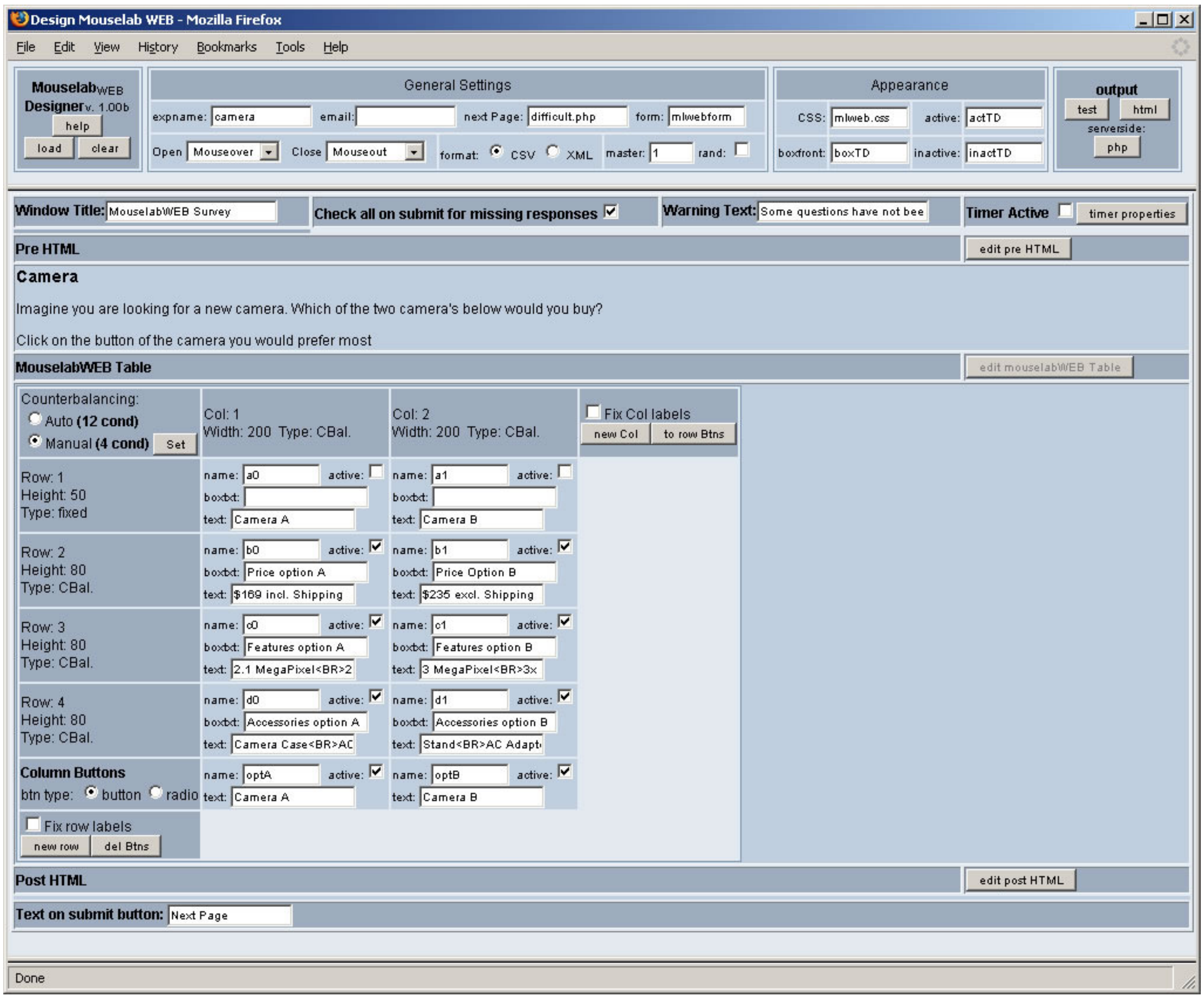
MouselabWEB Documentation

Version 1.00beta, Aug 14, 2008



General Settings: expname: camera; email: ; next Page: difficult.php; form: mlwebform; Open: Mouseover; Close: Mouseout; format: [✓ CSV] XML; master: 1; rand: ☐

Appearance: CSS: mlweb.css; active: actTD; boxfront: boxTD; inactive: inactTD

Window Title: MouselabWEB Survey

Check all on submit for missing responses ✓

Warning Text: Some questions have not bee

Timer Active ☐

Pre HTML

Camera

Imagine you are looking for a new camera. Which of the two camera's below would you buy?

Click on the button of the camera you would prefer most

MouselabWEB Table

Counterbalancing: Auto (12 cond) [✓ Manual (4 cond)]

	Col: 1 Width: 200 Type: CBal.	Col: 2 Width: 200 Type: CBal.	☐ Fix Col labels
Row: 1 Height: 50 Type: fixed	name: a0 active: ☐ boxtxt: text: Camera A	name: a1 active: ☐ boxtxt: text: Camera B	
Row: 2 Height: 80 Type: CBal.	name: b0 active: ✓ boxtxt: Price option A text: $169 incl. Shipping	name: b1 active: ✓ boxtxt: Price Option B text: $235 excl. Shipping	
Row: 3 Height: 80 Type: CBal.	name: c0 active: ✓ boxtxt: Features option A text: 2.1 MegaPixel
2	name: c1 active: ✓ boxtxt: Features option B text: 3 MegaPixel
3x	
Row: 4 Height: 80 Type: CBal.	name: d0 active: ✓ boxtxt: Accessories option A text: Camera Case
AC	name: d1 active: ✓ boxtxt: Accessories option B text: Stand
AC Adapt	
Column Buttons btn type: [✓ button] radio	name: optA active: ✓ text: Camera A	name: optB active: ✓ text: Camera B	
☐ Fix row labels			

Post HTML

Text on submit button: Next Page

MouselabWEB is designed by

Martijn C. Willemsen and Eric J. Johnson

Contact: info@mouselabweb.org

Table of Contents

What is new in version 1.00(beta)?

If you have been using previous versions of MouselabWEB, you will be interested in finding out what is new since 0.99.1a.

1. Supporting other events for opening (mouseover and click) and closing (mouseout, click (any), click (same), nothing) of boxes.
2. Completely redesigned counterbalancing system: the 2 group counterbalancing scheme has been dropped because it was both complex and very inflexible. In version 1.00, every column or row can be assigned as being counterbalanced or fixed. The columns and rows that are set to 'counterbalanced' can be counterbalanced automatically (full design) or by means of a manual schema. This allows for very specific sets of presentation orders, and makes counterbalancing order in MouselabWEB very flexible. On top of that, labeling on the boxes can be fixed, allowing only the underlying text to change.
3. Possibility to activate a timer (for time pressure studies) with a visible and configurable progress bar (CSS-based, so layout can be changed in the style sheet).
4. Choice buttons are now part of the CSS style sheet. A pressed button now gets a specific layout (which is flexible: it can be changed in the stylesheet), rather than a color around it.
5. Designer layout and interface has been changed on several points: button settings at the rows/columns, automatic button labeling, open- and close event settings, interface for entering manual counterbalancing schemes.
6. Direct download of page code when creating an HTML or PHP page (no more cut and pasting from an Dialog window)
7. An improved datalyser which sophisticated event extraction and processing!
8. Bug fixes, including:
    a. Fixed TEXTAREA errors in output (textarea in the pre- and postHTML was not parsed correctly in the output)
    b. Fix for problems with opening the test page in the designer in mozilla/firefox.
    c. If form check is turned off, the mouselabWEB choice buttons are now also not checked.

