PUPIL

User Documentation

Contents

1. Background

PUPIL ("Pedagogiskt Utvecklings-Project I Labbet", approximately "pedagogical development project for the laboratory") is a teaching tool intended to teach experimental psychology to students within distance education. The system is completely browser based, and does not require and locally installed software, neither for the person designing the experiments nor the students running the experiments.

1.1. *History*

PUPIL began as a pedagogical project in 2009 at MIUN. At the time, the psychology students were divided into two types of classes: students studying at the local campus, and students following the education on a distance.

Demonstrating common psychological experiments to local students were no problem, since they could all be placed in a computer lab in front of pre-configured computers. However, the distance students cause significant headache. At the time, another common software (which we will not mention by name here) was used. It had the ability to save distributable files that could be sent to the students via email or CDs. However, this required that the students had a computer with a reasonably standard configuration, and also that the student was reasonably confident with installing software. Both these requirements were often not fulfilled.

At the time, the university granted funding for pedagogical projects aimed at expanding the teachers' toolboxes and an application was made to develop a system for solving the problems experienced with the distance students. The funding was granted, and the psychology department borrowed personnel from the computer science department to conduct the actual programming.

Since that time the PUPIL project has been expanded upon in several stages and is still being actively developed.

1.2. *The parts of PUPIL*

PUPIL largely consists of two parts: An interface for designing projects, and an interface for running projects. The technical description of these are available in the "system documentation" and will not be repeated here. Suffice to say that both parts are written using Flash (a technique installed on 98% of modern browsers) and they communicate with a central server.

An experiment always consists of a set of "scenes" displayed to the student in a specified order. Each scene presents some stimulus (usually a set of images) and expects the student to react by pressing a key. The data saved are timings paired with what key the student pressed.

This document mainly deals with the design part of the system, as the experiment part should be self-explanatory.

1.3. Terms

PUPIL utilizes a terminology which in some parts reflect its origin as a teaching tool. For reference, here are the terms used in this document:

- **Project, experiment**: A project or an experiment is a set of scenes paired with instructions and settings. This is what a student runs and a teacher designs.
- **Scene**: A scene is what is visible to the student at a single time.
- **Block**: A block is a grouping of scenes within a project.
- **Student, respondent, test person**: A student, respondent or test person, is a person running an experiment. This person usually does not have access to designing experiments.
- **Teacher, author, designer**: A teacher, author or designer, is the person responsible for an experiment, and is usually the person designing it.
- **Image**: An image is an atomic part of a scene. It is an image file (PNG, JPEG or GIF) uploaded and placed in specified spot on a scene.
- **Category**: A set of images thematically belonging to each other, for example “spiders” or “flowers”. When designing scenes, one can assign a category to a position and let the system randomly pick an individual image from the category.
- **Pattern**: A layout approach for placing images on a scene. The “2x2” pattern for example, places images in two rows and two columns on the scene.

2. Logging in

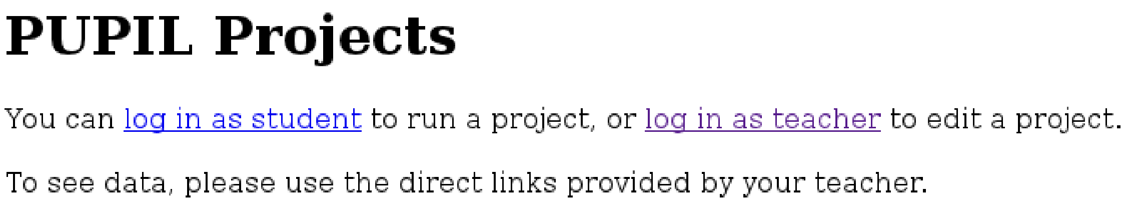
In order to log in to the system, you will need:

- A login name and a password
- The address to the system

It is here assumed that you have got these from your system administrator. Inside MIUN, the address is

http://freud.itm.miun.se/pupil/

When entering the link to the system in your web browser's address field, you should see a page looking a bit like this:



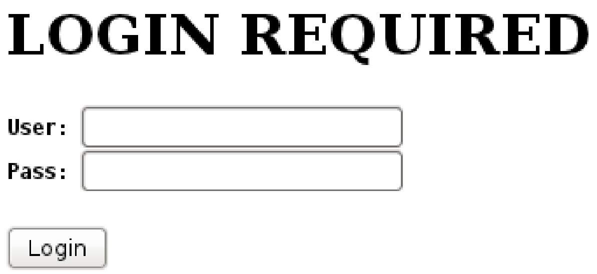
PUPIL Projects

You can log in as student to run a project, or log in as teacher to edit a project.

To see data, please use the direct links provided by your teacher.

You have two links to choose between. In practice, “log in as student” means that you want to *run* a project, and “log in as teacher” that you want *design* or *edit* a project. Also note that student and teacher logins are different: You cannot use a student login for editing projects and not use a teacher login to run a project.

To use the system we will here start by clicking “log in as teacher”. Note that the logins are different for teacher and student, and you cannot use a teacher login to run an experiment, nor can you use a student login to edit projects.

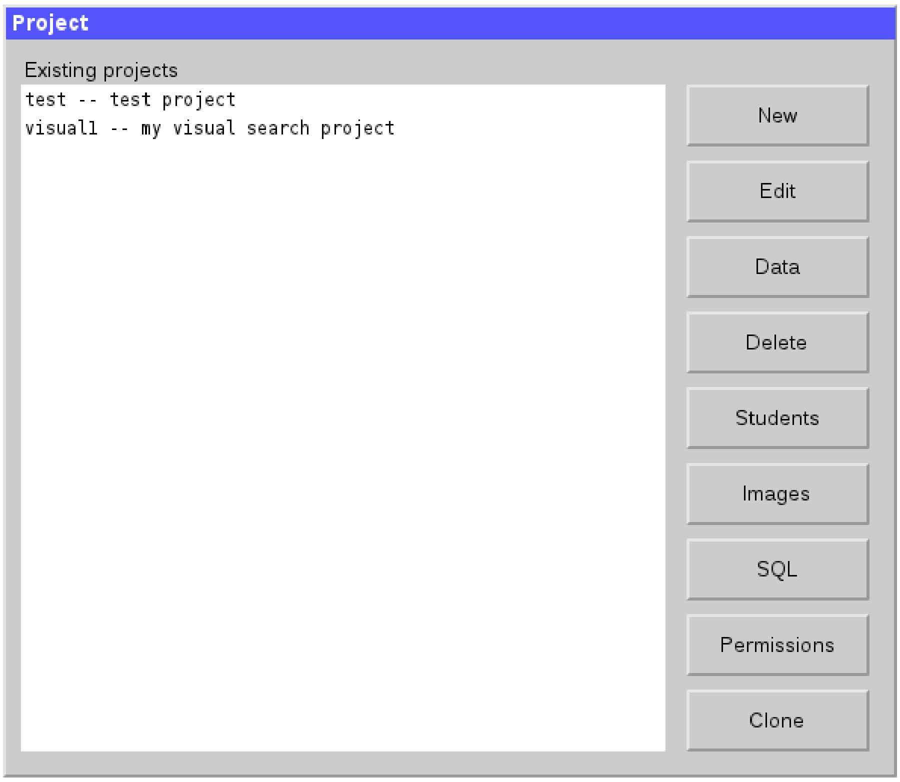


LOGIN REQUIRED

Enter the login name and password that you got from your system administrator.

3. The main window

After having entered correct credentials you will be greeted with the main window of the PUPIL system:



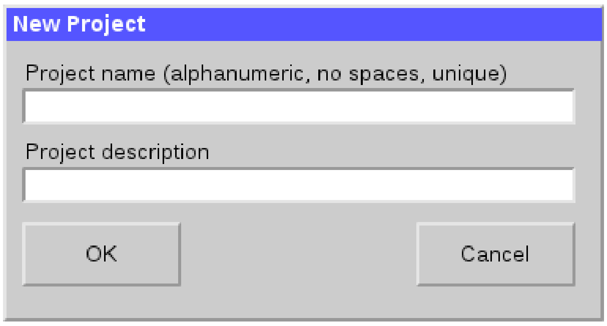
```
test -- test project
visual1 -- my visual search project
```

This window is the list of currently existing projects (experiments) and a list of buttons representing things you might want to do with the projects. The buttons are (from top down) :

- *New*. Create a new project from scratch
- *Edit*. Change or create scenes in a project
- *Data*. Download or clear data created by running a project
- *Delete*. Completely delete a project and everything relating to it. This cannot be undone.
- *Students*. Manage student logins
- *Image*. Manage stimulus material
- *SQL*. Write custom data queries
- *Permission*. Manage permissions for students, ie who is allowed to run a certain project
- *Clone*. Create a new project as a copy of an existing project

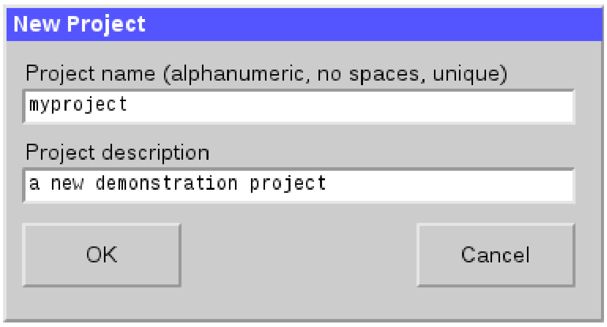
4. Creating a new project

As we want to create a new project from scratch here, we click the “new” button. A window looking like this appears:



The project name is a unique identifier for the project. This should ideally be a single short word. The identifier is used in links and data exports, so keep it simple. The project description is an arbitrary string used to describe the project we create here. In multiuser context, an email to the author of the project might be appropriate information to enter here for example.

Figure out a name and a description you want to use here. For demonstration purposes, we have entered “myproject” in the following screenshot. This will be used from here onwards. If you are following this document as a step-by-step instruction, you might want to use the same name.

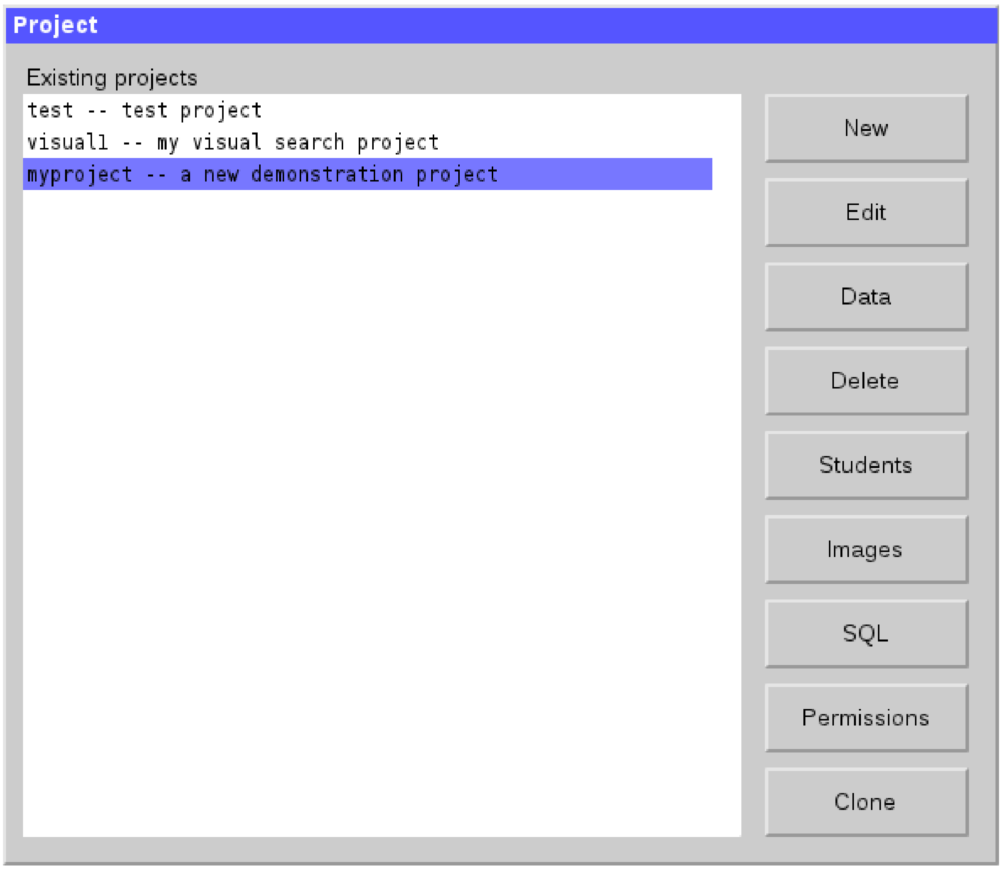


Project name (alphanumeric, no spaces, unique)
myproject

Project description
a new demonstration project

When finished, click “OK”.

