

RatBat

A Whisker client

by Rudolf Cardinal

www.whiskercontrol.com

Copyright (C) Cambridge University Technical Services Ltd.

Distributed by Campden Instruments Ltd (www.campden-inst.com)



RatBat

© Cambridge University Technical Services Ltd

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: March 2024 in Cambridge, UK

Creator (Whisker)

Rudolf N. Cardinal

Design and Programming (Whisker)

Rudolf N. Cardinal

Michael R. F. Aitken

Legal Advisor (CUTS)

Adjoa D. Tamakloe

Sales (Campden)

Julie Gill

Contacting the authors:

For information about Whisker, visit <http://www.whiskercontrol.com/>.

If you have sales enquiries about Whisker, contact Campden Instruments Ltd at <http://www.campden-inst.com/>.

If you have comments or technical enquiries that cannot be answered by the sales team, contact the authors:

Rudolf Cardinal (rudolf@pobox.com)

Mike Aitken (m.aitken@psychol.cam.ac.uk)

Table of Contents

Foreword	1
Part I RatBat	2
1 About RatBat	2
2 Required devices	3
3 Using RatBat	5
4 Configuring RatBat (general settings)	7
General parameters	10
5 The visual object library	12
Defining components of objects	15
Size, coordinates, and grids	16
Arc	18
Bezier spline	19
Bitmap	21
Chord	21
Ellipse	22
Line	22
Pie	23
Polygon	23
Rectangle	24
Rounded rectangle	24
Text	25
Pen options	26
Brush options	26
6 Configuring individual tasks	27
Reinforcement Familiarization	27
Touch Training	28
Reversal Learning	31
Delayed Matching/Non-matching to Sample	34
Paired-Associates Learning	37
Simple Schedules of Reinforcement	43
Notes on reinforcement timing	46
N-Pair Concurrent Visual Discrimination	46
N-Pair Concurrent Auditory Discrimination	49
Audiovisual ID/ED	51
About set-shifting	54
7 Before you start the task	56
8 Results	56
Text-based results file	56
Creating a new ODBC source	63
Using the Microsoft Access database for RatBat	67
Relational databases in general	69
Database structure	71
Index	72

Foreword

WARNING

Whisker is a system designed for research purposes only, and should never be used to control medical apparatus or other devices that could endanger human life.

DISCLAIMER

The authors, copyright holders, and distributors disclaim all responsibility for any adverse effects that may occur as a result of a user disregarding the above warning.

1 RatBat

1.1 About RatBat



Purpose

Cognitive test battery for rats.
Written for the University of Cambridge.
The battery includes:

- [Reinforcement familiarization](#)
- [Touchscreen training](#)
- [Reversal learning](#)
- [Delayed matching and non-matching to sample](#)
- [Paired-associates learning](#)
- [Simple schedules of reinforcement](#)
- [N-pair concurrent visual discrimination](#)
- [N-Pair Concurrent Auditory Discrimination](#)
- [Audiovisual intradimensional \(ID\)/extradimensional \(ED\) set shifting](#)

Software requirements

Requires Whisker v2.7 or greater (www.whiskercontrol.com).

Data storage

- Text-based output to disk.
- ODBC data storage to a database (supplied).

Author

Rudolf Cardinal (rudolf@pobox.com).
Some configuration/code changes to ID/ED task by John Earl (August 2005).

Acknowledgements

Thanks to Mike Aitken, Shibley Rahman, and Hannah Clarke for helpful discussions regarding the PAL and and reversal learning tasks.

Copyright

Copyright © Cambridge University Technical Services Ltd

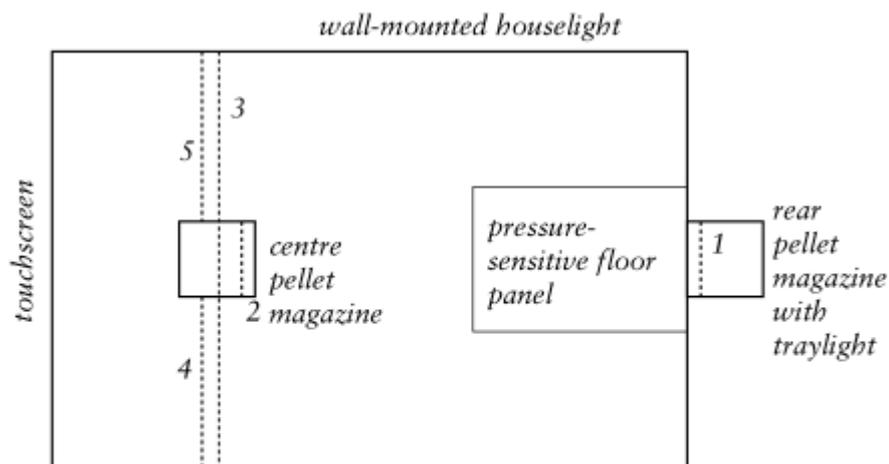
Revision history

- 19 June 2003. Started (version 0.1).
- 26 June 2003. Finished (version 0.1). Released within the University of Cambridge.
- 6 Aug 2003: Version 0.2 (interim release with known bugs). In the process of updating to version 0.3. Short lifespan, within U of C only.
- 7 Aug 2003. Version 0.3: tasks altered to be primarily two-alternative choice tasks, enabling use of IR beams as well as touchscreen. Miscellaneous bugfixes.

- 12 Jan 2004. Version 0.4: writes version number and compilation date to summary file; new summary statistics for reversal learning; new correction procedure ("VeryHarsh") for reversal learning.
- 22 Nov 2003. Version 0.5: change in method of centring stimuli; fixed bug (couldn't edit first point in a polygon list)
- 5 July 2004. Version 0.6: improved end-of-task on-screen display for reversal learning.
- 24 October 2004. Version 0.7: recording of all events; auditory discriminations; audiovisual ID/ED task; more status information for reversal task; new correction procedure (ANTIBIAS_NONCORRECTIONLIMIT) for reversal task.
- 9 Dec 2004. Version 0.8. Improved bitmap centring function. *N*-pair discrimination tasks (visual, auditory) have an option to finish when *X* trials from the last *Y* are performed correctly.
- 8 Feb 2005. Version 0.9. Bugfixes: (1) traylight could remain on between modules; (2) config version number reading was wrong; (3) minor errors fixed in manual.
- 21 Feb 2005. Version 1.0. Bugfixes: quasi-random selection procedures in NPair/NPairAuditory crashed.
- 1 May 2005. Version 1.1. Able to specify only the stimuli that will actually be required in the ID/ED task.
- 29 June 2005. Version 1.2. Bugfix: facility to trigger trials on a particular event was missing in the audiovisual ID/ED task.
- August 2005. Version 1.3. Some configuration/code changes to ID/ED task by **John Earl**.
- 8 March 2007. Version 1.4. Easier compilation for users.
- 12 Jan 2009. Version 2.0. Server default changed from "loopback" to "localhost" (Windows Vista compatibility and more general standardization).

1.2 Required devices

This program was designed for use with chambers like these:



Infrared beams: (1) rear magazine; (2) centre magazine; (3) centre locomotor beam; (4) left locomotor beam; (5) right locomotor beam.

... though less well-equipped chambers are support (through WhiskerServer's support for 'fake' devices) and better-equipped chambers are also supported (e.g. those with peristaltic pumps and lick sensors).

The program requires to claim devices in groups named **box0**, **box1**, **box2**... with device names as listed below in bold:

```
# ----- Box 0 definition
```

```

# INPUTS
# Lick sensors are used with peristaltic pump reinforcement.

line    0      box0    LICKSENSOR
line    3      box0    FLOORPANEL_REAR
line    6      box0    MAGNOSEPOKE_CENTRE
line    9      box0    MAGNOSEPOKE_REAR
line   12      box0    LOCOBEAM_LEFT
line   15      box0    LOCOBEAM_CENTRE
line   18      box0    LOCOBEAM_RIGHT

# OUTPUTS
# The houselight is on during the tasks.
# Pumps and/or pellet dispensers are used for reward.
# "Pump" delivers nice juice; "Pump2" delivers punishment (e.g. saline).
# The "extra reward device" might be a tone generator installed in the box, to be
associated with reward.
# The "extra punishment device" can be activated when the subject is punished.

line   24      box0    HOUSELIGHT
line   27      box0    PUMP
line   30      box0    PELLET_CENTRE
line   33      box0    PELLET_REAR
line   36      box0    TRAYLIGHT_REAR
line   39      box0    EXTRAREWARDDEVICE
line   42      box0    EXTRAPUNISHMENTDEVICE
line   45      box0    PUMP2

# DISPLAY
# This is the monitor with a touchscreen attached to it.

display 0      box0    SCREEN

# AUDIO
# This is one of the computer's sound device (typically connected to a
loudspeaker in the box).
# Used to reward successful rats with Mozart.

audio   0      box0    SOUND
audio   0      box0    LEFTSOUND
audio   1      box0    RIGHTSOUND

# ... and so on for other boxes

```

Please ensure that these devices are available and listed in the device definition file in use by the server. (The snippet above shows an extract from a typical definition file.) **If you do not have a particular device, and do not need it, then you should configure "fake" lines on the Whisker server, and assign the non-existent device to a fake line.**

Special note on monophonic and stereophonic sound systems

I suggest that if you have **two speakers, left and right** (e.g. audio devices 0 and 1), you do this:

```

audio   0      box0    SOUND
audio   0      box0    LEFTSOUND
audio   1      box0    RIGHTSOUND

```

... and tick the [Stereo system: play centred sounds to LEFTSOUND+RIGHTSOUND \(rather than](#)

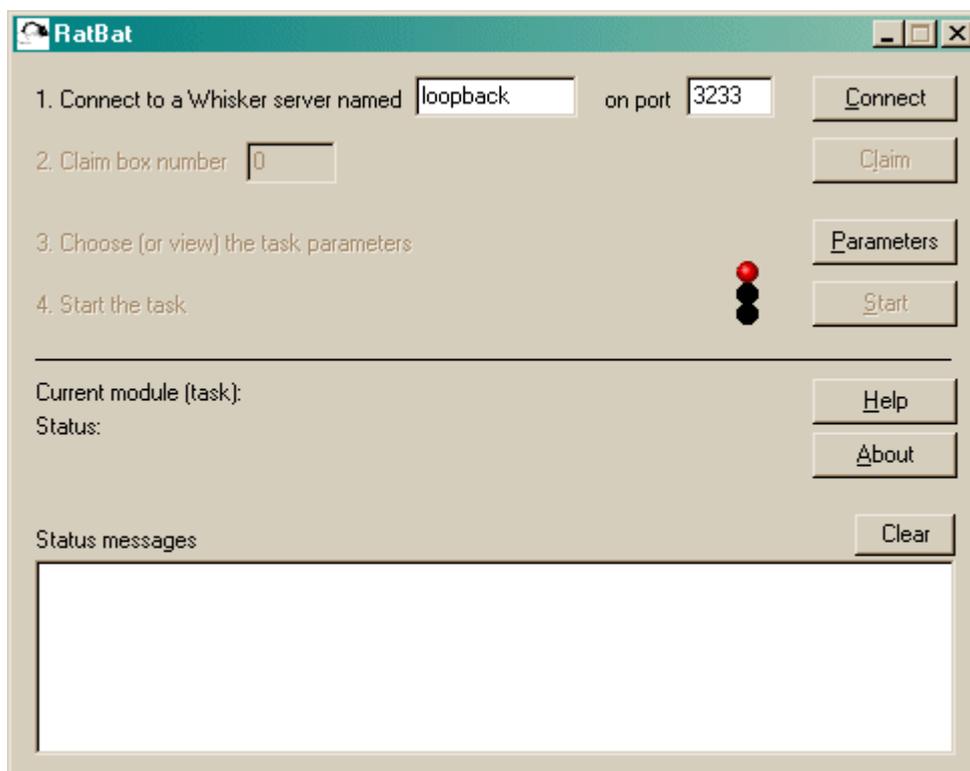
[SOUND](#) device box in the General Parameters. On the other hand, if you have a **single speaker** (e.g. audio device 0), you should untick this box and do this in the device definition file:

```
audio    0    box0    SOUND
audio    0    box0    LEFTSOUND
audio    0    box0    RIGHTSOUND
```

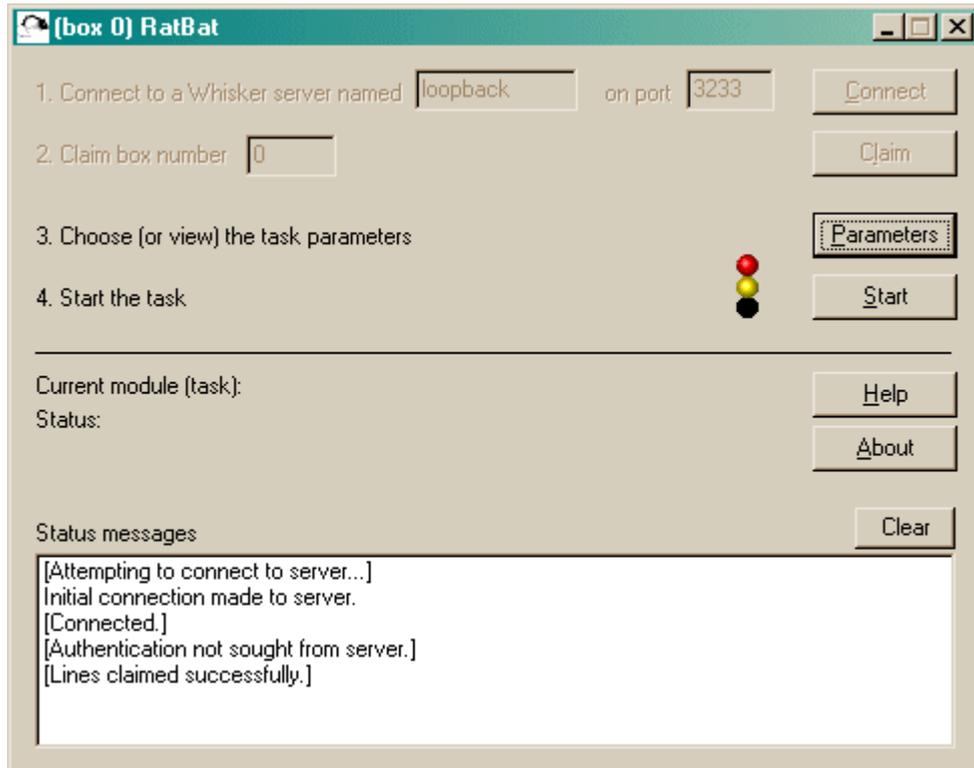
... and bear in mind that lateralized sound discrimination tasks won't work!

1.3 Using RatBat

When you run the task, the main screen looks as follows:



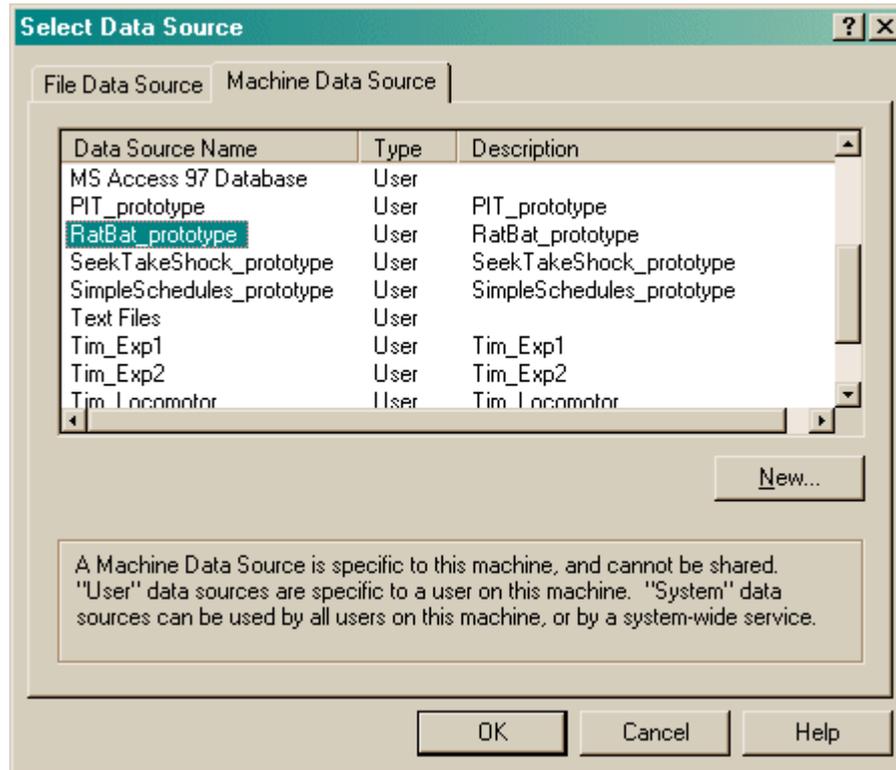
You must connect to a Whisker server, claim an operant chamber (box), and set up the [parameters](#) for your tasks. Configuration is explained on a [separate page](#). Once that's done, the traffic lights will turn amber.



When you are ready, press *Start* to begin RatBat.

The traffic lights will turn green and RatBat will then work through all the modules (tasks) in your task list.

When the task finishes, it saves data to disk and pops up a new dialogue box for you to select a database to store the data to. (The data sources are configured under *Control Panel* → *ODBC*.) If you previously specified an ODBC data source in the parameters, that data source is used automatically and you will only see a dialogue box if something goes wrong and the program needs your input.



Your data will be saved and RatBat has then finished.

1.4 Configuring RatBat (general settings)

To configure RatBat, choose **Parameters** from the [main screen](#).

RatBat allows you to run a selection of tasks. You must therefore tell RatBat which tasks you want to run, and in what order.

Set parameters for RatBat

Subject details

Load config Subject ID: test Session number: 17
 Save config Comment: (add your comment here)

Data recording

Set data file test-17Jul2003-1707-RatBat-summary.txt
 ODBC data source name (see Control Panel). Blank to choose later: RatBat_prototype Pick

Current module (task) list for this subject

Configure general parameters Configure visual objects

Module order

Add module Remove module Move up Move down

N-Pair Concurrent Visual Discrimination
 Paired-Associates Learning
 Delayed Matching/Non-matching To Sample
 Reinforcement Familiarization
 Touch Training
 Reversal Learning
 Simple Schedules of Reinforcement

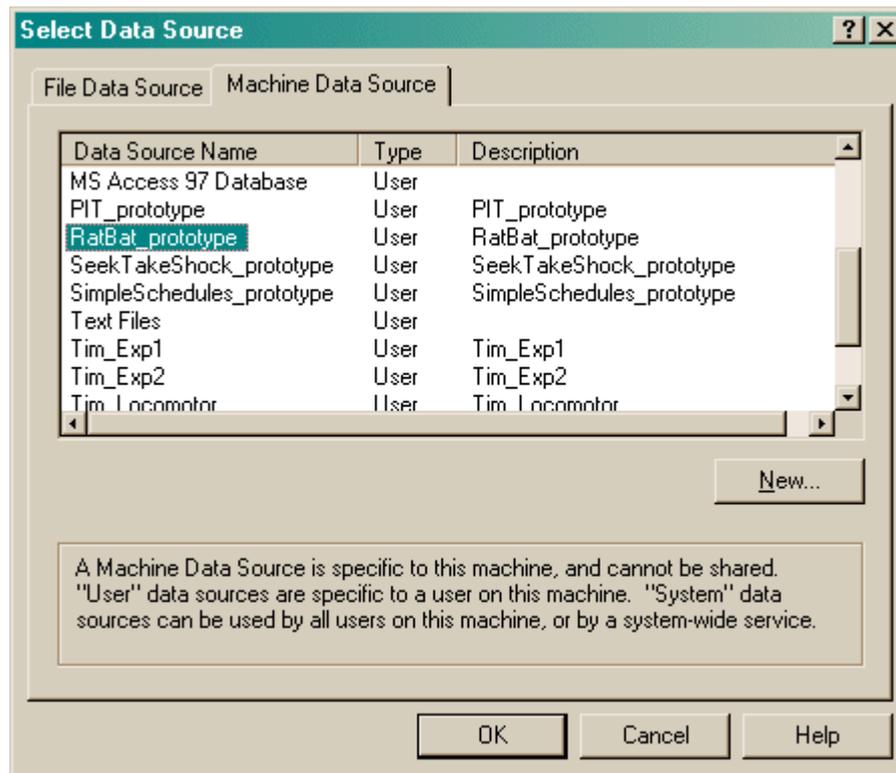
Configure module

You may **load** a configuration file or **save** it again. (If you load a configuration file, alter the settings, and click OK, your changes will be saved automatically for next time.) Loading may take a few seconds if the configuration files contain many stimuli, such as the supplied ID/ED task stimuli.

You may set any of the subject details (**ID**, **session number**, **comment**) by typing in the relevant boxes. You may **set a data file** if you choose; if you load a configuration file, the program will choose a default data file for you. This data file contains a textual summary of your results. The full result set is saved in a database via the ODBC (Open Database Connectivity) protocol. You may select the database at this point, or when the program finishes running.

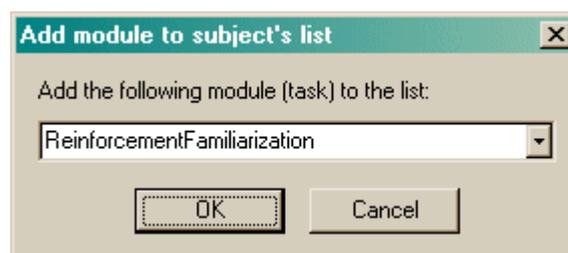
A sample configuration is supplied with RatBat. It's called "RatBat_TestConfig_And_SampleObjectLibrary.xml" and it lives in the directory you installed RatBat into (typically C:\Program Files\RatBat).

To pick an ODBC database *in advance* of finishing, click **Pick** and you will be offered the ODBC Data Source picker (below). Your choice will be recorded and will apply to this subject from now on (or until you specify a different source).



If you don't specify an ODBC data source now, or you delete the value in the **ODBC data source name** box, you'll be asked to choose when the task ends (and that choice will only apply to the session in progress).

The bottom half of the screen contains the module (task) list for the subject. You may specify any number of tasks from those that RatBat provides. They may be executed in any order. Click **Add Module** to add a new module to the end of the list. Click a module in the list to select it, and click **Remove Module** to or remove that module from the list. Click **Move Up** or **Move Down** to move a module up/down the list.



Click [Configure general parameters](#) to set up parameters that apply to all tasks.

Click [Configure visual objects](#) to set up objects used by the tasks.

Select a module from the list and click **Configure module** to set up parameters for the chosen module. Parameters that are specific to each task are explained with each task's description (click on the links below). The module names are:

- [Reinforcement Familiarization](#)
- [Touch Training](#)
- [Reversal Learning](#)
- [Delayed Matching/Non-matching To Sample](#)

- [Paired-Associates Learning](#)
- [Simple Schedules of Reinforcement](#)

1.4.1 General parameters

RatBat: General parameters

Links between tasks

Duration (s) Play sound Use houselight

Begin trials... (only applicable to trial-based tasks)

on rear panel activation on rear nosepoke on central nosepoke

on central locomotor beambreak on completion of preceding ITI, regardless of where the subject is

Subject is considered to have responded to a stimulus when it... approaches it (breaks L/R IR beam) or touches it touches it only

Use rear traylight when appropriate (to mean "poke your nose here")

Reward

Give pellet(s) # pellets: Pulse length (ms): Time between pellets (s): Dispenser: Rear Centre

Turn on pump Pump duration: Pump contingent upon licking during this time Each lick triggers pump for (s):

Play sound

Extra reward device Duration (s):

Punishment

Darkness Darkness time (s):

Turn on pump 2 (with pumping/licking parameters as for reward) - e.g. mildly aversive saline pump

Play sound

Extra punishment device Duration (s):

Stereo system; play centred sounds to LEFTSOUND+RIGHTSOUND (rather than SOUND) device

Default media directory (for sounds and bitmaps): Set

Link	Use WAV	Filename	Set	Frequency (Hz)	Sound type	Duration (s)	Level (0-100)
Link	<input type="checkbox"/>		Set	100	Sine	1	100
Reward	<input type="checkbox"/>		Set	1500	Tone	1	100
Punishment	<input type="checkbox"/>		Set	1000	Square	2	100
Marker 1	<input type="checkbox"/>		Set	500	Tone	1	100
Marker 2	<input type="checkbox"/>		Set	700	Tone	1	100
Marker 3	<input type="checkbox"/>		Set	900	Tone	1	100

Links between tasks

- **Duration (s).** The duration of the link between tasks.
- **Play sound?** If you select this option, the Link sound will be played at the start of the link (see *Sounds* below).
- **Houselight on?** By default, the houselight is on during tasks but switched off during linking periods. Choose this option to keep it on during the links.

Begin trials...

In trial-based tasks (i.e. most of those in RatBat), you may wish to ensure that your subject is located centrally before trials begin. You can therefore opt to begin trials when the subject activates a rear floor panel, nosepekes at a rear magazine, nosepekes at a central magazine, breaks a central infrared locomotor beam, or simply when it's time to do so (i.e. ignoring the subject's current location). In all but the last case, the program will wait for any ITI to elapse and will *then* wait for the subject's response before beginning the next trial.

What counts as responding to a stimulus?

RatBat is designed primarily around two-choice tasks. You may find you get more reliable detection of responding if a "response" to a stimulus is counted if the rat walks up to the screen on the appropriate side (left/right) (**approach**) in addition to making contact with the stimulus on the screen (**touch**). Approaches are presumed to be detected by appropriate infra-red beams (see [Required devices](#)).