Introduction

Welcome to MouselabWEB, the process-tracing tool that can be used to monitor the information acquisition process of decision makers. This document tells you everything you need to know to set up, run and analyze an experiment. To start, you need a system that's ready to run MouselabWEB. There are two options for this. The first one, the **Webmailer** version (which sends data via email), is very simple to set up and can be used on any webpage, without php-facilities. A basic computer with an Internet connection and cookies enabled will suffice. The downside of the version however, is that is requires a lot of work to combine the data from the emails into an analyzable dataset. This version is therefore only recommended as a first test or as a way to familiarize you with MouselabWEB.
The other setup makes use of **PHP** and a **MySQL** database. This version makes analyzing your data very easy. Tools (the Datalyser) are provided to put the data in tables or download it directly in CSV format. You can even see a live replay of what each participant did.
If you are unfamiliar with PHP and MySQL, don't worry; anyone with some basic knowledge of FTP and HTML will be able to figure it out with the help of this document. If these last two concepts make you panic even more, it's probably wise to ask for some help. This document is an installation guide and reference and also contains a walkthrough example that shows all of the key features of MouselabWEB. It's highly recommended to do this walkthrough after installing to get to know MouselabWEB in a swift and easy fashion.

How does MouselabWEB work?

The design philosophy behind MouselabWEB is to allow process tracing in ordinary web browsing, with a minimum of hassle on the part of the decision-maker. Technically, MouselabWEB uses technology already in the browser, dynamic HTML and javascript technology, so the user (participant) does not need to download plugins or other software: just one of a large set of fairly recent and common browsers is sufficient to view and use a MouselabWEB page.

An experiment in MouselabWEB will consist of a set of HTML pages, which are linked. Each MouselabWEB page can contain regular question types (e.g., text fields, drop down boxes, radio buttons) as well as MouselabWEB boxes. MouselabWEB pages can be generated using the *MouselabWEB designer*, an online editor for generating MouselabWEB pages. Underlying these pages are javascripts that contain the MouselabWEB functionality, as well as php-scripts that take care of saving the data in a background database. This is all transparent for the person creating the MouselabWEB page; she only needs to create the pages with the questions.

A typical MouselabWEB experiment will consist of more than one page, and the underlying code makes it very easy to link these pages; for every page a nextURL can be given, to which will be linked, passing subject ID and condition numbers automatically.

The Designer program allows you to generate these pages, and also to reload previously generated pages to adjust them. The main purpose of the Designer is to generate the MouselabWEB boxes structure, but it also has basic functionality to add additional HTML code to the page, such as titles, instructions and additional questions. In this respect, relatively complicated Internet experiments can be constructed quickly and efficiently.

A MouselabWEB page can contain:

- Ordinary questions: scales, checkboxes, radio buttons, and text fields, everything contained in HTML form elements because it uses just plain HTML!
  - Tools are included to quickly create standard type questions (scales, options, text fields)
  - The background scripting automatically saves the data and can also check before page submission if questions are answered
  - Process data can be captured (mouse over events and clicks on buttons and text fields can be registered in the process data that is also used for the MouselabWEB boxes!)
  - a timer can be set to control total amount of time available for the participant.
- MouselabWEB structure of boxes.

  Features of boxes
  - Counterbalancing of layout (taking care of reading order effects), boxes can be fixed or counterbalanced (relative to other boxes in the same group)
  - Labels on outside of a box.
  - Box-text is HTML (and thus can contain HTML stuff like images and such)
  - Choice buttons for columns/rows (that are counterbalanced with the MouselabWEB boxes)
  - Boxes can be active or inactive (for example for header rows containing labels) in which case they are open all the time. Inactive boxes can still be part of the counterbalancing scheme, such that headers change simultaneously with the active boxes.
  - Version 1 allows for other events than mouseover/mouseouts to open and close boxes.
  - Layout of boxes and choice buttons are made in CSS: so the looks can be changed easily.

When using the php-version of MouselabWEB, downloading and replaying data is easy using the Datalyser program.

Installation

Both versions of MouselabWEB use the Designer to create the HTML or PHP pages with MouselabWEB features. You can either use the online Designer or download the Designer and install it on your server or local PC.

For the first option, you simply go to http://www.mouselabweb.org/designer .
For the second option, download the designer in a zip file here: http://www.mouselabweb.org /download.php.

For the installation package itself, you can also go to the download page:
http://www.mouselabweb.org/download.php
Choose the HTML version for the Webmailer and the PHP version for the offline or online server version.

Webmailer

Installation is very straightforward. Download and save the html zip-file from the download page and unzip it to a local folder on your own computer (e.g., a folder called mlweb somewhere in your own documents folder). To install the HTML files onto the internet, just FTP the entire package (all the files you unzipped to your own mouselabweb folder on your local computer) to a directory on the web from which you want to run the experiments.
Make sure that any page you generate will also be in that folder and it should work.

See the readme file in the package for a description of the files contained in the package.

The webmailer version uses a dedicated formmailer on the MouselabWEB server to send the data directly to you by email. When making pages for the Webmailer, make sure that you have filled in your e-mail address. This address has to be an address that is registered with MouselabWEB (most likely your own email that you used when registering). Every completed experiment page will yield an e-mail at this address containing the data from the experiment. Furthermore all you need to do to create Webmailer pages is to use the HTML button in the output box in the *Designer* (see the *Designer* section).
If you need to run an experiment with more than 1 page, you will need the *mlweb_start.html* starting page to set a cookie with the subject ID and condnumber (see how to run an experiment later in this documentation)

PHP / MySQL

This section will explain installation of the php-package.

Offline

It's very easy to put the experiment in an offline version on, for example, a laptop computer. The nice feature of it is that you won't need internet access to run the experiment. It's also a great way to test your MouselabWEB or to work on it before you put it online.

To get your system ready you need to add webserver and PHP/MySQL capabilities to your system. You can do this by hand, or install a handy package like Apache Friends' XAMPP. The latter makes the job a breeze and is thus highly recommended. You can find this package for several operating systems here: http://www.apachefriends.org/en/xampp.html.

After installing it you can test if the webserver is working by opening a web browser and entering the URL: http://localhost (or the name you specified instead of localhost when you were installing the package).

This localhost is actually a folder on your hard drive that serves as the root directory of your webserver[1]. The page you are now viewing is the index.html in that folder. Everything you

[1] Note that opening php files from a file explorer window does not work (different from ordinary html pages): you have to access the php pages through the webserver, using a browser and http://localhost/[some path and name].php

want to run from the web server has to be located within this folder structure. If you installed the XAMPP package in C:\Program Files\ then C:\Program Files\xampp\htdocs\ is this folder.
We recommend putting separate (sets of) experiments in separate folders within localhost. So if you create a subfolder in the xampp\htdocs folder named [yourfoldername], the url to the specific folder will be: http://localhost/[yourfoldername]. Unzip the whole php-package into this subfolder. Every time you create a new experiment in a new folder, you should put all the files from the package again in this folder, because all files (scripts/images) have to be located in the same folder as the experiment pages themselves.

