PCP Manual Project

By Christian Forkstam

Content

1. Pipeline Code Project - Stimulus Presentation Lab (PCP-SPL)

Language and instruction manual for the Stimulus Presentation Lab part of the Pipeline Code Project.

1.1 License

PCP-SPL is developed by Christian Forkstam affiliated to:

- Cognitive Neurophysiology Group, Stockholm Brain Institute, Karolinska institute, Sweden.
- Max-Plank Institute for Psycholinguistics, the Netherlands.
- FC Donders Centre for Cognitive Neuroimaging, the Netherlands.
- University of Algarve, Portugal.

To contribute to the development of the code please indicate any problem thoroughly to simplify updates of the code. Indicate in the accompanying file 'Updates.log' any modifications of the code, your name and date, and send a zipped set of your experiment files to christian attsign forkstam dot se.

1.2 Installation

This code release for experiment presentation consists of control functions and scripts written for the commercial software Presentation (www.neurobs.com). The latest release of PCP-SPL can be downloaded from http://forkstam.se/PCP/. (An unsupported early version of the Presentation software can be found at http://forkstam.se/PCP/Presdl071092403.exe). Unzip and install.

1.3 Structure of the template scripts

Function names are coded in **small letters**. References to function names are coded in **CAPITALS**. Variables and presentation objects are coded in **ThisWay**.

The template set of experiment files include the following files:

• **Experiment files:**	$Experiment-Template / PCP-Experiment-3Key-Template.exp
	$Experiment-Template / PCP-Experiment-30Key-Template.exp
• **Scenario file:**	$Experiment-Template / Files / PCP-Scenario-Template.sce
• **Header file:**	$Experiment-Template / Files / PCP-Header-Template.pcl
• **Stimuli material file:**	$Experiment-Template / Files / PCP-Items-Template.pcl
• **Instruction file:**	$Experiment-Template / Files / PCP-Instruction-Template.txt

- **Questionnaire files:** $Experiment-Template / Files / PCP-Question1.txt

  $Experiment-Template / Files / PCP-Question2.txt

2. Experiment files

NB! The experiment .exp-files gets corrupted when transferred between different language specific OS (and sometimes also between different Presentation versions). Remember to check!

The experiment file response button setups are:

PCP-Experiment-3Key-Template.exp

- Button 1 = Left shift button
- Button 2 = Right shift button
- Button 3 = Space E.g. optional as key for experiment leader response

PCP-Experiment-30Key-Template.exp

- Button 1 = Left shift button
- Button 2 = Right shift button
- Buttons {3:28} = {A,B,...,Z}
- Button 29 = Space Defined as key to erase subject keyboard input in function GETKEYBOARDINPUT
- Button 30 = Enter Defined as key to finish subject keyboard input in function GETKEYBOARDINPUT

3. Scenario file

3.1 Scenario header

The Scenario file include a header and a Presentation objects definition part.

The general purpose with the header is to define experiment settings, such as file name of associated .pcl-Header file, response button codes, FMRI pulse information for FMRI experiment and FMRI testing, and sending output port code.

The objects definition part includes several Presentation object definitions for various object types:

PCP-Scenario-Template.pcl

```
################################################################
# HEADER
################################################################
scenario = "PCP-TemplateExperiment";                           # Name of scenario (for logfile use)
pcl_file = "PCP-Header-Template.pcl";                          # Name of accompanying pcl-file
                                                               #
active_buttons            = 3;                                 # Number of response keys: {1|2|3|30}
button_codes              = 1,2,9;                             # Code for response keys: {1|1,2|1,2,9|1:30}
#button_codes              = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30;
#target_button_codes       = 3,4,9;                            # Code for response with the CORRECT key {3|3,4|3,4,9|1:30}
                                                               #
$Black                    = "0,0,0";                           # Black colour
$White                    = "255,255,255";                     # White colour
$Red                      = "255,0,0";                         # Red colour
$SoundAttenuation         = 0.1;                               # Sound level attenuation
$NumSoundFiles            = 1;                                 # Number of sound files
$NumBitmapFiles           = 1;                                 # Number of bitmap files
                                                               #
default_monitor_sounds    = false;                             # Default setting to monitor sounds
                                                               #
default_font_size         = 30;                                # Default font size
default_font              = "Tahoma";                          # Default font
default_background_color = $White;                             # Default background colour font (default = white)
default_text_color        = $Black;                            # Default font colour (default = black)
screen_height             = 768;                               # Screen height (pixels)
screen_width              = 1024;                              # Screen width (pixels)
screen_bit_depth          = 16;                                # Colour depth (pixels)
                                                               #
scenario_type             = trials;                            # ---    {trials|fMRI_emulation|fMRI}
scan_period               = 2600;                              #|FMRI-  TR (ms)
pulses_per_scan           = 1;                                 #|mode   Number of pulses per scan
pulse_code                = 15;                                # ---    FMRI pulse code
                                                               #
default_output_port       = 1;                                 # Default output port
pulse_width               = 1;                                 # Pulse width (ms)
write_codes               = true;                              # Send codes through output port
                                                               #
begin;                                                         # End of header
```