If approaches are counted as responses, be aware that when a single stimulus is presented at the centre of the screen, an approach on either side is counted as a response. (In the [Touch Training](#) program, if a moving box stimulus is used in the last stage of training, an approach on either side is similarly accepted; however, this is probably not a training combination that is likely to be used, realistically.)

Traylight

If **Use rear traylight** is ticked, the tray (magazine) light at the back of the chamber is switched on (1) when the subject is required to nosepoke there to initiate a trial; (2) when pellets are delivered there. In either case, the light switches off as soon as the subject nosepokes.

Reward

- **Give pellet?** If you select this option, pellets will be delivered when the subject is rewarded.
 - **Pellets per reinforcement.** Choose the number of pellets per reinforcement. (This option is only applicable if you choose to give pellets in the first place.)
 - **Pellet pulse length (ms).** Select the length, in milliseconds, of the electrical pulse that will successfully activate your pellet dispenser. (For typical Med Associates 45-mg pellet dispensers, 45 ms works quite well, but you will have to experiment to find the best value for your device.)
 - **Interpellet gap (s).** Only applicable if you are giving multiple pellets per reinforcement. This determines the length of time, in seconds, that the program will wait between giving each pellet in a multi-pellet reward. Choose a value that is long enough to let your pellet dispenser recover from the previous delivery - very short interpellet gaps can cause pellet dispensers to jam.
 - **Dispenser:** choose from a centre or a rear pellet dispenser.
- **Turn on pump?** Select this to use pump reinforcement.
 - **Pump reinforcement duration.** How long should the pump run?
 - **Pump contingent upon licking during this time?** If this option is *not* ticked, then the pump simply runs for the time specified. If you tick this option, the pump is *available* for this duration, but is only actually activated when the subject licks.
 - **Each lick delivers liquid for this duration (s):** If the pump is made contingent upon licking, then each lick activates the pump for a certain time. Specify that time here.
- **Play sound?** If you select this option, the Reward sound will be played when the subject is rewarded (see *Sounds* below).
- **Extra reward device?** To have another device activated when the subject is rewarded (e.g. an external tone generator), tick this option.
 - **Duration (s).** How long should this extra reward device be activated for?

Note that you could give rewards with no sounds, or sounds with no rewards, or both.

Punishment

- **Darkness?** If you select this option, the houselight will be switched off as part of the punishment.
 - **Darkness time (s).** This sets the length of time the houselight will be off (only applicable if you selected the previous option).
- **Turn on pump 2?** Optionally, you can use the PUMP2 device to deliver mildly aversive substances (e.g. saline), instead of the rewarding substance that you normally deliver as reward

(via the PUMP device). Since this technique only works if the manner of collecting reward and punishment is identical (so the subject can't tell in advance what it's going to get), the other parameters for pumping (e.g. lick contingency, duration) are exactly the same as those for the reward pump.

- **Play punishment sound?** Chooses whether or not to play the Punishment sound as part of the punishment.
- **Extra punishment device?** To have another device activated when the subject is punished (e.g. an external tone generator), tick this option.
 - **Duration (s).** How long should this extra punishment device be activated for?
- Shocks are not explicitly implemented as a punishment option (for ethical reasons: this seems pretty severe). Contact rudolf@pobox.com if this causes problems.

Default media directory

If the server needs WAV files or bitmaps (.BMP) and cannot find them, it looks in this directory. If you have a collection of multimedia files (.WAV, .BMP) that you are using with RatBat, we suggest you select that directory here. Click **Set** to browse for the directory.

Sounds

For the predefined sounds (Link, Reward, Punishment, Marker1, Marker2, Marker3), you may set the following options:

- **Use WAV file.** Sounds may either be played as simple tones or as WAV files.
- **Filename.** To specify WAV files. Click **Set** to browse for the file. (Only applicable to WAV sounds.)
- **Frequency (Hz).** Specifies the sound's frequency in Hertz. (Only applicable to non-WAV sounds.)
- **Sound type.** Choose the waveform of your sound. "Tone" is similar to "Sine" but contains more energy. (Only applicable to non-WAV sounds.)
- **Duration (s).** The duration of the sound, in seconds. (Only applicable to non-WAV sounds.)
- **Level (0-100).** The volume of the sound. Maximum volume is 100; minimum volume is 0. More specifically, this number is 100 minus the sound attenuation in decibels (dB). (Only applicable to non-WAV sounds.)

Marker 1 is typically used to indicate the start of a trial.

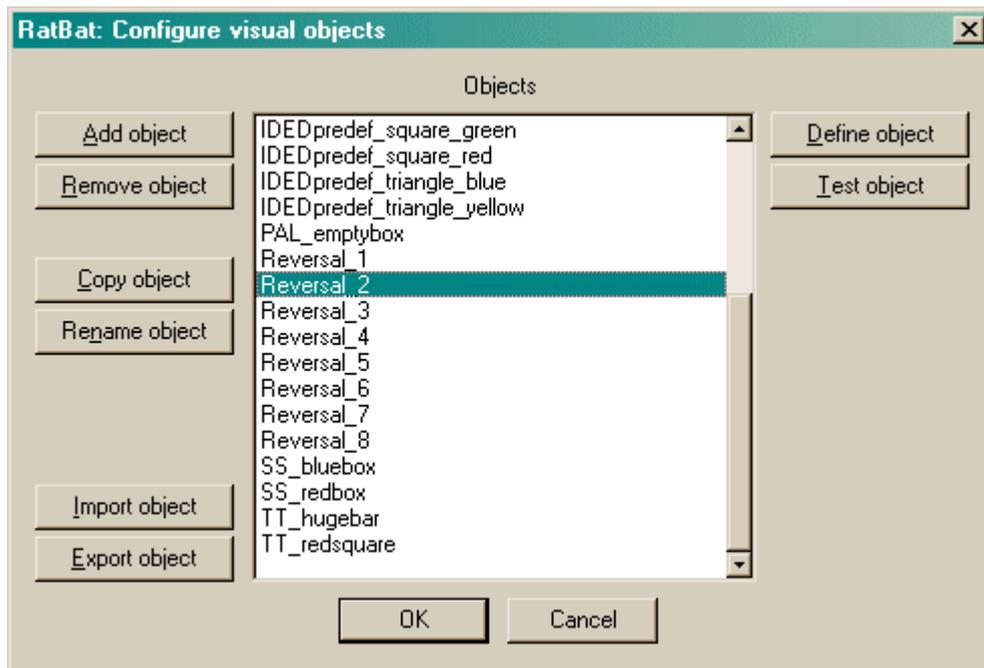
Marker 2 is typically used to indicate the start of a second phase of a trial.

Marker 3 is typically used to provide response feedback.

1.5 The visual object library

Editing the library

Every visual object (picture) used by RatBat lives in the **visual object library** and has a **unique name** associated with it.



- You can **add** and **remove** objects from the list - though you cannot remove an object that is being used by one of your tasks. You can make a **copy** of an object and you can **rename** objects. If you rename an object, all references to it by other tasks in your task list will be amended accordingly.
- Click **Define object** to configure the object itself.
- If you are already connected to a Whisker server, you may click **Test object**, and the object will be displayed in a new window (a "virtual device window") on the server's desktop for you to inspect it.
- Click **OK** to accept your changes, or **Cancel** to abort.
- Sometimes (as shown above) when you select an object, a message appears saying "**WARNING: This object has no touchable components.**" In this case, the object will not respond to being touched, so you are probably best avoiding it in all your tasks until you have added a component to it that is touchable (see below)!
- You may **import and export objects from other configuration files**. You may find it convenient to keep one configuration file as a master object library (for example, you could give it a dummy subject name and not use it to run tasks). If you design a handy object for one subject, you could export it to the library. When you create a new configuration file, you can either load one that's quite similar and save it under a new name, or start from scratch and import objects from your library. Click **Import** and **Export** to import/export objects. You will be asked to choose the configuration file for import/export, and then to choose the objects to copy across. When importing, you can select multiple objects (click on several objects). If objects with the same name exist in the file you are importing into (or exporting to), the objects will be renamed upon arrival.

A sample object library is supplied with RatBat. It's called **RatBat_TestConfig_And_SampleObjectLibrary.xml**.

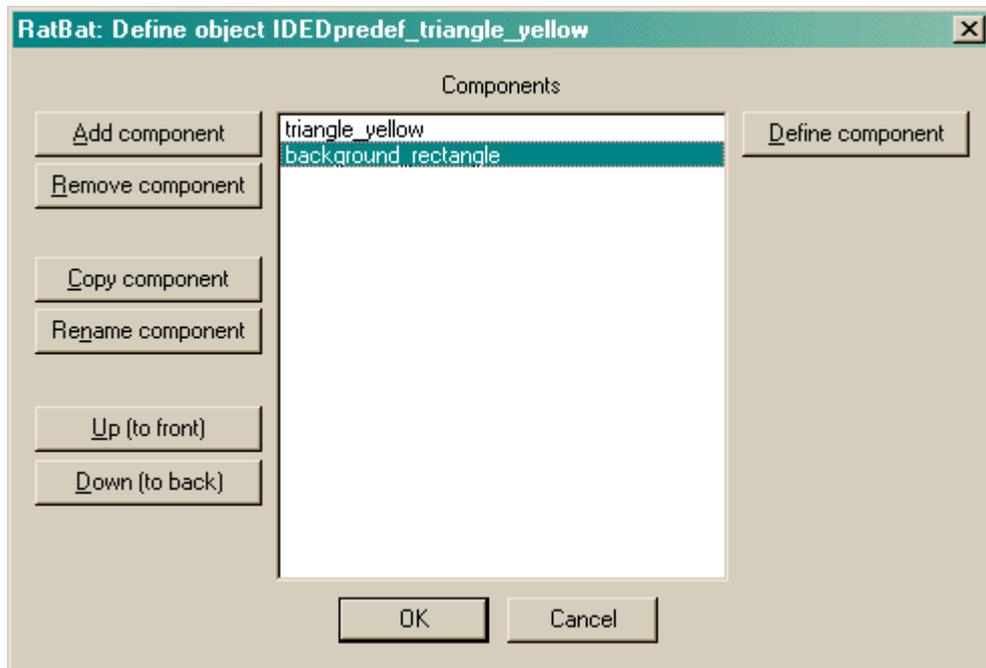
Editing individual objects

When you use your new object in a RatBat task, the task may place your object anywhere on the screen. Each task defines certain locations that it uses (for example, the ThreeChoice task displays them in one of three locations in a horizontal line in the middle of the screen). The task's locations are defined as rectangles. The tasks automatically try to centre your objects in these rectangles. (Some objects, like text, give them more difficulty!) This centring system relies on you basing the

top-left corner of your objects at the point (0,0). If you don't, your objects will be offset in the tasks.

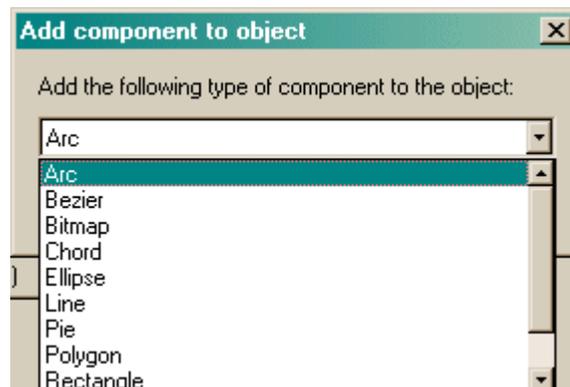
Editing components of individual objects

When you clicked *Define* in the dialogue box above, you can define components of one particular object in the dialogue box shown below.



Here we are defining the object called "IDEDpredef_triangle_yellow", and at present it has two components, named "triangle_yellow" and "background_rectangle". The objects are in a stack: "triangle_yellow" is at the top of the stack, so it will be the object in front of all the others. "Background_rectangle" is at the bottom of the pile.

- Click **Add** or **Remove** to add/remove components from the list. When you click Add, you will be offered a choice of the various types of component that are available, and then be asked to give your new component a name. (Note that two components can't have the same name.)

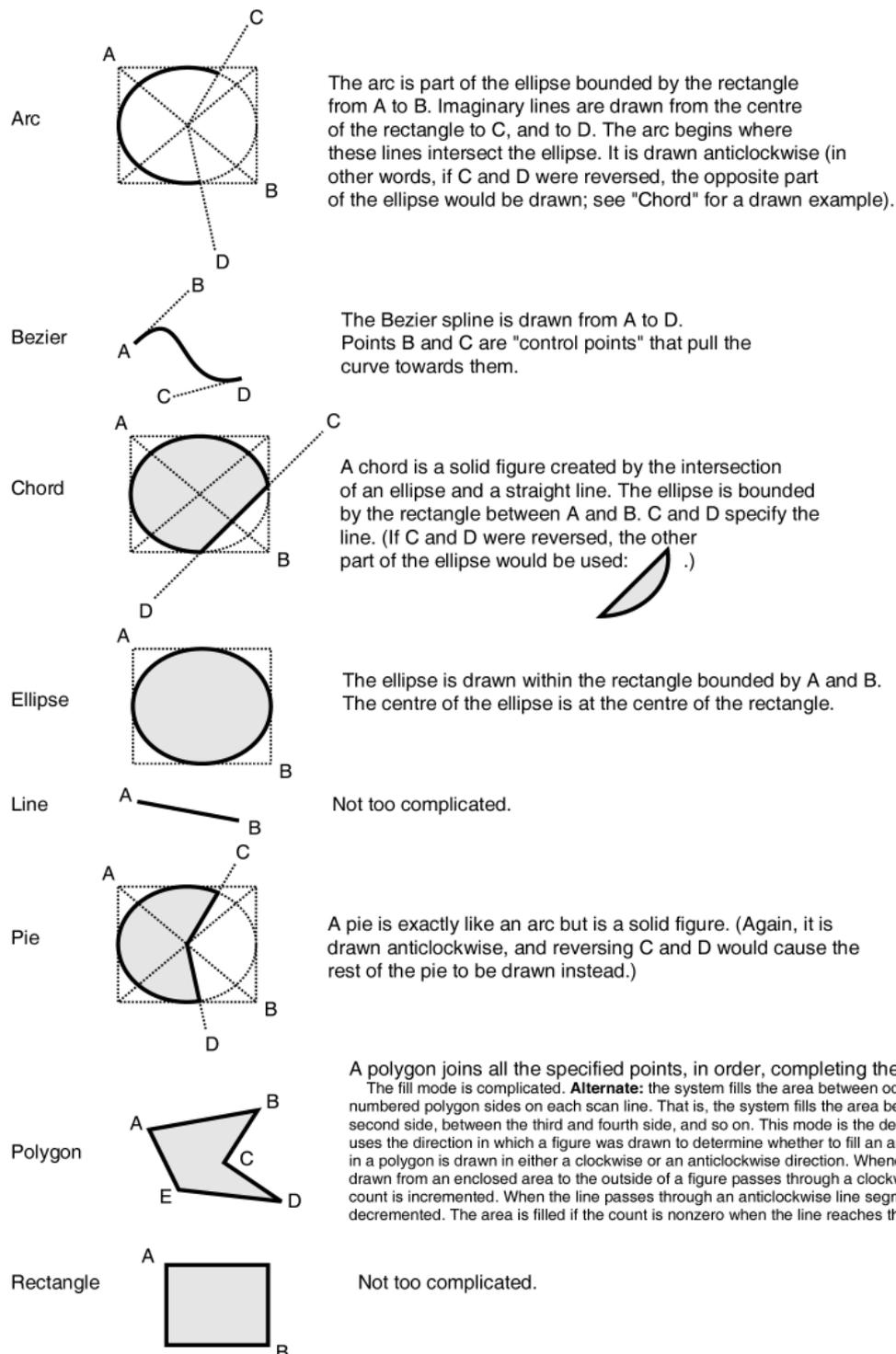


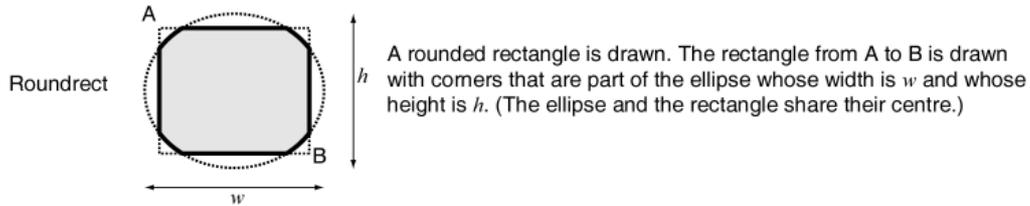
- Click **Copy** or **Rename** to copy or rename a component. (Note that two components can't have the same name.)
- Click **Up** or **Down** to move components up or down the stack.
- Click **Define** to [specify a component's details](#).

1.5.1 Defining components of objects

What do the components look like, and how do we define them?

Aside from bitmaps and text, all component types are illustrated in the picture below. Your objects should begin at the top-left point (0,0). Increasing x/y coordinates move to the right and down.





Which components are touchable?

Note that **arcs, bezier splines, lines, and text CANNOT support mouse or touchscreen events**. Everything else (bitmaps, chords, ellipses, pies, polygons, rectangle, rounded rectangles) can.

What part of the object is touchable?

What you see is what you can touch. If you want a larger area to be touchable, define a black rectangle of the desired size (all RatBat tasks use a black background), giving it a black or null pen, and place it at the bottom of your object's stack of components.

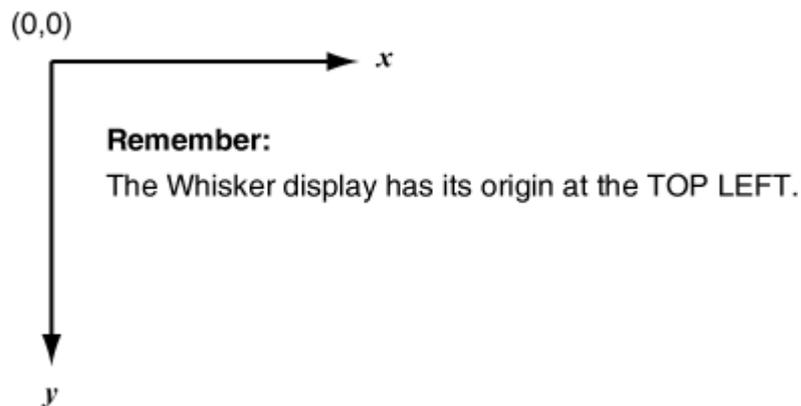
Defining the components

When you click **Define** in the [Component Definition dialogue](#), you can set the options for a particular component. Here are the possible components:

- [Arc](#)
- [Bezier spline](#)
- [Bitmap](#)
- [Chord](#)
- [Ellipse](#)
- [Line](#)
- [Pie](#)
- [Polygon](#)
- [Rectangle](#)
- [Rounded rectangle \(roundrect\)](#)
- [Text](#)

1.5.2 Size, coordinates, and grids

What coordinate system does RatBat use?



How do the tasks position objects on the screen?

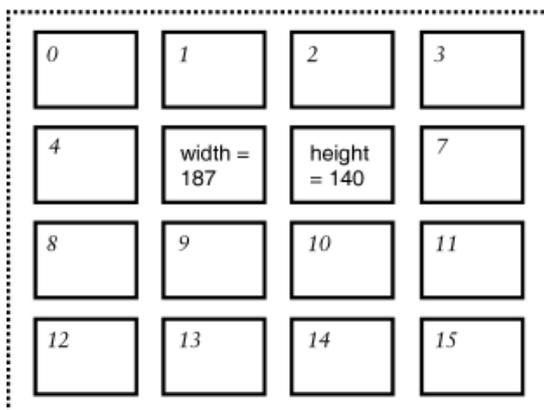
All RatBat tasks treat the screen as if it were a 1000x750 grid. They divide the screen up into rectangles, shown below. When called upon to display a stimulus, they attempt to work out the stimulus size and then they try to position the stimulus in the centre of the grid square that they are using, assuming that the stimulus begins at (0,0) in its internal coordinate system (the system with which you define objects).

The tasks will have problems determining the size of text, and of bitmaps if you don't force the bitmap to a specified size. You should therefore test these stimuli before using them.

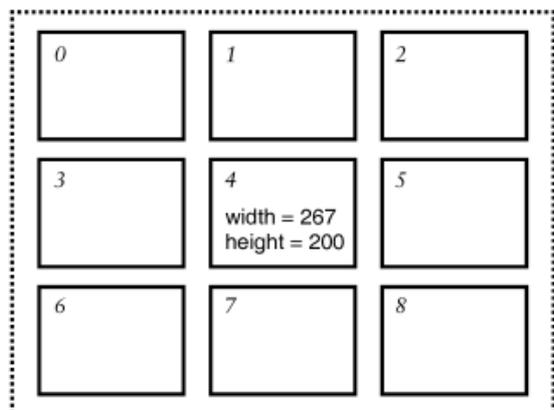
- [Reinforcement Familiarization](#) doesn't display any stimuli.
- [Touch Training](#) uses the two-way grid.
- [Reversal Learning](#) uses the two-way grid for two-stimulus tasks, and the middle row of the nine-way grid for three-stimulus tasks.
- [Delayed Matching/Non-matching To Sample](#) uses the nine-way grid (location 4 for the presentation phase, and locations 0, 2, 6, and 8 for the choice phase).
- [Paired-Associates Learning](#) uses the nine-way grid.
- [Simple Schedules of Reinforcement](#) uses the centre of the nine-way grid.
- [N-Pair Concurrent Visual Discrimination](#) uses the two-way grid.

The sizes of these grids are shown below. In the 16-, 9-, and 4-way grids, the space between the grid squares and between the grid and the edge of the screen is 5% of the screen's extent. Refer to the width/height of individual cells; for example, stimuli designed for use with tasks that use the nine-way grid should fit in a 267x200 rectangle. *Not all the grids shown below are used at present.*

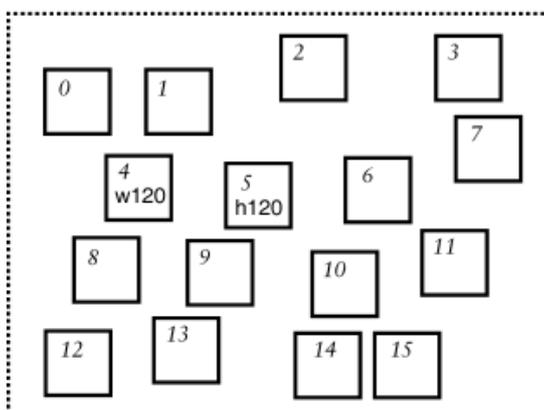
Sixteen-way grid



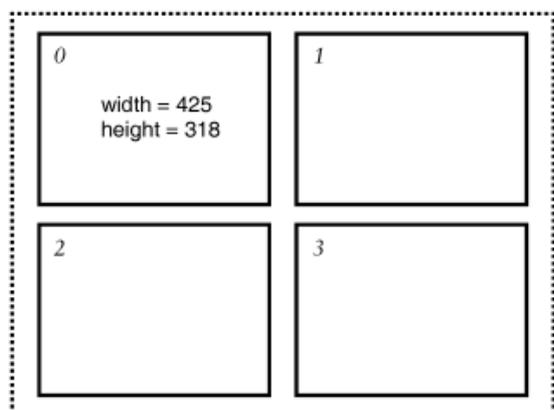
Nine-way grid

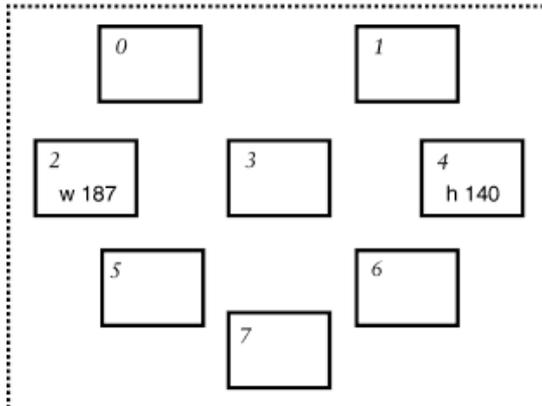


Sixteen-way scattered (approx.)

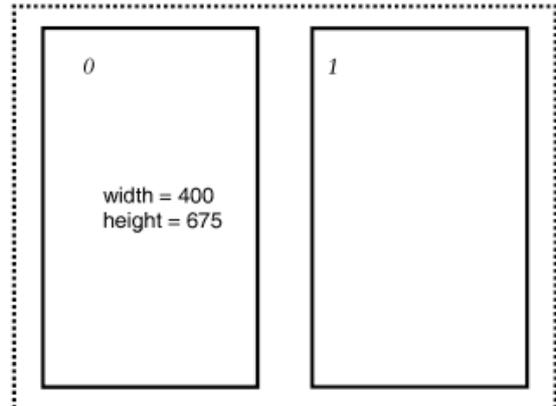
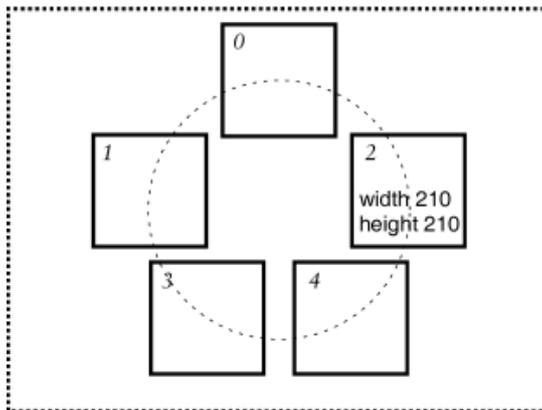
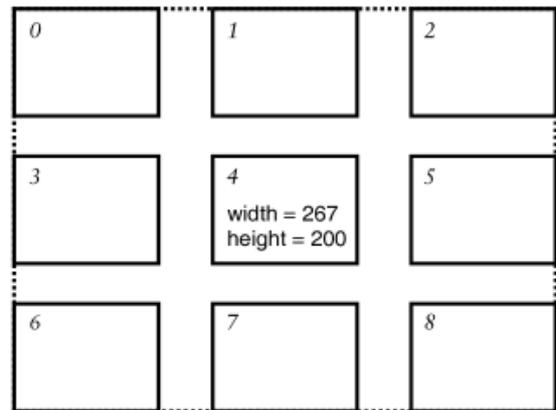


Four-way grid



Eight-way scattered**Two-way grid**

whole screen
width = 1000
height = 750

**Five-choice pattern****Nine-way (extreme) grid****1.5.3 Arc**

Define an ARC ✕

Press F1 for help (including pictures)

	x	y
A (top left of rectangle)	<input type="text" value="0"/>	<input type="text" value="0"/>
B (bottom right of rectangle)	<input type="text" value="100"/>	<input type="text" value="100"/>
C (intersection point 1)	<input type="text" value="0"/>	<input type="text" value="100"/>
D (intersection point 2)	<input type="text" value="0"/>	<input type="text" value="0"/>

- The meaning of the points is explained in the [figure above](#).
- Click [Pen options](#) to determine how the edge of the object is painted.