After installing XAMPP and running it, you need to go through these last steps to make MouselabWEB work on your system.

1. Go to http://localhost/phpmyadmin and create a database; name it MouselabWEB for example.

2. Open *mlwebdb.inc.php* in a text editor and change the values there in the appropriate ones. (This file contains the database information and is use by all other php-files to access the database (for example, the save script uses it, but the datalyser also uses it). In most cases—xampp default settings have no password on the database, which might be a security issue on your local network— the values will be:

   ```
   $DBhost = "localhost";
   $DBuser = "root";
   $DBpass = "";
   $DBName = "MouselabWEB";
   $table = "mlweb";
   ```

   The last line assigns a name for the table with your experiment data within the database. You could set this to any value you like.

3. Run *create_table.php* on the server to create a table in the database based on the settings in mlwebdb_inc.php. This would for example be http://localhost/[yourfoldername]/create_table.php
   This script creates a table named in line: `$table = "mlweb";`
   You could use differently named tables within the same database for different sets of experiments (which would have different subfolders, each with their own *mlwebdb.inc.php*

4. Once the table is created you are set. Any mouselabweb php file you generate with the *Designer* can be copied to the current experiment folder and will automatically save the data into the database in the table you just created.

5. The Datalyser program has a fixed password on the first line of *datalyser.php*.
   ```
   // set password
   $password = "mlweb";
   ```
   Change this password if you are running the experiment online. Otherwise other

people knowledgeable of MouselabWEB will be able to download your data. To be able to download CSV data sets from the download page, you'll have to make sure the tmp directory (that is in the zip file) in your current experiment folder has read/write access (unix chmod 777) enabled; otherwise the php script underlying the download page cannot write to it. Typically, in the xampp package itself on your local machine, directories are freely accessible and this is not needed, but this is not true by default for most online folders.

Online

Online basically requires the same demands as offline, but often you don't have access rights to the server to create databases and such. If this is indeed the case, make sure to ask your administrator to give you the following:

- Access to run PHP scripts.
- A mysql-database with password in which you can create the mlweb table (using the *create_table.php* script).
- A subdirectory called tmp with read/write access for all users. You might have to set these priorities yourself using an FTP program (chmod 777 on the tmp folder). This needed for the datalyser. Also remember to change the datalyser password when online, otherwise, anyone with the right URL could download your data…

The Designer

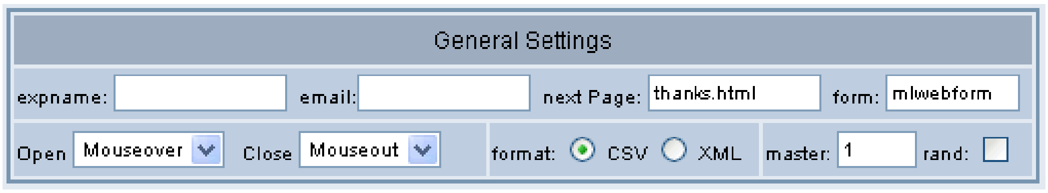
This section will explain all the options and settings in the Designer. The Designer makes it very easy to create your MouselabWEB experiment as well as adding some general HTML elements where subjects can enter other data, like radio buttons and text fields.

Starting from scratch

When creating a new experiment, you first want to fill in some of the general settings before starting to create the MouselabWEB cells. The settings are all on top of the screen and are discussed in this section. After that, clicking on the button '*edit MouselabWEB table'* will enable you to start inserting the experiment data. You can also insert HTML code above and beneath the MouselabWEB table. The '*edit pre HTML' and 'edit post HTML'* buttons will activate these sections.

General Settings

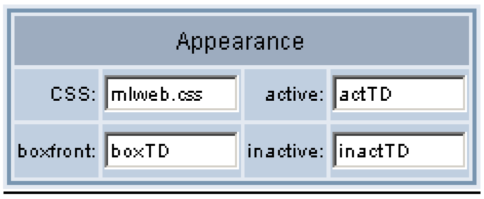
In the General Settings part you have to set some global settings important for proper functioning of the experiment.


General Settings

expname:		email:		next Page:	thanks.html	form:	mlwebform
Open	Mouseover	Close	Mouseout	format:	[✓ CSV] XML	master: 1	rand: ☐

Expname	The name or number of the experiment (passed on with submission of the form to identify this specific page in the database). For the replay function in the datalyser program, it is easy if file name and expname are identical. (The player defaults to expname.php, but a different name can of course be entered.), Also when you download/save the page code, it will default to ‘expname.html’ or ‘expname.php’
Email	The e-mail address used by the webmailer. Make sure this is the same address you used for registration. If you use the PHP version, this can be left blank.
Next Page	The URL of the page the browser should link to, if the MouselabWEB page is submitted. The header variables (e.g., condition and subject variables) are passed to the next page, allowing a chain of MouselabWEB pages that pass the same subject and the same counterbalancing number. Default value is thanks.html
Form	The name of the form on the page (default value is normally fine, but this has to be set to the existing form name if MouselabWEB table is inserted in an existing web page with an existing form).
Open/Close	Select what action should open and close a box. **Opening:** mouseover or click **Closing:** mouseover, click (any): this means any click on another box will close the current box), click (same): only clicking the same box can close the current box), noClose: (meaning no action, every box stays open) If closing is set to noclose, more than one box is allowed to stay open. (note: Process data will also be generated for mouse moves over open boxes). In any other situation, a box will have to be closed before a next one can open (one box open at a time).
Format	How the process data should be outputted: CSV is comma-separated, used for import in many programs (e.g., Excel) and statistical packages such as R, SAS and SPSS. Note however that when you open it directly as an Excel file, all the data will be put in one column. This can be prevented by importing it or replacing the comma’s by semicolons. XML is a format that uses an XML-like structure, which does result in more overhead (larger files). The datalyser program accepts both types when downloading and extracting data from the database.
Master	The special feature can be used to put a division on the counterbalancing number: e.g., if master is set to 2, two consecutive condition numbers will result in the same order, and only the third will have the next order. This can be very useful if the experiment has some experimental conditions that people are assigned to by a number that goes along with the counterbalancing number (e.g., if you use the same number to counterbalance as to divide your experimental conditions). Otherwise, the experimental conditions will be confounded with the presentation order (counterbalancing) conditions, having condition 1 always with order 1, condition 2 always with order 2, etc.
Rand	This feature forces randomization of the counterbalancing variables (thus resulting in random orders of presentation on the screen), even if a condition number is supplied.