The project is now available in the project list:

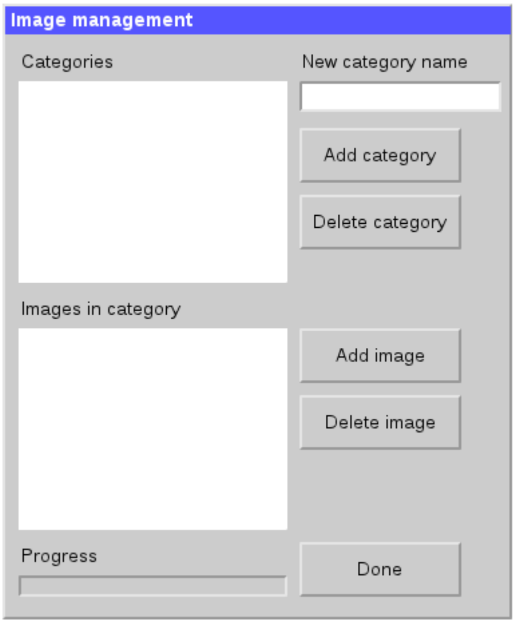


```
test -- test project
visual1 -- my visual search project
myproject -- a new demonstration project
```

5. Managing images

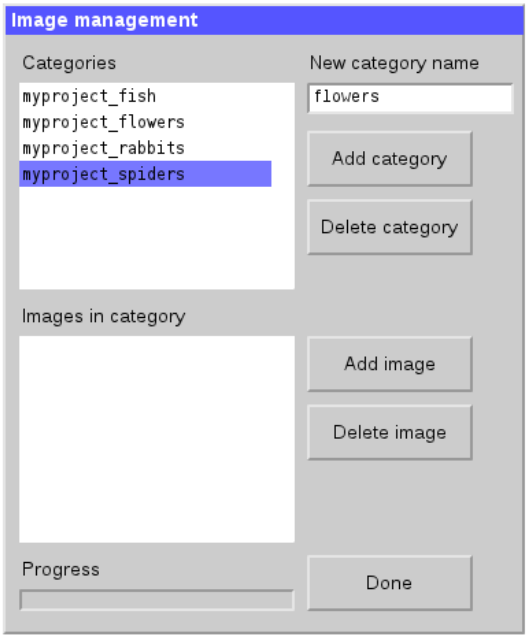
The first thing we are likely to want to do with a project of this kind is to upload stimulus material. For purely text-based experiments this is not necessary.

Mark the “myproject” line in the list and click on the “images” button. A new window appears:



As a reminder from the introduction: All stimulus material consists of images. Images are organized into categories. An image always belong to a category.

In the “image management” window we can see there is an (empty) list of categories and an empty list of images. We will start by creating a new category. Enter “spiders” in the box under “new category name” and click “add category”. Do the same for “fish”, “flowers” and “rabbits”. We now have a list of four categories:

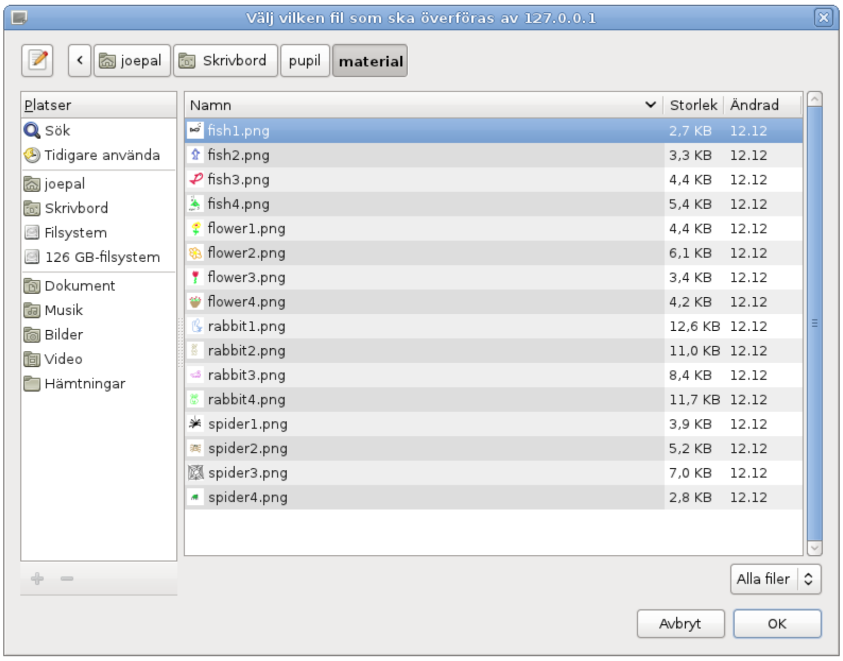


Categories:

myproject_fish
myproject_flowers
myproject_rabbits
myproject_spiders

New category name: flowers

Since every image must belong to a category, we mark the “spiders” category and click “add image” to select an image to upload to the server.



Namn	Storlek	Ändrad
fish1.png	2,7 KB	12.12
fish2.png	3,3 KB	12.12
fish3.png	4,4 KB	12.12
fish4.png	5,4 KB	12.12
flower1.png	4,4 KB	12.12
flower2.png	6,1 KB	12.12
flower3.png	3,4 KB	12.12
flower4.png	4,2 KB	12.12
rabbit1.png	12,6 KB	12.12
rabbit2.png	11,0 KB	12.12
rabbit3.png	8,4 KB	12.12
rabbit4.png	11,7 KB	12.12
spider1.png	3,9 KB	12.12
spider2.png	5,2 KB	12.12
spider3.png	7,0 KB	12.12
spider4.png	2,8 KB	12.12

A window appears where you can select an image to upload. Note that this window can look very different depending on what operating system and browser you are using.

Double-click on the desired image. Repeat the steps (clicking “add image” and selecting an image file) until you have uploaded all the images in the spider category:

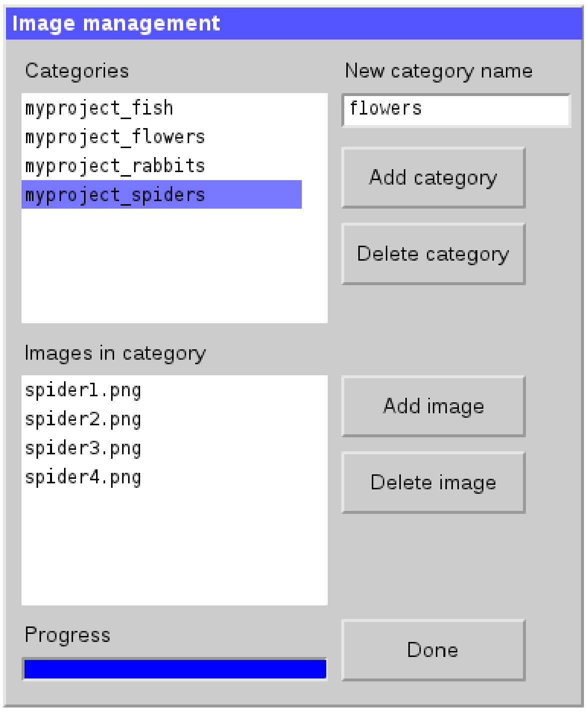


Image management

Categories

myproject_fish
myproject_flowers
myproject_rabbits
myproject_spiders

New category name: flowers

Images in category

spider1.png
spider2.png
spider3.png
spider4.png

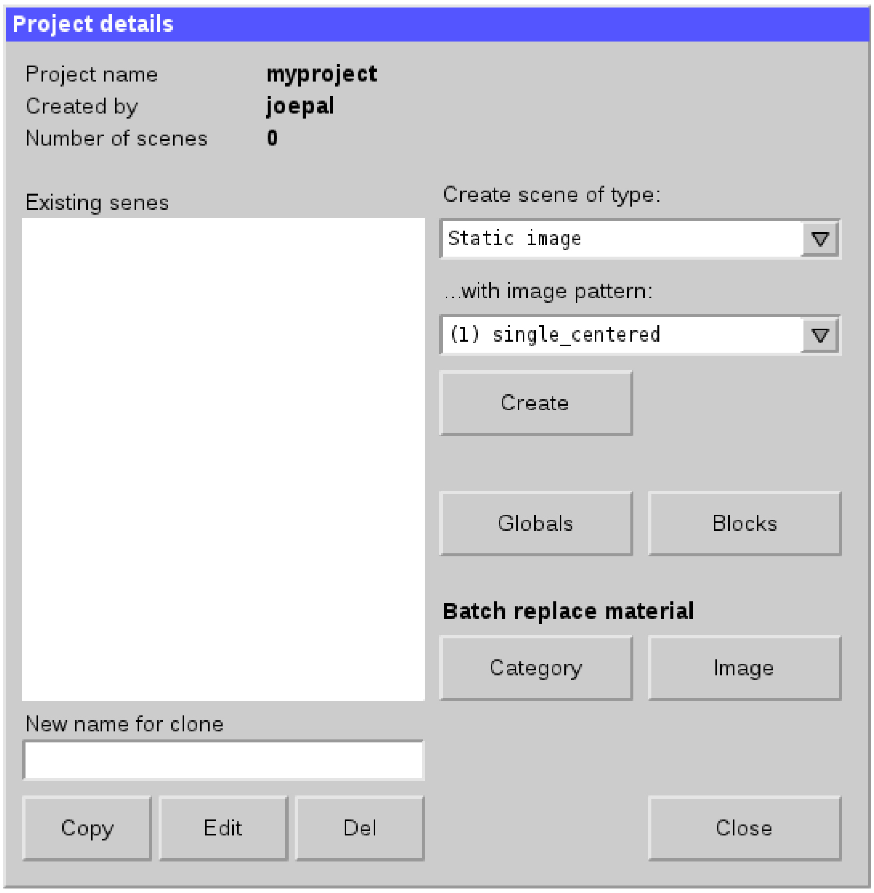
To upload material for another category, mark the category in the list and click “add image”.

When finished with the stimulus material, click “done”.

6. Creating scenes

At this point you should be back at the project list. The “myproject” project should still be marked in the list. If not, mark it.

Click on the “edit” button. The scenes list window appears:



Project details

Project name	**myproject**
Created by	**joepal**
Number of scenes	**0**

Create scene of type: Static image

...with image pattern: (1) single_centered

The main purpose of this window is to manage scenes, but there are some other functions too. The “blocks”, “globals” and “batch replace materials” will be discussed in later sections.

In order to create a scene you need to decide two things: The “scene type” and the “pattern”.

6.1. Types of scenes

The following scene types are available:

- *SI – Static Image*. A scene consisting only of image material. Images are selected explicitly so that one specified image always appears in a designated position.
- *SCRI – Static Category, Random Image*. A scene consisting only of image material. Here categories are tied to positions on the screen. When the scene is displayed, an image is picked at random from the category.
- *SOSI – Static Option, Static Image*. A scene consisting of image material and (optionally) lines of text describe possible options to choose between for the user. For the image material, images are chosen explicitly so that a specific image always appears in a designated position.
- *SORI – Static Option, Random image*. A scene consisting of image material and (optionally) lines of text describe possible options to choose between for the user. In the image material, categories are tied to positions on the screen. When the scene is displayed an image is picked at random from the category.
- *Text-based*. An information screen only consisting of text.

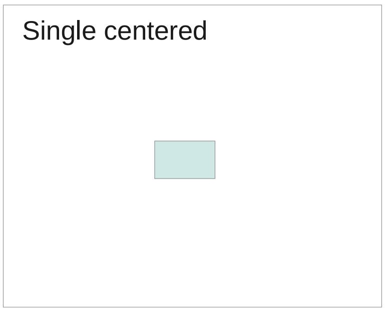
The scene types and how to use them will be discussed in more detail below.

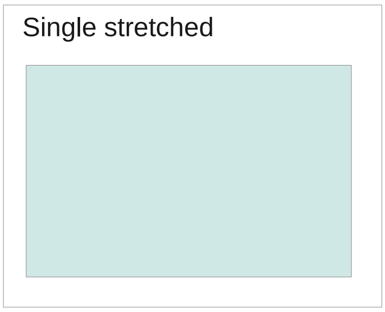
6.2. Patterns

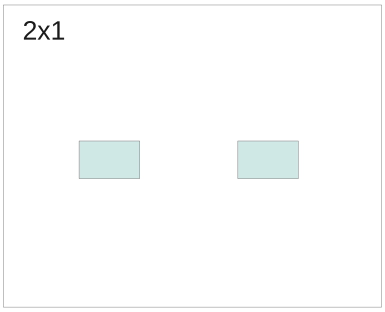
A pattern is a set template for how material is laid out on a scene, ie on what position on the screen an image is show. Currently there are 14 defined patterns, here listed together with how many images they require in total.

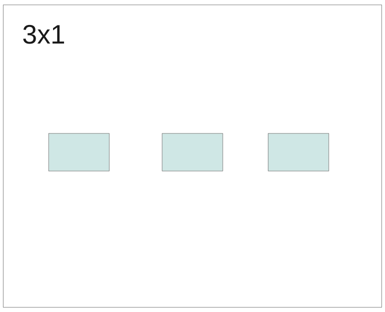
Pattern name	**Number of images**
Single centered	1
Single stretched	1
2x1	2
3x1	3
1+2	3
2+1	3
2x2	4
4 as plus	4
5 as x	5
5 as plus	5
3 x 2	6
3+1+3	7
3+2+3	8
3x3	9

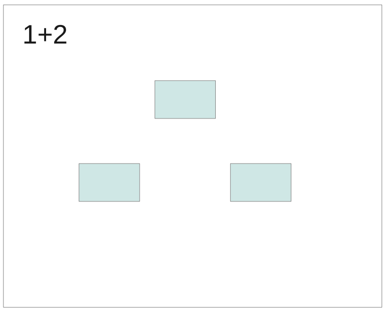
The following page demonstrates how the patterns are laid out in practice.

