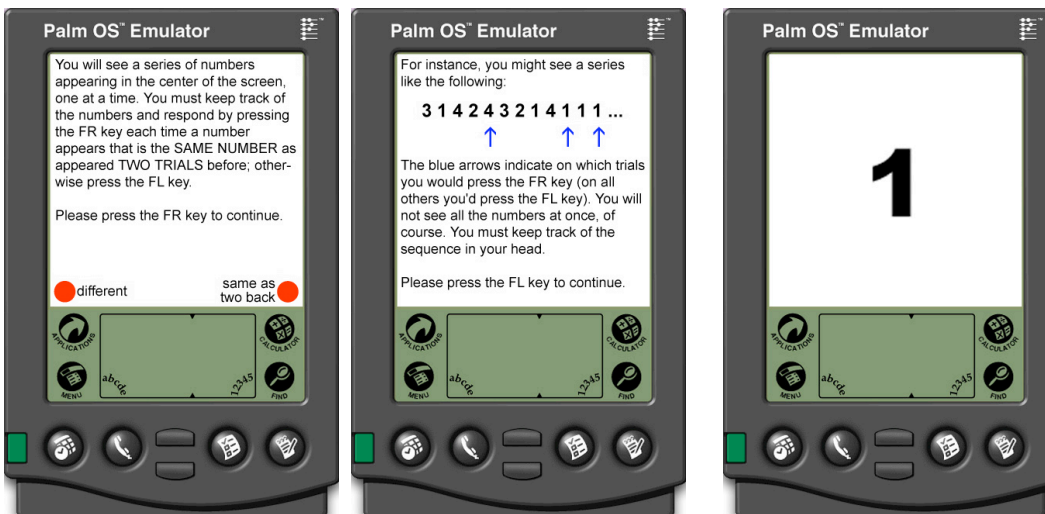
Screenshots of Divided Attention test (03-DA)



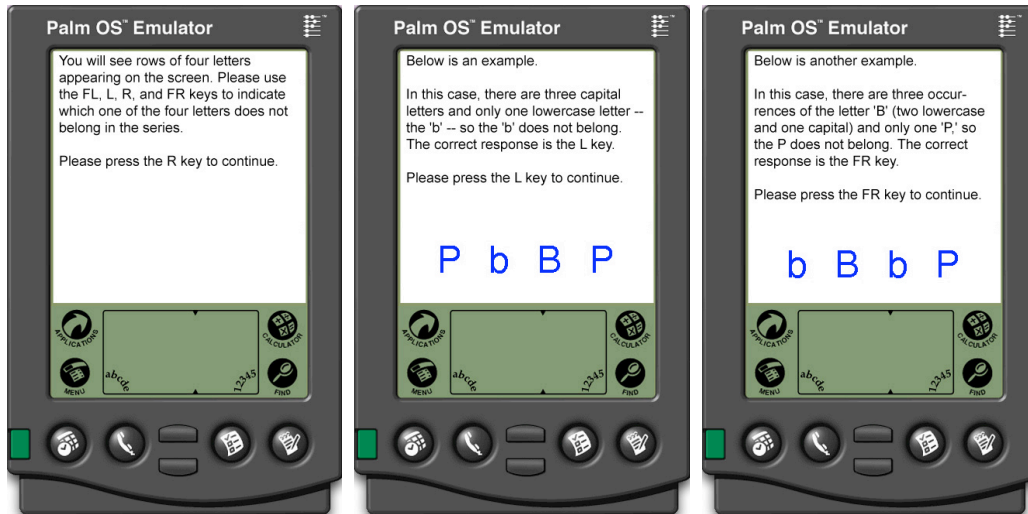
Screenshots of Spatial Problem-Solving (Mental Rotation) test (04-MR)