Appearance

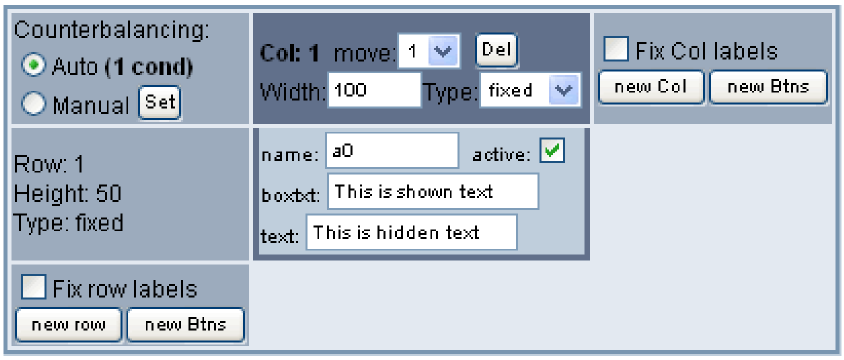


CSS: mlweb.css
active: actTD
boxfront: boxTD
inactive: inactTD

These boxes can be used to change the default CSS classes used for the appearance of the MouselabWEB boxes.

CSS	The name of the style sheet that should be used by MouselabWEB for this page. Default is mlweb.css.
Active	The style used for the display of the inside of active boxes.
Boxfront	The style used for inactive/open boxes (e.g. the style used for headers and such).
Inactive	The style used to draw a nice front on the outside of the box.

Creating Mouselab Cells



Counterbalancing: [✓ Auto (1 cond)] Manual
Col: 1 move: 1
Width: 100 Type: fixed
Fix Col labels
Row: 1
Height: 50
Type: fixed
name: a0 active: ✓
boxtxt: This is shown text
text: This is hidden text
Fix row labels

If the MouselabWEB Cells part is not visible yet, click on the '*edit MouselabWEB table'* button and you can start to create the cells. Simply use the '*new row*' and '*new col*' to add rows and columns. You can then start filling in the values and such. At all times it's possible to swap and/or delete the rows and columns. Click on the header of the particular row or column you wish to edit and it will become selected (and the properties can be edited).

The Cell Options:

Name	This is the name of the box in the process data (so this name will become an important variable in your data file, so make it descriptive)
Boxtxt	The text visible to the subjects on the outside of the box
Text	The hidden text that becomes visible after opening a box, (e.g. by moving the mouse cursor into the box.)
Active	The active checkbox sets whether a box is always open (unchecked, e.g.,

	for header rows) or closed (active checked)

Table Options:

CounterBalancing	**Auto:** if set to auto, columns and rows which type is set to CBal will be counterbalanced against each other. For example, if there are 3 columns and 2 rows that are set to CBal, 3x2x1 * 2x1 =24 orders are possible. Columns that are set to fixed will remain in fixed position (e.g. header rows or columns should have this property: for example: header rows should have the row set to fixed, but the columns and thus the labels, can still be counterbalanced together with the other cells). **Manual:** This allows to set all counterbalancing orders used manually. Pressing the **set** button will open a new window in which orders can be entered. This feature is handy when you want to have more control over the orders presented. For example when using gambles with separate outcome and probability boxes, one typically wants to keep the outcome and probability of one event together and in the same order. (more on this feature in the walkthrough)
Move	Move the selected row or column to a different position
Height / Width	Sets the width or the height of the cells in pixels
Type	The Type item sets the counterbalancing group (fixed, CBal). Columns and Rows can be counterbalanced or fixed.Columns and rows set to CBal will be counterbalanced according to the Countebalancing settings (manual or auto, see above)
Del	Delete the current active column or cell
new Btns	create new column or row buttons (if buttons are already present, one button will say 'del btns' (to delete the current set of buttons, this but is shown on the row of column the buttons are in) and the other will say 'to row btns' or 'to col btns', to move the buttons from rows to cols, or vice versa.

Pre & Post HTML

The pre and post HTML can be used to add additional text (in HTML format) to the MouselabWEB page. If this HTML contains form elements, the background scripts will automatically capture the responses for these elements.
MouselabWEB also provides templates which make it very easy to quickly insert scales, option boxes, and text fields. You can give the scales, boxes, and textfields variable names and the process of mouseovers and clicks on the scales can be captured, like for the MouselabWEB cells. There is a checkbox in each option pane for this.

Insert Scale	HTML scale with custom amount of points, labels and values
Insert Option / Choice	Radio or Checkbox HTML with custom amount of options, labels and values.
Insert Textfield	Simple textfield (1 row or multiple rows (textarea))

Undo last insert	Removes the latest inserted HTML generated code

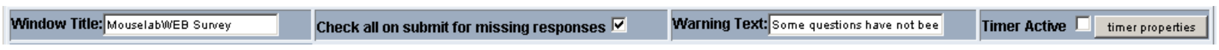
Opening / Saving an experiment

Press *load* to import a previously made MouselabWEB page. Most features of the existing page will be captured, but specific manual formatting might get lost. Pages from older versions should typically load fine (if not let us know). In this way you can upgrade older pages to a new version with more features.

Use the ouput buttons on the right side of the screen to save the page (as HTML or PHP); In older versions you needed to copy-paste the code into an editor (This is due to the nature of the *Designer*, which is an HTML-based program and therefore does not have access to your file system.) In the new online version you can download the code directly. If you have installed the designer locally and running it on a php-enalbed webserver, you should also be able to download the file.

Other settings

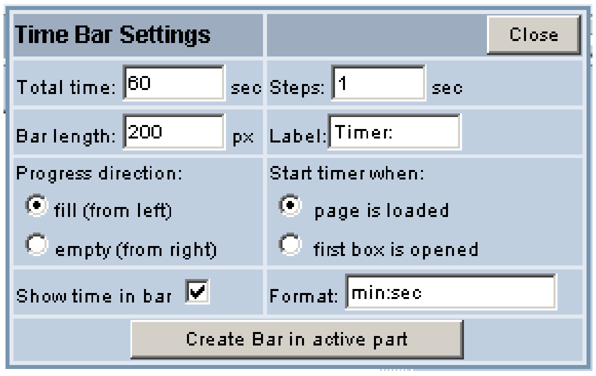
The title of the MouselabWEB page (which will typically appear in the window title bar) can be changed in the top row of the editing section.



Window Title: MouselabWEB Survey | Check all on submit for missing responses ✓ | Warning Text: Some questions have not bee | Timer Active ☐

If you have added additional questions to the MouselabWEB page besides a MouselabWEB table, you can select the form-check option to make sure people have inputted an answer before going to the next page. The warning text displayed when the subject has not met this condition is also customizable. This bar also allows for activating the timer and setting the properties of the timer.

Timer



Time Bar Settings

Total time: 60 sec | Steps: 1 sec

Bar length: 200 px | Label: Timer:

Progress direction: [✓ fill (from left)] empty (from right)

Start timer when: [✓ page is loaded] first box is opened

Show time in bar ✓ | Format: min:sec

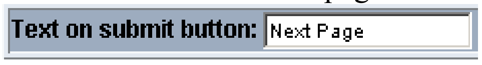
Version 1.00 includes a timer that will show a timer bar progressing, and will disable new acquisitions if the total time has passed. By pressing the timer properties, you can set some of the properties of the timer.

- **Total time:** total time a user is allowed to access boxes.