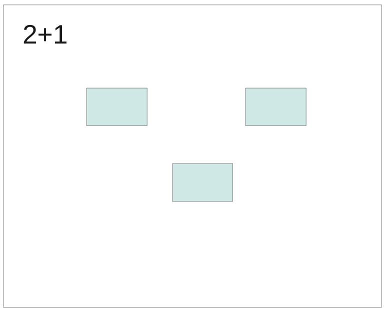

Single centered

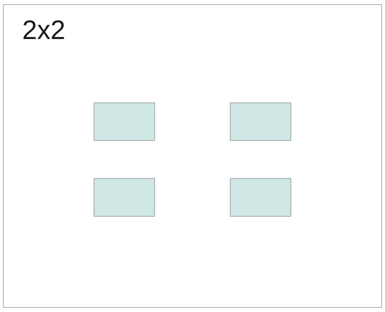

Single stretched

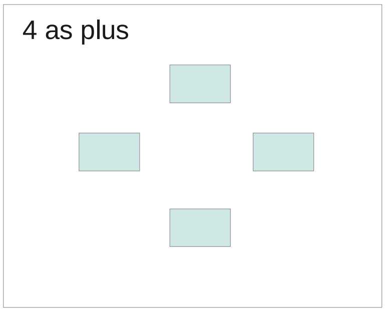

2x1

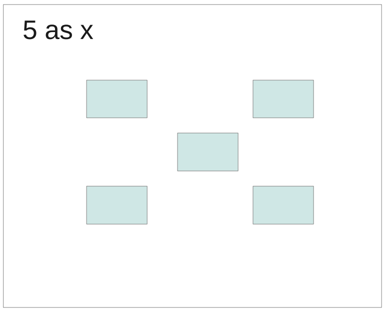

3x1

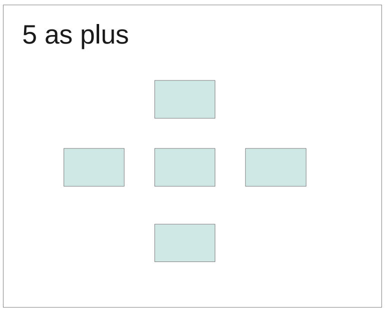

1+2

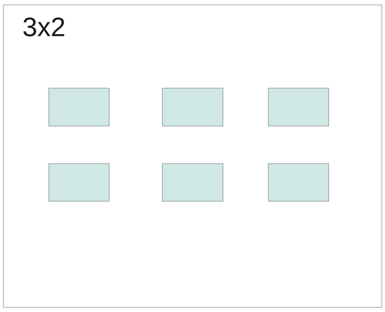

2+1

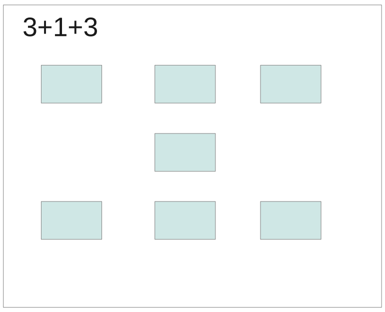

2x2

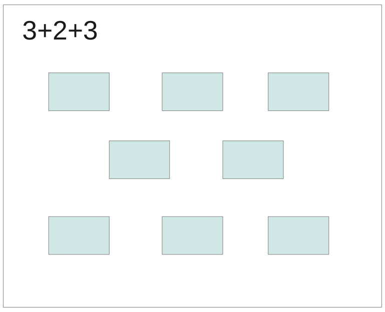

4 as plus

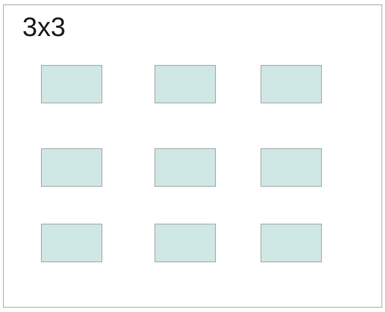

5 as x


5 as plus


3x2

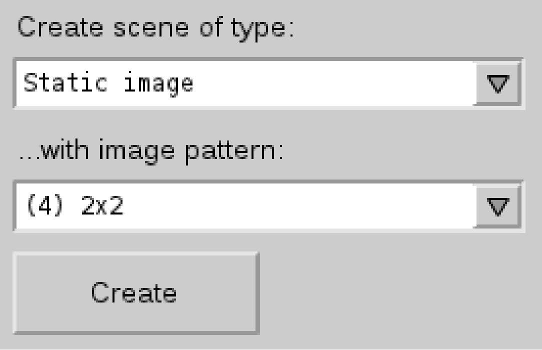

3+1+3


3+2+3


3x3

A pattern has to be chosen when creating a scene, as it determines how many images are tied to the scene. This is also the only way to control the layout of the scene, apart from the fundamental layout determined by the scene type.

In the following sections, the pattern “2x2” has been chosen, and the scene type is varied.



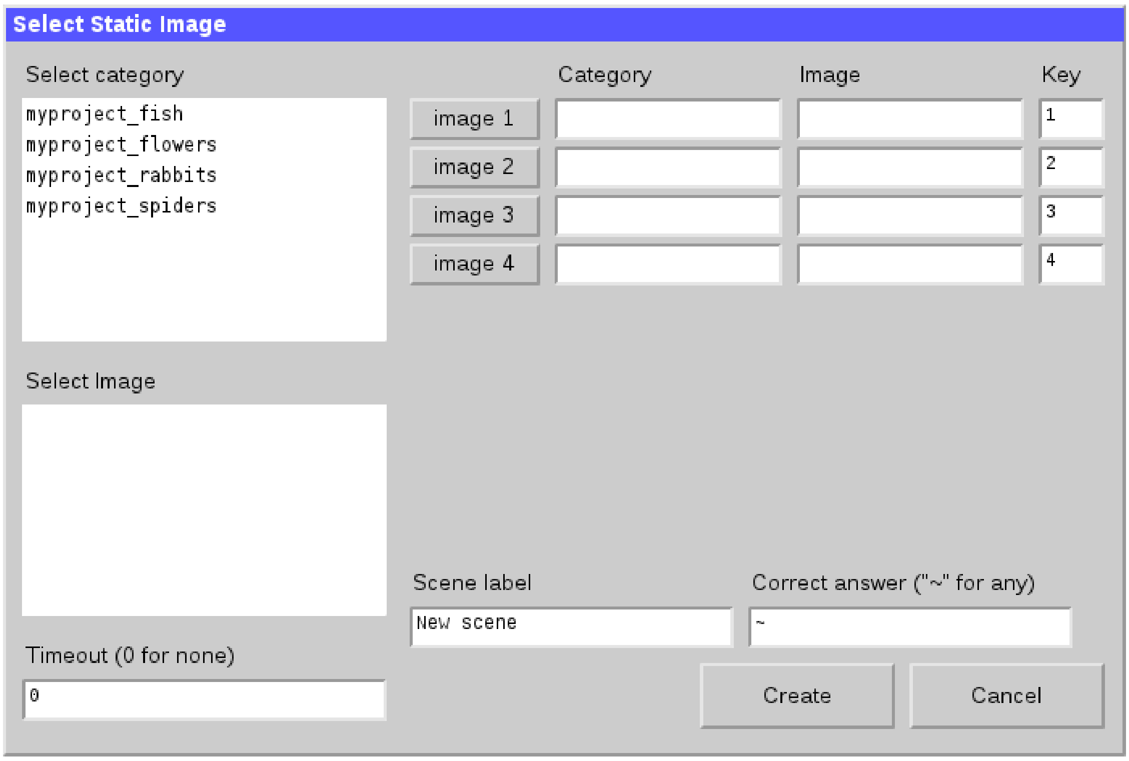
Create scene of type: Static image

...with image pattern: (4) 2x2

In the image pattern drop-down box, the number in parenthesis is the number of images required for the pattern.

6.3. SI – Static image

In the static image scene type we lay out exact images in specified positions. The scene will thus always look the same and there is no randomization. When entering the scene settings we get empty fields for each of the required images. Since the pattern requires fours images, we have four lines of images to specify:



Select Static Image

Select category

myproject_fish
myproject_flowers
myproject_rabbits
myproject_spiders

Select Image

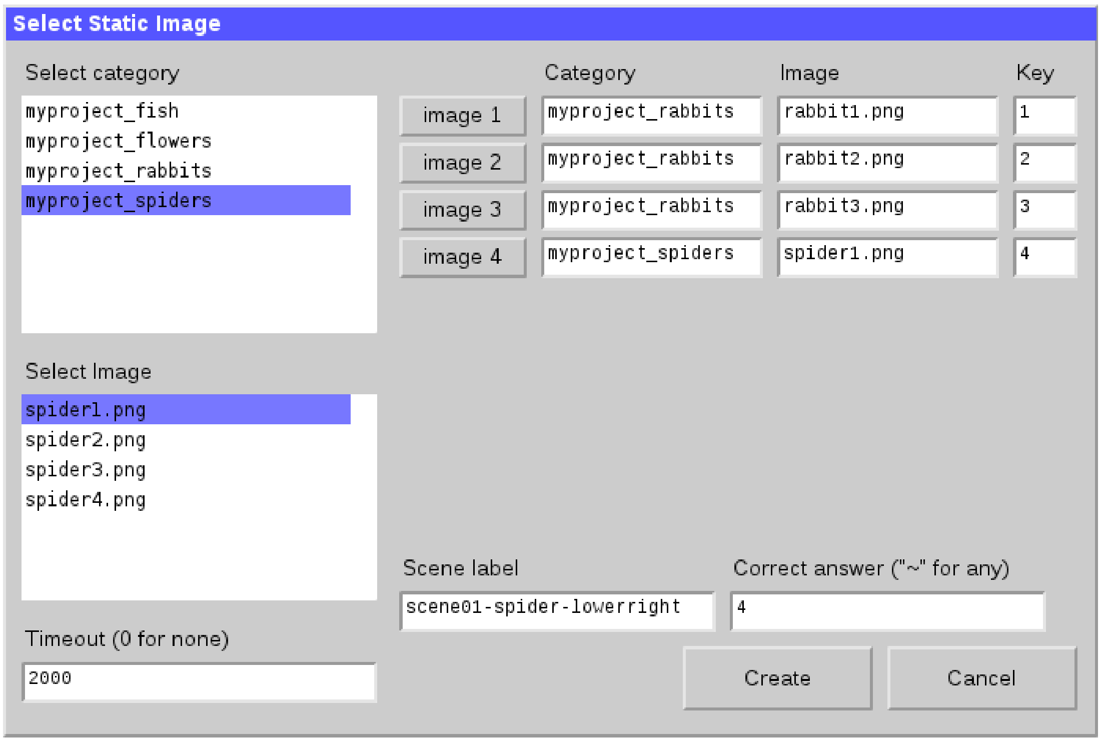
	Category	Image	Key
image 1			1
image 2			2
image 3			3
image 4			4

Scene label: New scene

Correct answer ("~" for any): ~

Timeout (0 for none): 0

In our example we want the respondent to find a spider amongst a set of rabbits. For the first three lines we thus first mark the rabbit category and select a rabbit image. After having selected the rabbit image we click on the “image” button, for example “image 1” to fill the first line. For the final line we click the spider line, select a spider image and click the image 4 button.



Select category

- myproject_fish
- myproject_flowers
- myproject_rabbits
- myproject_spiders

Select Image

- spider1.png
- spider2.png
- spider3.png
- spider4.png

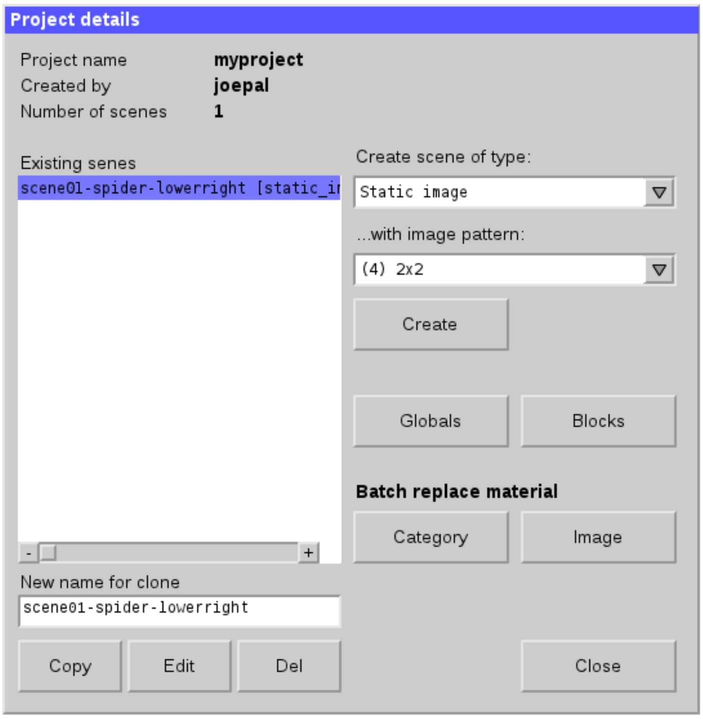
	Category	Image	Key
image 1	myproject_rabbits	rabbit1.png	1
image 2	myproject_rabbits	rabbit2.png	2
image 3	myproject_rabbits	rabbit3.png	3
image 4	myproject_spiders	spider1.png	4

Timeout (0 for none): 2000

Scene label: scene01-spider-lowerright

Correct answer ("~" for any): 4

Since scenes will be showed in alphabetical order unless we enable randomization, we set the scene name to something including a number. Further, we set the correct answer to 4 since that is the image that should be found. Finally we set the timeout to 2000 milliseconds. If the respondent doesn't click anything within two seconds, the scene is marked as a fail. Once done with the settings, click create. The scene now appears in the scene list:



Project name	**myproject**
Created by	**joepal**
Number of scenes	**1**

Existing senes

scene01-spider-lowerright [static_i

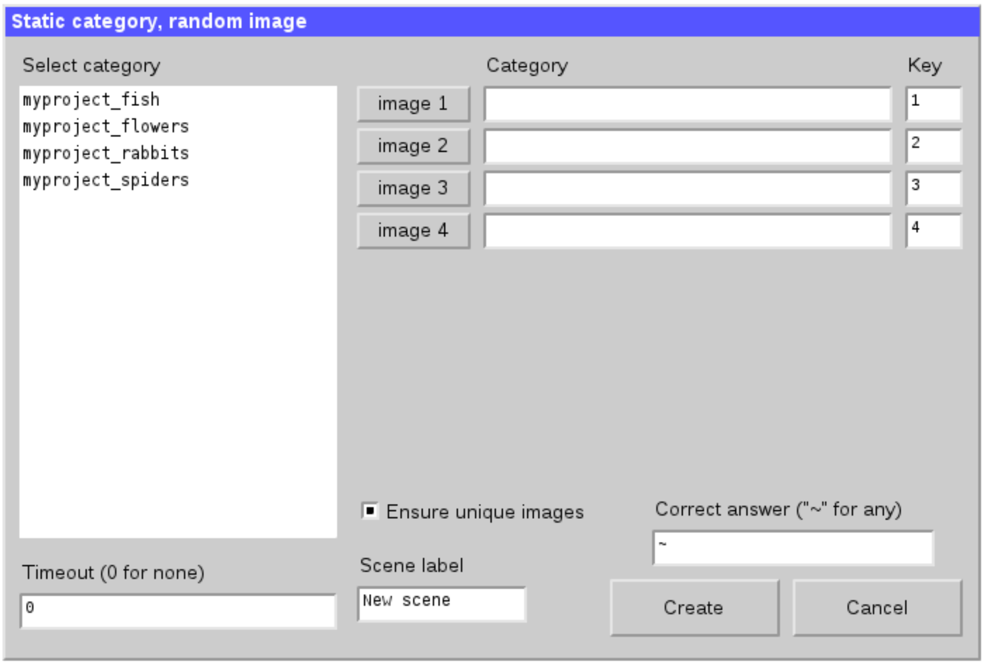
Create scene of type: Static image

...with image pattern: (4) 2x2

New name for clone: scene01-spider-lowerright

6.4. SCRI – Static Category, Random Image

The next scene type is the “Static Category, Random Image”. It looks basically the same as the static image, but here we lay out categories in designated position and randomly pick an image from the category each time the scene is displayed. The scene will thus look different between trials. This is likely the most commonly used scene type.