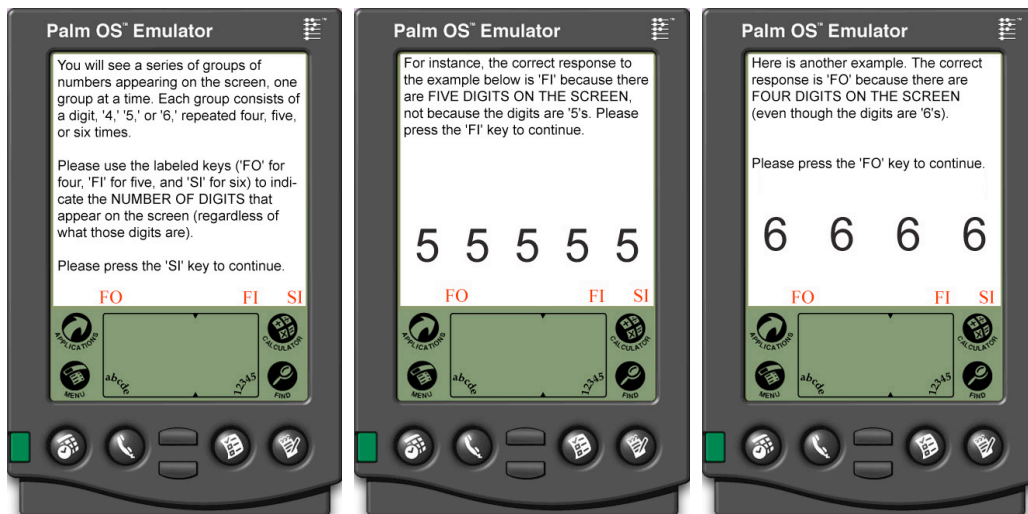
Screenshots of Spatial Working Memory test (05-SWM)



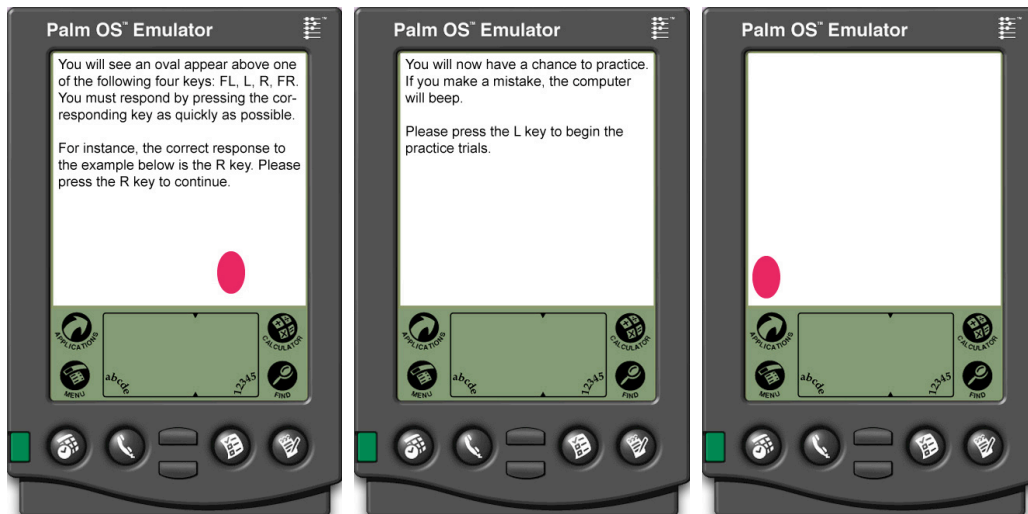
Screenshots of Verbal Working Memory test (06-VWM)



Screenshots of Cognitive Set Switching test (07-CS_X)



Screenshots of Filtering test (08-FN)



Screenshots of Perceptual Reaction Time test (09-PRT)

Appendix A

Notes about Windows-Mobile Version

Bugs and Known Problems

The Windows-Mobile version is still an 'alpha' version. It may be too buggy to be useful for most purposes. It received limited testing on the Dell Axim x51v and HP iPaq hx2790b. Below is a summary of some **known problems** with the Windows version:

1. The license agreement is not displayed to Users, but they are still bound by it. The PI of the research project is responsible for his/her users.
2. The "ratings" and "comments" are not recording properly, so you should change the "Options" (viewable from the Login screen) so that Users are not asked to rate or comment on their performance.
3. Memory issues: At the end of each task, an "end of task" screen will appear (or a ratings and/or comments screen will appear, if those options are selected, which they should not be, per note above). During this time, the program is saving the task data into memory. If the "next test" button at the bottom of these screens is pressed too soon after the last trial of any of the tasks, the program *may* give an error message (try counting 5 mississippis to avoid pressing it too soon). Likewise, if users are just pressing buttons as fast as they can at the end of the task (i.e., not actually doing the task), the program may give them an error message. If you get a chance to play around with the tasks yourselves before giving them to users you can try different things and see what happens; maybe you won't get any error messages!

The error message will be something like "OutOfMemory Exception". There will be a little "OK" button on the message window, and sometimes when one clicks it, one simply proceeds to the next test on the list (skipping the one that was trying to load while data from the previous task was trying to save) or goes back to the list of tests. However, other times, the device just crashes, and you need to use the stylus to do a soft reset, which naturally quits one out of MiniCog (and loses data).

4. The automatic data-scoring **does not work!** You will need to calculate your subjects' scores based on their User files.
5. The passcode is required in order for users to see their results, BUT it should not be shared with any such users, because the results that are displayed are often WRONG! (See #3; the automatic scoring is not working right.)

Location Of Minicog/Tasks On PDA

Use your normal installation method (active-sync, bluetooth, whatever) for transferring the files to a meaningfully named folder on the PDA (I call my folder “minicogfiles”); this will be in your “my documents” folder by default. (Note, I always use Bluetooth, since the other installation methods didn’t work for me, given that my desktop computer is a Mac.)

If you want the data to be stored to the SD (memory) card, copy the folder containing the MiniCog files and paste it on your card; you must *run* this version (or a shortcut to it) in order for the data to be stored on the card.

Copy MiniCog itself (the version that is on the card) and “paste [a] shortcut” somewhere that is easy for the users to find. You can rename the shortcut so it’s easier to read (the default “Shortcut to MiniCog” doesn’t fit on my PDA screen, so I renamed it “MiniCogShortcut” – you probably do not want to rename it just MiniCog, lest you (or the PDA) get confused about which is the real program). (I “copied” the program and then “pasted [the] shortcut” at the highest level directory (“my device”) that I could view in File Explorer.... Not sure if there is a way to make a shortcut that appears on the PDA ‘desktop’ because I am not that familiar with the Windows Mobile operating system myself.)

Running MiniCog

When you open MiniCog (either the version on the memory card, or the shortcut to that version [remember, the version on the device itself will save the data, but will not save it to the memory card]), at the bottom of the Login screen click on the Options and Tests tabs (and enter administrator password* “83202138”) to set:

- a. [Options tab] Whether or not users are asked to ‘rate’ and ‘comment’ on their performance at the end of each task (checkboxes). Once these options are set, they apply to all tasks for all administrations until the options are changed again. **Make sure these are unchecked** (see note above; these options don’t record properly).
- b. [Tests tab] Which tasks to display and in what order.
To order the tasks, select one (they should all be in the ‘visible’ column, but if not, “show” them), and press the U or D button on the screen to move it up or down the list.

Exiting the MiniCog Program

When users reach the end of the last task on the list and press the “next test” button, they will be asked if they want to start over. Users should be able to exit MiniCog by pressing the stylus to the screen and clicking on the resulting “exit” window that comes up; this will bring them back to the list of tasks, where they can click the “OK” button at the very top right of the screen, and this will quit MiniCog. They should only do this once they have completed all the tasks on the list on gone back to the beginning.

Memory Notes

1. Allocate as much of your PDA's memory to running programs as possible. This has nothing to do with the memory card – I am referring to the built-in memory, which is the only kind that can run programs; the cards are just for data storage (like the difference between RAM and ROM on a regular computer).
2. On a related note, you want to make sure that you quit all other programs (except the File Explorer, if you need it to find MiniCog) before running MiniCog.