- **Steps:** time interval of updating the progress bar.
- **Bar length:** length of the progress bar in pixels
- **Label:** Label shown on the left of the time bar
- **Progress Direction:** set whether the bars fills from the left side (and timer counting up from 0 if "show time" is checked), or empties from right (and counting down from total time if "show time" is checked)
- **Start timer when:** select whether timer starts when page is loaded or when first box is opened.
- **Show time in bar:** prints the current time in the progress bar
- **Format:** how time is displayed in the time bar, when there is a colon (:), the bar will show minutes and seconds separately (if the current time is more than 60 seconds). If there is no colon, it will only show secs. It will use the labeling as given in the format box (before and after the colon). Examples:
  let's say the current time is 80 sec
  format: "min:sec", time bar will show **"1 min : 20 sec"**
  format: "M:s", time bar will show **"1 M : 20 s"**
  format: "secs", time bar will show **"80 secs"**

To actually create a time bar, the timer must be set to active (in the top bar). When pressing the create button, it will put in some code in the current active part (preHTML, mouselabwebtable (bottom), postHTML). Note that this actually inserts code in the preHTML or postHTML part if this is active, at the position of the current cursor (like inserting a scale).
Whenever the "create" button is used again, the previous code is removed. However, if you deactive the timer in between, the code inserted in a pre or postHMTL part will not be removed automatically.

Submit button
On the bottom of the editing section, a field can be adapted to change the text printed on the next/submit button of the page.


Text on submit button: Next Page

Running the experiment

HTML version

When running the HTML version, one needs a start page that sets a cookie, such that the subject-ID and condnum can be saved and passed on to the next page. Open **mlweb_start.html** and edit the line containing: *linkstr = " "*;. Between the quotes you need to fill in the name of your experiment page. This file and the experiment file need to be in the same directory.

PHP version

Depending on the version you use, there are several ways to run an experiment. If you use the php-version, you can call the first page of the experiment by passing a subject ID and condnum in the header. The URL would then be:
http://[yourdomain]/[yourfolder]/[filename].php?subject=[subjname]&condnum=[condition number]
A way to do this would be to send each subject a personalized email message with a direct link to the survey, using unique Ids and condition numbers for each subject.
Another way is to use the **mlwebstart_php.html**.
Open the file in an editor and adjust the name of the starting page in this file (this variable is called pagename). This page will ask for a subject name and condition number and then start the experiment by loading <pagename> with the subject and condnum passed in the header, just as shown above. An important feature of this page is that it will open a new window resized to full screen, without buttons and address bars.

Once the experiment is started, MouselabWEB will link from one page to the next in the experiment until the last page is reached, continuously passing on the subject name and condition number via the header (in the php version) or the cookie (for the HTML version).

About the counterbalancing number:

The MouselabWEB script will counterbalance the layout based on the number provided in condnum. If the number provided is larger, it will use the modulus of the number (e.g., if you have 4 conditions, condnum=0 will give you condition 1, condnum=3 will give you condition 4 and condnum=5 will give you condition number 1 again.)

Retrieve the experiment data

The data inside a MouselabWEB page is temporarily stored in an HTML form on the page, and by pressing submit (or next page), this form gets posted to a script that will save the data in some way (e.g., storing in the database, or sending them by email to the experimenter).
The form contains the following fields:

- **expname:** The name of this MouselabWEB page
- **subject:** The name of subject, as passed in the header/URL
- **condnum:** The number of the counterbalancing condition, as passed through in the header/URL or via a cookie
- **IP:** address of the computer

- **choice:** The choice made by the participant if choice buttons are added to the MouselabWEB table.
- **procdata:** The process data
- **nextURL:** The name of the next page, the URL the script should browse to after submission of the form
- **any additional form element will be captured as well and stored in the database or sent via email to you.**

PHP / MySQL : Datalyser

The php version includes the Datalyser, which can be used to download, view and replay your process data. Open 'datalyser.php' on your site (in the folder where your pages are). The page has a password which might be needed to prevent other people from downloading your data. Try the replay option on one or more sites. It's a very nice way to see what subjects really do in the decision making process. Show table shows all the captured invents of the table and you can also download the tables to CSV files (comma separated values). Make sure the subdirectory tmp is write-enabled; this is the directory the CSV files will be created in for downloading, so the script should be able to write to it.

Data format of process data

The process data is event data. That means that it is just a serial list of events that occurred. Each event contains four subelements:

Event: type of event: mouseover, mouseout, click
Name: name of the event (e.g., the name of the MouselabWEB cell, or the name of the form element that was captured)
Value: value of the event, which equals the boxtext, or the value of the form element captured.
Time: time in milliseconds measured from the moment the page was started.

Special events in the process data:
Besides events that are associated with MouselabWEB box openings/closings or form elements captured, there are also special events, indicated by special event names:

onLoad: indicates the time the page was fully loaded and displayed.
Subject: indicates how the subject number is passed to the page. (If the name of the subject event is "header", the subject name and counterbalancing condition number was passed in the header (URL); if it says "cookie" was passed using a cookie. The value indicates the current condition number.
Events: indicates the type of open and close event used:
For opening: 0=mouseover, 1=click
For closing: 0=mouseout, 1=click(any), 2=click(same), 3=noclose

Order: This event tells you the exact order in which the columns and rows were presented.

Submit: if the submit button of the present page is pressed, it will generate this event. If the submit succeeds (which only happens when the form check is positive), then another submit event with “succeeded” as its value will be generated.

The process data is captured in one single field in the database/email sent, in either CSV or XML format. If you use the Datalyser in the php package, the download script can unpack the events into a list of events, which will certainly make data analysis easier.

New: as of version 1.00, the datalyser can also process the events file. It will output one processed event file (labeled [expname]_proc.csv) and one aggregated process file (labeled [expname]_proc_sum[div].csv), summarizing all data of a participant across a trial into 1 or more divisions [div]. For processing, two variables can be set: threshold and division.

The threshold is used to drop acquisitions with short times. This can take care of spurious acquisitions made by people moving the mouse over boxes without looking at the content. You can use your own threshold here, but typically 100ms or 200ms acquisitions can hardly have been read by the participant, so these would be save thresholds.

The division is used for the summary file. For example, if division is 2, the output file will have two rows per participant, one for the first half of the acquisitions in the data, and one for the second half of the data. these data files are great for doing aggregate analyses using repeated measures or multilevel regression models.

Webmailer

All the results will be mailed to your specified email address. Note, however, that this has to be the address that you used to register at the MouselabWEB site. The email will be in the format of an HTML table. This should be relatively easy to paste into Excel or other spreadsheet or statistical programs. However, this is a very time-consuming process, which is why we recommend using the php-version.
If you use the webmailer version and would like to get the data in unpacked format, you can use a different formmailer on the server that does not send you the data in HTML format, but in text format, with the events unpacked. Even more, if you have additional form elements, this data is also unpacked. So this provides you with a dataset in a format that is very easy to import and analyze in many statistical programs.

How can you use this new formmailer instead of the standard formmailer?

In the <FORM> tag of your own mouselabWEB page, change the name of action

```
<FORM name="mlwebform" method="POST”
action=http://www.mouselabweb.org/mlwebmailer.php>
```

from mlwebmailer.php to the new formmailer name: **mlwebmailer_evt.php.**
This new formmailer has not been fully tested; let us know about any bugs or problems with it.