1.5.4 Bezier spline

Define a BEZIER spline (a wiggly line) ✕

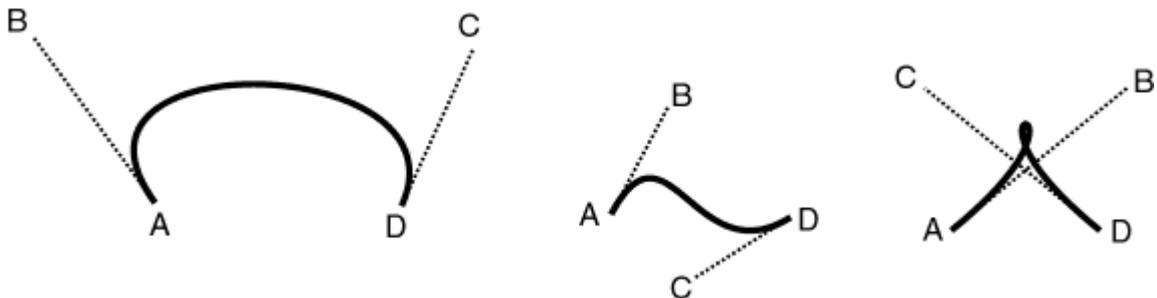
Press F1 for help (including pictures)

	x	y
A (start point)	<input type="text" value="100"/>	<input type="text" value="100"/>
B (control point 1)	<input type="text" value="0"/>	<input type="text" value="50"/>
C (control point 2)	<input type="text" value="250"/>	<input type="text" value="50"/>
D (end point)	<input type="text" value="200"/>	<input type="text" value="100"/>

This curve passes through these points:
 (100,100) - (92,66) - (174,66) - (200,100)

- The meaning of the points is explained in the [figure above](#) and in the examples and mathematical description below.
- Click [Pen options](#) to determine how the edge of the object is painted.

Some more examples:



Definition of a Bezier curve (de Casteljaou, 1959; Bezier, 1962)

Let's start simple. Imagine a line that begins at A and ends at D. Let t be a variable from 0 to 1. We can define a point $P_{AD}(t)$ on the line segment AD as

$$P_{AD}(t) = (1-t)A + tD$$

If we add another point, B, into the picture, we can define $P_{AB}(t)$ as a point between A and B, and $P_{BD}(t)$ as a point between B and D. If we apply the same method to define $P_{AB-BD}(t)$ as a point between $P_{AB}(t)$ and $P_{BD}(t)$, we get

$$\begin{aligned}
 P_{AB}(t) &= (1-t)A + tB \\
 P_{BD}(t) &= (1-t)B + tD \\
 P_{AB-BD}(t) &= (1-t)\{(1-t)A + tB\} + t\{(1-t)B + tD\} = (1-t)^2A + 2t(1-t)B + t^2D
 \end{aligned}$$

This quadratic equation in t defines a quadratic Bezier curve - a parabola. For computer graphics purposes, cubic Bezier curves are more often used. As you might expect, these are defined by four points A, B, C, and D. If the cubic Bezier function is defined as a point between $P_{AB-BC}(t)$ and $P_{BC-CD}(t)$ in the same manner as we've been doing so far...

$$\begin{aligned}
 P_{AB}(t) &= (1-t)A + tB \\
 P_{BC}(t) &= (1-t)B + tC \\
 P_{CD}(t) &= (1-t)C + tD \\
 P_{AB-BC}(t) &= (1-t)\{(1-t)A + tB\} + t\{(1-t)B + tC\} = (1-t)^2A + 2t(1-t)B + t^2C \\
 P_{BC-CD}(t) &= (1-t)\{(1-t)B + tC\} + t\{(1-t)C + tD\} = (1-t)^2B + 2t(1-t)C + t^2D \\
 \text{CubicBezier}(t) &= (1-t)P_{AB-BC}(t) + tP_{BC-CD}(t) = \dots = (1-t)^3A + 3(1-t)^2tB + 3(1-t)t^2C + t^3D
 \end{aligned}$$

Summary: Where we end up

Cubic Bezier splines are usually defined with endpoints A and D and control points B and C that are not on the curve, as above. The equation for a point on this curve is given by

$$\text{CubicBezier}(t) = (1-t)^3A + 3(1-t)^2tB + 3(1-t)t^2C + t^3D$$

where t is the curve's parameter and ranges from 0 to 1. This curve can be expressed in a different way: as a curve passing *through* four points, PQRS, where $P = \text{CubicBezier}(0)$, $Q = \text{CubicBezier}(1/3)$, $R = \text{CubicBezier}(2/3)$, and $S = \text{CubicBezier}(1)$. From the formula above,

$$\begin{aligned}
 P &= A \\
 Q &= 1/27(8A + 12B + 6C + D) \\
 R &= 1/27(A + 6B + 12C + 8D) \\
 S &= D
 \end{aligned}$$

and therefore

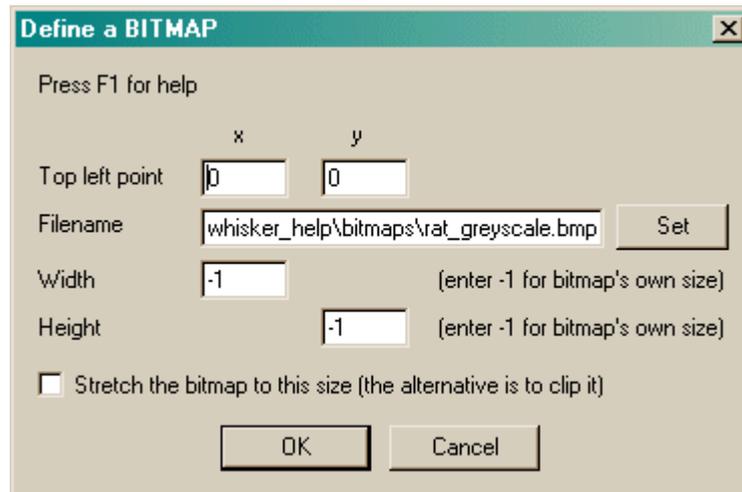
$$\begin{aligned}
 A &= P \\
 B &= 1/6(-5P + 18Q - 9R + 2S) \\
 C &= 1/6(2P - 9Q + 18R - 5S) \\
 D &= S
 \end{aligned}$$

In RatBat's Bezier dialogue box (above), you enter the points ABCD and RatBat shows you the points PQRS ("The curve passes through these points: ...").

Some properties of Bezier curves

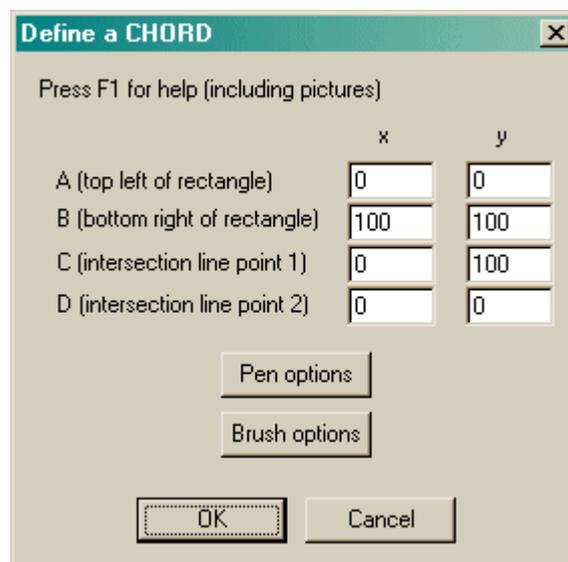
- Bezier curves always pass through their first and last points (A and D here), but not necessarily through their other control points (B and C).
- A Bezier curve always lies fully within the convex hull defined by its control points.
- The line AB has the same tangent at the curve at A, and the line CD has the same tangent as the curve at D.
- Bezier curves are always divisible into two Bezier curves (in a manner which makes them easy to draw iteratively). See Yuan, F. (2001), *Windows Graphics Programming*, Hewlett-Packard/Prentice Hall, New Jersey (p481 onwards).

1.5.5 Bitmap



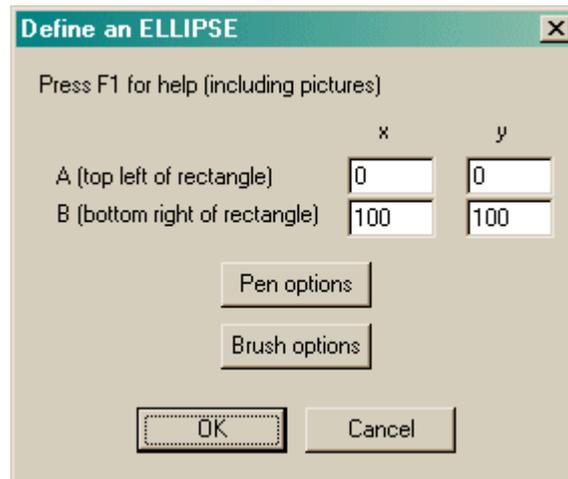
- Choose the **(x, y) coordinates of the top left point**. This is normally (0,0).
- Choose the **filename** of the bitmap. Click **Set** to browse for the file. If you are running RatBat on a different computer to WhiskerServer, remember that the filename must be accessible by the server, not the client.
- Choose a width and height to force the bitmap to, or leave the values at -1 to use the bitmap's intrinsic height.
- Choose whether to **stretch** or **clip** the bitmap (leave the tickbox unticked for clipping). Stretching means that the bitmap is deformed to fit your specified width/height. Clipping means that the bitmap's size isn't changed, but that the right/bottom edges may be cut off if the width/height you specify are smaller than the bitmap's intrinsic size.

1.5.6 Chord



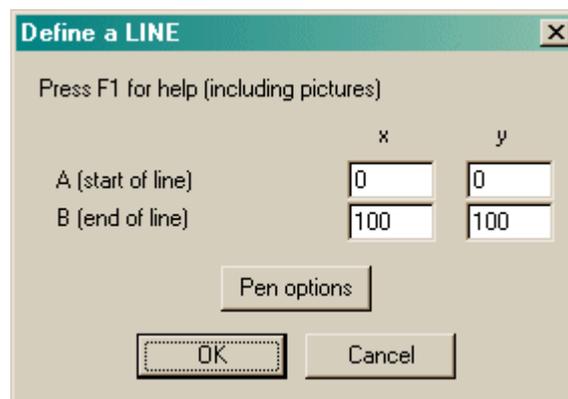
- The meaning of the points is explained in the [figure above](#).
- Click [Pen options](#) to determine how the edge of the object is painted.
- Click [Brush options](#) to determine how the inside of the object is filled.

1.5.7 Ellipse



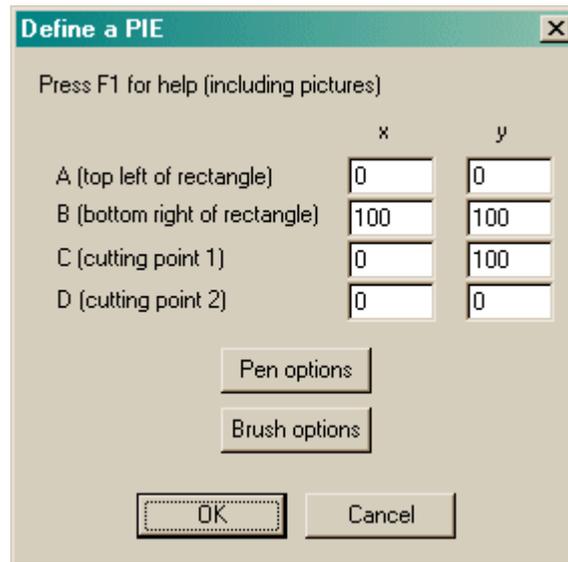
- The meaning of the points is explained in the [figure above](#).
- Click [Pen options](#) to determine how the edge of the object is painted.
- Click [Brush options](#) to determine how the inside of the object is filled.

1.5.8 Line



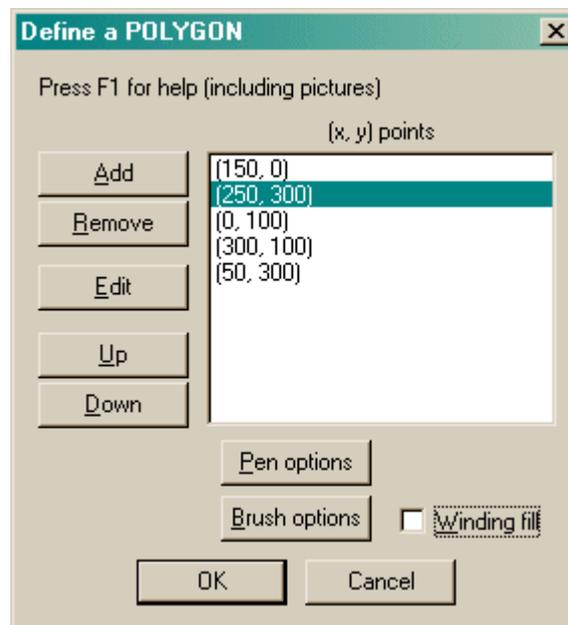
- The meaning of the points is explained in the [figure above](#).
- Click [Pen options](#) to determine how the edge of the object is painted.

1.5.9 Pie



- The meaning of the points is explained in the [figure above](#).
- Click [Pen options](#) to determine how the edge of the object is painted.

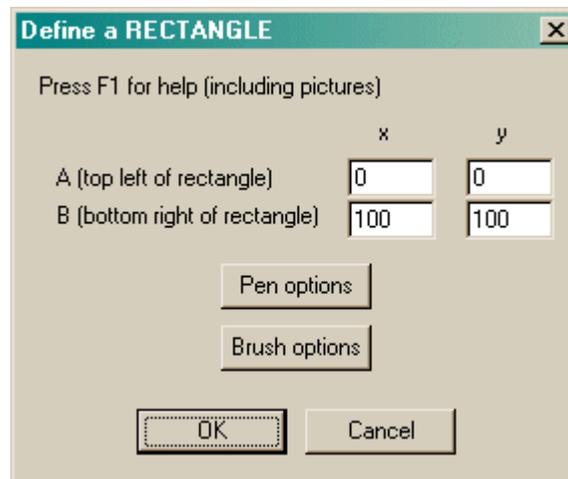
1.5.10 Polygon



- The meaning of the points is explained in the [figure above](#). You need at least three points for a polygon component.
- Click **Add** or **Remove** to add points to the polygon or remove them.
- Click **Edit** to alter a point.
- Click **Up** or **Down** to re-order the points.
- The meaning of the rather complicated **winding/alternate fill setting** is also explained in the [figure above](#).

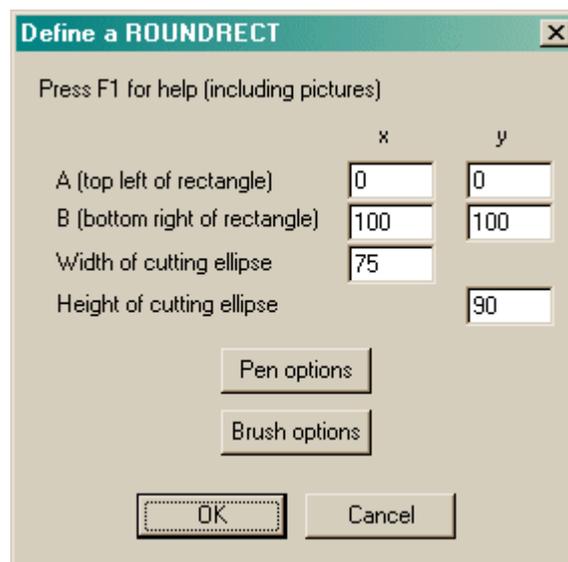
- Click [Pen options](#) to determine how the edge of the object is painted.
- Click [Brush options](#) to determine how the inside of the object is filled.

1.5.11 Rectangle



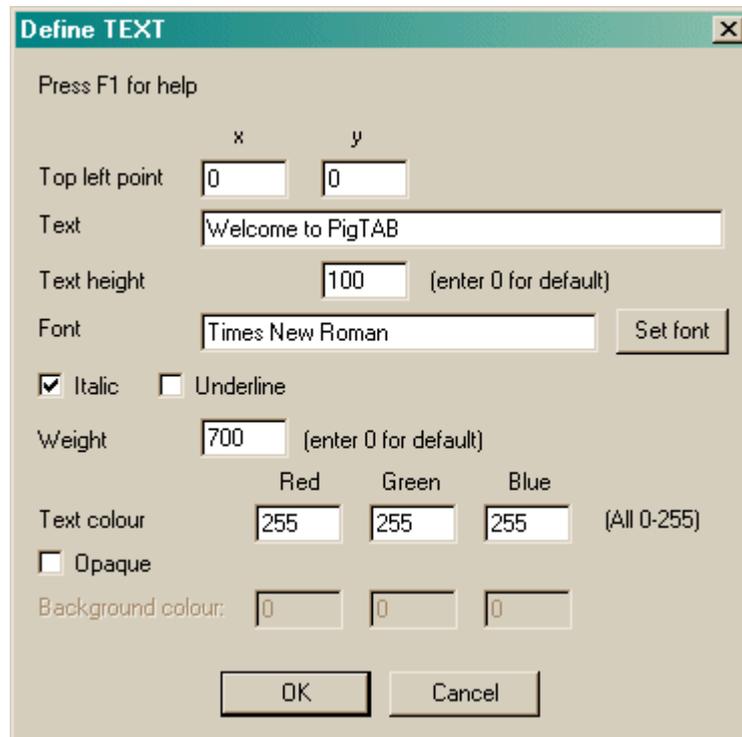
- The meaning of the points is explained in the [figure above](#).
- Click [Pen options](#) to determine how the edge of the object is painted.
- Click [Brush options](#) to determine how the inside of the object is filled.

1.5.12 Rounded rectangle



- The meaning of the points is explained in the [figure above](#).
- Click [Pen options](#) to determine how the edge of the object is painted.
- Click [Brush options](#) to determine how the inside of the object is filled.

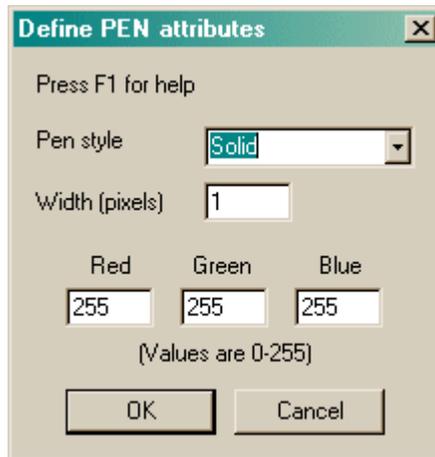
1.5.13 Text



- Choose the **(x, y) coordinates of the top left point**. This is normally (0,0).
- Set the **text** itself.
- Set the **text height** (in pixels, not points), or use 0 for a default setting.
- Choose the **font name** to be used by the server.
- Choose whether the font should be **italic** or **underlined**.
- Choose a font **weight** (equivalent to "boldness", and ranging from 1-1000), or use 0 for a default weight.
- Choose the **text colour**. Bear in mind that picking a black font (as would be common for wordprocessing and the like) will make the font invisible on the default black background of Whisker screens. Alter the background colour, place the text on another object, or change the font colour.
- Many of the settings listed above can be set by clicking the **Set font** button, which lists the fonts available on your system.
- Choose whether the font should be **opaque** or not. If it is opaque, the gaps between the letters are filled in with a **background colour**, in which case you may choose this too.

1.5.14 Pen options

Pens draw round the edges of components.



- Choose a **pen style**. The options *solid*, *dash*, *dot*, *dashdot*, and *dashdotdot* should be fairly self-explanatory. *Null* gives an invisible pen. *Insideframe* is relevant when the pen is thick; for example, if you draw a circle of diameter 100 units with a pen of width 20 units, the circle will normally end up having an external diameter of 120 units and an internal diameter of 80 units (as the pen overlaps by 10 units on the inside and the outside of the circle). If you specify *insideframe*, the circle's outside diameter is 100 units in this situation.
- Choose a **pen width** in pixels.
- Choose a **pen colour** by specifying values from 0-255 for the red, green, and blue components of the colour.

1.5.15 Brush options

Brushes are used to fill the insides of solid components with colours or patterns.



- Choose a **brush style**. This may be *hollow* (invisible), *solid*, or *hatched* (in which case you can specify the hatching style and colour).
- For solid and hatched brushes, choose the **hatching colour**.

- If you select a hatched brush, choose the **hatching style**. The hatching styles are *back diagonal* (lines at 45° anticlockwise from the horizontal axis), *cross* (horizontal and vertical lines); *diagcross* (lines at 45° clockwise and anticlockwise from the horizontal); *fdiagonal* (lines at 45° clockwise to the horizontal); *horizontal* (horizontal lines); *vertical* (vertical lines).
- A hatched brush may either be **opaque or transparent**. If it is transparent, you can see through the hatching to whatever is beneath. If it is opaque, you may set the **background colour** used to fill in the gaps in the hatching.

1.6 Configuring individual tasks

Choose a task to see how to configure it:

- [Reinforcement Familiarization](#)
- [Touch Training](#)
- [Reversal Learning](#)
- [Delayed Matching/Non-matching to Sample](#) (D[N]MTS)
- [Paired-Associates Learning](#) (PAL)
- [Simple Schedules of Reinforcement](#)
- [N-Pair Concurrent Visual Discrimination](#)
- [N-Pair Concurrent Auditory Discrimination](#)
- [Audiovisual intradimensional \(ID\)/extradimensional \(ED\) set shifting](#)

1.6.1 Reinforcement Familiarization

About the task

Purpose: to train the subject to take rewards, and optionally associate them with a sound. The task presents reward at random intervals

Configuring the task

What constitutes "a reinforcer" is defined in the [General Parameters](#). This allows you to set the reward sound and define what sound is associated with the presentation of a reward.

Parameters for Reinforcement Familiarization

Special options for licker training

Pump switched on at start of session
Pump runs freely until subject has made licks (0 for unlimited)

Lick-contingent juice on FR1 schedule for rest of session
Each lick triggers the pump for seconds.
(No sounds or extra devices are active during this time.)

Subsequently, rewards are given as specified in the General Parameters

Fixed-time (FT) schedule Random-time (RT) schedule

Schedule parameter (s):

Maximum number of rewards (0 for no limit):

Maximum time allowed (min) (0 for no limit):

OK Cancel

- **Special options for licker training.** These options are only applicable if you have a pump delivering liquid reinforcement and a lick sensor. If you wish, you can proceed through two initial stages before the main schedule starts. These extra options do NOT use the reward parameters specified in the [General Parameters](#) - for example, no sounds are played. If you select one or more of these options, they are triggered in the order they are listed in the parameters dialogue box: (1) pump switched on... (2) lick-contingent juice on FR1 schedule... (3) FT/RT schedule.
 - **Pump switched on at start of session.** If selected, the pump is activated without a pause at the start of the session, until the subject has made the number of licks specified in the **licks** box. At this point, the pump is switched off. If you specify zero licks here, the pump will run for the *entire* session and the program will never proceed to another phase.
 - **Lick-contingent juice on FR1 schedule for rest of session.** This option allows you to reward licking for the rest of the session. Each lick activates the pumps for a fixed time (specified in the **Each lick triggers...** box).
- **Schedule parameter (s).** Now begins the proper schedule; all rewards delivered here follow the settings specified in the [General Parameters](#).
 - **Fixed-time schedule.** Reward is delivered at regular intervals. Reward is delivered; the program waits for the end of reinforcement; this parameter specified how long it will *then* wait before delivering the next reward. (Note that this is not quite the standard parameter of an FT schedule: the standard way is to specify the time between the start of reward 1 and the start of reward 2; here, this parameter specifies the time between the end of reward 1 and the start of reward 2. This is an easy way to prevent the second reinforcer being scheduled while the first is still being delivered! You can think of this parameter as the inter-reinforcement interval.)
 - **Random-time schedule.** An imaginary clock ticks once a second. Every clock tick, the program flips a coin that will turn up heads with a probability of $1/parameter$. If the coin turns up heads, a reinforcer is delivered. Thus, if the parameter is 30, reinforcement is delivered with a probability of $1/30$ every second, meaning that on average reinforcement is delivered once every 30 seconds. (However, for as long as it takes to deliver the reinforcer or the reinforcer-associated sound, the clock stops ticking in order to prevent another reinforcer being delivered at the same time - if you need a precise RT schedule, you must take this time into account too.)
 - **Maximum number of rewards.** Once this number of rewards has been delivered, the task stops. (Specify 0 for no limit.) This number includes rewards earned on the FR1 licking part of the task, but not on the "free juice" part.
 - **Maximum time permitted during task.** Once this time limit has been reached, the task stops. (Specify 0 for no limit.)