Static category, random image

Select category

myproject_fish
myproject_flowers
myproject_rabbits
myproject_spiders

	Category	Key
image 1		1
image 2		2
image 3		3
image 4		4

☑ Ensure unique images

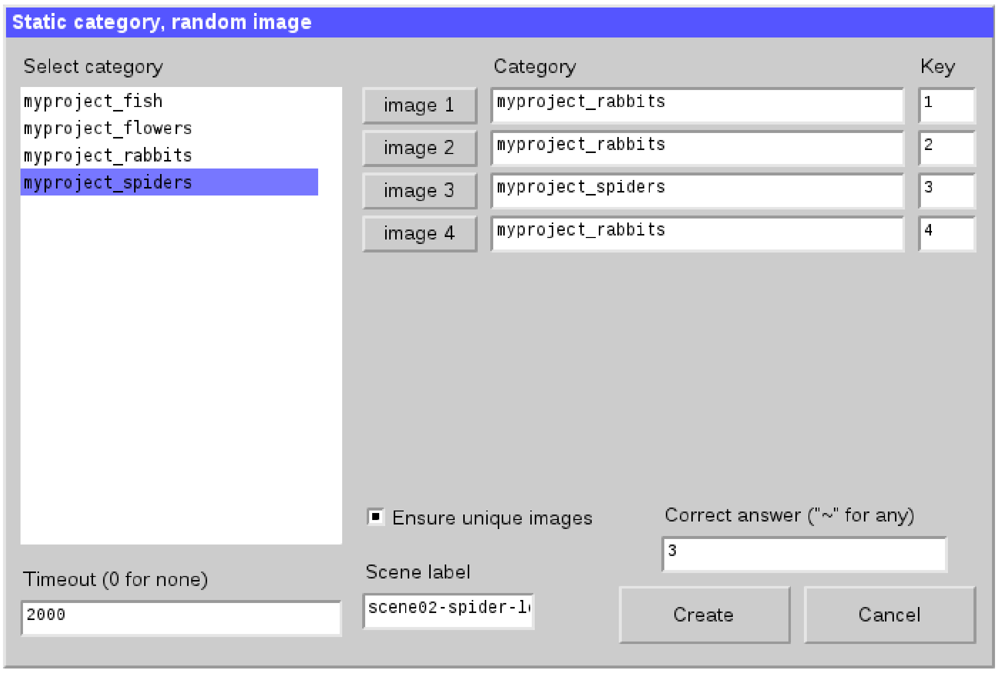
Correct answer ("~" for any): ~

Timeout (0 for none): 0

Scene label: New scene

Create Cancel

We will create a scene with three rabbits and one spider. Thus we select rabbit and click the image button for each line that should be a rabbit line. Then we select spider and click the remaining line.



Static category, random image

Select category

myproject_fish
myproject_flowers
myproject_rabbits
myproject_spiders

	Category	Key
image 1	myproject_rabbits	1
image 2	myproject_rabbits	2
image 3	myproject_spiders	3
image 4	myproject_rabbits	4

☑ Ensure unique images

Correct answer ("~" for any): 3

Timeout (0 for none): 2000

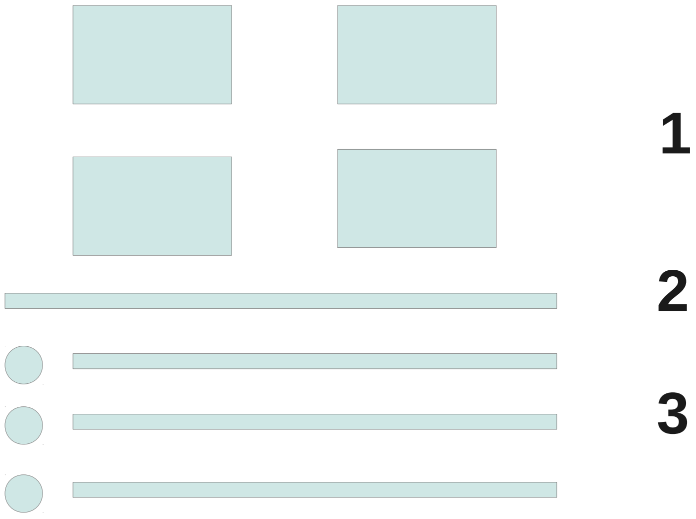
Scene label: scene02-spider-l

Create Cancel

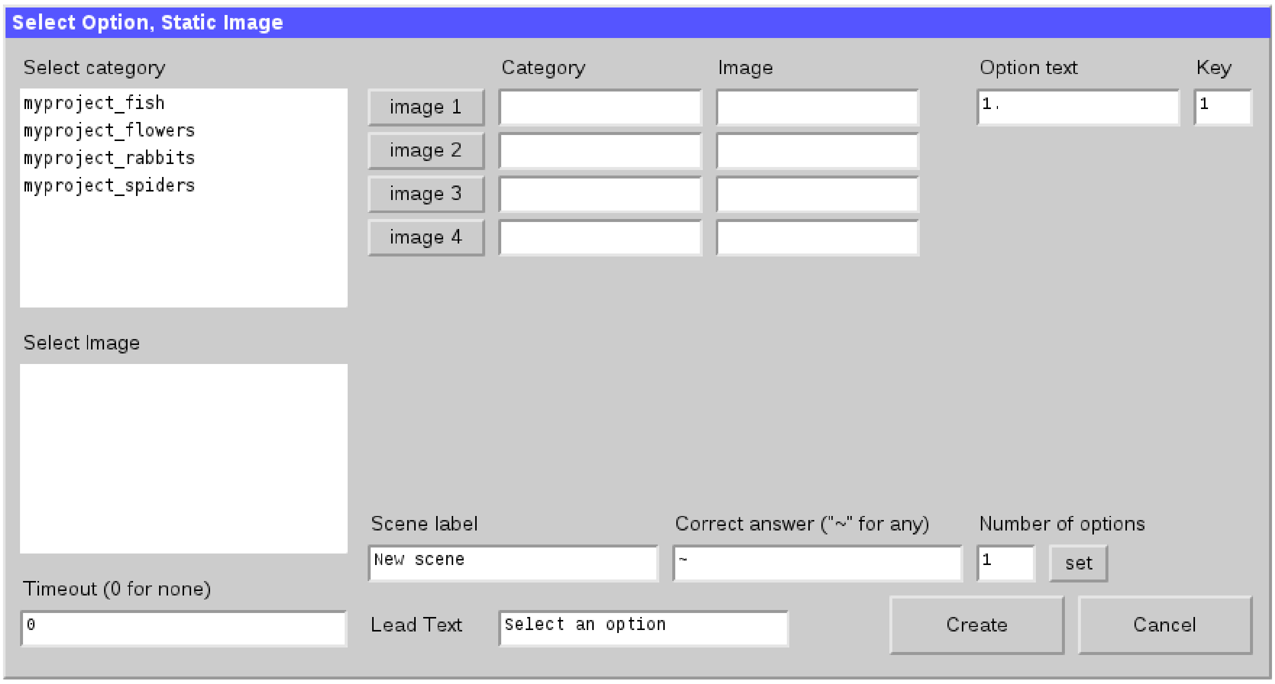
The only option otherwise differing from the static image scene type is the "ensure unique images", which is pre-selected. If this is checked, the system will ensure that the same image cannot be displayed more than once within the same scene. Note that this will result in an error when running the experiment if there are not enough images within a category to fill the scene.

6.5. SOSI – Static Option, Static Image

In the "Static Option, Static Image" we combine the basic static image scene with textual options. These can be of a different number than the images. Schematically, the scene will look like this:



At the top (1) we layout images as with the previous scene type. Then we have a lead text (2), usually a question. Finally we have a number of possible answers (3) each with a key and a text string. When creating a scene the settings window will look like this:



Select Option, Static Image

Select category:
myproject_fish
myproject_flowers
myproject_rabbits
myproject_spiders

Option text: 1.
Key: 1

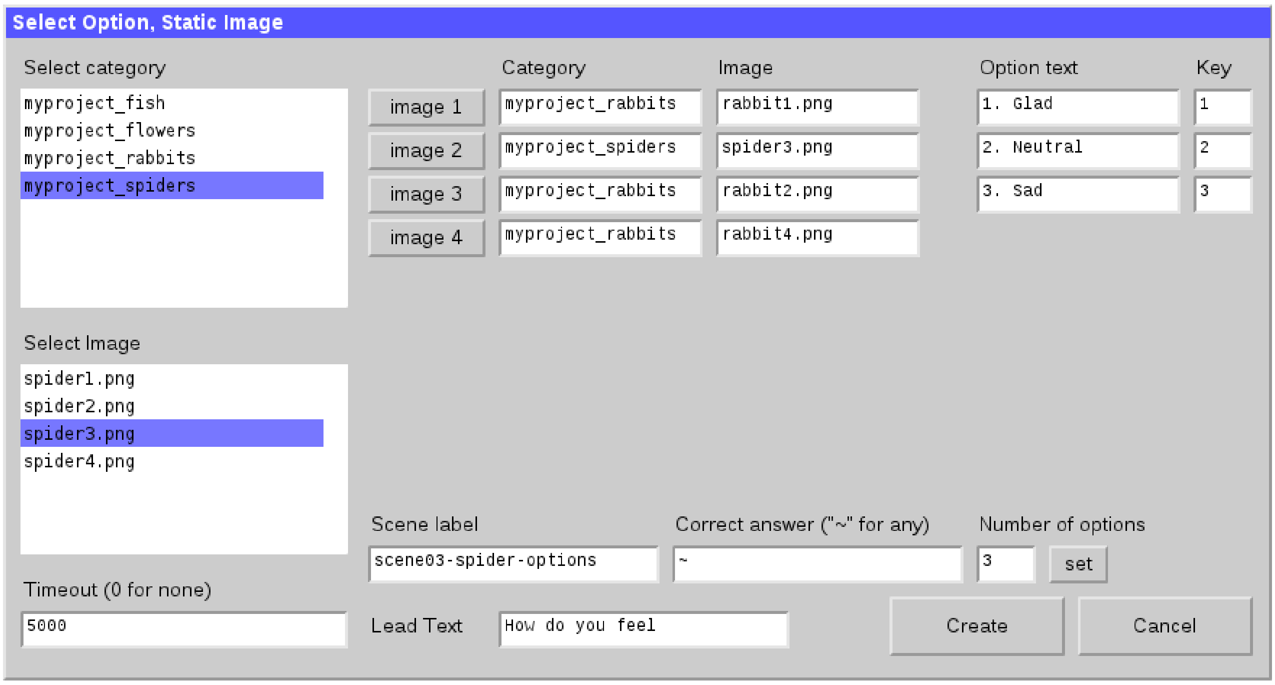
Scene label: New scene
Correct answer ("~" for any): ~
Number of options: 1

Timeout (0 for none): 0

Lead Text: Select an option

The number of image lines are set up as with the previous scenes, but we also have a column with option texts. The number of options can be changed by editing the text box and clicking "set".