Capturing other form elements

The save.php script that is used to save the data into the database also saves additional, non-mouselabWEB data into the database. Any form element on the page will be saved. For this the database contains to two additional variables: **addvar** contains the names of the additional form elements, and **adddata** contains the values of these additional form elements. If there is more than one additional element that is captured, names and values are separated by semi-colons. The Datalyser already unpacks these form elements when you download you data, so you will get this data automatically. This additional feature allows you to put additional questions on any mouselabWEB page, without the need for additional scripting to save the data!

Walkthrough

In this walkthrough you'll get to know most of the MouselabWEB features and possibilities. This walkthrough assumes you have the php type MouselabWEB system ready, although this will also work in plain HTML with some adjustments.

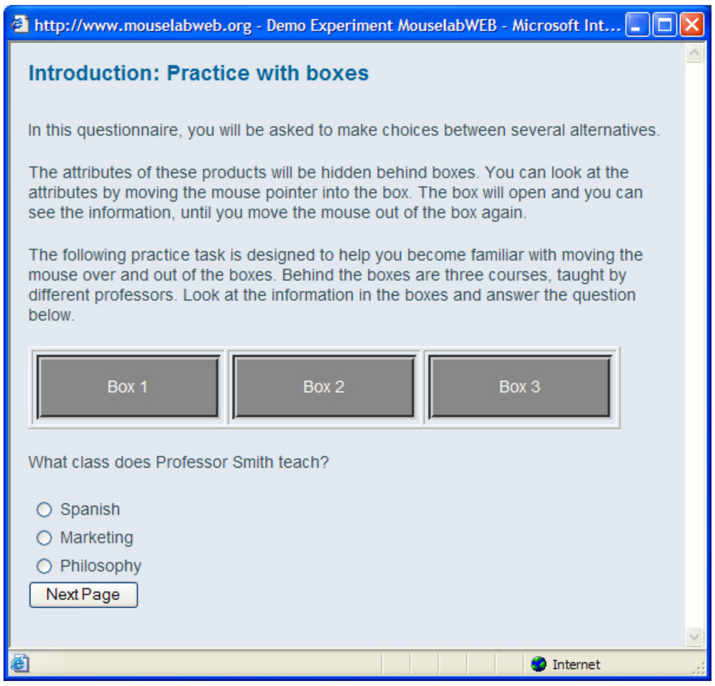
.

It will show you how to make a set of pages that link to each other. Subjects are identified by a condition number and an id. The condition number is used for counterbalancing; the id can be alphanumeric or numeric (and even identical to the condition number). MouselabWEB passes the ids and condition numbers from one page to another. In between the data is saved (sent by email or put in database)

First, you're going to make the pages of the experiment. The first page will contain a simple MouselabWEB structure. The second page is more complex, and the third is a page with only a scale. The last page is just a plain thank you note.

This walkthrough is based on the demo pages you can find on the MouselabWEB site. On this site you can also download a zip file with the php-scripts, if you want to look at them more carefully. But you can also generate them yourself using this walkthrough!

Page 1 : intro.php


http://www.mouselabweb.org - Demo Experiment MouselabWEB - Microsoft Int...
Box 1
Box 2
Box 3
Next Page
Internet

Introduction: Practice with boxes

In this questionnaire, you will be asked to make choices between several alternatives.

The attributes of these products will be hidden behind boxes. You can look at the attributes by moving the mouse pointer into the box. The box will open and you can see the information, until you move the mouse out of the box again.

The following practice task is designed to help you become familiar with moving the mouse over and out of the boxes. Behind the boxes are three courses, taught by different professors. Look at the information in the boxes and answer the question below.

What class does Professor Smith teach?

- Spanish
- Marketing
- Philosophy

Let's take a look at the general settings. Choose 'intro' as experiment name and because this experiment contains more pages, we have to fill in what page the current page will link to. Enter camera.php here, which is Page 2. This page will not use Buttons, so that setting can be left alone.

Now we're going to create the MouselabWEB cells. Add one row and three columns. The settings of the cells can be adjusted by clicking on them. Try to adjust the width of the cells to 160 and the height of the cells to 60 and stop the rows and columns from linking.
Inside the cells are the fields where the text can be inserted. The box text field is the text shown when the MouselabWEB cells are closed. Enter Box 1, Box 2 and Box 3 in these fields. In the other text field, the hidden field, insert these texts:

```
Professor Marx<BR>Spanish
Professor Jones<BR>Marketing
Professor Smith<BR>Philosophy
```

As you might have noticed, the fields contain an HTML tag. The cells can contain HTML (even images by using the <IMG> tag). The <BR> tag in this case creates a line break.

MouselabWEB is also able to add HTML above and beneath the cells we just created.
Insert the following HTML code in the *Pre html* field:

```
<H2>Introduction: Practice with boxes</H2>
<P>In this questionnaire, you will be asked to make choices between several
alternatives.</P>
<P>The attributes of these products will be hidden behind boxes. You can look
at the attributes by moving the mouse pointer into the box. The box will open
and you can see the information, until you move the mouse out of the box
again.</P>
<P>The following practice task is designed to help you become familiar with
moving the mouse over and out of the boxes.

Behind the boxes are three courses, taught by different professors. Look at
the information in the boxes and answer the question below.
</P>
```

Note again here the usage of HTML. Everything between <H2> and </H2> will be displayed as header 2 and <P> </P> marks paragraphs.

The page will end with a question and an option field for the user to answer this question. The Question is "What course does Professor Smith teach?".
Go to the *post HTML* part, enter this question.

```
<P>What course does Professor Smith teach?</P>
```

Then create an option set by clicking on the insert Option button. In the pop up, enter a variable name and make sure three options are created. Because there is only one right answer, we are using the type single (radio). In the next screen enter the three occupations in the value and text field.
After the options are created, don't forget to insert the question itself. The HTML code for the option set will be inserted at the current position of the cursor, so before clicking on the create option box, make sure the cursor is in the right spot.

Page 1 is almost done now. The last addition we're making is that we're making sure the subjects won't leave the answer option empty. This is done by putting a check right above the Pre HTML field. You can change the warning text if necessary.

Try out the test button to see what the page will look like. After that, save the page under the name 'intro.php' by pressing the php button in the output menu (top right). You can directly download the page and save it or copy/paste into an editor.