You must specify either a maximum number of rewards, or a maximum time, or both.

The houselight is on during the task.

1.6.2 Touch Training

About the task

Purpose: to train the subject to touch stimuli.

A trial begins and marker sound 1 is played. Object(s) are presented on the screen. If the subject responds correctly within a criterion time, reward it. Punish the subject if it fails to respond in time (if you've set a time limit), or if it misses the stimulus and touches the background instead (if you've chosen to punish this).

Configuring the task

Parameters for Touch Training (v2) [X]

Maximum number of trials (0 for no limit):

Maximum time (min) (0 for no limit):

Response criterion time (s) (0 for no limit):

Missing the stimulus (and touching the background) terminates the trial as a failure

Time between trials (s): from to s

Stimuli

Specify stimuli

Stimulus:

How many stimuli should be shown at once?

Possible locations:

A single square will be used as the stimulus.

Colour (red, green, blue):

Starting size: 120x120 240x240 360x360 480x480
 600x600 800x675 Whole screen (1000x750)

Final size: 120x120 240x240 360x360 480x480
 600x600 800x675 Whole screen (1000x750)

Shrink stimulus one step after this many consecutive correct trials:

Move box around once the final size has been reached.

If approach is scored as touching (see General Params), approach EITHER side is permitted.

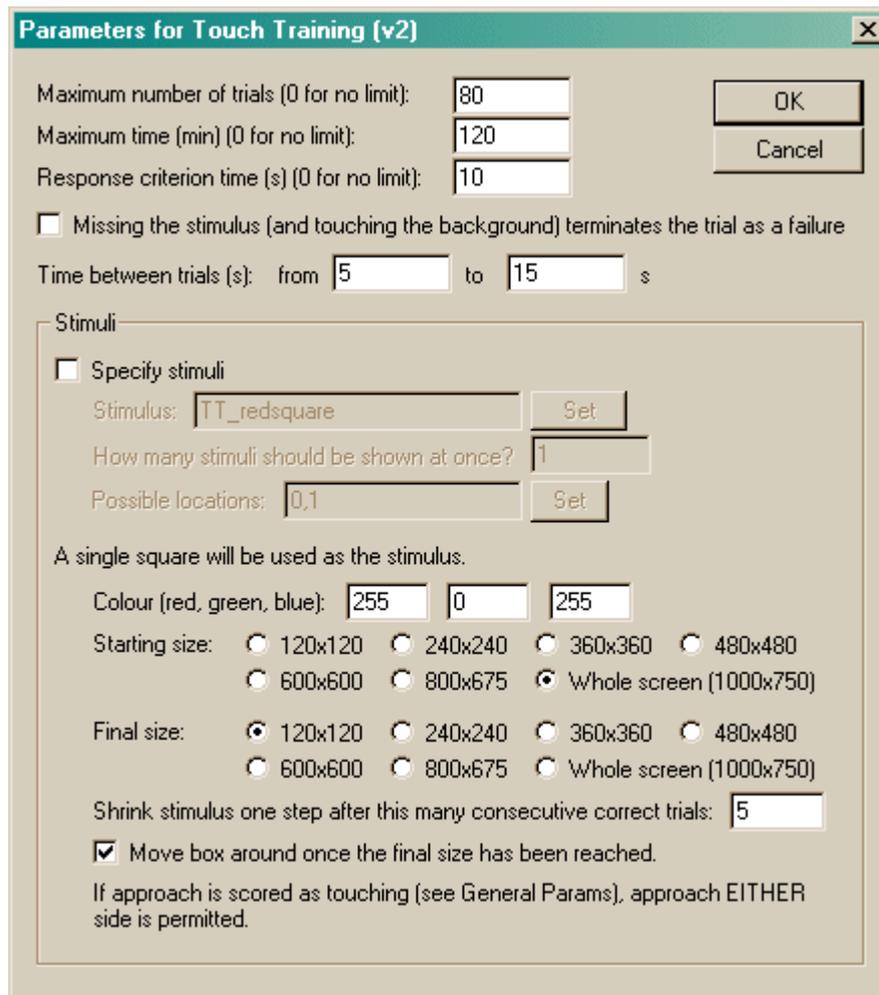
- **Maximum number of trials.** When the subject has performed this number of trials, the task ends. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Maximum time.** When this time elapses, the task is terminated as soon as the current trial has finished. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Response criterion time.** If the subject fails to respond to a stimulus within this time, an omission is scored.
- **Missing the stimulus terminates the trial as a failure.** If ticked, the trial is terminated if the subject touches the background. Otherwise, the subject can touch the background first and later touch the stimulus, or can 'slide' onto the stimulus.
- **Time between trials.** Specify a minimum and a maximum intertrial time (they may be the same). The actual time is chosen with a rectangular probability distribution within these values.
- **Specify stimuli.** If ticked, you specify the stimuli exactly. If not, a square is used (and it can shrink as time goes by).

Options available if "Specify stimulus" is ticked:

- **Stimulus.** Choose the stimulus (click **Set** to select from the list of available stimuli).
- **How many stimuli should be shown at once?** Self-explanatory.
- **Possible locations.** Click **Set** to choose the set of locations at which the stimulus (or stimuli) should be presented. Locations 0 and 1 represent the left and right sides of the screen; see the map in Size and coordinates (nine-way grid) or the map shown when you click **Set**.

Options available if "Specify stimulus" is not ticked:

- **Colour (red, green, blue).** Specify the square's colour. Each value (red, green, blue) can be from 0 to 255.
- **Starting size.** Specify the size that the stimulus starts at.
- **Final size.** Specify the size that the stimulus finishes at.
- **Shrink stimulus... after this many consecutive correct trials.** When the subject performs this many trials correct in a row, the stimulus shrinks one step (until it reaches the final size).
- **Move box around once the final size has been reached.** If this is ticked, then once the stimulus has reached the final size and the subject performs n correct trials in a row (where n was defined in *Shrink stimulus...*) then the stimulus starts to move around randomly on each trial.



Parameters for Touch Training (v2)

Maximum number of trials (0 for no limit):

Maximum time (min) (0 for no limit):

Response criterion time (s) (0 for no limit):

Missing the stimulus (and touching the background) terminates the trial as a failure

Time between trials (s): from to s

Stimuli

Specify stimuli

Stimulus:

How many stimuli should be shown at once?

Possible locations:

A single square will be used as the stimulus.

Colour (red, green, blue):

Starting size: 120x120 240x240 360x360 480x480
 600x600 800x675 Whole screen (1000x750)

Final size: 120x120 240x240 360x360 480x480
 600x600 800x675 Whole screen (1000x750)

Shrink stimulus one step after this many consecutive correct trials:

Move box around once the final size has been reached.

If approach is scored as touching (see General Params), approach EITHER side is permitted.

What constitutes "reward" and "punishment"?

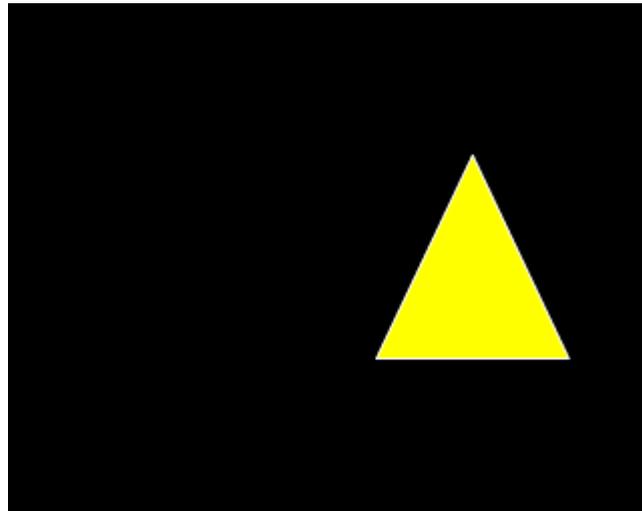
Options for reward and punishment are set in the General Parameters section; visual objects are defined in the Visual Object Library.

More on displaying objects

The task works with an internal scaling system based on a display that is 1000 units (pixels) wide and 750 high. If you specify the stimulus, it should fit into a rectangle that is 350 (w) x 600 (h). Imagine that the screen is divided into two boxes. (See this diagram ("Size and coordinates").)

Screenshot from the task

TOUCH IT!



1.6.3 Reversal Learning

About the task

Provides facilities for simple or serial reversal learning, with either two stimuli ($A+B- \rightarrow A-B+$) or three ($A+B-C- \rightarrow A-B+C-$).

There is an option to use three objects. In this task, a subject is trained with $A+B-C-$ and then reversed to $A-B+C-$; perseveration can then be measured directly as the degree to which subjects respond to A more than to C. For examples of this task in the recent neurobiological literature, see Arnsten et al. (1997; *Neurobiology of Aging* 18: 21-28) or Jentsch et al. (2002; *Neuropsychopharmacology* 26: 183-190). I'm sure this form of the task has a much longer history, but I don't have my copy of Mackintosh (1974; "The Psychology of Animal Learning") to hand!

Configuring the task

Parameters for Reversal Learning [X]

Maximum number of trials (0 for no limit): Maximum time (min) (0 for no limit):

Max time to wait for a response (s) (0 for no limit): Play Marker 1 sound at start of trial?

Time between trials (s): from to s Leave correct stimulus on screen during reward

Reverse within a session ... whenever subject performs of the last trials correctly

Stimulus A:

Stimulus B:

Begin with B+ (rather than A+)

p(reward | correct) = 1; p(reward | incorrect) = 0 p(reward | correct) = p(reward | incorrect) =

Spatial location of correct stimulus randomized in pairs rather than being fully random

What form of correction procedure should be used?

None

ANTIBIAS. When A consecutive presses to one side, CP starts.
CP presents correct stim on other side. CP runs until a total of B correct responses (not necessarily consecutive). At end of CP, trial sequence resumes exactly as it was. Max #trials applies to ALL types of trial.
 Session begins with CP.
... in which case C correct trials needed before the initial CP stops, and correct stim is initially on
 Left Right

HARSH. Whenever a trial is performed incorrectly, CP starts.
CP consists of re-presenting the same trial up to A times. As soon as the subject gets a CP trial correct, the CP terminates. Max #trials applies only to NON-CORRECTION trials.

VERYHARSH. Just like Harsh, but max #trials applies to ALL types of trial.

ANTIBIAS_NONCORRECTIONLIMIT. Just like Antibias, but max #trials applies only to NONCORRECTION trials

parameter A = parameter B = parameter C =

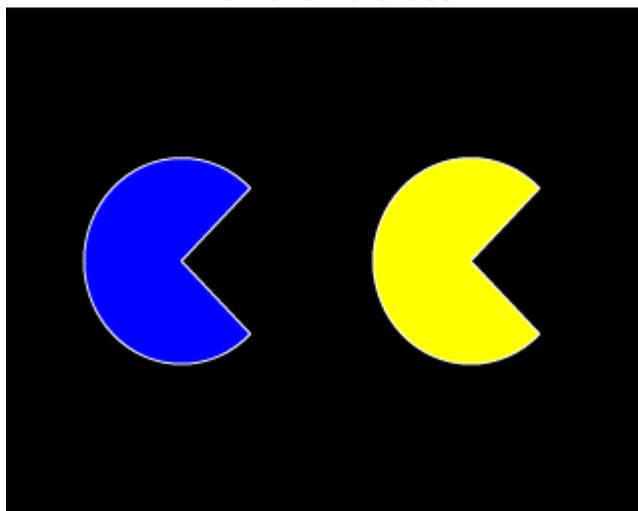
- **Maximum number of trials.** When the subject has performed this number of trials, the task ends. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Maximum time.** When this time elapses, the task is terminated as soon as the current trial has finished. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Maximum time to wait for a response.** If the subject fails to make a response within this time, the subject fails the trial. (You may specify 0 for no limit.)
- **Play Marker 1 sound at start of trial?** Fairly obvious.
- **Time between trials.** Specify a minimum and a maximum intertrial time (they may be the same). The actual time is chosen with a rectangular probability distribution within these values.
- **Leave correct stimulus on screen during reward.** If unticked, all stimuli vanish when one is chosen. If ticked, then if the incorrect stimulus is chosen both vanish; if the correct stimulus is chosen, the incorrect stimulus vanishes and the correct stays for a while (while reward is being delivered).
- **Reverse within a session... when subject performs X of the last Y trials correctly.** Fairly obvious, I hope. Set the value of X and Y in the boxes.
- **Use three stimuli rather than two?** Choose whether you want a two-stimulus task or a three-stimulus task.
- **Stimuli.** Choose the stimuli required by the task (A, B, +/- C). In a three-stimulus task, stimulus C is never correct.
- **Begin with B+ (rather than A+)?** If you want stimulus B to be correct initially, tick this box. Otherwise, A will be correct.
- **Probability of reward given a correct/incorrect response.** A conventional reversal procedure has $p(\text{reward} | \text{correct}) = 1$ and $p(\text{reward} | \text{incorrect}) = 0$. However, if you would like a

fully probabilistic reversal task, untick this box. You may then specify $p(\text{reward} | \text{correct})$ and $p(\text{reward} | \text{incorrect})$ directly. For example, if you specify 0.8 and 0.2, then correct responses would be rewarded 80% of the time, while incorrect responses would be rewarded 20% of the time.

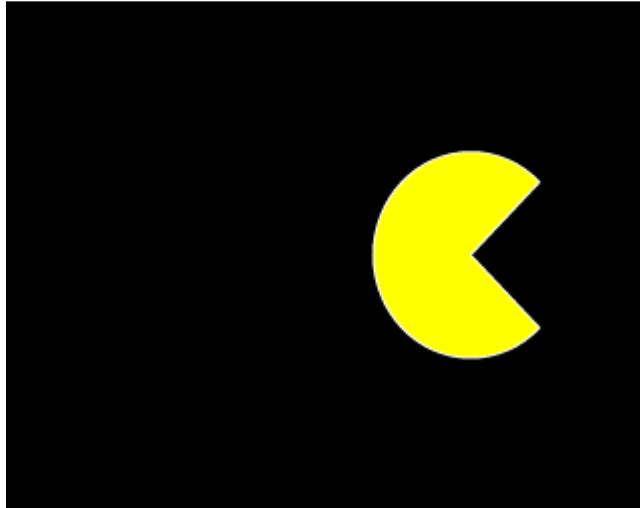
- If you reward some "incorrect" responses, and you have chosen the option "Leave correct stimulus on during reward", the program will leave the **chosen** stimulus on (i.e. one that is notionally "incorrect", but is being rewarded on this trial). This seems the only consistent thing to do. Essentially, a probabilistic task blurs the definition of "correct" and "incorrect", so the option is best described as "Leave chosen stimulus on if it's rewarded"!
- **Spatial location of correct stimulus randomized in pairs...**
 - If this box is not ticked, the location of the correct stimulus is chosen at random for each trial (and, in the three-stimulus task, the location of the "incorrect but correct in the past" and the "never correct" stimuli are similarly chosen at random).
 - If this box is ticked, then the locations are randomized in pairs (for the two-stimulus task), meaning that in every pair of trials, the correct stimulus is on the left on one trial and on the right in the other, but the order of those two trials within the pair is random. For the three-stimulus task, there are six possible spatial combinations (ABC, ACB, BAC, BCA, CAB, CBA) and in every six trials one of these combinations will be used, with the order within the group of six being random.
- **Correction procedure.** Choose the type of correction procedure (CP) you wish to use. The meaning of the types of correction procedure is explained carefully in the dialogue box. Note that if you allow within-session reversals and your subject achieves the criteria for reversing, any ongoing correction procedure is cancelled, and all correction procedure counts are reset.
 - If the "antibias" correction procedure is employed with a three-stimulus task and the subject perseverates in the middle, then the correct stimulus is randomly assigned to the left or the right location for the correction procedure. If it perseverates on the left, then the correct stimulus is assigned to the right-hand side; if it perseverates on the right, the correct stimulus is assigned to the left-hand side. In all cases, once the correction procedure has determined where the correct stimulus is to be, it chooses the location of the "incorrect but once correct" stimulus and the "never correct" stimulus (C) at random.

Screenshots of the task

The moment of choice.



In this case, the subject responded correctly and correct stimuli are being left up during reward:



1.6.4 Delayed Matching/Non-matching to Sample

About the task

The Marker 1 sound is played to signal the start of a trial. An object is shown in the centre of the screen (Phase 1). (Optionally, the subject has to touch this object; optionally, it can be rewarded for doing so.) The object vanishes, and a delay ensues. After this delay, the object is re-presented together with one or more other objects (Phase 2), heralded by the Marker 2 sound.

In **delayed matching to sample (DMTS)**, the subject must touch the object that was shown first. In **delayed non-matching to sample (DNMTS)**, it must touch the new object. (Both test the subject's ability to remember information about the first object during the delay; the matching/nonmatching option is typically used to account for or overcome a subject's species-specific natural tendency to select either familiar or novel stimuli.)

Correct responses are rewarded; incorrect responses are punished. Optional correction procedure: if correction is switched on, failed trials are repeated once (see below).

Configuring the task

Parameters for Delayed Matching/Non-matching To Sample (DMTS/DNMTS)

Maximum number of trials (0 for no limit): Maximum time (min) (0 for no limit):

Max time to wait for a response when one required (s):

Time between trials (s): from to s

Phase 1

Must touch Phase 1 stimulus

Rewarded for touching Phase 1 stimulus

Phase 1 stimulus duration (s):

Phase 2

Matching

Correction phase if subject fails Phase 2

Stimuli

Add Remove Up Down

0. IDEDpredef_circle_black
1. IDEDpredef_circle_green
2. IDEDpredef_circle_red
3. IDEDpredef_hat_cyan
4. IDEDpredef_hat_magenta
5. IDEDpredef_pentagram_cyan
6. IDEDpredef_pentagram_magenta

Cyclical Start with stimulus # and work down

Random Try to avoid stimuli used in last trials

Levels (determining the delay between phase 1 and 2)

Starting level:

Do not alter the level

Choose the level randomly for each trial

Increase level every X trials

Increase level when X of last 20 trials performed correctly

X:

Delays

Add Remove Up Down

0.00 (Level 1)
5.00 (Level 2)
10.00 (Level 3)
20.00 (Level 4)

OK Cancel

- **Maximum number of trials.** When the subject has performed this number of trials, the task ends. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Maximum time.** When this time elapses, the task is terminated as soon as the current trial has finished. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Maximum time to wait for a response.** If the subject fails to make a response within this time, the subject fails the trial. (This time limit applies to Phase 1, if you require your subjects to touch the Phase 1 stimulus, and Phase 2, and the correction procedure if one is used.)
- **Time between trials.** Specify a minimum and a maximum intertrial time (they may be the same). The actual time is chosen with a rectangular probability distribution within these values.
- **Must touch phase 1 stimulus.** If this is selected, then the subject must respond to the Phase 1 stimulus in order to proceed to phase 2. If you choose this option, you may also choose whether or not the subject should be **rewarded for touching the Phase 1 stimulus**. If you do not want your subject to have to touch the stimulus, you must specify the **Phase 1 stimulus duration** instead.
- **Matching.** If this is ticked, the task is delayed matching to sample. Otherwise, it's delayed non-matching to sample.
 - For DNMTS, the original object is shown together with one novel object.
 - For DMTS, too, the original object is shown together with one novel object (distractor).
- **Correction phase if subject fails Phase 2.** Optionally, phase 2 can be repeated immediately (once) if the subject fails it the first time.
- **Stimuli.** This shows the list of available stimuli. You cannot put a stimulus into the list more than once. Click **Add** and **Remove** to add/remove stimuli. You can also choose one of two

methods for choosing the stimuli for each trial:

- **Cyclical.** The program begins with a specified **stimulus number** and selects stimuli for each trial by working down the list, resuming at the start of the list if/when it runs out of stimuli to use at the bottom of the list.
- **Random.** The program picks a set of stimuli to use at random on each trial. You will need to fill in "**Try to avoid stimuli used in last X trials**"; the program will try not to choose any stimuli that have appeared (or were scheduled to appear, in the case of failure at phase 1) in the last X trials.

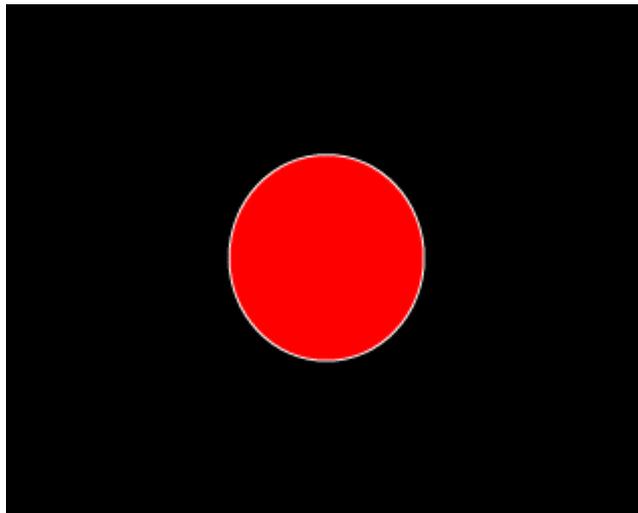
It's your responsibility to ensure that enough stimuli are in the list!

- **Levels.** On the right-hand side of the screen is the list of memory delays (delays between phase 1 and phase 2) that correspond to levels in the task. You may **add, remove, or re-order** the levels with the buttons next to the list. You may then choose the **starting level**, and the **method** by which the task chooses a level for each trial. (You can have the level fixed, or chosen randomly for each trial, or you can increase the level by one every X trials, or you can increase the level by one when X of the last 20 trials have been performed correctly. Set your chosen value of X in the box.)

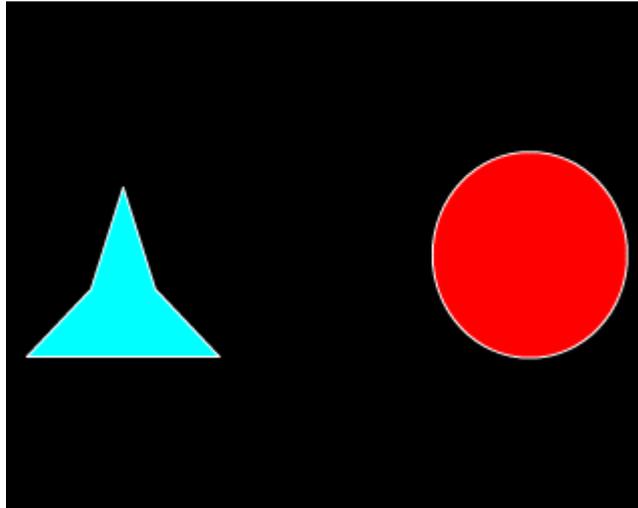
Options for reward and punishment are set in the [General Parameters](#) section; visual objects are defined in the [Visual Object Library](#).

Screenshots from the task

Phase 1



Phase 2. DMTS or DNMTS? You just can't tell...



1.6.5 Paired-Associates Learning

About the task

The basic principle of a paired-associates learning task is to associate multiple pairs of stimuli, and test recall. An example from human games is the Klimt Memo game; you have a deck of 72 cards with 36 images from paintings by Gustav Klimt. You scatter the cards face-down on a table. Each player turns over cards two at a time, trying to remember where each image is; if you turn over two identical cards, you remove the pair from the table; otherwise, you replace them in the face-down position. The objective is to turn over as many matching pairs as possible. The basic association being tested is a {stimulus, location} pair, and you must remember many of these to do well in the game.

Both the human and monkey versions of CANTAB use a stimulus-location pair. One might call the test "delayed matching-sample-to-location". In human CANTAB, subjects are first shown up to 8 stimuli in different locations around the edge of the screen. In the test phase, they are then shown a series of single stimuli in the centre of the screen and asked to indicate the location in which each stimulus was presented in the first phase. In monkey CANTAB, only 4 locations are used and the exemplar is not in the centre of the screen; instead, the exemplar is shown in every possible location and the monkey must choose the one location in which that stimulus was previously shown. For the task to be described as PAL, >1 stimulus must be used (otherwise it's just delayed matching to location).

The rat version presented here offers up to three locations.

The sample stage begins with the Marker 1 sound; thereafter, individual stimuli are presented. The subject may be required to touch them. After all the stimuli have been sampled, there is a "memory delay". The choice phase begins with the Marker 2 sound. The subject will be offered multiple choices, one for each stimulus that was seen in the sample phase. Each time, the subject must touch the location in which that stimulus appeared in the sample phase.