A finished scene could look like this:



Select Option, Static Image

Select category

myproject_fish
myproject_flowers
myproject_rabbits
myproject_spiders

Select Image

spider1.png
spider2.png
spider3.png
spider4.png

Timeout (0 for none): 5000

	Category	Image
image 1	myproject_rabbits	rabbit1.png
image 2	myproject_spiders	spider3.png
image 3	myproject_rabbits	rabbit2.png
image 4	myproject_rabbits	rabbit4.png

Option text	Key
1. Glad	1
2. Neutral	2
3. Sad	3

Scene label: scene03-spider-options

Correct answer ("~" for any): ~

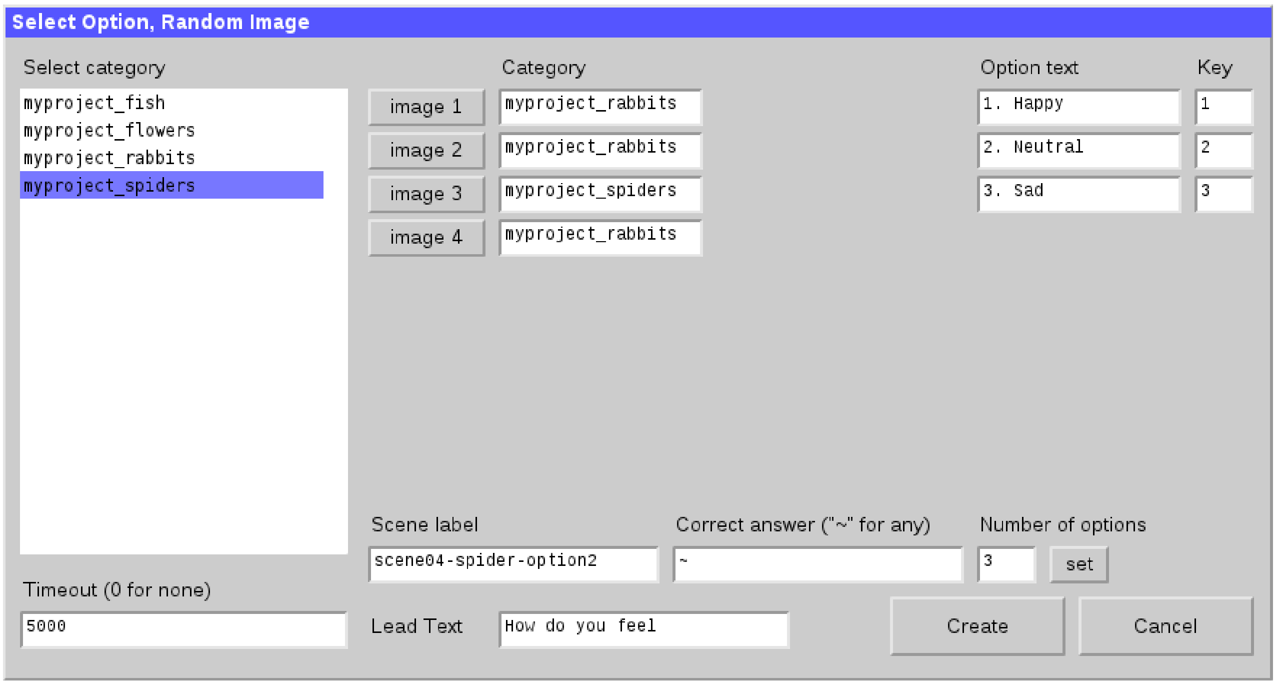
Number of options: 3

Lead Text: How do you feel

Here we have laid out images as before but created three options and written a lead text. Since any key is valid, we have left the “correct answer” as “~”, which means any answer will be marked as correct.

6.6. SORI – Static Option, Random Image

The “Static Option, Random Image” is the category correspondent of the scene type with options. It works the same as SOSI, but here we lay out categories rather than images:



Select Option, Random Image

Select category

myproject_fish
myproject_flowers
myproject_rabbits
myproject_spiders

Timeout (0 for none): 5000

	Category
image 1	myproject_rabbits
image 2	myproject_rabbits
image 3	myproject_spiders
image 4	myproject_rabbits

Option text	Key
1. Happy	1
2. Neutral	2
3. Sad	3

Scene label: scene04-spider-option2

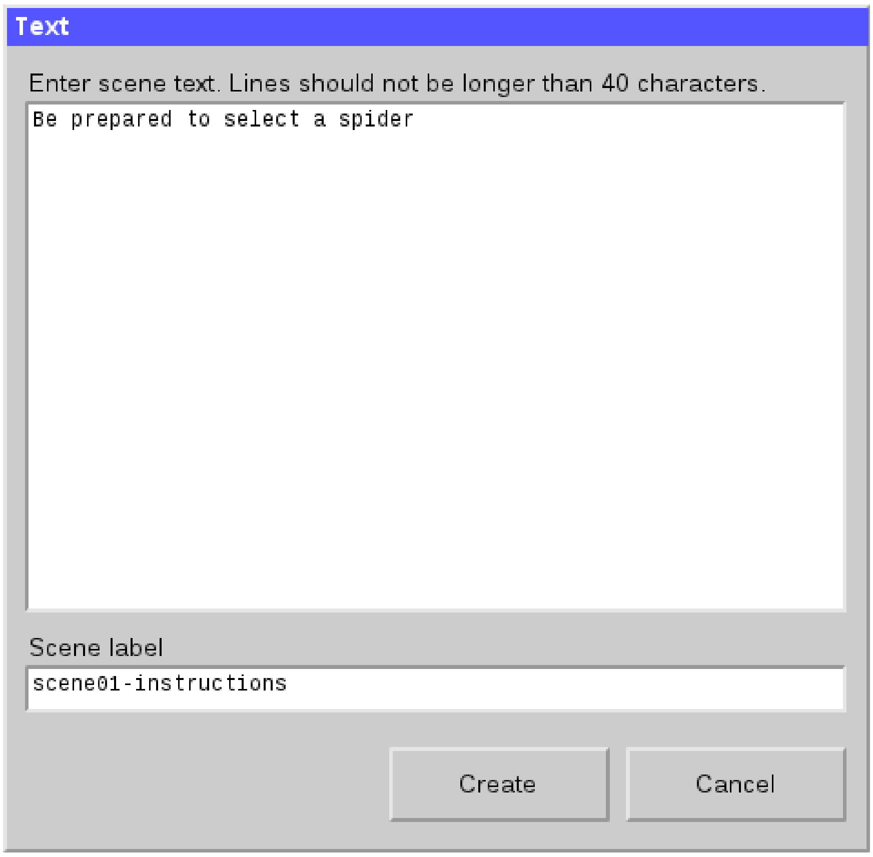
Correct answer ("~" for any): ~

Number of options: 3

Lead Text: How do you feel

6.7. Text-based

The odd bird in the collection is the text-based scene. Here we do not lay out any images at all. Instead we only show a set of text strings. The strings will be centered on the screen when displayed to the respondent.



Enter scene text. Lines should not be longer than 40 characters.

```
Be prepared to select a spider
```

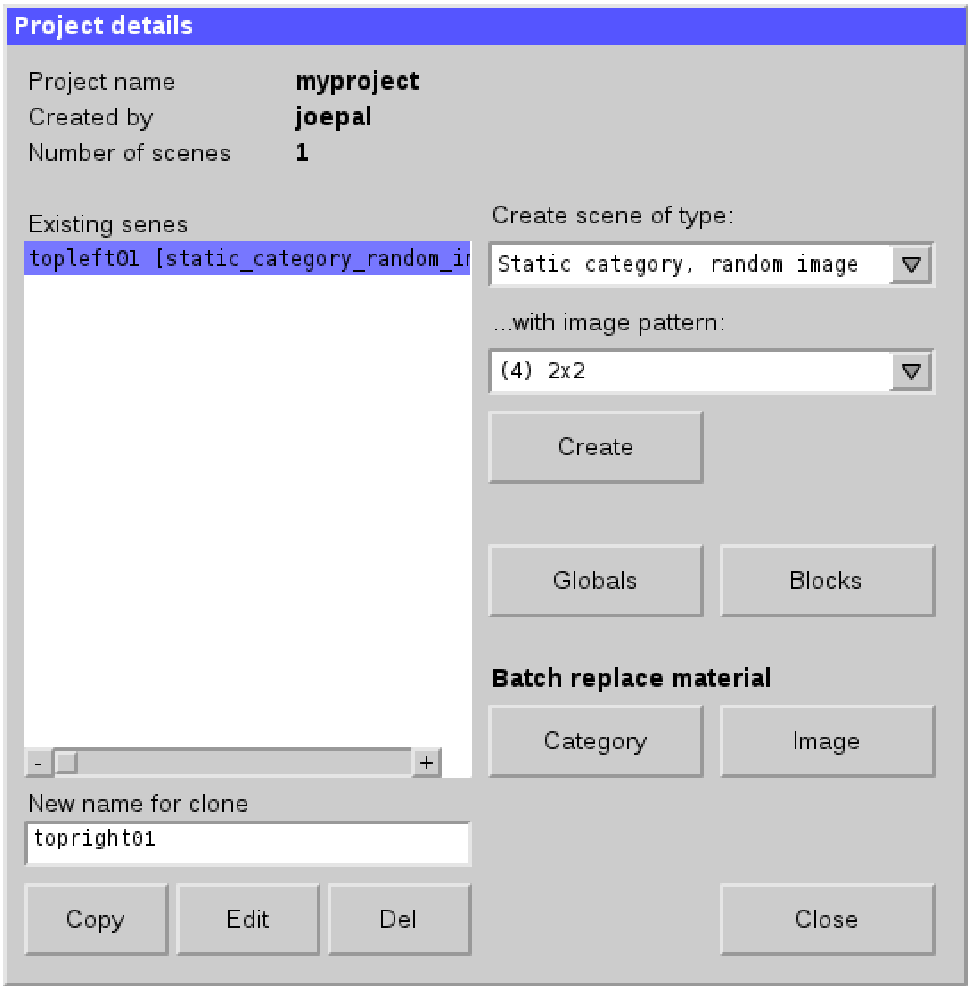
Scene label

```
scene01-instructions
```

Each line in the text field above will end up as a separate line of text on the scene as displayed to the respondent.

6.8. Cloning scenes

Manually creating scenes can be tedious. Luckily it's possible to clone a scene and, if necessary, edit the clone.



Project details

Project name myproject
Created by joepal
Number of scenes 1

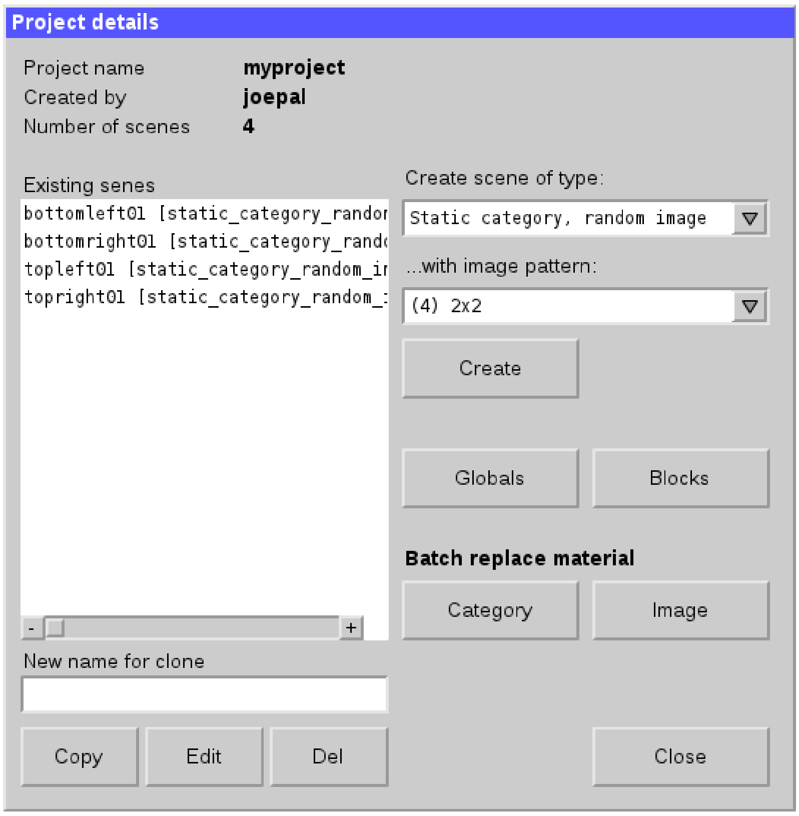
Existing senes
topleft01 [static_category_random_in

Create scene of type: Static category, random image

...with image pattern: (4) 2x2

New name for clone
topright01

Here I have created a 2x2 SCRI scene with a spider in the top left position and rabbits in the other positions. Since we want scenes with spiders in each position respectively, we mark the line with the scene, enter a new name in the “new name for clone” box and click “copy”. Repeat for each position.