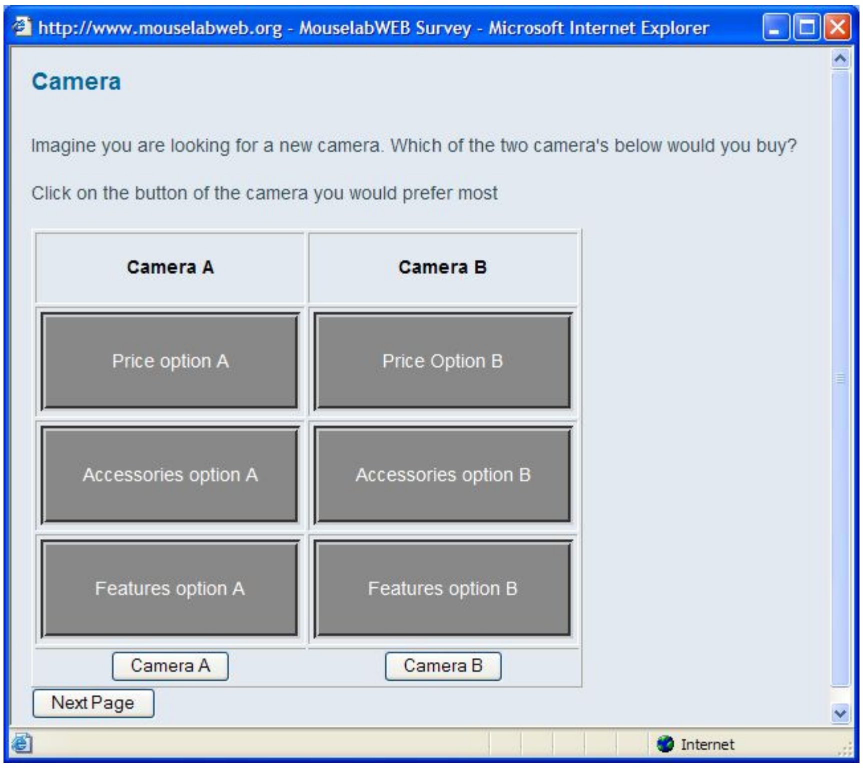
Page 2 : camera.php

In gathering process data, most of the time it is necessary to counterbalance the order of presentation to correct for main effects in data caused by the natural reading order (i.e., top-left boxes are more likely to be opened first and therefore get higher frequency of acquisition and somewhat longer looking times). A MouselabWEB structure can be counterbalanced in very many ways.

For example, let's assume we have the following stimuli. Two camera's with three features. We want to counterbalance the two options and the attributes in the rows. This would result in 2x3x2x1=12 versions. However, we would like the price on top or below, but not in between the two features. This means we have 4 row versions: PFA, PAF, FAP, AFP and likewise a total of 8 conditions. In MouselabWEB (from version 1.00) you can set these conditions manually yourself, as we will discuss below.

	Camera A	Camera B
price	$169 incl. Shipping	$235 excl. Shipping
features	2.1 MegaPixel 2x Optical Zoom	3 MegaPixel 3x Optical Zoom
Accessories	Camera Case AC adapter	Stand AC Adapter Spare battery

Let's start: Create this structure by making 2 columns and four rows. Put the above shown texts in the text fields. Uncheck the Active options on the first row; this will make the text in these


http://www.mouselabweb.org - MouselabWEB Survey - Microsoft Internet Explorer
Internet

Camera

Imagine you are looking for a new camera. Which of the two camera's below would you buy?

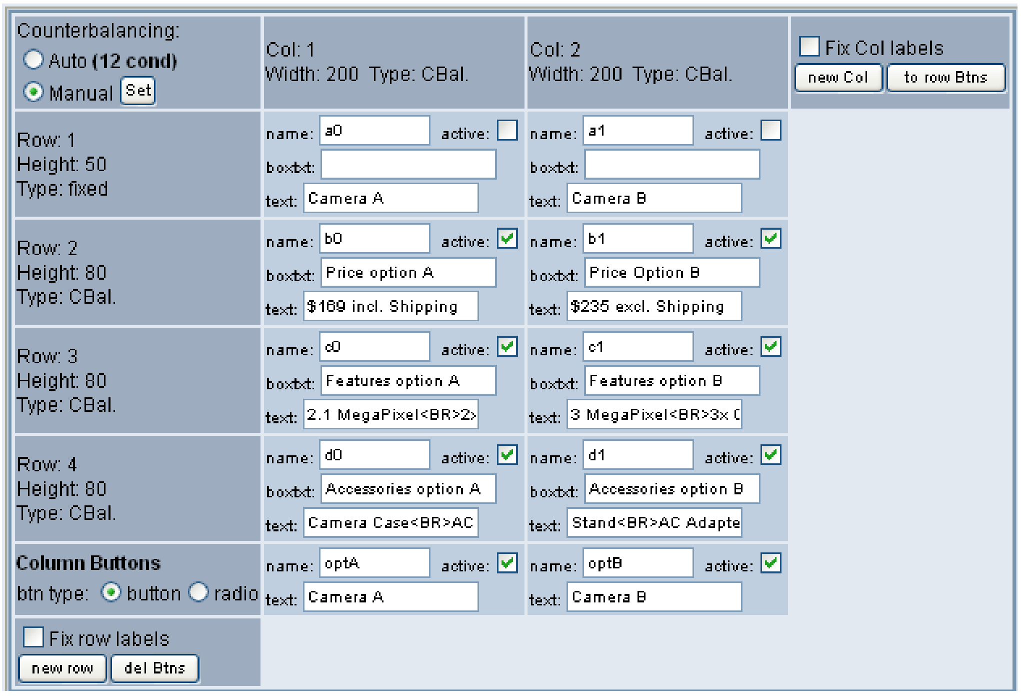
Click on the button of the camera you would prefer most

Camera A	Camera B
Price option A	Price Option B
Accessories option A	Accessories option B
Features option A	Features option B
Camera A	Camera B

Next Page

boxes always shown. For the other 6 cells, enter boxtext's like "Price, Properties and Extra's".

Now we are going to set the counterbalancing conditions. Make sure that the first row is set to fixed, and the other rows and the two columns to 'counterbalanced'. The auto counterbalancing would now result in 12 conditions (see upper left corner), but as we have a specific set of conditions in mind, we will use the manual setting.



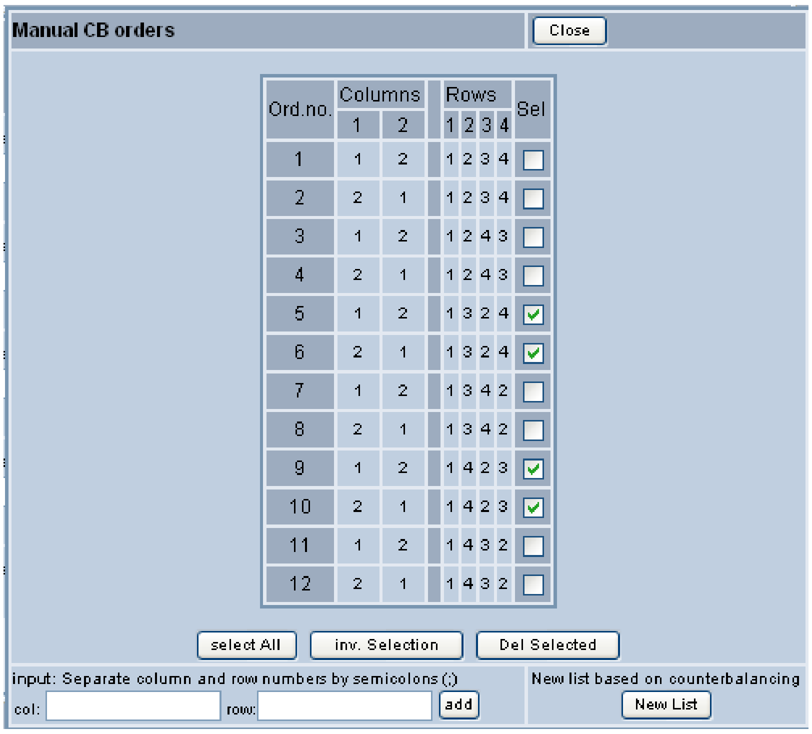
Counterbalancing: Auto (12 cond) [✓ Manual] Set	Col: 1 Width: 200 Type: CBal.	Col: 2 Width: 200 Type: CBal.	Fix Col labels ☐ new Col / to row Btns
Row: 1 Height: 50 Type: fixed	name: a0 active: ☐ boxtxt: text: Camera A	name: a1 active: ☐ boxtxt: text: Camera B	
Row: 2 Height: 80 Type: CBal.	name: b0 active: ✓ boxtxt: Price option A text: $169 incl. Shipping	name: b1 active: ✓ boxtxt: Price Option B text: $235 excl. Shipping	
Row: 3 Height: 80 Type: CBal.	name: c0 active: ✓ boxtxt: Features option A text: 2.1 MegaPixel<BR>2>	name: c1 active: ✓ boxtxt: Features option B text: 3 MegaPixel<BR>3x (	
Row: 4 Height: 80 Type: CBal.	name: d0 active: ✓ boxtxt: Accessories option A text: Camera Case<BR>AC	name: d1 active: ✓ boxtxt: Accessories option B text: Stand<BR>AC Adapte	
Column Buttons btn type: [✓ button] radio	name: optA active: ✓ text: Camera A	name: optB active: ✓ text: Camera B	
☐ Fix row labels new row / del Btns			