Configuring the task

Parameters for Paired-Associates Learning (PAL)

Maximum number of trials (0 for no limit): Maximum time (min) (0 for no limit):

Max time to wait for responses (s) (0 for no limit): Time between trials (s): from to s

Sample phase

Must touch stimuli in sample phase

Rewarded for touching sample phase stimuli

Sample phase stimulus duration (s):

Time between sample stimuli (s):

Memory delay (s): from to s

Choice phase

Reward each correct choice

Time between choice presentations (s):

Stimuli

"Empty box" object:

Target stimuli:

0. IDEDpredef_circle_black
1. IDEDpredef_circle_green
2. IDEDpredef_circle_red
3. IDEDpredef_hat_cyan
4. IDEDpredef_hat_magenta
5. IDEDpredef_pentagram_cyan
6. IDEDpredef_pentagram_magenta

Cyclical Start with stimulus # and work down

Random Try to avoid stimuli used in last trials

Mark responses that aren't rewarded or punished with the Marker 3 sound

End trial on first error (of omission or commission)

Repeat failed trials ... up to times (0 for no limit) Shuffle sample/choice order when repeating trials

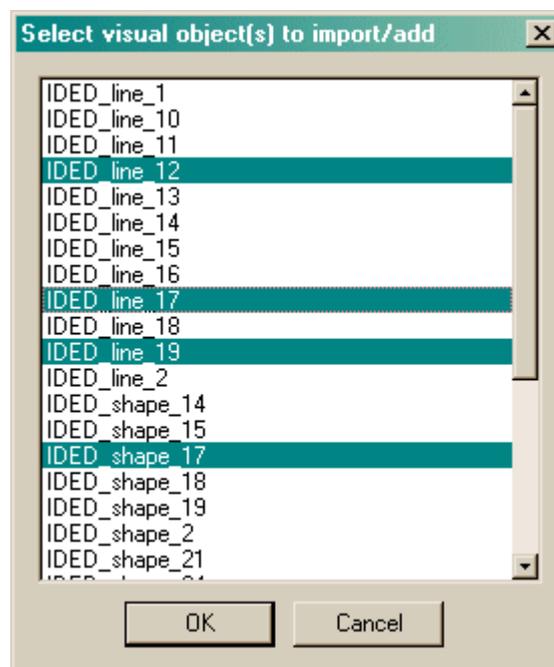
Trial scheme:

- **Maximum number of trials.** When the subject has performed this number of trials, the task ends. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Maximum time.** When this time elapses, the task is terminated as soon as the current trial has finished. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Maximum time to wait for a response.** If the subject fails make a response within this time, the subject fails the trial. (This time limit applies to the sample phase only if you require your subjects to touch the sample stimulus; it always applies to the choice phase.)
- **Time between trials.** Specify a minimum and a maximum intertrial time (they may be the same). The actual time is chosen with a rectangular probability distribution within these values.
- **Must touch stimuli in sample phase.** If this is selected, then the subject must respond to each sample stimulus. If you choose this option, you may also choose whether or not the subject should be **rewarded for touching sample stimuli**. If you do not want your subject to have to touch the stimulus, you must specify the **Sample stimulus duration** instead.
- **Time between sample stimuli.** This is the time between consecutive sample stimuli.
- **Memory delay.** This is the time between the last sample stimulus and the first of the choices. Specify a minimum and a maximum time (they may be the same). The actual time is chosen with a rectangular probability distribution within these values.
- **Reward each correct choice.** If this is selected, every time the subject makes a correct choice (in the choice phase, obviously), it gets reward. If this option is not selected, subjects only get rewarded at the end of the trial if they have not made any choice errors. Regardless of this option, any time the subject makes an error, it is punished (as specified in the [General Parameters](#)).
- **Time between choice presentations.** This is the time between consecutive choice presentations.
- **Mark responses that aren't rewarded or punished with the Marker 3 sound.** If this is chosen, then any responses that are neither rewarded nor punished are marked with the Marker 3 sound (as specified in the [General Parameters](#)).
- **End trial on first error.** If this is selected, then whenever the subject makes a mistake, the trial ends.

- **Repeat failed trials.** If this is selected, then if the subject fails to get all the choice presentations correct (or if you have "End trial on first error" selected and it gets something wrong in the sample phase) then the trial will be repeated. Set the **number of times a trial may be repeated**, too. Optionally, the order in which samples and choices are presented can be shuffled for repeated trials (while holding the stimulus-location pairs constant); to do this, tick **Shuffle sample/choice order when repeating trials**.
- **Trial scheme.** Trial schemes define the number and type of trials you will use. Click **Set** to choose a trial scheme. Click **Define** to define trial schemes. See below for details.
- **"Empty box" stimulus.** Choose the stimulus to be used as an "empty box" marker (for more details, see "Block Specification", below).
- **Stimuli.** This shows the list of available stimuli. You cannot put a stimulus into the list more than once. Click **Add** and **Remove** to add/remove stimuli. You can also choose one of two methods for choosing the stimuli for each trial:
 - **Cyclical.** The program begins with a specified **stimulus number** and selects stimuli for each trial by working down the list, resuming at the start of the list if/when it runs out of stimuli to use at the bottom of the list.
 - **Random.** The program picks a set of stimuli to use at random on each trial. You will need to fill in "**Try to avoid stimuli used in last X trials**"; the program will try not to choose any stimuli that have appeared (or were scheduled to appear, in the case of failure at phase 1) in the last X trials.

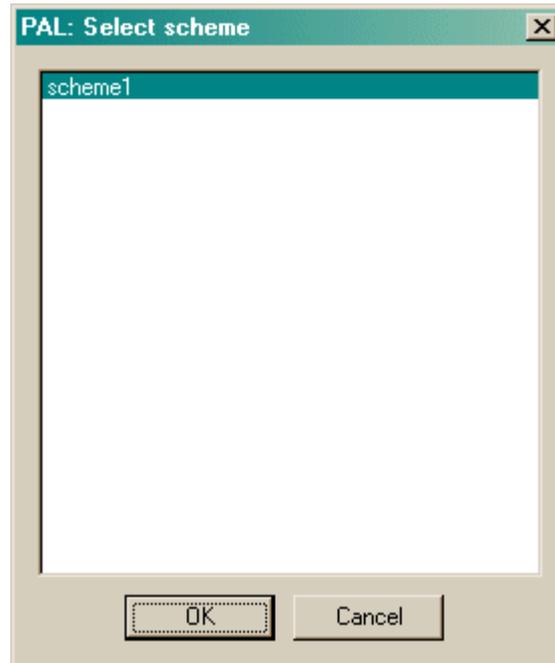
It's your responsibility to ensure that enough stimuli are in the list! The program will complain if you try to start it and there aren't enough stimuli to provide a unique stimulus in each location for every trial in your scheme. It won't complain if it needs to re-use stimuli from trial to trial.

Incidentally, when you **Add** stimuli, you can choose several at once by holding down the Shift key as you click on stimuli:



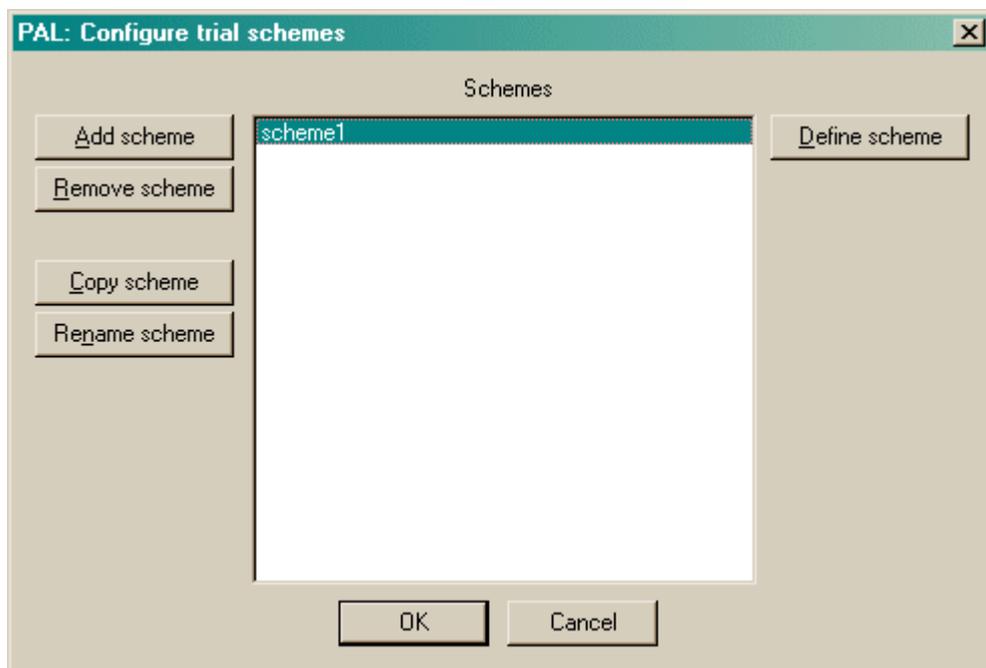
Set trial scheme

When you click this button, you can choose from a list of defined trial schemes.



Trial schemes

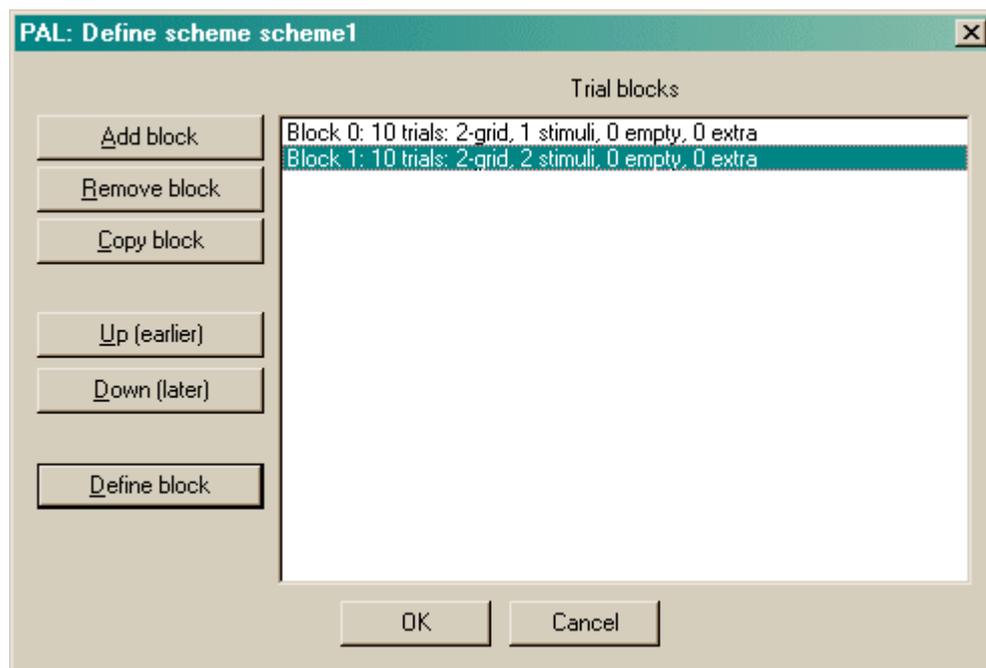
When you click **Define schemes** from the main parameters dialogue box, you can see the schemes. You may add, remove, copy, and rename schemes. Click a scheme and then click **Define scheme** to edit one particular scheme.



If a scheme is mis-configured, a warning message will appear when you click on the scheme.

Define scheme

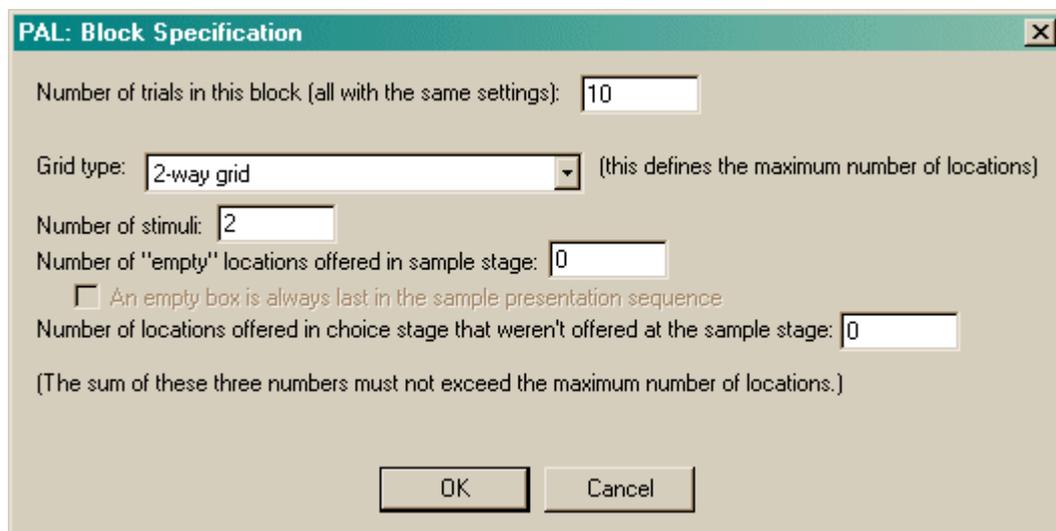
When you click this button, you can edit an individual scheme. Schemes consist of **blocks** of trials. Here, there are two blocks. You may add, remove, copy, and re-order blocks. Click **Define block** to edit one particular block.



If a block is mis-configured, a warning message will appear when you click on the block.

Block specification

Here's where you can edit the block.



- **Number of trials.** Set the number of trials in this block. All these trials will run with the same settings (though individual stimuli and locations will be chosen at random for each trial within the block).
- **Grid type.** Choose from a variety of grid patterns; this setting determines the number of available locations (actually, in the rat version of the task, it's always 2). The three numbers that follow must not add up to more than the number of available locations!
- **Number of stimuli.** The number of stimuli presented in the sample and choice phases.
- **Number of "empty" locations offered in sample stage.** You may show the "empty box" object at a number of locations in the sample phase, if you wish. This can be useful to enforce responding to particular areas of the screen. Set this number here.

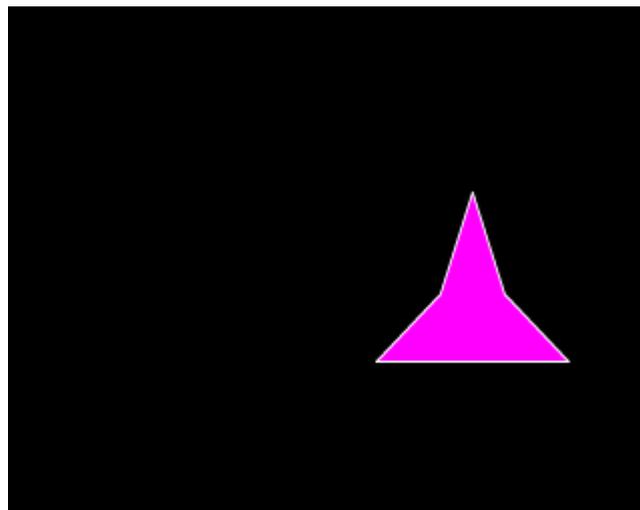
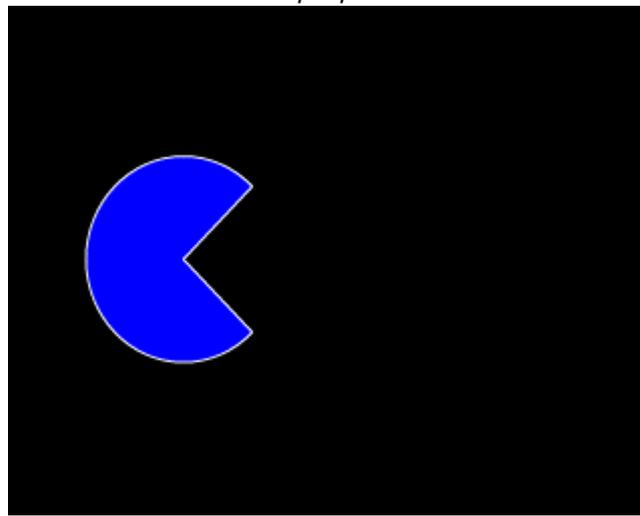
- **An empty box is always last in the sample presentation sequence.** This option allows you to enforce the rule that one of the empty boxes will always be shown last in the sample phase.
- **Number of locations offered in the choice stage that weren't offered at the sample stage.** In the choice phase, the task will always use the locations in which sample stimuli were presented (that's obvious), and it'll also use the locations at which you displayed the "empty box" in the sample phase, if any. You can also offer locations in the choice phase that were never offered in the sample phase. Choose that number of extra locations here.

To remind you: the sum of the last three numbers musn't exceed the number of locations available in the grid type you've selected; otherwise, warnings will appear showing that your scheme/block is mis-configured.

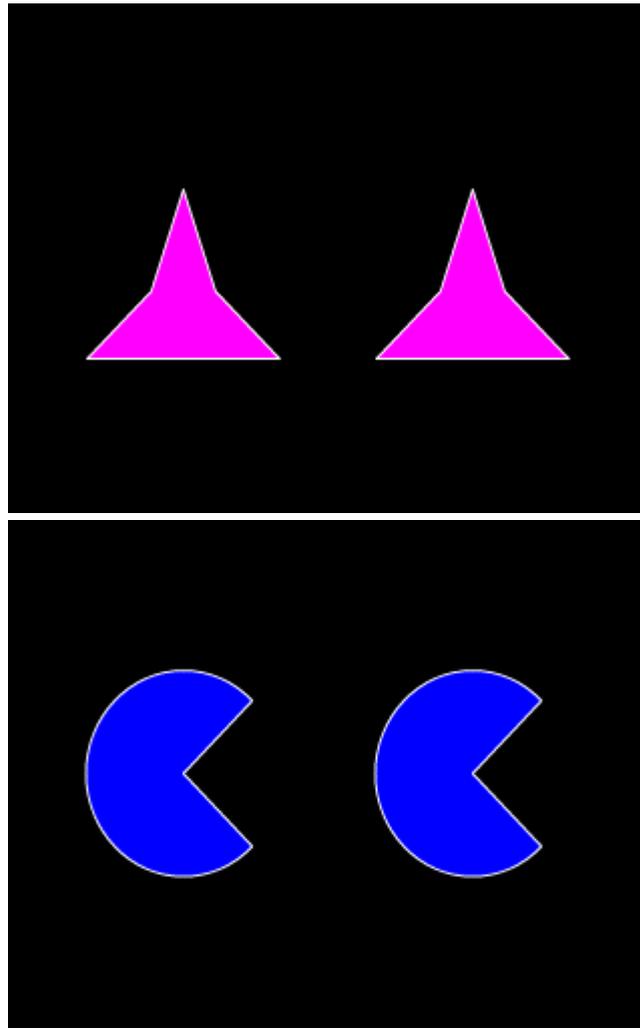
Screenshots from the task

This is a very simple example, with two stimuli, no "empty boxes" shown in the sample phase, and no extra locations offered in the choice phase.

Sample phase



Choice phase.



1.6.6 Simple Schedules of Reinforcement

About the task

Schedules of reinforcement using a touchscreen manipulandum.

Configuring the task

What constitutes "a reinforcer" is defined in the [General Parameters](#). This allows you to set the reward sound and define what sound is associated with the presentation of a reward.

Parameters for Simple Schedules of Reinforcement

Maximum number of reinforcers (0 for no limit):

Maximum time (min) (0 for no limit): picked between and

Response object:

(800,600)

Mark responses aurally (with the Marker 3 sound)

Mark responses visually

Marker object:

Time to show marker object for (s):

Schedule:

Parameters: Minimum interval (s)

Maximum interval (s)

Timeout at reinforcement Timeout duration (s):

PR schedules end based on time since last response (not reward)

- **Maximum number of reinforcers.** Once this number of rewards has been delivered, the task stops. (Specify 0 for no limit.) You must specify either a maximum number of rewards, or a maximum time, or both.
- **Maximum time (min).** Once this time limit has been reached, the task stops. (Specify 0 for no limit.) You must specify either a maximum number of rewards, or a maximum time, or both.
- **Response object.** The picture that the subject has to touch. Click **Set** to choose an object from the [visual object library](#).
- **Mark responses aurally.** If this is selected, the Marker 3 sound will be played to inform the subject that it has touched the picture successfully. (The schedule pauses while this sound is played.)
- **Mark responses visually.** If this is selected, you may replace the response object with another picture for a brief period of time, to indicate visually that the subject has made a successful response. The marker object is shown here; click **Set** to choose one from the [visual object library](#) and choose the **Time to show marker object for (s)**. The schedule is paused while the marker object is being shown.
- **Schedule.** Choose the schedule, and up to three **parameters** associated with the schedule. The schedules are:
 - **CRF - continuous reinforcement (FR-1).** One reinforcer per response.
 - **EXT - extinction.** No reinforcers.
 - **FR x - fixed ratio.** One reinforcer per x responses.
 - **VR x to y - variable ratio (specifying min, max).** After a variable number of responses (randomly chosen from *min* to *max* inclusive), one reinforcer is delivered.
 - **RR x - random ratio.** $P(\text{reinforcer} | \text{response}) = 1/x$.

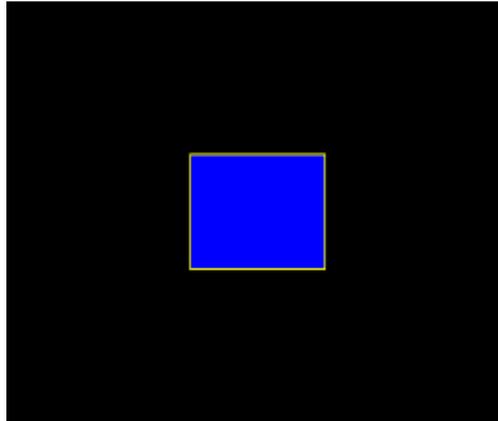
- **PROB p - probabilistic.** $P(\text{reinforcer} \mid \text{response}) = p$.
- **FI x - fixed interval.** The first response after x seconds is reinforced. The *first response* of the schedule is also reinforced.
- **RI x - random interval.** Reinforcement is set up on a random-time schedule (see below); after reinforcement has been set up, the next response is reinforced.
- **VI x to y - variable interval (specifying min, max).** After a variable time (from *min* to *max* seconds), the next response is reinforced.
- **FT x - fixed time (NONCONTINGENT).** No object is shown on the screen. Reinforcement is delivered every x seconds.
- **VT x to y - variable time (specifying min, max) (NONCONTINGENT).** No object is shown on the screen. The schedule waits for between *min* and *max* seconds, then delivers a reinforcer, then repeats.
- **RT x - random time (NONCONTINGENT).** No object is shown on the screen. Every second, $p(\text{reinforcer delivered this second}) = 1/x$. Thus, on average, reinforcement is delivered once every x seconds, but the subject cannot predict the likelihood of reinforcement based on how long it has waited (unlike a typical VT schedule).
- **PR - progressive ratio - add one (1,2,3,4...)** - progressive ratio schedule, adding one to the ratio requirement at each step. The schedule termination is determined by the parameter; if *parameter* is >0 , then when *parameter* minutes have elapsed since the last reinforcer, the schedule stops. I suggest 60 as a sensible value. Progressive ratio schedules were invented by William Hodos [Hodos W (1961); Progressive ratio as a measure of reward strength, *Science* 134: 934-944].
- **PR - progressive ratio - double (1,2,4,8...)** - progressive ratio schedule, doubling the ratio requirement at each step. The schedule termination is determined by the parameter; if *parameter* is >0 , then when *parameter* minutes have elapsed since the last reinforcer, the schedule stops. I suggest 60 as a sensible value.
- **PR - progressive ratio - Fibonacci (1,1,2,3,5...)** - progressive ratio schedule with a Fibonacci progression. The schedule termination is determined by the parameter; if *parameter* is >0 , then when *parameter* minutes have elapsed since the last reinforcer, the schedule stops. I suggest 60 as a sensible value.
- **PR - progressive ratio - Roberts exponential ($A * \exp(\text{reinfnum} * B) - A$)** - progressive ratio schedule with an exponential progression, based on Roberts DCS & Richardson NR (1992), Self-administration of psychomotor stimulants using progressive ratio schedules of reinforcement, *Neuromethods* 24: 233-269 (eds Boulton A, Baker G, Wu PH; Humana Press). The ratio requirement is $(A * \exp(\text{reinforcer number} * B)) - A$, rounded to the nearest integer. Typically, A is 5. A typical schedule might have $B=0.2$; these values yield ratio requirements {1, 2, 4, 6, 9, 12, 15, 20, 25, 32, 40, 50, 62, 77, 95, 118, 145, 178, 219, 268, 328, 402, 492, 603, 737, 901, 1102, 1347, ...}. A steeper PR schedule is obtained with $B=0.25$, giving {1, 3, 6, 9, 12, 17, 24, 32, 42, 56, 73, 95, 124, 161, 208, 268, 346, 445, 573, 737, 948, 1218, 1566, 2012, 2585, 3321, 4265, 5478, ...} The schedule termination is determined by the other parameter (on the left, labelled (min)); if this parameter is >0 , then when this many minutes have elapsed since the last reinforcer, the schedule stops. I suggest 60 as a sensible value.
- **DELAYFR1 - FR1 with delayed reinforcement.** This is an FR1 schedule, but there is a delay between responding and reinforcement. This delay is the sole parameter (specified in seconds).

Special case: the first response on contingent interval schedules (FI, RI, VI) is always reinforced.

For noncontingent schedules, nothing is shown on the display.

See also [Notes on reinforcement timing](#).

Screenshot from the task



1.6.6.1 Notes on reinforcement timing

It is possible to cause reinforcement conflicts in this task. For example,

- You can respond on two schedules which share a reinforcement device, such that both schedules want to use the reinforcement device simultaneously.
- You can respond more rapidly than your reinforcement device allows (e.g. if you have FR1 for a 7-s pump infusion with no timeout, you can respond again while the pump is still pumping from your first response).

Exactly the same is true if your reinforcement is delayed from the response that caused it (though it's much harder to perceive what's going on).

SimpleSchedules ignores requests for reinforcement for devices that are busy. It records in the event log (text-based and ODBC) whether a scheduled reinforcement was *actually* given.

What happens when you request a timeout on a delayed-reinforcement schedule? Should the timeout occur at the time of the response, or at the time of the reinforcement? Well, a timeout is to stop you responding, so it should occur at the time you respond. This is what SimpleSchedules does.

What happens when your schedule wants to reinforce, and to give you a timeout, but your reinforcement device is busy? SimpleSchedules will not reinforce (because the device is busy reinforcing you anyway) but it *will* implement your timeout.