Project details

Project name **myproject**
Created by **joepal**
Number of scenes **4**

Existing senes

```
bottomleft01 [static_category_rando
bottomright01 [static_category_rand
topleft01 [static_category_random_i
topright01 [static_category_random_
```

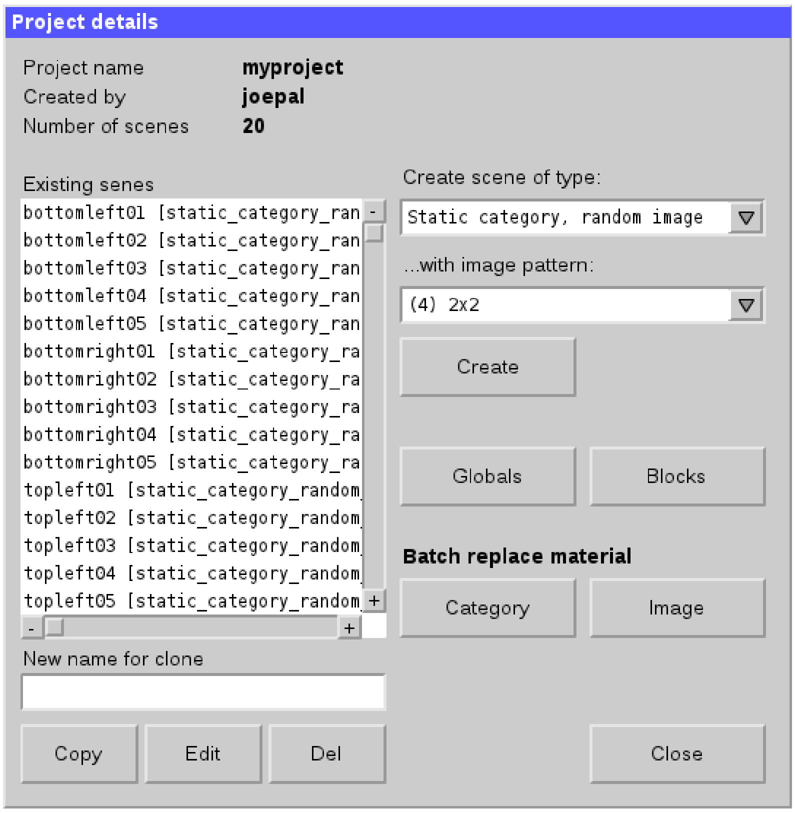
Create scene of type: Static category, random image

...with image pattern: (4) 2x2

Batch replace material

New name for clone

Having made the copies, we can mark a scene and click “edit”. A window looking like the create scene window will pop up and we can change the scene as applicable. Having made the changes we can then clone the scenes again to get the number of scenes we want to display to the respondent. Here we plan on ensuring each position is shown to the respondent five times:



Project details

Project name **myproject**
Created by **joepal**
Number of scenes **20**

Existing senes

```
bottomleft01 [static_category_ran
bottomleft02 [static_category_ran
bottomleft03 [static_category_ran
bottomleft04 [static_category_ran
bottomleft05 [static_category_ran
bottomright01 [static_category_ra
bottomright02 [static_category_ra
bottomright03 [static_category_ra
bottomright04 [static_category_ra
bottomright05 [static_category_ra
topleft01 [static_category_random
topleft02 [static_category_random
topleft03 [static_category_random
topleft04 [static_category_random
topleft05 [static_category_random
```

Create scene of type: Static category, random image

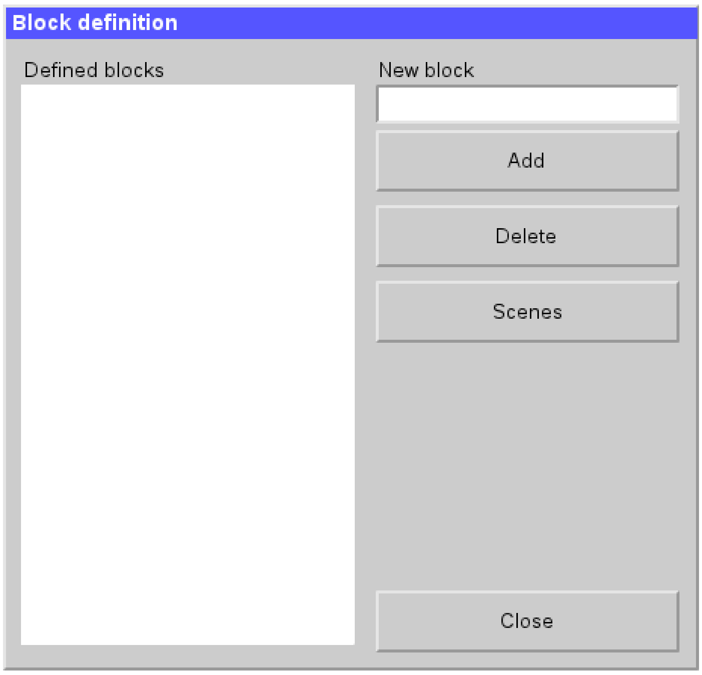
...with image pattern: (4) 2x2

Batch replace material

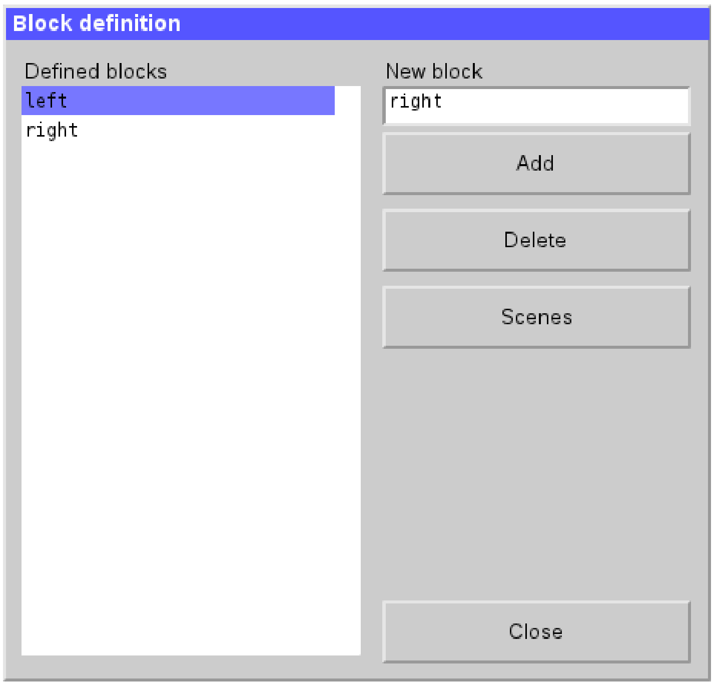
New name for clone

7. Creating blocks

In order to display a scene, it needs to belong to a “block”. To open the block settings click the “blocks” button in the scene list window.



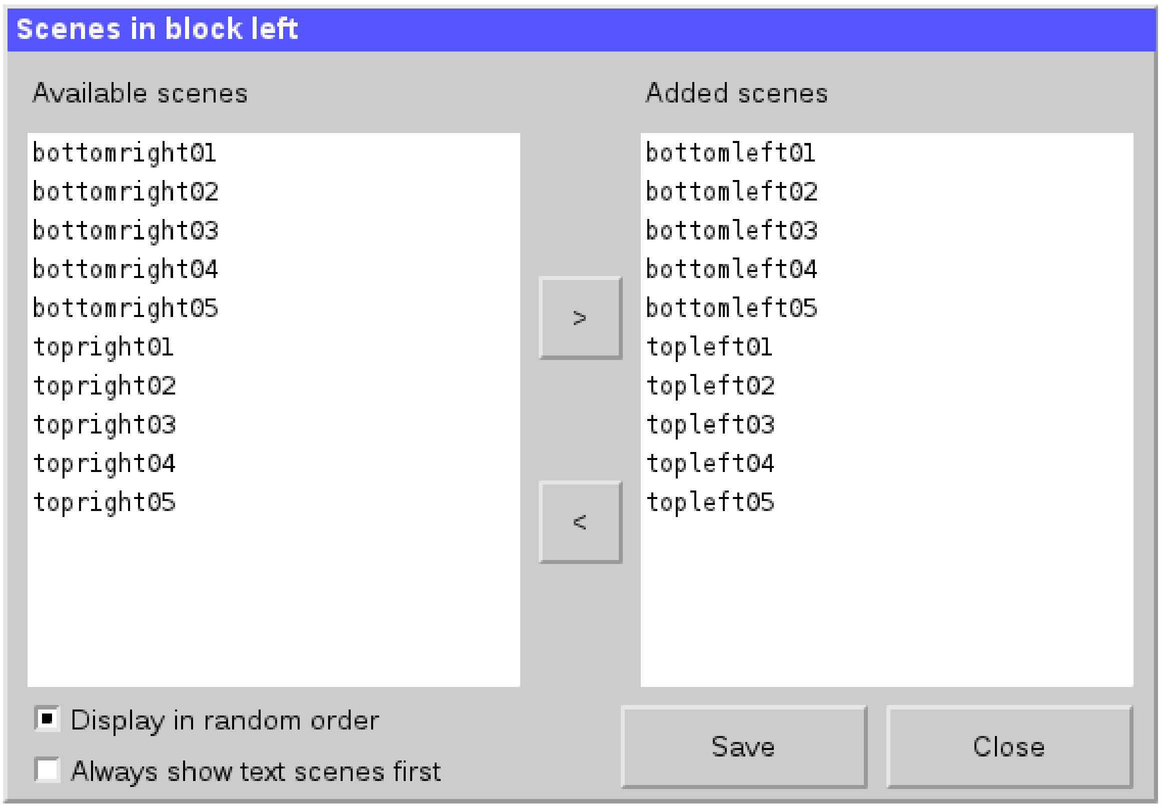
Blocks are usually shown in alphabetical order. This can be overriden, but we will return to that later. For now we will for demonstration purposes create two blocks which we will organize our scenes in. We will have one block where spiders appear to the left in the image and one block where spider appear to the right:



Defined blocks: left, right

New block: right

To assign scenes, we click the “left” block and click “scenes”. A window with a list of available scenes appears.



Scenes in block left

Available scenes	Added scenes
bottomright01	bottomleft01
bottomright02	bottomleft02
bottomright03	bottomleft03
bottomright04	bottomleft04
bottomright05	bottomleft05
topright01	topleft01
topright02	topleft02
topright03	topleft03
topright04	topleft04
topright05	topleft05

☒ Display in random order

□ Always show text scenes first

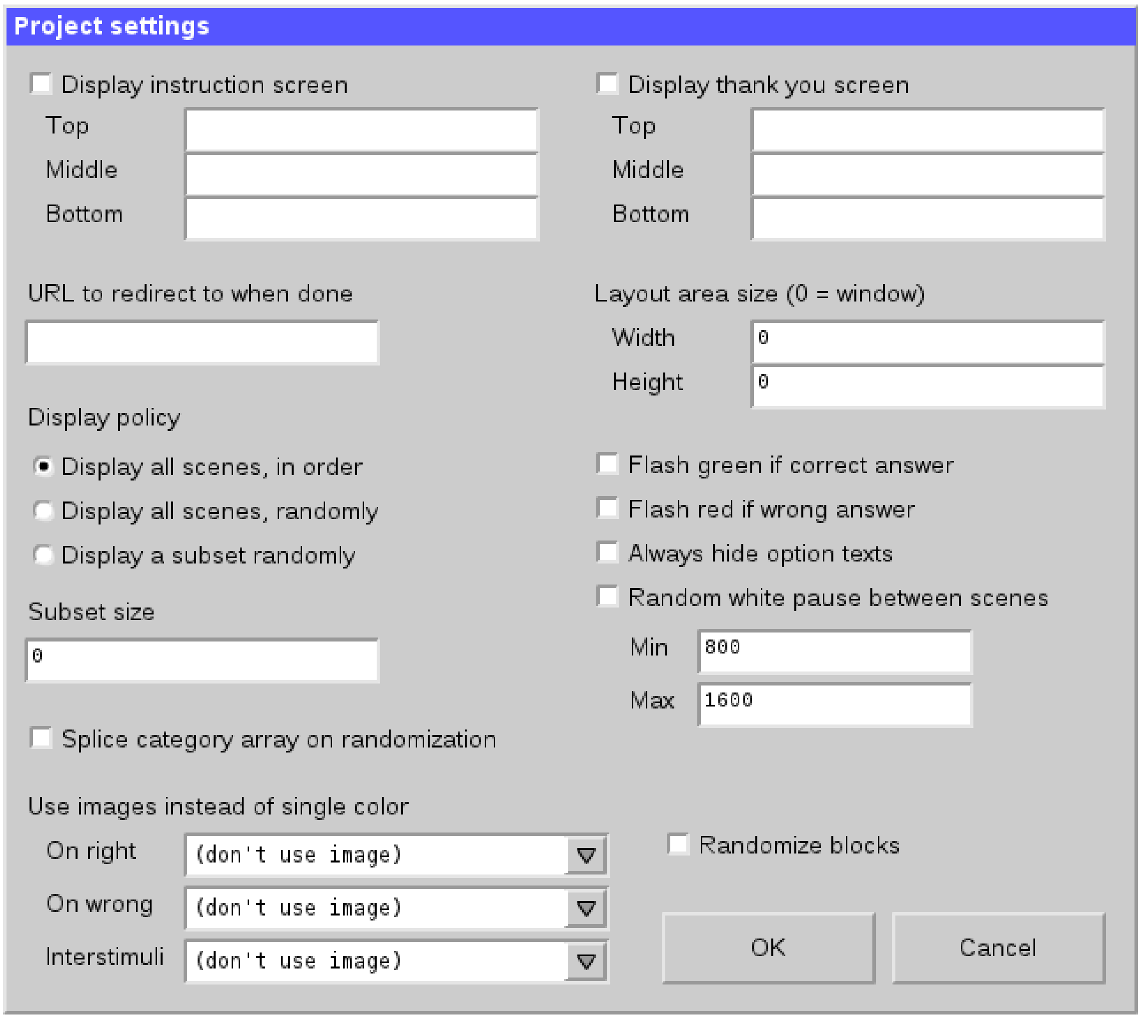
To assign a scene to a block we mark the image and click “>”. The scene is then moved to the list of added scenes.

Apart from the scenes we have to checkboxes. “Display in random order” applies to the *scenes within the block*. Note that this is different from displaying the *blocks* in random order (the setting for displaying the blocks in random order is somewhere else).

If we opt for displaying the scenes in random order we can use “always show text scenes first” to ensure that instruction scenes appear before the other scenes.

8. Project settings

Having created scenes and assigned them to blocks we can finish up our project by setting global options. To open this window go to the scene list window and click “Globals”.



Project settings

□ Display instruction screen

Top

Middle

Bottom

□ Display thank you screen

Top

Middle

Bottom

URL to redirect to when done

Layout area size (0 = window)

Width 0

Height 0

Display policy

● Display all scenes, in order

○ Display all scenes, randomly

○ Display a subset randomly

Subset size

0

□ Flash green if correct answer

□ Flash red if wrong answer

□ Always hide option texts

□ Random white pause between scenes

Min 800

Max 1600

□ Splice category array on randomization

Use images instead of single color

On right (don't use image)

On wrong (don't use image)

Interstimuli (don't use image)

□ Randomize blocks

8.1. *Text screens*

As an alternative to creating text scenes, text can be shown before and after the project is run. In the setting window these are called “instruction screen” and “thank you screen”.