Change associated Header file:

```
pcl_file = "PCP-Header-Template.pcl";
```

Change button settings:

for PCP-Experiment-30Key-Template.exp:

```
active_buttons           = 30;                                 # Use 30 active button
button_codes             = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30;# Response code {1:30}
```

Change sound level attenuation to 30% during sound presentation:

```
$SoundAttenuation = 0.3;
```

Change number of sound files to 10 files during sound presentation:

```
$NumSoundFiles = 10;
```

Change number of bitmap files to 10 files during bitmap presentation:

```
$NumBitmapFiles = 10;
```

Change to FMRI experiment presentation:

```
scenario_type = fMRI;
```

Change to test (e.i. emulate scanner pulse) FMRI experiment presentation:

```
scenario_type = fMRI_emulation;
```

Change to 10 pulses per scan during FMRI experiment presentation:

```
pulses_per_scan = 10;
```

Change to scanner pulse code 10 during FMRI experiment presentation:

```
pulse_code = 10;
```

Change to not send codes through parallel port:

```
write_codes = false;
```

3.2 Scenario objects definitions

PCP-Scenario-Template.pcl

```
################################################################
# OBJECTS DEFINITIONS
################################################################
```

Create trial object **Trial** with stimulus event object **StimulusEvent** and picture object **Picture**.

```
#---Trial, StimulusEvent, and Picture objects------------------# ---
picture { } default;                                           #|Picture object default
trial { stimulus_event { picture {                             #|Template trial Trial
    } Picture;                                                 #|Picture object Picture
  } StimulusEvent;                                             #|StimulusEvent object StimulusEvent
} Trial;                                                       #|Trial object Trial
```

Create text object array **Texts** with 15 text objects.

```
#---Text objects array-----------------------------------------# ---
array {                                                        #|Text objects
  LOOP $i 15;                                                  #|Loop over text objects
    $k = '$i + 1';                                             #|
      text { caption = " "; };                                 #|Text object
  ENDLOOP;                                                     #|End loop
} Texts;                                                       #|Texts array name
```

Create box object array **Boxs** with 5 box objects.

```
#---Box objects array------------------------------------------# ---
text { caption = " "; } Bar;                                   #|Text object Bar
array {                                                        #|Box objects
  LOOP $i 5;                                                   #|Loop over text objects
    $k = '$i + 1';                                             #|
      box { height = 1; width = 1; };                          #|Box object
  ENDLOOP;                                                     #|End loop
} Boxs;                                                        #|Boxs array name
```

Create bitmap object array **Bitmaps** with **$NumBitmapFiles** box objects.

```
#---Bitmap objects array---------------------------------------# ---
array {                                                        #|Picture bitmap(s)
  LOOP $i $NumBitmapFiles;                                     #|Loop over bitmapfile(s)
    $k = '$i + 1';                                             #|
    bitmap {                                                   #|
      preload = false;                                         #|Preload bitmap in memory
      filename = "P$k.bmp";                                    #|Bitmap file name
    };                                                         #|
  ENDLOOP;                                                     #|End loop
} Bitmaps;                                                     #|Bitmap array name
```

Create sound object array **Sounds** with **$NumSoundFiles** sound objects:

```
#---Sound objects array----------------------------------------# ---
array {                                                        #|Sound wavefile(s)
  LOOP $i $NumSoundFiles;                                      #|Loop over soundfile(s)
    $k = '$i + 1';                                             #|Update
    sound {                                                    #|
      wavefile {                                               #|
        preload = false;                                       #|Preload wavefile in memory
        filename = "S$k.wav";                                  #|Wavefile file name
      };                                                       #|
      attenuation = $SoundAttenuation;                         #|
    };                                                         #|
  ENDLOOP;                                                     #|End loop
} Sounds;                                                      #|Sound array name
```