1.6.7 N-Pair Concurrent Visual Discrimination

About the task

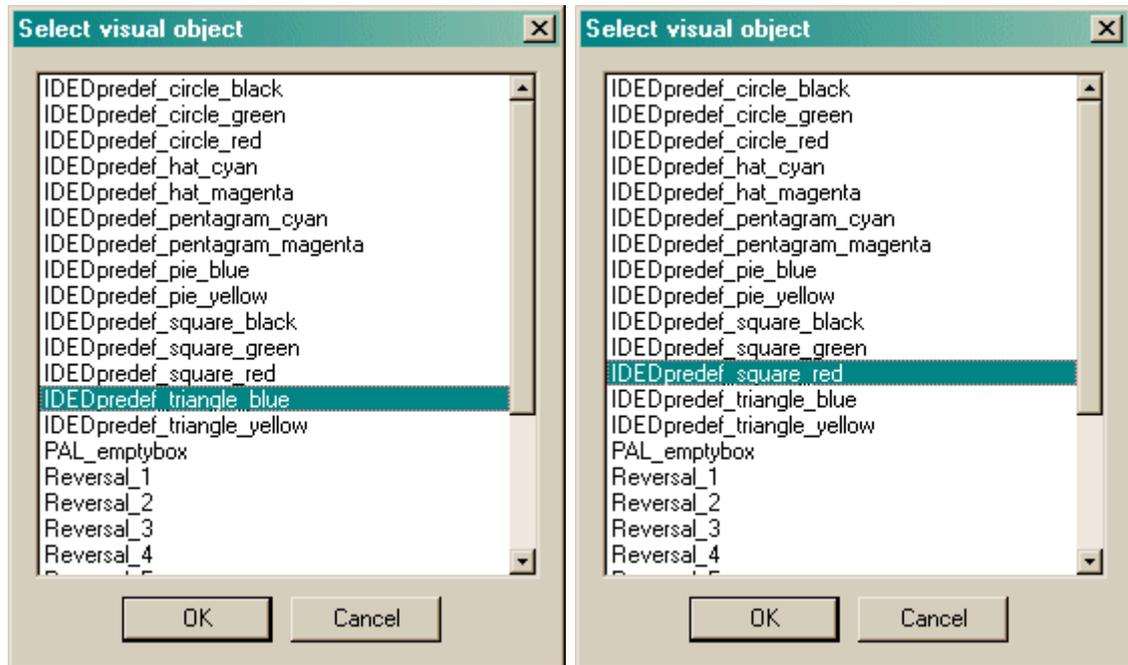
Implements n -pair concurrent visual discrimination. The subject is presented with a series of pairs of stimuli; in each pair, one is correct (+) and one is wrong (-).

Configuring the task

- **Maximum number of trials.** When the subject has performed this number of trials, the task ends. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Maximum time.** When this time elapses, the task is terminated as soon as the current trial has finished. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Finish when (X) of the last (Y) trials were performed correctly.** I hope this is self-explanatory.
- **Response criterion time.** If the subject fails to make a response within this time, the subject fails the trial. (You may specify 0 for no limit.)
- **Time between trials.** Specify a minimum and a maximum intertrial time (they may be the same). The actual time is chosen with a rectangular probability distribution within these values.
- **Leave correct stimulus on screen during reward.** If unticked, all stimuli vanish when one is chosen. If ticked, then if the incorrect stimulus is chosen both vanish; if the correct stimulus is chosen, the incorrect stimulus vanishes and the correct stays for a while (while reward is being delivered).
- **Pairs.** This lists the pairs in use. Each pair will have one correct stimulus (+) and one incorrect stimulus (-). Click **Add** or **Remove** to add or remove pairs. Click **Toggle** to swap round which of the pair is correct.
- **Pair choice method.** Which pair will be used on any given trial? You may either have the program choose a pair at random, or quasi-randomly (such that each of the pairs appears once in every n trials, where n is the number of pairs).

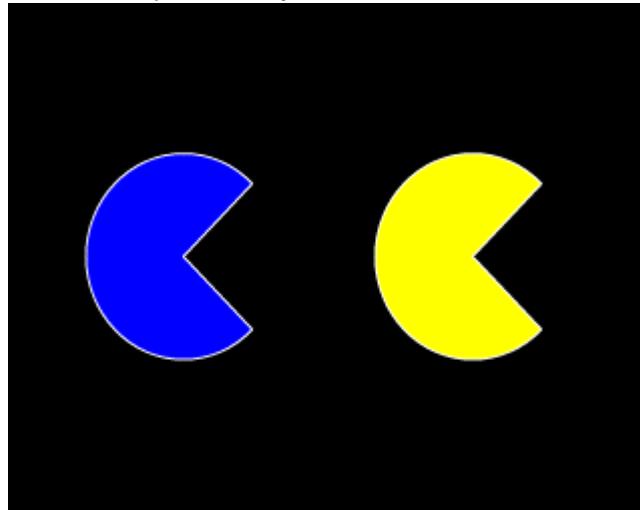
When you click **Add** to add a pair, you see the following dialogue box (just as a reminder):

Then this dialogue box appears twice, once for each stimulus to be chosen:

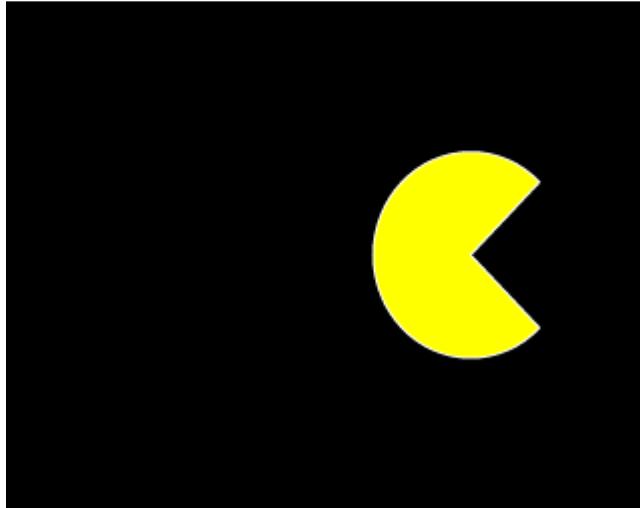


Screenshots from the task

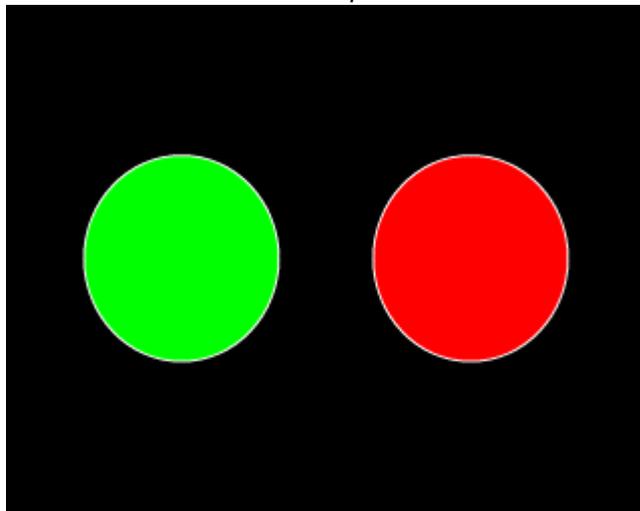
A pair... The yellow one is correct.



If you wish, the correct stimulus can remain behind (during reward) if the subject chooses correctly.



Another pair.



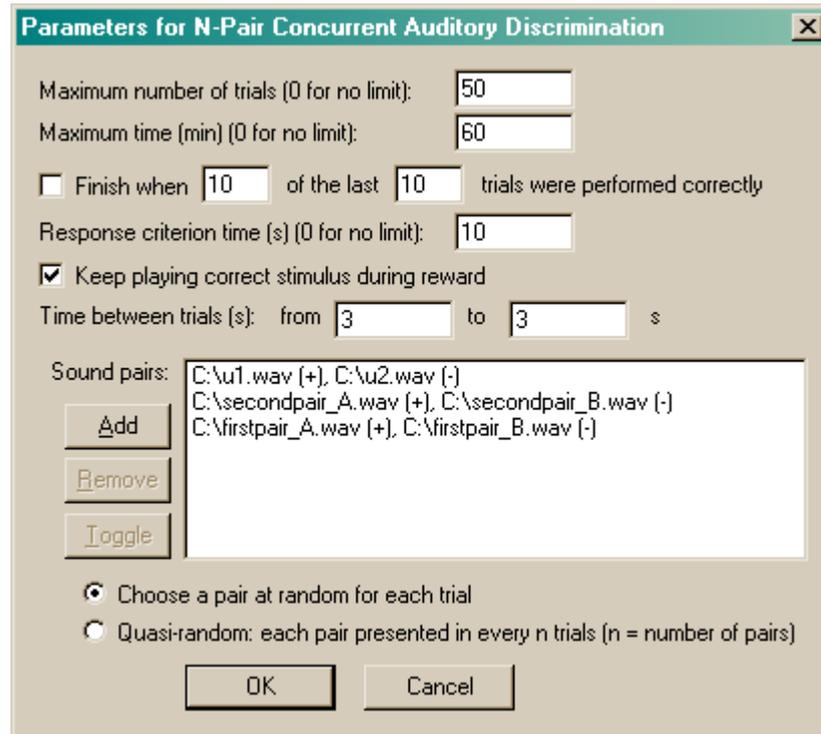
1.6.8 N-Pair Concurrent Auditory Discrimination

About the task

Implements n -pair concurrent auditory discrimination. The subject is presented with a series of pairs of stimuli; in each pair, one is correct (+) and one is wrong (-). One stimulus is played through the left speaker, the other is simultaneously played through the right.

Configuring the task

This task involves stereo auditory discrimination. Therefore, both "**Stereo system...**" and "**Subject is considered to have responded to a stimulus when it... approaches it**" must be selected in the [General Parameters](#).

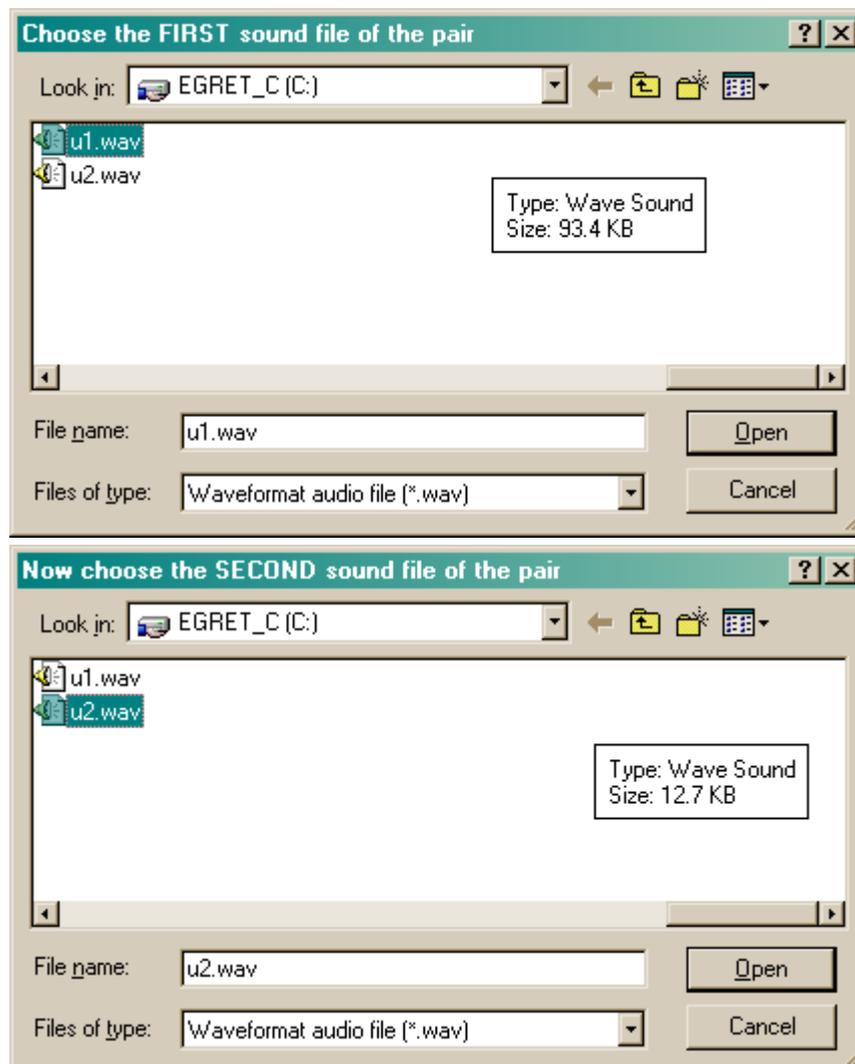


- **Maximum number of trials.** When the subject has performed this number of trials, the task ends. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Maximum time.** When this time elapses, the task is terminated as soon as the current trial has finished. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Finish when (X) of the last (Y) trials were performed correctly.** I hope this is self-explanatory.
- **Response criterion time.** If the subject fails to make a response within this time, the subject fails the trial. (You may specify 0 for no limit.)
- **Time between trials.** Specify a minimum and a maximum intertrial time (they may be the same). The actual time is chosen with a rectangular probability distribution within these values.
- **Keep playing correct stimulus during reward.** If unticked, all stimuli vanish when one is chosen. If ticked, then if the incorrect stimulus is chosen both vanish; if the correct stimulus is chosen, the incorrect stimulus vanishes and the correct stays for a while (while reward is being delivered).
- **Pairs.** This lists the pairs in use. Each pair will have one correct stimulus (+) and one incorrect stimulus (-). Click **Add** or **Remove** to add or remove pairs. Click **Toggle** to swap round which of the pair is correct.
- **Pair choice method.** Which pair will be used on any given trial? You may either have the program choose a pair at random, or quasi-randomly (such that each of the pairs appears once in every n trials, where n is the number of pairs).

When you click **Add** to add a pair, you see the following dialogue box (just as a reminder):



Then this dialogue box appears twice, once for each stimulus to be chosen:



Screenshots from the task

None; it's an auditory task! The screen is blank.

1.6.9 Audiovisual ID/ED

A rat attentional set-shifting task in which one stimulus dimension is auditory, and one visual. Stimuli are presented to the left or right of the chamber - so a "left" compound stimulus is a visual stimulus on the left-hand side of the screen, and an auditory stimulus from the left-hand speaker.

For details of the principles of intradimensional and extradimensional attentional set-shifting, see [About Set-Shifting](#).

Configuring the task

Note: changes by John Earl (August 2005, RatBat v1.3) not yet documented.

This task involves stereo auditory discrimination. Therefore, to run, both "**Stereo system...**" and

"Subject is considered to have responded to a stimulus when it... approaches it" must be selected in the [General Parameters](#).

Parameters for Audiovisual Intradimensional (ID)/Extradimensional (ED) Set-Shifting

Maximum number of trials (0 for no limit): Maximum time (min) (0 for no limit):

Max time to wait for a response (s) (0 for no limit): Play Marker 1 sound at start of trial? (Not advised.)

Time between trials (s): from to s Leave correct stimulus on during reward

Touches during ITI are punished by restarting the ITI

Increase stage ... when subject performs of the last trials correctly

Reversal stage after SD Reversal stage after CD Reversal stage after ID shift Reversal stage after ED shift

Starting stage:

Simple discrimination (A1+A2-)

Reversal (A2+A1-)

Compound discrimination (A1B1+A2B2- and A1B2+A2B1-)

Reversal (A2B1+A1B2- and A2B2+A1B1-)

Intradimensional (ID) shift (A3B3+A4B4- and A3B4+A4B3-)

Reversal (A4B3+A3B4- and A4B4+A3B3-)

Extradimensional (ED) shift (A5B5+A6B6- and A6B5+A5B6-)

Reversal (A5B6+A6B5- and A6B6+A5B5-)

What form of correction procedure should be used?

None

ANTIBIAS. When A consecutive presses to one side, CP starts. CP presents correct stim on other side. Compound stimulus composition is random. CP runs until a total of B correct responses (not necessarily consecutive). At end of CP, trial sequence resumes exactly as it was. Max #trials applies to ALL types of trial.

Session begins with CP.
... in which case C correct trials needed before the initial CP stops, and correct stim is initially on

Left Right

HARSH. Whenever a trial is performed incorrectly, CP starts. CP consists of re-presenting the same trial up to A times. As soon as the subject gets a CP trial correct, the CP terminates. Max #trials applies only to NON-CORRECTION trials.

A = B = C =

visual 1: Set

visual 2: Set

visual 3: Set

visual 4: Set

visual 5: Set

visual 6: Set

auditory 1: Set

auditory 2: Set

auditory 3: Set

auditory 4: Set

auditory 5: Set

auditory 6: Set

How should dimensions A and B be realized physically?

A is visual, B is auditory

A is auditory, B is visual

How should compound stimulus composition (e.g. A1B1/A2B2 v. A1B2/A2B1) and {left or right correct} be determined?

Stim. comp. random; L/R correct random

Stim. comp. randomized in pairs; L/R correct random

Stim. comp. random; L/R correct randomized in pairs

{Stim. comp., L/R correct} randomized in quads

OK Cancel

- **Maximum number of trials.** When the subject has performed this number of trials, the task ends. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Maximum time.** When this time elapses, the task is terminated as soon as the current trial has finished. (You may specify 0 for no limit, though you must specify a limit on the number of trials, the time, or both.)
- **Maximum time to wait for a response.** If the subject fails to make a response within this time, the subject fails the trial.
- **Play Marker 1 sound at start of trial?** Fairly obvious.
- **Time between trials.** Specify a minimum and a maximum intertrial time (they may be the same). The actual time is chosen with a rectangular probability distribution within these values. The time between trials starts **after** the reward or punishment from the previous trial has finished.
- **Leave correct stimulus on during reward.** If unticked, all stimuli vanish when one is chosen. If ticked, then if the incorrect stimulus is chosen both vanish; if the correct stimulus is chosen, the incorrect stimulus vanishes and the correct stays for a while (while reward is being

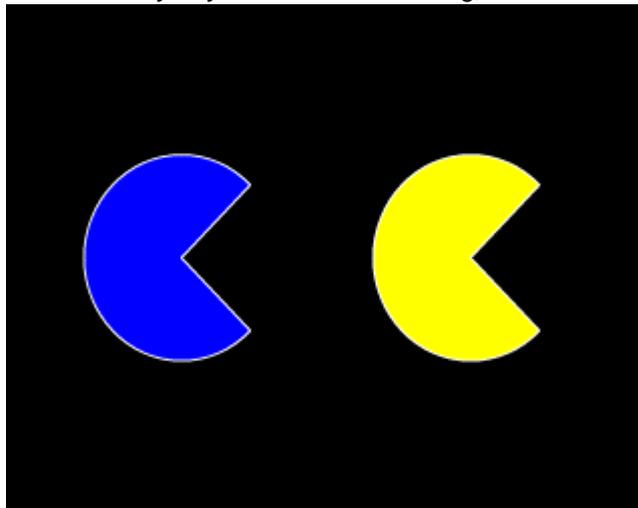
delivered).

- **Increase stage... when subject performs X of the last Y trials correctly.** Fairly obvious, I hope. Set the value of X and Y in the boxes.
- **Give a reversal stage after the simple discrimination.** Enables/disables the "SD reversal" phase.
- **Give a reversal stage after the compound discrimination.** Enables/disables the "CD reversal" phase.
- **Give a reversal stage after the ID shift.** Enables/disables the "ID reversal" phase.
- **Give a reversal stage after the ED shift.** Enables/disables the "ED reversal" phase.
- **Starting stage.** Choose the stage to start at for this session.
- **Correction procedure.** Choose the type of correction procedure you wish to use. The meaning of the types of correction procedure is explained carefully in the dialogue box. Note that if your subject shifts up a stage, any ongoing correction procedure is cancelled, and all correction procedure counts are reset.
- **Stimuli.** Choose the stimuli required by the task (visual 1, visual 2, auditory 1, auditory 2, etc.). An example is shown above.
- **Physical realization of dimensions A and B.** Dimensions A and B (as referred to in the stages shown on the left) can be assigned such that A=visual, B=auditory, or the other way round.
- **Randomization.** Choose the randomization technique used for the task. For compound stimuli, in any given trial you can present either A1B1 / A2B2 compounds, or A1B2 / A2B1 compounds. Furthermore, the correct stimulus can be on the right or on the left. The stimulus composition and the "left/right correct" assignment can be fully random (with a very small chance that you get ten on the left consecutively correct, because that's what random means). Or, the "left/right correct" assignment can be randomized in pairs (meaning that in pair of trials you get one "left correct" trial and one "right correct" trial, e.g. L,R - R,L - R,L - L,R - R,L - R,L - R,L - ...), with the stimulus composition totally random. Or the stimulus composition can be randomized in pairs (e.g. A1B1[/A2B2], A1B2[/A2B1] - A1B2[/A2B1], A1B1[/A2B2] - A1B2[/A2B1], A1B1[/A2B2] - ...) with the L/R assignment fully random. Or the four possible combinations (L versus R correct and A1B1/A2B2 versus A1B2/A2B1) can be randomized in groups of four trials (quads).

Screenshots from the task

Screenshots are not terribly informative, since this is a combined audiovisual task! But two stimuli are shown on the screen, so it could look like this:

Choose one. But maybe you should be listening, rather than looking...



1.6.9.1 About set-shifting

About the task

Presents two objects. One is correct and one is incorrect. If the subject touches the correct one, it is rewarded; if it touches the wrong one, it is punished. The discrimination varies in difficulty. The subject learns over several trials which stimuli are correct and which are incorrect; then we confuse it by changing the rules.

Stimuli can be *simple* (when they vary along only one dimension, such as shape, colour, intensity, or location), or *compound* (when they vary along more than one dimension).

Dimensional shifting in humans

In human testing, typical dimensions might be colour, shape, and number. A famous example of this kind of test is the Wisconsin Card Sorting Test (Grant & Berg, 1948, *J Exp Psychol* 38: 404), in which subjects must sort cards with a variable number of coloured shapes. They must discover the sorting rule solely by reinforcement from the experimenter. The rule is then changed: for example, they may be required to switch from "red correct, blue wrong" to "blue correct, red wrong" - a reversal. Alternatively, they may be given a new set of colours to learn - an intradimensional shift. Again, they may also be required to switch from "blue correct, red wrong" to "circles correct, squares wrong" - an extradimensional shift.

Dimensional shifting in non-human primates

In primate testing, two typical dimensions are "blue shapes" and "white squiggly lines". [See Dias R, Robbins TW, Roberts AC (1997). Dissociable forms of inhibitory control within prefrontal cortex with an analog of the Wisconsin Card Sort Test: restriction to novel situations and independence from "on-line" processing. *Journal of Neuroscience* 17: 9285-9297.] They use a large number of exemplars in each dimension (shapes and lines).

Basic test sequence in the general case

Assume there are two dimensions, A and B. There is a large catalogue of objects. Each object is assigned a value for each dimension (or is stated not to possess that dimension). For example, if the dimensions are colour and shape, then each object would be assigned a value for colour (e.g. "red") and shape (e.g. "triangle"). If there were a third dimension, of "superimposed wiggly line", a red triangle stimulus might have a value of "nonexistent" for this third dimension.

OK. For the sake of the following general prototype, keep in mind the primate tests in which the dimensions are typically "background shape" (values e.g. square, circle, triangle...) and "superimposed wiggly line" (similarly, with a variety of values).

- **1. Simple discrimination.** Displayed objects possess dimension A but not dimension B. Within dimension A, value 1 is designated correct and value 2 is designated incorrect. In other words, we reward subjects if $A()=1$ and punish them if $A()=2$. (We can run this discrimination for as long as we can find target objects such that $A(\text{object})=1$ and distractor objects such that $A(\text{object})=2$, as long as in both cases $B(\text{object})=\text{undefined}$.)
- **2. Simple reversal.** We carry on except that we now designate $A()=1$ as incorrect and $A()=2$ as correct. The subject has to reverse its responses to a previously-learned discrimination.
- **3. Compound discrimination.** We now introduce a second dimension, B, but ignore its value. Now objects are selected for presentation on the basis that $((A(\text{object})=1 \text{ or } A(\text{object})=2) \text{ and } B(\text{object}) \neq \text{undefined})$. The symbol " \neq " means "does not equal". The subject is rewarded if it selects an object such that $A(\text{object})=2$ and are punished if they select an object such that $A(\text{object})=1$. This continues the "A" discrimination from the previous, reversal stage.
- **4. Reversal of compound discrimination.** We reverse the discrimination along dimension A and continue to ignore B. The trials are just the same as the previous stage, but $A()=1$ is

correct and A() $=$ 2 is incorrect once more.

- **5. Intradimensional shift.** We introduce new values of dimension A; dimension B is still ignored. Objects are selected such that ((A(object) $=$ 3 or A(object) $=$ 4) and B(object) \neq undefined). Subjects are rewarded if A(object) $=$ 3 and punished if A(object) $=$ 4.
- **6. Extradimensional shift.** We switch our attention to dimension B and ignore A. Objects are selected such that ((B(object) $=$ 1 or B(object) $=$ 2) and A(object) \neq undefined). Subjects are rewarded if B(response) $=$ 1 and punished if B(response) $=$ 2.

What's ignored? By whom? A caveat. When a dimension B (or anything else) is present and being ignored, we would be wise to ensure that p(response is correct) is not dependent on the value of this dimension. Otherwise, our subject may well learn to discriminate on the dimension we (and it) are supposed to be ignoring. (Other dimensions like location or presentation order are important to consider here.) A dimension can be made irrelevant by holding the value of the dimension constant (e.g. always presenting stimuli in a single location, always making the background shape purple, etc.), or by randomizing it (e.g. always randomizing the locations of pairs of objects presented). If a dimension is randomized rather than held constant, the subject may *attempt* to learn the discrimination based on this dimension but cannot succeed.