To enable one of these screens check the checkbox and enter text in any or all of the provided text boxes.

The text here will be showed first and last in the project, independent of randomization and block settings

8.2. Redirecting

The normal behavior of the system is to show a popup window with a message informing the respondent once all scenes have been displayed.

As an alternative, the system can redirect the web browser to another web page. In order to do so, enter the address in the “URL to redirect to” text box. If left empty, the system will go with the default behavior.

8.3. Layout area

When laying out images, PUPIL will normally use all available screen space. However, this will cause the project to look different depending on the screen size of the respondent. It may also look strange on very large monitor.

As an alternative, it is possible to specify an area centered on the screen. If specified, PUPIL will lay out all images within these boundaries.

To go for this approach, specify width and height (in pixels) in the width and height text boxes. If left at 0, the entire screen space will be used.

8.4. Pauses and feedback

Normally, all scenes are displayed with no delays. However, it is possible to specify both that there should be a pause with a feedback and a random delay before the next scene is displayed.

For feedback, the easiest approach is to check “flash green” and “flash red”. If these are checked, the entire screen will be red or green respectively for a short period of time after a scene is responded to.

For pauses, an empty screen can be shown between each scene. To enable this, check the “random white pause” checkbox and decide between what time intervals the pause length should be randomized. The min/max numbers are specified in milliseconds.

As an alternative to the red/green/empty screens, it is possible to instead display an image centered on the screen. To specify this, use the “use image instead of single color” drop-down boxes. In order to select an image, it has to have been uploaded using the normal stimulus material upload approach (see above).

8.5. Randomization

For randomization, note that the “display policy” and “subset” settings conflict with the block randomization settings. These options will be removed in a future version.

To randomize the order in which blocks are shown, check the “randomize blocks” checkbox.

To ensure that an image is only shown once within the entire project, check the “splice category” checkbox. Note that this can lead to problems if there are too few images within a category.

9. Managing students and permissions

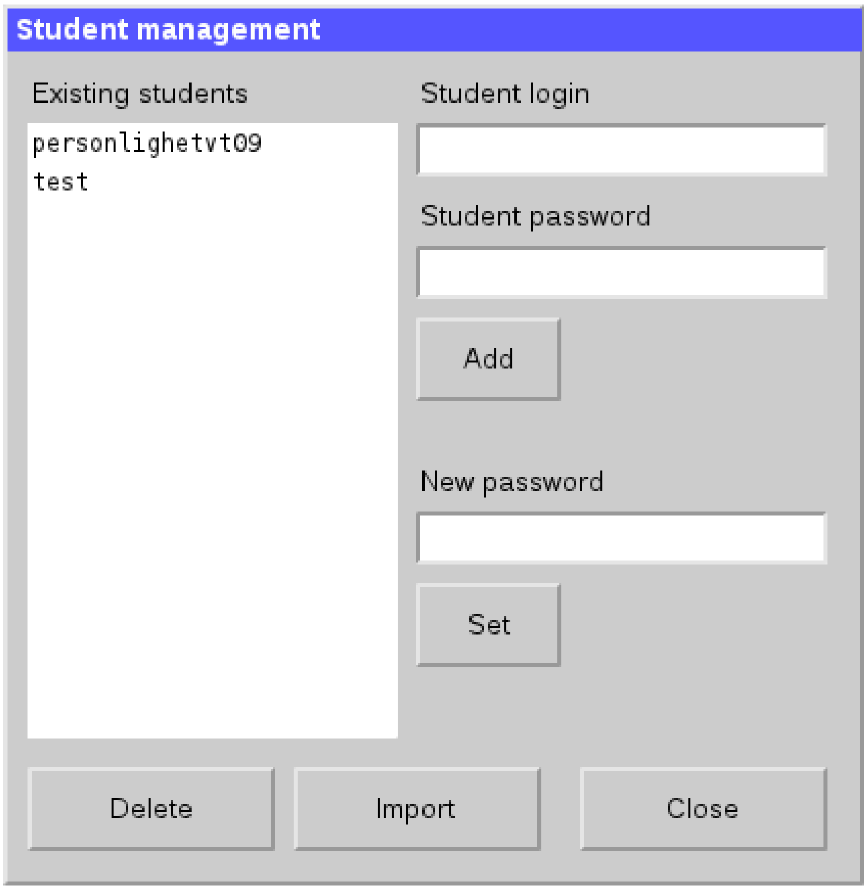
True to its origin as a teaching system, respondents are called “students”. In order to run an experiment, the respondent need to have a student login.

Some parts of the operations within these settings are specific for Mid Sweden University. If the local password does not match for a student, an LDAP-lookup will be made to the university directory service to see if the student is valid centrally. These settings are hard-coded and will not work outside the MIUN firewall.

To enter the student settings, click the “students” button in the project list window.

9.1. Adding students manually

Each student has a login name and a password:



Student management

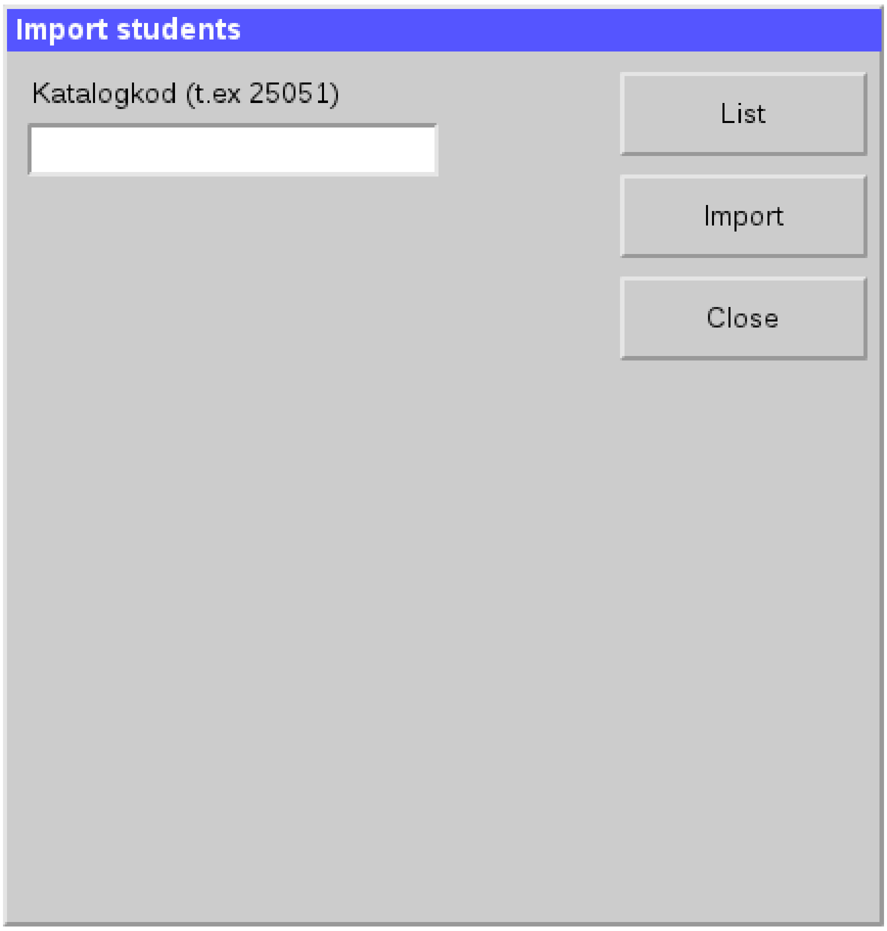
Existing students

personlighetvt09
test

To add a student, enter the name and the password in the respective text boxes and click “add”.

9.2. *Batch imports*

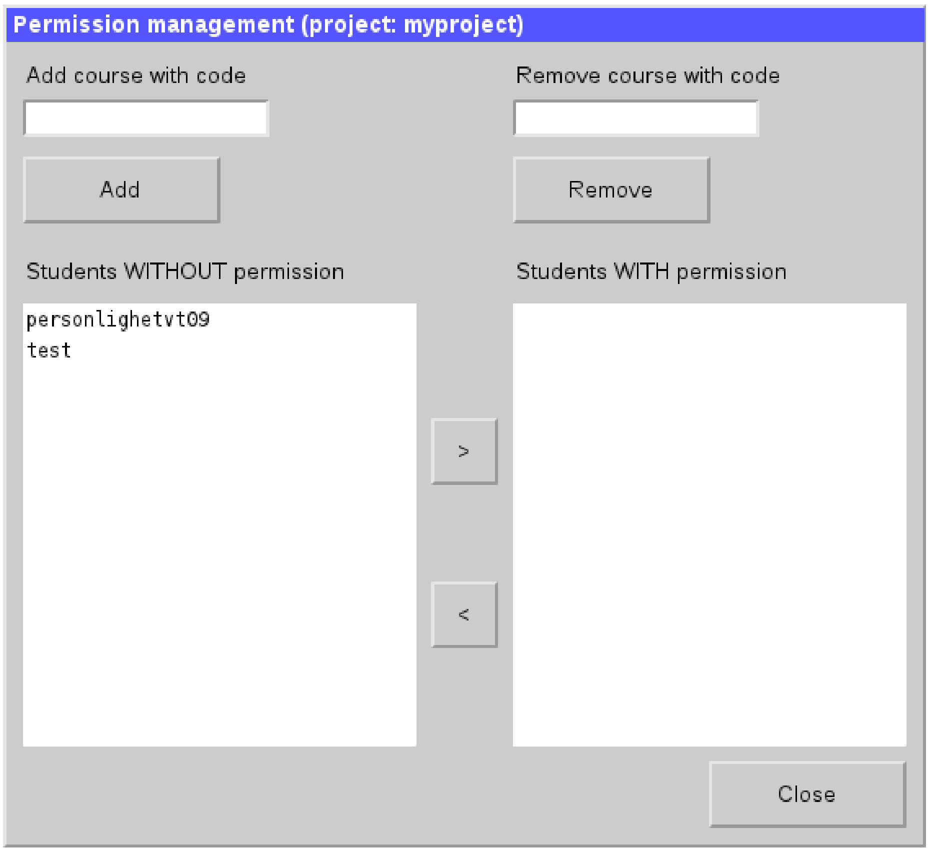
Another MIUN-specific setting is the possibility to import students from the central database. In order to do so, enter the course code for the course and click “list”. If it looks likely that the set of students are correct, click “import”.



If this does not work, it is always possible to add the students manually using the previous approach. The passwords will still be checked against the central database.

9.3. Permissions

In order for a student to be allowed to run an experiment, a permission need to be granted. In order to manage permissions, click the “permissions” button in the project list window:



In the normal approach, mark a student and assign permission by clicking “>”. As an alternative within MIUN, all student belonging to a certain course can be added automatically by entering the course code and clicking “add”.

10. Running the experiments

Once permissions have been granted, students can be let in to run the project.

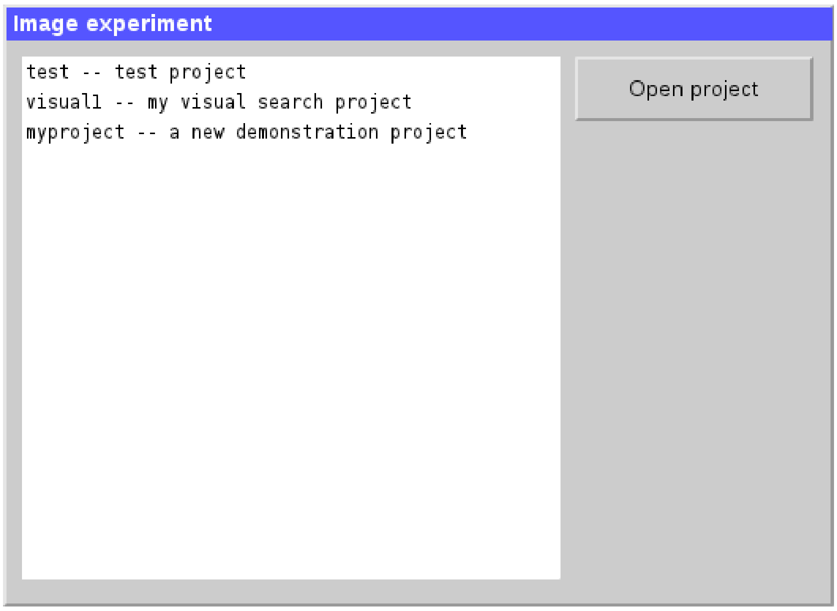
10.1. *Distributing a link*

There are two ways to lead students to the correct project. The first is to have them go the start page of PUPIL. Within MIUN this link is:

http://freud.itm.miun.se/pupil/

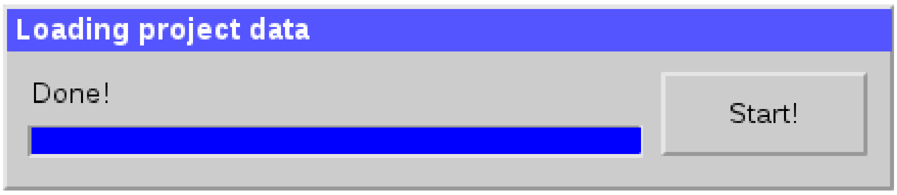
Outside MIUN, ask your system administrator for the correct address.

Once the student has logged in, he can choose the project from a list:



```
test -- test project
visual1 -- my visual search project
myproject -- a new demonstration project
```

The student here has to select a project and click "open project". If all is well, the material will load and a start project popup will be shown:



As an alternative, the project can be selected automatically by including it in the link given to the student.