The structure we discussed above would have the following structure.

orderno.	0	1	2	3	4	5	6	7
rows	P, F, A	P, F, A	P, A, F	P, A, F	F, A, P	F, A, P	A, F, P	A, F, P
columns	A, B	B, A	A, B	B, A	A, B	B, A	A, B	B, A

Does this look complicated? The manual setting allows setting this easily. Click on set (upper left corner). A dialog will open, with no orders assigned, yet. By clicking on the 'new list' a full set of 12 conditions is generated. We can now easily delete those we do not want, by selecting these rows: this concerns all orders in which row 3 is in the third position (prices are not allowed to be in the middle). See the print screen of the dialog box below were these rows are marked. By clicking "del selected", the four rows will be deleted and we have the 8 conditions we wanted.

Note that you can also manually add more conditions, by adding them one by one in the two boxes at the bottom, separated by semi-colons.


Manual CB orders | Close

Ord.no.	Columns		Rows	Sel
	1	2	1 2 3 4	
1	1	2	1 2 3 4	☐
2	2	1	1 2 3 4	☐
3	1	2	1 2 4 3	☐
4	2	1	1 2 4 3	☐
5	1	2	1 3 2 4	✓
6	2	1	1 3 2 4	✓
7	1	2	1 3 4 2	☐
8	2	1	1 3 4 2	☐
9	1	2	1 4 2 3	✓
10	2	1	1 4 2 3	✓
11	1	2	1 4 3 2	☐
12	2	1	1 4 3 2	☐

select All | inv. Selection | Del Selected

input: Separate column and row numbers by semicolons (;)
col: row: add

New list based on counterbalancing
New List

When all these settings are done, MouselabWEB will control the counterbalancing automatically. The page also shows you the total number of conditions in the upper left corner.

Note: There are two ways in which MouselabWEB can decide which of these orders to show:

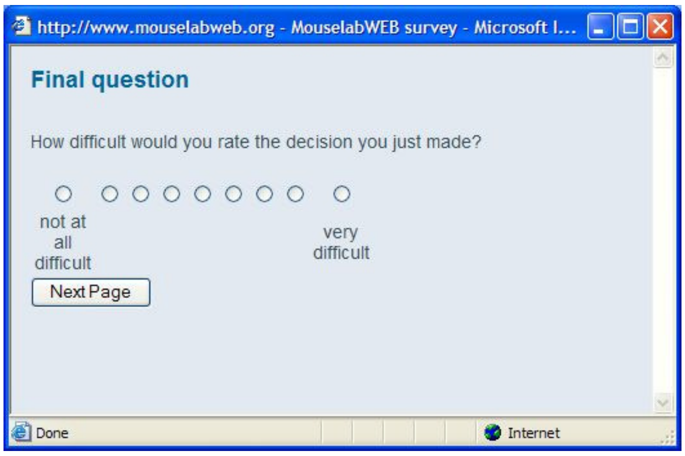
1. By randomly picking one of the orders: The disadvantage is if the sample size is not very large, the number of subjects in each order will not be uniformly distributed. By clicking the randomize button in the general settings, the script will automatically randomize the boxes.
2. By selecting the appropriate order based on a number (e.g., a sequential number assign to the variable condnum) set by the experimenter. MouselabWEB actually selects the order based on the remainder of this number divided by the number of conditions (e.g., in the present camera example, with 8 conditions, if the number is 10 the selected order is 2). This will reduce the chance that the number of subjects in each box is not uniformly distributed, as we could assign the subjects to each of the orders sequentially based on their subject number.

That said, let’s add some more functionality. Naturally, the subject needs to make a choice for one of the cameras. Therefore, click on new buttons on the last row ans select button as type. This will create two buttons underneath the columns. Name them optA and optB and enter Camera A and B for text.

Also, we need to specify the name in the general settings and the follow up page. Name it camera and the next page will be ‘difficult.php’.

To help you somewhat further, this is how the structure should look like:
The MouselabWEB part of the page is done now, only some PRE and POST HTML is needed to explain the subject what is expected of him. Save the page as camera.php and go to the next page.

Page 3 : difficult.php


http://www.mouselabweb.org - MouselabWEB survey - Microsoft I...
Next Page
Done
Internet

Final question

How difficult would you rate the decision you just made?

not at all difficult

very difficult

Contrary to what you might fear given the name, this page is very simple. It contains no MouselabWEB cells, just a scale. The data entered on this page will also be captured and put in the database.

In the general settings, insert the name ‘difficult’ and next page ‘thanks.html’. Also, we again turn on the formcheck to be sure the subject is required to provide an answer.

In the *pre HTML* field, insert a scale of nine points and select *capt. Events*. This way, mouse over etc will also be logged. *Show scale values* can be turned off. On the next part, put 1 to 9 in the first column, and add a label ‘very easy’ in the first box in the second column, and ‘very hard’ in the last.

After creating it, put the following above the scale:

```
<H2>Final question</h2>
<P>How difficult would you rate the decision you just made?</P>
```

Then save the page as *difficult.php*.

Page 4 : thanks.html

This page is already in the MouselabWEB package; it’s nothing more than an end page.

Running the pages

Upload (FTP) the file to your MouselabWEB folder on the Internet. You can start the set of pages with the link:
http://[yourwebfolderlink]/intro.php?subject=[subjectname]&condnum=[condnum].
The names in [brackets] depend on your settings and file names; it could be something like this: http://mywebsite.org/mlweb/intro.php?subject=me&condnum=2

This link will be starting your intro.php page with “me” as subject name and it will use the counterbalancing order associated with the number 2.

Retrieving the data

Run datalyser.php on your site. You’ll see the page names you created in the walkthrough in the download table. Try the replay option on one or more pages/experiments. It’s a very nice way to see what subjects really do in the decision making process. Show table shows all the captured invents of the table and you can also download the tables to CSV files (comma separated values). Make sure the subdirectory tmp is written enabled, this is the directory the CSV files will be created in.