Therefore, we need a library of objects with at least one object in each of the following categories:

A=1; B=undefined
 A=2; B=undefined
 A=1; B=anything except "undefined", constant or randomized
 A=2; B=anything except "undefined", constant or randomized
 A=3; B=anything except "undefined", constant or randomized
 A=4; B=anything except "undefined", constant or randomized
 A=anything except "undefined", constant or randomized; B=1
 A=anything except "undefined", constant or randomized; B=2

Having lots of objects in each of these categories may be a good thing behaviourally, as it may lead the subject to extract the general features of that dimension/value (for example, if A(1) is equivalent to colour(red), then having lots of red objects may lead the subject to learn that "red" is correct, and not "red triangle"). This is the approach taken by the primate tasks, but not by tasks used for rats (or, now, pigs).

Dimensional shifting in rats

Birrell & Brown (2000) implemented a set-shifting task in rats (Birrell JM & Brown VJ, 2000. Medial frontal cortex mediates perceptual attentional set shifting in the rat. *Journal of Neuroscience* 20: 4320-4). Rats dug in two bowls for food. The bowls has dimensions of (A) odour; (B) filling medium; (C) surface texture. They adopted a policy of changing all stimuli at times of ID or ED shifting (a "total change design", p4321, which is required for accurate interpretation of the difference between reversal learning and ED shifts; see p4323). Their test sequence was as follows (+ indicates correct stimuli, - incorrect, bold indicates the correct part of the stimulus):

Simple discrimination	A1(+) , A2(-)
Compound discrimination	A1/B1(+) , A2/B2(-) A1/B2(+) , A2/B1(-)
Reversal	A2/B1(+) , A1/B2(-) A2/B2(+) , A1/B1(-)
ID shift	A3/B3(+) , A4/B4(-) A3/B4(+) , A4/B3(-)
Reversal	A4/B3(+) , A3/B4(-) A4/B4(+) , A3/B3(-)
ED shift	B5/A5(+) , B6/A6(-) B5/A6(+) , B6/A5(-)

Reversal	B6/A5(+), B5/A6(-) B6/A6(+), B5/A5(-)
----------	--

This illustrates the general test sequence nicely.

The sequence used by RatBat

Much the same:

Simple discrimination	A1(+), A2(-)
Reversal	A2(+), A1(-)
Compound discrimination	A1/B1(+), A2/B2(-) A1/B2(+), A2/B1(-)
Reversal	A2/B1(+), A1/B2(-) A2/B2(+), A1/B1(-)
ID shift	A3/B3(+), A4/B4(-) A3/B4(+), A4/B3(-)
Reversal (optional)	A4/B3(+), A3/B4(-) A4/B4(+), A3/B3(-)
ED shift	B5/A5(+), B6/A6(-) B5/A6(+), B6/A5(-)
Reversal (optional)	B6/A5(+), B5/A6(-) B6/A6(+), B5/A5(-)

How long shall we test for? The usual measure on this task is trials (or errors) to criterion. Birrell & Bowman (2000) used a criterion of 6 consecutive correct responses. That seems reasonable (though make the number configurable).

Left/right position should be chosen randomly for each trial. Order of presentation of the two alternative pairs (e.g. which to present of A1/B1-A2/B2 or A1/B2-A2/B1) should be randomized in pairs of trials.

1.7 Before you start the task

Have you remembered to make your own copy of the supplied [database](#) and [set it up under ODBC](#)?

1.8 Results

RatBat always stores results in two places. One is a human-readable [text file](#). The other is a [database](#). (You choose the name of this file in the [main parameters dialogue box](#), and you can choose the database here as well.)

1.8.1 Text-based results file

A sample results file is shown below. The configuration information is shown first; the results follow. (There aren't very many results, because I got bored creating the file.) The results section is shown in bold.

Tasks generally produce results in a comma-delimited format with a header line giving the field names. (This format is itself suitable for importing into a relational database.) Some tasks produce other human-readable summary information.

I encourage you to think of this file as a backup. The [database](#) contains all this information

and can be used to retrieve both simple and highly detailed information about a subject's performance.

RatBat -- SUMMARY FILE

IDENTIFICATION

Subject: test
 Session: 9
 Date/time code: 26-Jun-2003 (20:47)
 Comment: (add your comment here)
 Summary file name: test-26Jun2003-2049-RatBat-summary.txt
 Default ODBC database:

Box: 0
 Client computer name: EGRET
 Server computer name: loopback

GENERAL PARAMETERS

Default media directory: C:\Program Files\RatBat\Media
 Link - Duration (s): 20
 Link - Play sound during link? N
 Link - Houselight on during link? N
 Adjust stimuli for central feeder? N
 Reward - Give pellet? N
 Reward - Pellets per reinforcement: 1
 Reward - Pellet pulse length (ms): 45
 Reward - Interpellet gap (s): 0.5
 Reward - Give pump? Y
 Reward - Pump duration (s): 5
 Reward - Pump contingent upon licking? Y
 Reward - If cont., lick pump time (s): 1
 Reward - Play sound? Y
 Reward - Activate extra reward device? N
 Reward - Extra reward device time (s): 5
 Punishment - Darkness? Y
 Punishment - Darkness time (s): 10
 Punishment - Play sound? Y
 Punishment - Extra punishment device? N
 Punishment - Extra device duration (s): 10
 Punishment - Give pump 2? N
 Sounds - Link sound is WAV? N
 Sounds - Link sound filename:
 Sounds - Link sound frequency (Hz): 200
 Sounds - Link sound type: Sine
 Sounds - Link sound duration (s): 1
 Sounds - Link sound level (0-100): 100
 Sounds - Reward sound is WAV? N
 Sounds - Reward sound filename:
 Sounds - Reward sound frequency (Hz): 1000
 Sounds - Reward sound type: Tone
 Sounds - Reward sound duration (s): 1
 Sounds - Reward sound level (0-100): 70
 Sounds - Punishment sound is WAV? N
 Sounds - Punishment sound filename:
 Sounds - Punishment sound frequency (Hz): 1500
 Sounds - Punishment sound type: Sawtooth
 Sounds - Punishment sound duration (s): 2

```

Sounds - Punishment sound level (0-100): 70
Sounds - Marker1 sound is WAV?          N
Sounds - Marker1 sound filename:
Sounds - Marker1 sound frequency (Hz):  500
Sounds - Marker1 sound type:             Tone
Sounds - Marker1 sound duration (s):     1
Sounds - Marker1 sound level (0-100):   70
Sounds - Marker2 sound is WAV?          N
Sounds - Marker2 sound filename:
Sounds - Marker2 sound frequency (Hz):  650
Sounds - Marker2 sound type:             Tone
Sounds - Marker2 sound duration (s):     1
Sounds - Marker2 sound level (0-100):   70
Sounds - Marker3 sound is WAV?          N
Sounds - Marker3 sound filename:
Sounds - Marker3 sound frequency (Hz):  800
Sounds - Marker3 sound type:             Tone
Sounds - Marker3 sound duration (s):     1
Sounds - Marker3 sound level (0-100):   70

```

VISUAL OBJECTS

```
-- Object 0: TT_hugebar
```

```
biggreenbar,rectangle 0 0 800 100 -penstyle solid -penwidth 1 -pencolour 255 255
255 -brushsolid 0 255 0
```

```
-- Object 1: Reversal_1
```

```
bitmap,bitmap 0 0 "Reversal_1.bmp" -height -1 -width -1 -clip
```

```
-- Object 2: Reversal_2
```

```
bitmap,bitmap 0 0 "Reversal_2.bmp" -height -1 -width -1 -clip
```

```
-- Object 3: Reversal_3
```

```
bitmap,bitmap 0 0 "Reversal_3.bmp" -height -1 -width -1 -clip
```

```
-- Object 4: Reversal_4
```

```
bitmap,bitmap 0 0 "Reversal_4.bmp" -height -1 -width -1 -clip
```

```
-- Object 5: Reversal_5
```

```
bitmap,bitmap 0 0 "Reversal_5.bmp" -height -1 -width -1 -clip
```

```
-- Object 6: Reversal_6
```

```
bitmap,bitmap 0 0 "Reversal_6.bmp" -height -1 -width -1 -clip
```

```
-- Object 7: Reversal_7
```

```
bitmap,bitmap 0 0 "Reversal_7.bmp" -height -1 -width -1 -clip
```

```
-- Object 8: Reversal_8
```

```
bitmap,bitmap 0 0 "Reversal_8.bmp" -height -1 -width -1 -clip
```

```
-- Object 9: SS_bluebox
```

```
bluerectangle,rectangle 0 0 200 200 -penstyle solid -penwidth 2 -pencolour 255 255
```

```
255 -brushsolid 0 0 255

-- Object 10: SS_redbox

redrectangle,rectangle 0 0 200 200 -penstyle solid -penwidth 2 -pencolour 255 255
255 -brushsolid 255 0 0

-- Object 11: IDEDpredef_circle_black

circle_black,ellipse 0 0 300 300 -penstyle solid -penwidth 3 -pencolour 255 255
255 -brushsolid 0 0 0
background_rectangle,rectangle 0 0 300 300 -penstyle null -penwidth 1 -pencolour 0
0 0 -brushsolid 0 0 0

-- Object 12: IDEDpredef_circle_green

circle_green,ellipse 0 0 300 300 -penstyle solid -penwidth 3 -pencolour 255 255
255 -brushsolid 0 255 0
background_rectangle,rectangle 0 0 300 300 -penstyle null -penwidth 1 -pencolour 0
0 0 -brushsolid 0 0 0

-- Object 13: IDEDpredef_circle_red

circle_red,ellipse 0 0 300 300 -penstyle solid -penwidth 3 -pencolour 255 255 255
-brushsolid 255 0 0
background_rectangle,rectangle 0 0 300 300 -penstyle null -penwidth 1 -pencolour 0
0 0 -brushsolid 0 0 0

-- Object 14: IDEDpredef_hat_cyan

hat_cyan,polygon 5 150 50 200 200 300 300 0 300 100 200 -alternate -penstyle solid
-penwidth 3 -pencolour 255 255 255 -brushsolid 0 255 255
background_rectangle,rectangle 0 0 300 300 -penstyle null -penwidth 1 -pencolour 0
0 0 -brushsolid 0 0 0

-- Object 15: IDEDpredef_hat_magenta

hat_magenta,polygon 5 150 50 200 200 300 300 0 300 100 200 -alternate -penstyle
solid -penwidth 3 -pencolour 255 255 255 -brushsolid 255 0 255
background_rectangle,rectangle 0 0 300 300 -penstyle null -penwidth 1 -pencolour 0
0 0 -brushsolid 0 0 0

-- Object 16: IDEDpredef_pentagram_cyan

pentagram_cyan,polygon 5 150 50 250 300 0 100 300 100 50 300 -winding -penstyle
solid -penwidth 3 -pencolour 255 255 255 -brushsolid 0 255 255
background_rectangle,rectangle 0 0 300 300 -penstyle null -penwidth 1 -pencolour 0
0 0 -brushsolid 0 0 0

-- Object 17: IDEDpredef_pentagram_magenta

pentagram_magenta,polygon 5 150 50 250 300 0 100 300 100 50 300 -winding -penstyle
solid -penwidth 3 -pencolour 255 255 255 -brushsolid 255 0 255
background_rectangle,rectangle 0 0 300 300 -penstyle null -penwidth 1 -pencolour 0
0 0 -brushsolid 0 0 0

-- Object 18: IDEDpredef_pie_blue

pie_blue,pie 0 0 300 300 300 0 300 300 -penstyle solid -penwidth 3 -pencolour 255
255 255 -brushsolid 0 0 255
background_rectangle,rectangle 0 0 300 300 -penstyle null -penwidth 1 -pencolour 0
0 0 -brushsolid 0 0 0
```

```

-- Object 19: IDEDpredef_pie_yellow

pie_yellow,pie 0 0 300 300 300 0 300 300 -penstyle solid -penwidth 3 -pencolour
255 255 255 -brushsolid 255 255 0
background_rectangle,rectangle 0 0 300 300 -penstyle null -penwidth 1 -pencolour 0
0 0 -brushsolid 0 0 0

-- Object 20: IDEDpredef_square_black

square_black,rectangle 0 0 300 300 -penstyle solid -penwidth 3 -pencolour 255 255
255 -brushsolid 0 0 0

-- Object 21: IDEDpredef_square_green

square_green,rectangle 0 0 300 300 -penstyle solid -penwidth 3 -pencolour 255 255
255 -brushsolid 0 255 0

-- Object 22: IDEDpredef_square_red

square_red,rectangle 0 0 300 300 -penstyle solid -penwidth 3 -pencolour 255 255
255 -brushsolid 255 0 0

-- Object 23: IDEDpredef_triangle_blue

triangle_blue,polygon 3 150 0 0 300 300 300 -alternate -penstyle solid -penwidth 3
-pencolour 255 255 255 -brushsolid 0 0 255
background_rectangle,rectangle 0 0 300 300 -penstyle null -penwidth 1 -pencolour 0
0 0 -brushsolid 0 0 0

-- Object 24: IDEDpredef_triangle_yellow

triangle_yellow,polygon 3 150 0 0 300 300 300 -alternate -penstyle solid -penwidth
3 -pencolour 255 255 255 -brushsolid 255 255 0
background_rectangle,rectangle 0 0 300 300 -penstyle null -penwidth 1 -pencolour 0
0 0 -brushsolid 0 0 0

-- Object 25: TT_redsquare

redsquare,rectangle 0 0 100 100 -penstyle solid -penwidth 1 -pencolour 255 255 255
-brushsolid 255 0 0

-- Object 26: PAL_emptybox

whiteboxblackbackground,rectangle 0 0 260 190 -penstyle solid -penwidth 3 -
pencolour 255 255 255 -brushsolid 0 0 0

CONFIGURATIONS

-- Task 0: PAL

Maximum no. trials (0 for no limit):          50
Max. time (min) (0 for no limit):             60
Maximum time for responding (s) (for both phases): 10
Minimum intertrial time (s):                  5
Maximum intertrial time (s):                  15
Must touch sample stimuli?                    Y
Rewarded for touching sample stimuli?         N
Sample stimulus duration, if no touch required (s): 10
Time between samples (s):                     0.5
Minimum memory delay (s):                     5
Maximum memory delay (s):                     20

```

```

Reward each correct choice?                N
Time between choices (s):                  0.5
Mark responses that aren't rewarded/punished? Y
End trial on first error?                  N
Repeat failed trials?                      N
Repeat failed trials up to X times:        5
Shuffle sample/choice order when repeating trials? N
Trial scheme:                             testscheme
'Empty box' object:                       PAL_emptybox
Stimulus selection method:                 Cyclical
... (if cyclical) start with stimulus:     0
... (if random) avoid repeats over last X trials: 5

-- Task 1: NPair

Maximum no. trials:                       50
Maximum time (min):                       60
Response criterion time (s):               10
Leave correct stimulus up during reward? Y
Minimum intertrial time (s):               1
Maximum intertrial time (s):               5
Number of pairs used:                     2
Pair choice method:                       Random
... Pair 0 (correct, incorrect): IDEDpredef_circle_red, IDEDpredef_circle_green
... Pair 1 (correct, incorrect): IDEDpredef_pie_yellow, IDEDpredef_pie_blue

-- Task 2: ReinforcementFamiliarization

Licker option - pump on at start?         N
Licker option - pump until X licks:        16
Licker option - FR1 licking:               N
Licker option - pump time per lick (s):    1
Main schedule:                             RT
Parameter of main schedule (s):            30
Maximum number of rewards:                 120
Maximum time (min):                       60

-- Task 3: TouchTraining

Response criterion time (s):                10
Starting level:                            1
Increase level?                             N
... regardless of performance?             Y
... criterion (every X trials or if X/last 20 correct): 40
In levels 1+2, present stimuli on the left? N
In levels 1+2, present stimuli in the centre? Y
In levels 1+2, present stimuli on the right? N
Maximum no. trials:                        80
Maximum time (min):                        120
Minimum intertrial time (s):                5
Maximum intertrial time (s):                15
Min inter-phase pause (sequential level):   2
Max inter-phase pause (sequential level):   10
Number of objects used:                     1
... Object 0 (name, correct?): TT_hugebar, Y

-- Task 4: Reversals

Maximum no. trials (0 for no limit):        100
Max. time (min) (0 for no limit):          120
Maximum time for responding (s):            10
Begin trial with Marker 1 sound?           N
Leave correct stimulus up during reward?     Y

```

```

Minimum intertrial time (s):                5
Maximum intertrial time (s):                15
Reverse within a session?                   N
... Reverse when X of last Y correct... X:  6
... Reverse when X of last Y correct... Y:  6
Use three stimuli rather than two?          N
Stimulus A:                                Reversal_1
Stimulus B:                                Reversal_2
Stimulus C:
Begin with B+ (rather than A+)?             N
p(reward|correct)=1, p(reward|incorrect)=0?  Y
... otherwise, p(reward|correct) =          1
... and p(reward|incorrect) =                0
Spatial locn randomized in groups (not fully random)? Y
Correction procedure:                       None
Correction procedure parameter A:            10
Correction procedure parameter B:            3
Correction procedure parameter C:            3
Start session with correction procedure?     N
Correction proc.: correct stim initally on:  Left

-- Task 5: DMTS

Matching? (Alternative is non-matching.)    Y
If matching, number of novel (distractor) stimuli: 1
Maximum no. trials (0 for no limit):        50
Max. time (min) (0 for no limit):           60
Must touch phase 1 stimulus:                Y
Maximum time for responding (s) (for phases 1 + 2): 10
Phase 1 stimulus duration if no touch required: 5
Rewarded for touching phase 1 stimulus:     N
Correction phase if subject fails phase 2:   N
Starting level:                              1
Method of changing level:                    Level_fixed
... criterion (X), if applicable:            15
Minimum intertrial time (s):                 5
Maximum intertrial time (s):                 15
Number of stimuli available:                 14
Stimuli available:                          IDEDpredef_circle_black;
IDEDpredef_circle_green;IDEDpredef_circle_red;IDEDpredef_hat_cyan;
IDEDpredef_hat_magenta;IDEDpredef_pentagram_cyan;IDEDpredef_pentagram_magenta;
IDEDpredef_pie_blue;IDEDpredef_pie_yellow;IDEDpredef_square_black;
IDEDpredef_square_green;IDEDpredef_square_red;IDEDpredef_triangle_blue;
IDEDpredef_triangle_yellow
Stimulus selection method:                   Cyclical
... (if cyclical) start with stimulus:       0
... (if random) avoid repeats over last X trials: 5
Number of levels (values of the memory delay): 4
Level delay values:                          0.000;5.000;10.000;20.000

-- Task 6: SimpleSchedules

Maximum no. reinforcers: 50
Maximum time (min):      120
Manipulandum object:    SS_redbox
Mark responses aurally?  Y
Mark responses visually? N
Visual marker object:
Visual marking time (s): 0.2
Schedule type:          CRF [FR-1]
Use timeout?            N
Timeout duration (s):   0

```

RESULTS

```

*** TASK ABORTED (cancelled by user or contact with server lost)
Emergency save of outstanding results:

-- Results for Task 0: Paired-Associates Learning

Trial,TrialStartTimeMs,GridType,NumStimuli,NumEmptySamples,
NumExtraChoiceLocations,SampleCorrect,SampleOmissions,ChoiceStageReached,
MemoryDelayMs,ChoiceCorrect,ChoiceIncorrect,ChoiceOmissions,ITTimeMs
0,3132792478,3-grid(strip),3,0,0,3,0,Y,6187,3,0,0,9604
1,3132823487,3-grid(strip),3,0,0,3,0,Y,5453,2,1,0,5017

Trial,Presentation,PresentationStartTimeMs,Stage,Empty,StimulusPresented,
LocationsPresented,Responded,ResponseLatencyMs,ResponseLocation,ResponseCorrect,
Rewarded,Punished,LickLatencyMs
0,0,3132792490,Sample,N,IDEDpredef_circle_red,1,Y,1369,1,Y,N,N,N
0,1,3132794373,Sample,N,IDEDpredef_circle_black,0,Y,1048,0,Y,N,N,N
0,2,3132795936,Sample,N,IDEDpredef_circle_green,2,Y,1070,2,Y,N,N,N
0,3,3132803208,Choice,N,IDEDpredef_circle_red,0:1:2,Y,2315,1,Y,N,N,N
0,4,3132806040,Choice,N,IDEDpredef_circle_black,0:1:2,Y,1127,0,Y,N,N,N
0,5,3132807684,Choice,N,IDEDpredef_circle_green,0:1:2,Y,1187,2,Y,Y,N,N
1,0,3132823493,Sample,N,IDEDpredef_hat_magenta,2,Y,1331,2,Y,N,N,N
1,1,3132825339,Sample,N,IDEDpredef_pentagram_cyan,1,Y,676,1,Y,N,N,N
1,2,3132826529,Sample,N,IDEDpredef_hat_cyan,0,Y,716,0,Y,N,N,N
1,3,3132832714,Choice,N,IDEDpredef_hat_cyan,0:1:2,Y,1421,0,Y,N,N,N
1,4,3132834651,Choice,N,IDEDpredef_hat_magenta,0:1:2,Y,1238,2,Y,N,N,N
1,5,3132836405,Choice,N,IDEDpredef_pentagram_cyan,0:1:2,Y,1427,2,N,N,N,N

RatBat FINISHED

Started at:          26-Jun-2003 (20:49)
Finished at:        26-Jun-2003 (20:50)

Successfully wrote to database: ODBC:DSN=RatBat_prototype;DBQ=D:
\Whisker\CODE\clients\rnc - cambridge\RatBat\RatBat database (sample).mdb;
DriverId=281;FIL=MS Access;MaxBufferSize=2048;PageTimeout=5;

```

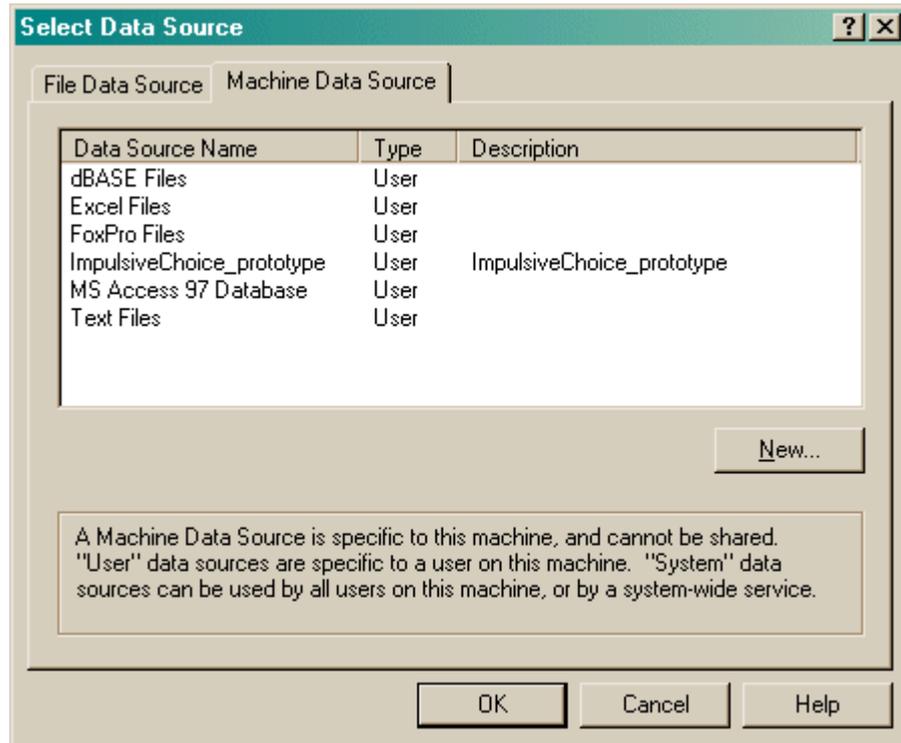
1.8.2 Creating a new ODBC source

What happens if you're using RatBat for the first time, and need to set up an ODBC (Open Database Connectivity) source for RatBat? You should configure it via **Control Panel** → **ODBC** [in Windows 2000, **Control Panel** → **Administrative Tools** → **Data Sources (ODBC)**]. Alternatively, you can set one up "on the fly", as explained here.

The example below is for a PIT (Pavlovian-instrumental transfer) database, as I couldn't be bothered to redo all the screenshots for a RatBat database. However, every step is identical! Just choose your copy of the RatBat database instead.

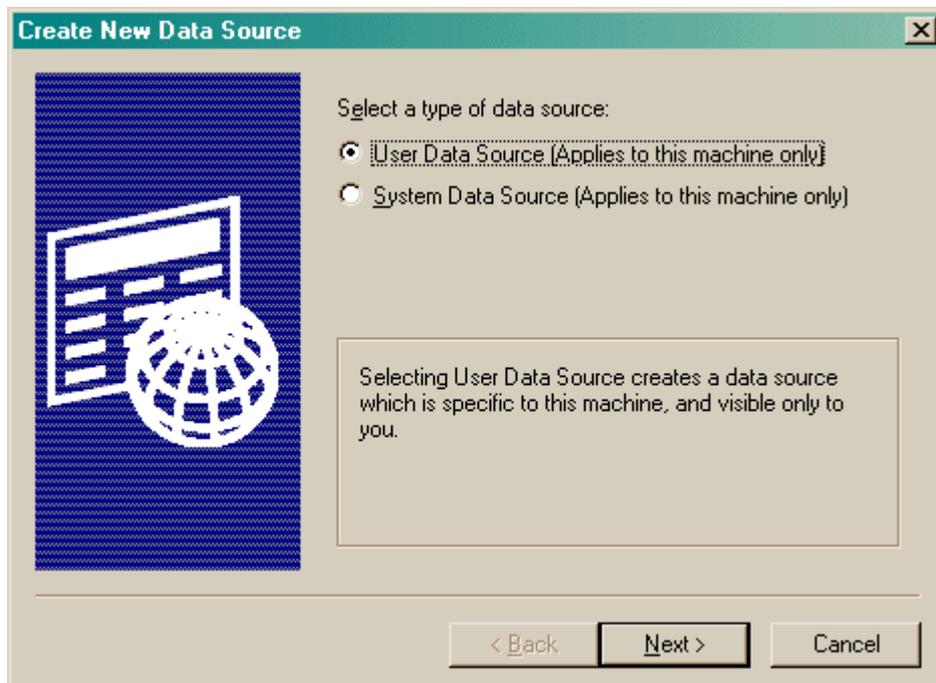
Remember: you shouldn't use the supplied database without making a copy for yourself. (It will work, but if you ever uninstalled or reinstalled RatBat, this file might be replaced or lost. It is much safer to make your own copy and set up ODBC to use your copy.)