In order to do so, use the following URL:

http://freud.itm.miun.se/pupil/flash/experiment.jsp?project=[project name]

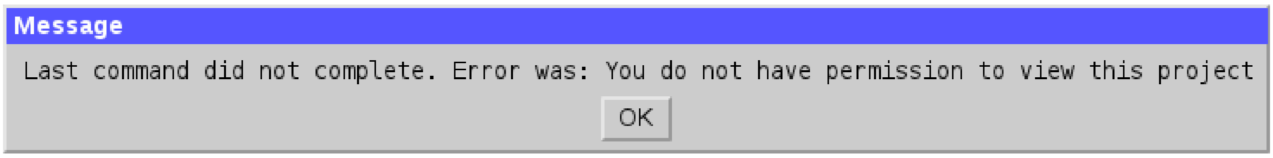
.. where “project name” is the name of your project, for example:

http://freud.itm.miun.se/pupil/flash/experiment.jsp?project=myproject

Outside MIUN, the base URL will be different.

10.2. Dealing with common errors

The by far most common error is that the student has not been granted permission to the project:



Message

Last command did not complete. Error was: You do not have permission to view this project

To fix this, grant the student permission as seen above.

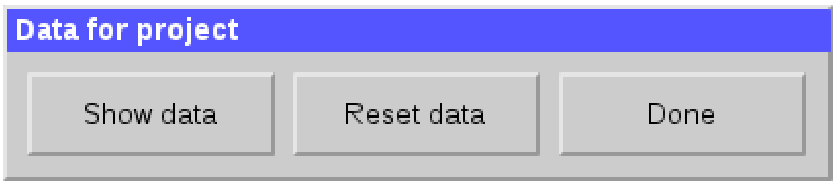
Other common errors normally occur because the author of the project has clicked “ensure unique images” in a scene setting or “splice category array” in the project setting, but there are not enough images withing a category to fulfill the request. In order to fix this, either uncheck the option or add enough material within the category.

11. Getting data

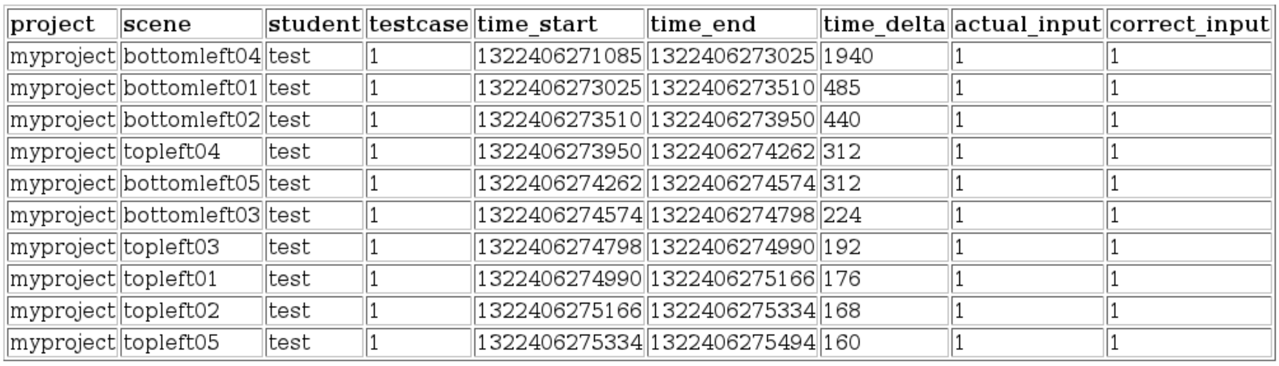
Once a set of students have run through the project, it is time to download the data. To access the data, click “data” in the project list (after having logged in as teacher).

11.1. Downloading all data

In the data window, click “show data”:



A new browser window with a table is displayed:



project	scene	student	testcase	time_start	time_end	time_delta	actual_input	correct_input
myproject	bottomleft04	test	1	1322406271085	1322406273025	1940	1	1
myproject	bottomleft01	test	1	1322406273025	1322406273510	485	1	1
myproject	bottomleft02	test	1	1322406273510	1322406273950	440	1	1
myproject	topleft04	test	1	1322406273950	1322406274262	312	1	1
myproject	bottomleft05	test	1	1322406274262	1322406274574	312	1	1
myproject	bottomleft03	test	1	1322406274574	1322406274798	224	1	1
myproject	topleft03	test	1	1322406274798	1322406274990	192	1	1
myproject	topleft01	test	1	1322406274990	1322406275166	176	1	1
myproject	topleft02	test	1	1322406275166	1322406275334	168	1	1
myproject	topleft05	test	1	1322406275334	1322406275494	160	1	1

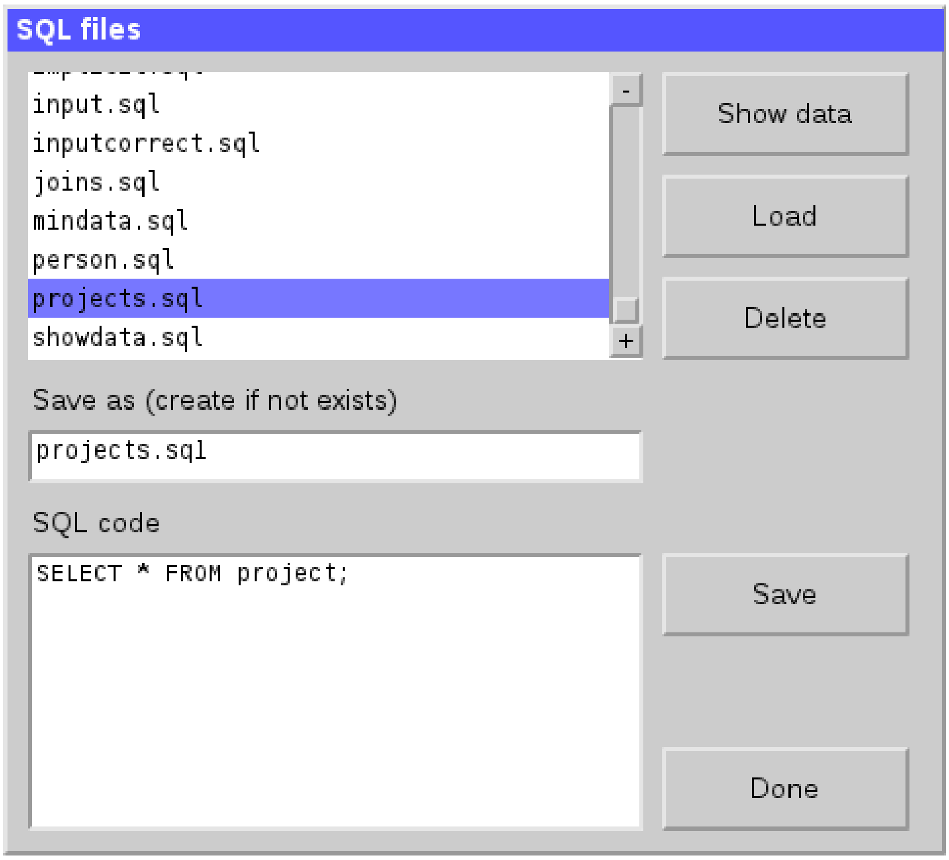
The columns in the table are:

- project: The name of the project for which you download data (this will always be the name of the project you selected).
- Scene: The name of scene for which the following timings were measured
- student: The name of the student causing the measurement
- testcase: The case number. The first student to run through the project is test case 1, the second test case 2. If the first student runs the project again after this, he would become test case 3.
- time_start: A “unix timestamp” for when the scene was shown. A unix timestamps is the number of milliseconds since january 1, 1970.
- time_end: A “unix timestamp” for when the respondent clicked a key responding to the scene.
- time_delta: How long the scene was visible
- actual_input: What exact key the student clicked. If he clicked “a”, there would be an “a” in the list.
- correct_input: If the input was correct. If “1” the input was correct, if “0” it was incorrect.

The easiest way to deal with the data is to simply mark everything in the browser window, select “copy” from the edit menu, open excel and select “paste” from the edit menu.

11.2. *Making custom queries*

As an alternative to using the default data download, it is possible to write custom SQL queries. To access this functionality click “SQL” from the project list window.



Save as (create if not exists)

projects.sql

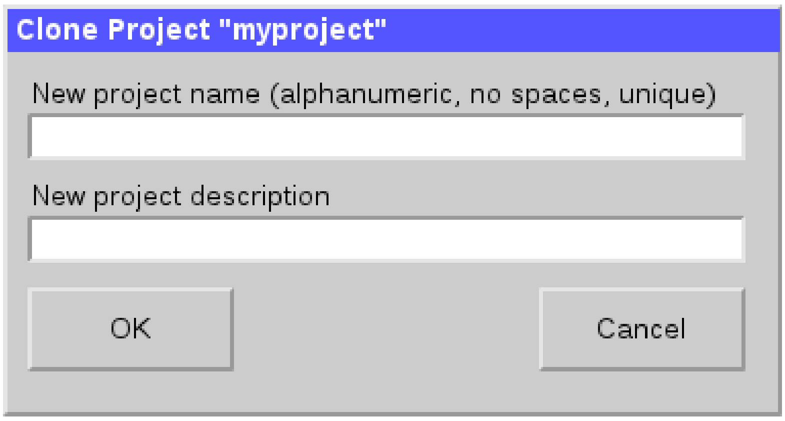
SQL code

```
SELECT * FROM project;
```

It is beyond this manual to explain how SQL works. It is recommended that the casual user stay away from this or ask a person knowledgable in database management for help.

12. Cloning a project

If a new project is very similar to an existing project, it can be beneficial to clone the existing project rather than starting from scratch. To clone a project, select it in the project list and click clone.

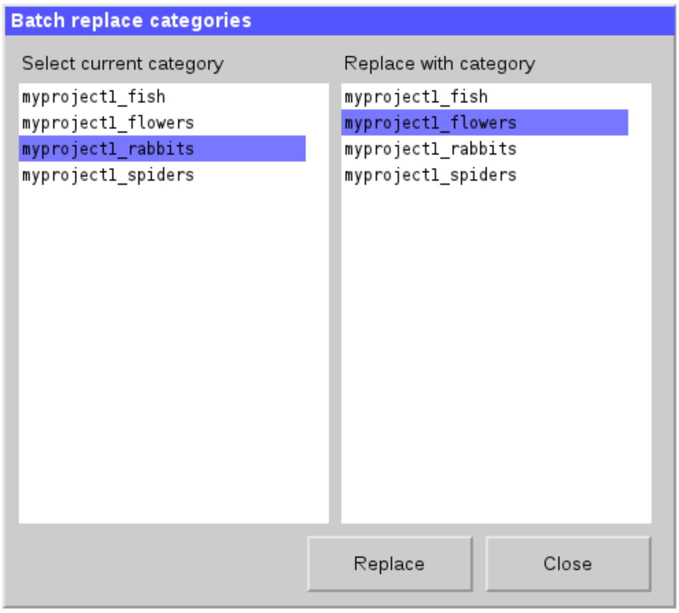

Clone Project "myproject"
New project name (alphanumeric, no spaces, unique)
New project description
OK
Cancel

Once this has been done, an exact copy of the previous project is available. This includes all stimulus material.

12.1. *Batch replacing material*

If you wish to replace stimulus material, upload it as usual using the image management. In the scene list there are two buttons under “batch replace material” for replacing existing material: image and category.

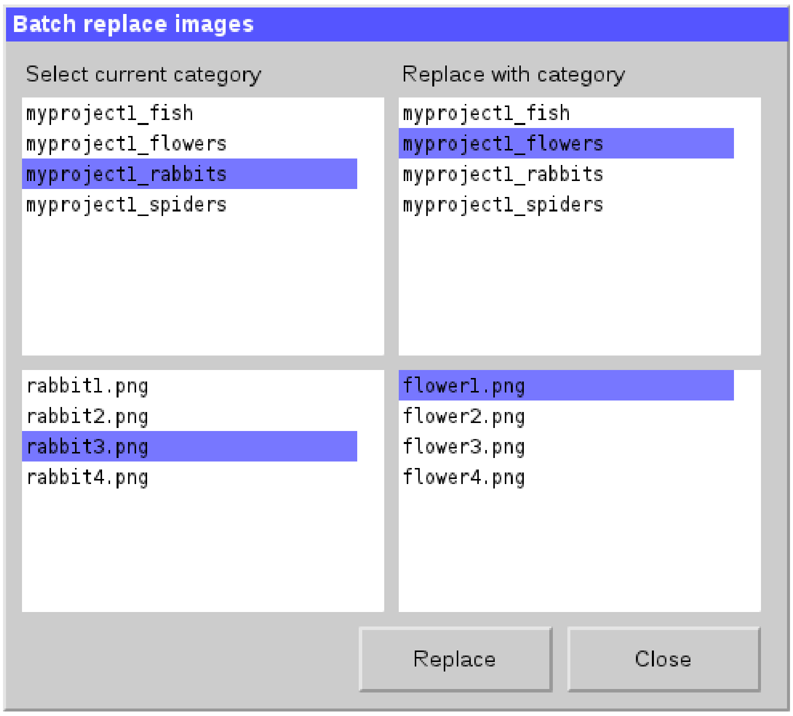
To replace categories, click the category button:



To replace all occurrences of the “rabbit” category with “flower” in SCRI and SORI scenes, click rabbit under current and flower under replace. Then click the “replace” button.

All scenes of SCRI and SORI now uses “flower” where they previously used “rabbit”. Repeat for each category you wish to repeat.

For SI and SOSI scenes, exact images need to be replace. To enter the image replacement, click image in the scene list window:



To replace “rabbit3” with “flower1”, click rabbit category and then the rabbit image in the left side of the window. Then click flower category and the flower image in the right part of the window. Then click “replace”. Repeat for all images you wish to replace.