Suppose you're looking for a PIT database. But there isn't one...

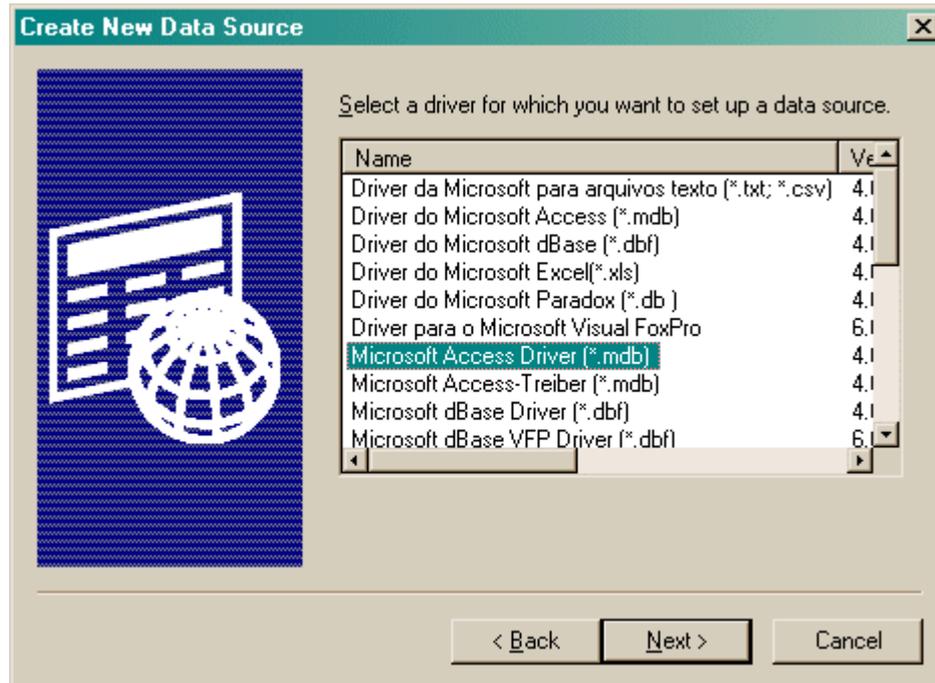


Let's assume that you have **already made a working copy of the prototype database supplied with the task**. How do we go about setting this up as an ODBC data source?

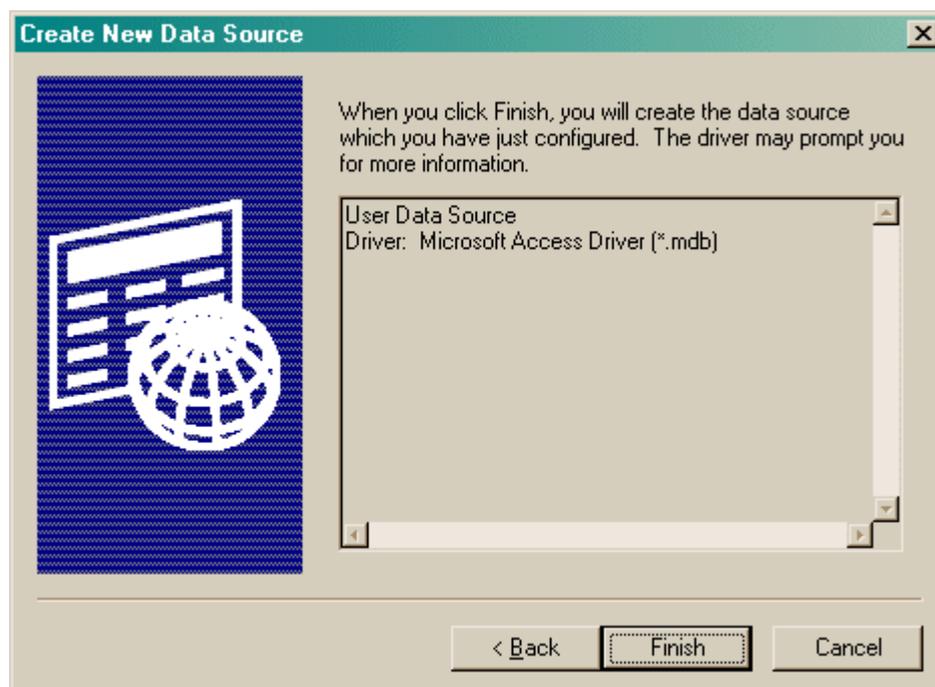
Click New.



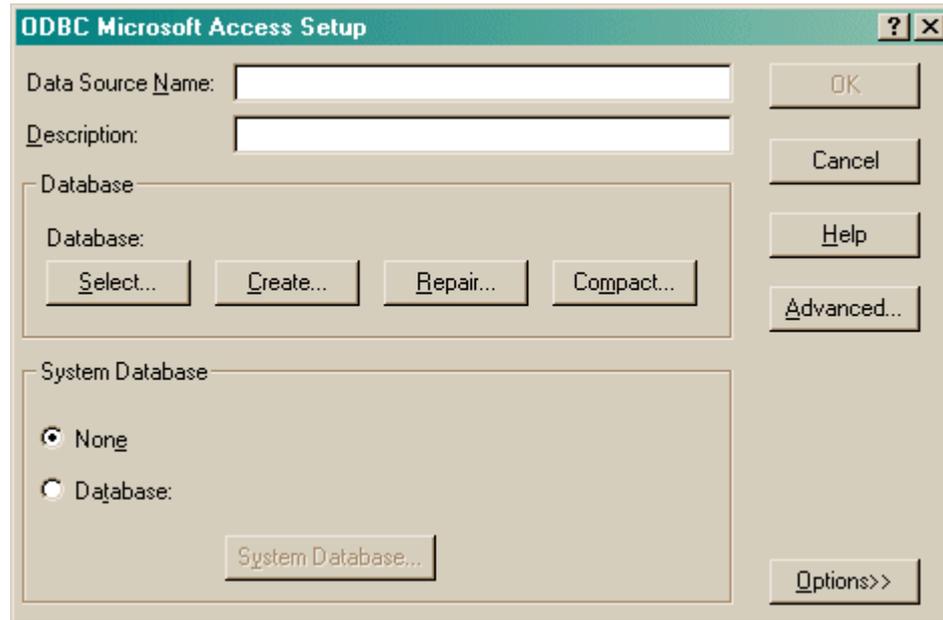
Choose a User or System data source. **User** is probably more sensible. Click Next.



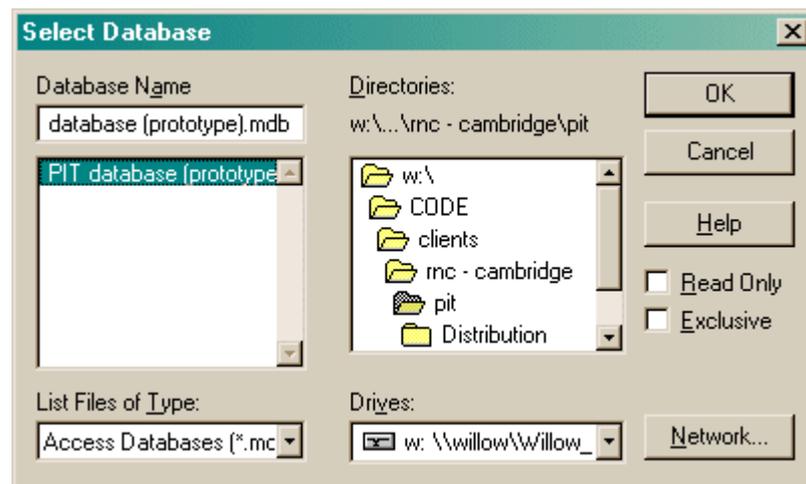
Choose your database driver. Click Next.



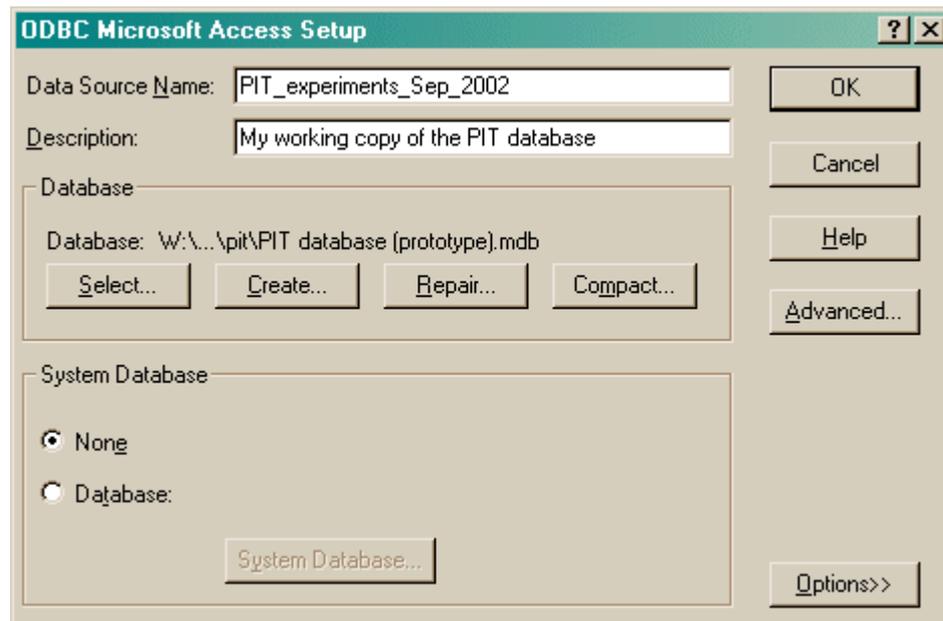
Click Finish.



You should fill in the **Data Source Name (no spaces)** and the **description**, and **Select** a database. When you click Select, this dialogue box appears:



Choose your database here and click OK. Your ODBC data source fields should now all be set up:



Click OK. You will be returned to the ODBC selection screen with your new data source now available.

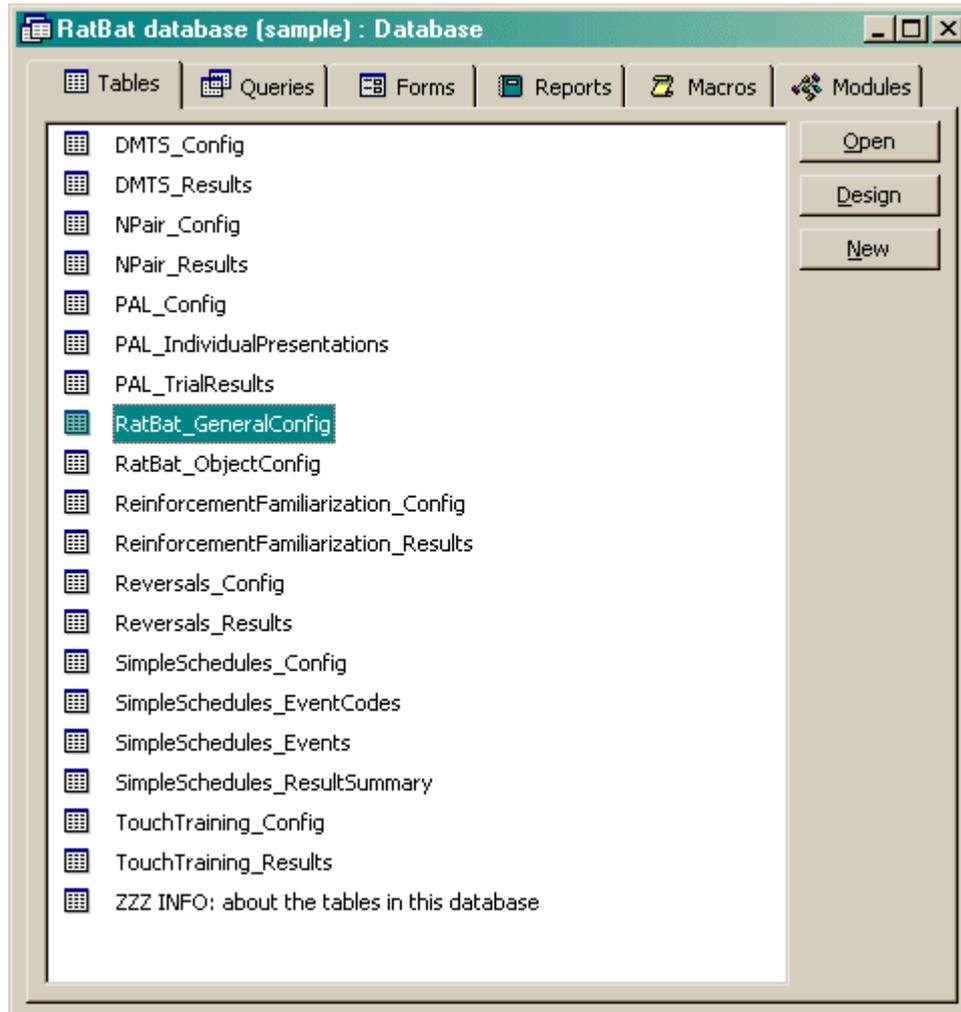
1.8.3 Using the Microsoft Access database for RatBat

Remember: you shouldn't use the supplied database without making a copy for yourself. (It will work, but if you ever uninstalled or reinstalled RatBat, this file might be replaced or lost. It is much safer to make your own copy and set up ODBC to use your copy.)

When supplied, the database is called "**RatBat database (sample).mdb**". Make a copy before using it!

You need Microsoft Access (97 or higher) to use this database. Sorry about that.

When you open the database, it looks like this:



RatBat will store its results here. The table "**ZZZ INFO: about the tables in this database**" summarizes the information held in each table. Click a table and click **Design** to view a list of all the fields.

NPair_Config : Table			
	Field Name	Data Type	Description
▶	RecordNum	AutoNumber	(Automatically generated.)
📅	DateTimeCode	Date/Time	Date/time the program was started.
🗃️	Subject	Text	Subject
🗃️	Box	Number	Box number
🗃️	ModuleNumber	Number	Order of this task in the master task list
	MaxTrials	Number	Maximum number of trials (0 for no limit)
	MaxTimeMinutes	Number	Maximum time in minutes (0 for no limit)
	ResponseCriterionTimeSec	Number	The time the subject has in which to respond (s) when given a stimulus/stimuli
	LeaveCorrectStimUpDuringRe	Yes/No	Leave the correct stimulus up during reward?
	MinIntertrialTimeSec	Number	Minimum intertrial time (s)
	MaxIntertrialTimeSec	Number	Maximum intertrial time (s)
	NumPairs	Number	Number of object pairs used
	Pairs	Memo	Names of objects/pairs used
	PairChoiceMethod	Text	How the pair order was chosen

Don't modify anything in Design view unless you know what you're doing!

If you close the Design view and click **Open** instead, you see the *contents* of this table. Here is the contents of the NPair_Results table (holding results for the *n*-pair concurrent visual discrimination task). I entered some sample results into this table.

NPair_Results : Table											
	RecordNum	DateTimeCod	Subject	Box	Modul	Trial	TrialStartTime	LeftStimulus	RightStimu	LeftCorrect	Responded
	1	17/2003 21:32:52	test	0	0	0	-1159410390	IDEDpredef_pi	IDEDpredef_pi	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	2	17/2003 21:32:52	test	0	0	1	-1159393088	IDEDpredef_pi	IDEDpredef_pi	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	3	17/2003 21:32:52	test	0	0	2	-1159382802	IDEDpredef_pi	IDEDpredef_pi	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	4	17/2003 21:32:52	test	0	0	3	-1159371872	IDEDpredef_pi	IDEDpredef_pi	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	5	17/2003 21:32:52	test	0	0	4	-1159361938	IDEDpredef_ci	IDEDpredef_pi	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	6	17/2003 21:32:52	test	0	0	5	-1159346905	IDEDpredef_pi	IDEDpredef_pi	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	7	17/2003 21:32:52	test	0	0	6	-1159330070	IDEDpredef_ci	IDEDpredef_pi	<input type="checkbox"/>	<input type="checkbox"/>
	8	17/2003 21:32:52	test	0	0	7	-1159307828	IDEDpredef_pi	IDEDpredef_pi	<input type="checkbox"/>	<input type="checkbox"/>
	9	17/2003 21:32:52	test	0	0	8	-1159284511	IDEDpredef_pi	IDEDpredef_pi	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Feel free to explore the tables.

When you want to extract data for analysis, you may want to create **queries** to do so. (Queries are listed in the "Queries" section of the main database screen.) Queries can be created using Access's visual query design system, or using the language SQL (Structured Query Language). A little on [relational database principles and SQL](#) follows.

1.8.4 Relational databases in general

I have found the most useful way to store data is in a **relational database**, often called a relational database management system (RDBMS). A relational database stores data in **tables**, which are made up of *fields* and *records*:

A table:	<i>five fields:</i>				
	Date	Rat	NumResponses	NumStimuli	NumReinforcements
<i>one record:</i>	17/2/00 12:29:00	M4	56	5	1
<i>another:</i>	17/2/00 14:37:06	M5	437	43	8
<i>... and so on</i>	17/2/00 12:54:00	M4	263	26	5

The driving principle behind a relational database is this: **never duplicate data**. Let's say our rats came from two groups, Sham and Lesion. If we wanted to record this in the database, so we could analyse data by group, we could store it like this:

Table BigData					
Date	Rat	Group	NumResponses	NumStimuli	NumReinforcements
17/2/00 12:29:00	M4	sham	56	5	1
17/2/00 14:37:06	M5	lesion	437	43	8
17/2/00 12:54:00	M4	<u>sham</u>	263	26	5

However, this introduces two problems. Firstly, it generates very large tables. Secondly, and more importantly, it is unclear what to do if the data is inconsistent – let's say the underlined 'sham' was changed to 'lesion' by mistake. The database would then not know whether rat M4 was in the Sham or Lesion group – there would be entries for both. The solution to both problems is to create two tables, *linked* on the smallest possible unit of information (in this example, the rat name):

Table Responses				
Date	Rat	NumResponses	NumStimuli	NumReinforcements
17/2/00 12:29:00	M4	56	5	1
17/2/00 14:37:06	M5	437	43	8
17/2/00 12:54:00	M4	263	26	5

Table Groups	
Rat	Group

M4	sham
M5	lesion

By using the rat name as a **key** (also known as a *foreign key*), the database can link the two tables together whenever we want to know how many responses the two groups made on average.

When we want to find out that sort of information, we **query** the database, specifying how we want to see the data. We could, for example, obtain the following (ignoring a glaring scientific error!):

Query AverageByGroups				
Group	NumberOfSubjects	MeanNumResponses	MeanNumStimuli	MeanNumReinforcements
sham	2	159.5	15.5	3
lesion	1	437	43	8

Summary of database principles

So relational databases split up the data (which should be entered in well-designed tables without any duplication of information) from queries that look at the data in an infinite variety of ways.

A concrete example: Microsoft Access 97

Microsoft Access 97 is a commonly-used relational database for PCs. It isn't perfect, by a long shot, but I've found it good enough. It supports **structured query language (SQL)** for designing queries; this is a powerful quasi-English language. For example, the query shown above would be written in SQL like this:

```
SELECT group,
       count(*) as NumberOfRats,
       avg(NumResponses) as MeanNumResponses,
       avg(NumStimuli) as MeanNumStimuli,
       avg(NumReinforcements) as MeanNumReinforcements
FROM responses, groups
WHERE responses.rat = groups.rat
GROUP BY group
;
```

If you find all this a bit cryptic, Access also provides a graphical interface for designing queries.

Getting data out of a database

Given a well-designed database, you should be able to get the data out in any conceivable way. The size of this manual doesn't permit a detailed look at relational database design or queries, but there are abundant sources. If you use Microsoft Access, there's the help system, but I also recommend Viescas JL (1997), *Running Microsoft Access 97*, Microsoft Press. Beyond that there is a whole field of database design.

Tip



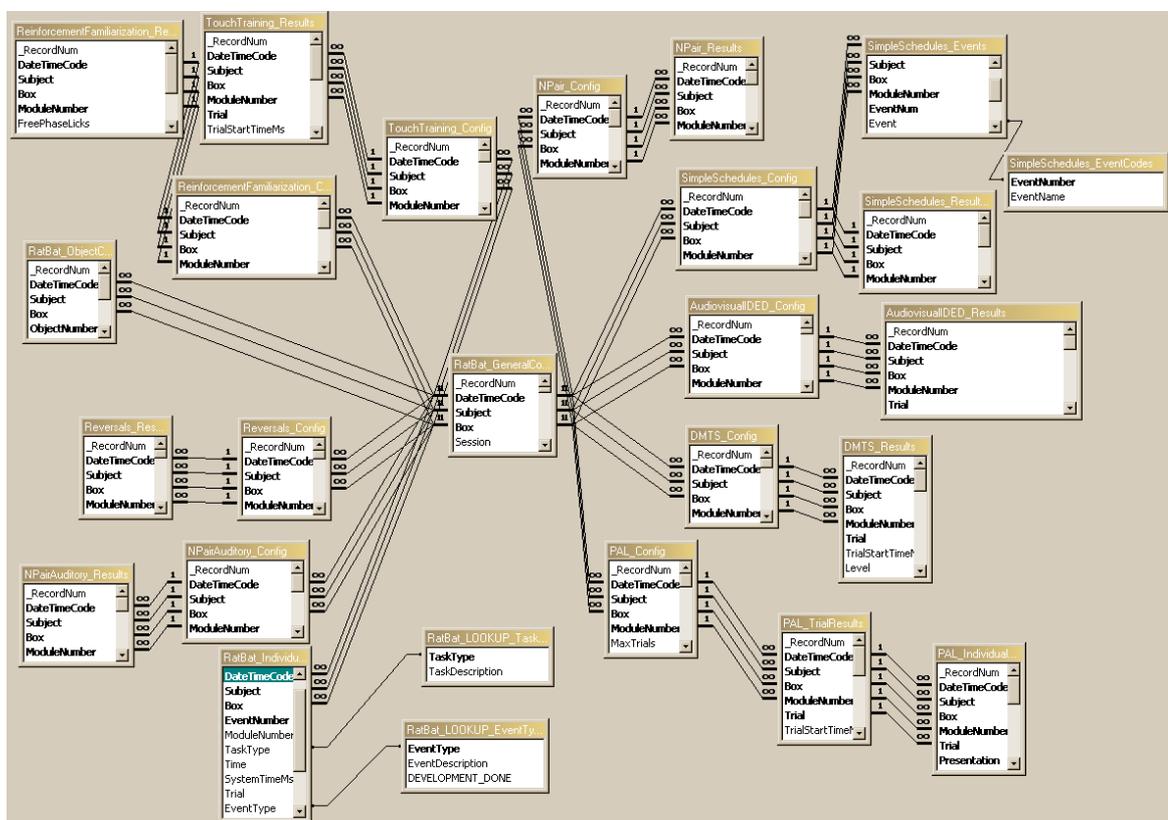
I operate on the principle that any view of the data is achievable. If the graphical query design can't do it, you can

use SQL. If SQL can't do it alone, you can use Visual Basic to augment it. If all that fails (and it hasn't failed me yet) you can always re-export the data and use a general-purpose programming language to analyse it. If the data's there, you can get at it.

One thing is worth noting: modern statistical packages (e.g. SPSS, <http://www.spss.com/>) are starting to support the ODBC standard for exchanging information with databases. You can set up database queries to create views of the data that your stats packages can use, then set up sequences of ODBC capture, analysis and graphical presentation in your stats package. Then whenever you import new data, you can run the entire analysis in a matter of seconds. If you handle large volumes of data, it easily repays the initial effort.

1.8.5 Database structure

This is the structure of the RatBat database:



Index

- R -

RatBat

- about 2
- about set shifting 54
- arc 15, 18
- Audiovisual ID/ED task 51
- before you start 56
- Bezier spline 15, 19
- bitmap 15, 21
- brush options 26
- chord 15, 21
- configuring 7
- coordinate systems 16
- database structure 71
- defining components of visual objects 15
- Delayed Matching/Non-Matching To Sample task 34
- ellipse 15, 22
- general parameters 10
- line 15, 22
- N-Pair Concurrent Auditory Discrimination 49
- N-Pair Concurrent Visual Discrimination 46
- Paired-Associates Learning task 37
- pen options 26
- pie 15, 23
- polygon 15, 23
- positioning objects 16
- rectangle 15, 24
- Reinforcement Familiarization task 27
- reinforcement timing 46
- relational databases 69
- required devices 3
- results 56
- Reversal Learning task 31
- rounded rectangle 15, 24
- setting up an ODBC source 63
- Simple Schedules of Reinforcement 43
- size of objects 16
- text 15, 25
- text-based results file 56
- Touch Training task 28
- using 5
- using the results database 67
- visual object library 12