Create trial object StopSoundTrial.

```
#---StopSoundTrial object--------------------------------------# ---
trial {                                                        #|Stop sound presentation trial
  monitor_sounds = true;                                       #|Monitor sounds
  trial_duration = 1;                                          #|Durate for 1 ms
} StopSoundTrial;                                              #|
```

Create trial object **StopVideoTrial**.

```
#---StopVideoTrial object--------------------------------------# ---
trial {                                                        #|Stop video presentation trial
  monitor_videos = true;                                       #|Monitor video
  trial_duration = 1;                                          #|Durate for 1 ms
} StopVideoTrial;                                              #|
```

4. Header file

The header file include several sub parts. Its general purpose is to define trial timing to present the stimulus. Each trial is composed of several subtrials periods, defined by the following main trial time bins:

TRIAL STRUCTURE: Sub-trial time periods within a trial.

Trial name	**Screen**	**Duration (ms)**	**Time (ms)**
BEGIN TRIAL			0
PRE-STIMULUS PERIOD	+	1000 ms	1000 ms
STIMULUS PERIOD (see below)	**XYZ**	3000 ms	4000 ms
POST-STIMULUS PERIOD	+	1000 ms	5000 ms
RESPONSE	**Response**	2000 ms	7000 ms
INTER-TRIAL PERIOD	+	4000 ms	11000 ms
NEXT TRIAL			

STIMULUS PERIOD STRUCTURE: Sub-stimulus periods within a stimulus.

Trial name	**Screen**	**Duration (ms)**	**Time (ms)**
PRE-STIMULUS PERIOD			0
Stimulus (1) period	**XYZ**	(1800 ms)	
- Substimulus 1	**X**	300 ms	300 ms
- Inter-substimulus period		300 ms	600 ms
- Substimulus 2	**Y**	300 ms	900 ms
- Inter-substimulus period		300 ms	1200 ms
- Substimulus 3	**Z**	300 ms	1500 ms
- Inter-substimulus period		300 ms	1800 ms
Inter-stimulus period	+	1000 ms	2800 ms
Stimulus (2) period	**YZX**	(1800 ms)	
- Sub-stimulus 1	**Y**	300 ms	3100 ms
- Inter-substimulus period		300 ms	3400 ms
...	...	...	...
POST-STIMULUS PERIOD			

4.1 Stimulus items

In PCP-Header-Template.pcl:

```
include "PCP-Items-Template.pcl";                       # Get trial items
```

In PCP-Items-Template.pcl:

The columns include specific information for each item to be presented:

1. **ITEM TYPE** is saved in the logfile and should be unique for each item.

   In addition, an @ sign followed by up to 5 triplets of digits will be interpreted by the RUNEXPERIMENT function (see below) to send these codes over the output port. **This is only true if both the Scenario header variable write_codes in PCP-Scenario-Template.sce is set as true and the global workspace variable SendCode in PCP-Header-Template.pcl is also set at true.**

2. **ITEM CODE** is the numeric code to be sent over the port for an external marker timing log (e.g. the EEG recording computer). It is divided in equally many triplets as sub-items (e.g. code 001 = port code 1, 010 = 10, and 100 = 100). For example, to present a sentence with 3 words while sending the codes **10**, **222**, and **13** is coded as **010222013**. An example of a code translation guide for trial items:

```
Code Translation Example Guide

Sentences:
A = Correct sentence                            First = 010, Target = 011, Other = 012, Last = 013
B = Sentence with local semantic violation      First = 020, Target = 021, Other = 022, Last = 023
C = Sentence with local syntactic violation     First = 030, Target = 031, Other = 032, Last = 033
D = Sentence with global semantic violations    First = 040, Target = 041, Other = 042, Last = 043
E = Sentence with global syntactic violation    First = 050, Target = 051, Other = 052, Last = 053
```

3. **CORRECT RESPONSE BUTTON** is the button code for correct response (e.g. key 1 corresponds to the first input device as defined in PCP experiment file

4. **NUMBER OF SUBITEMS** is the number of sub-items in complete item (e.g. words in sentence)

5. **SUBITEM1, SUBITEM2, …** is the sub-item 1, 2, ... in complete item (e.g. word 1, 2, ... in sentence), or as in the case of picture and sound presentation, the file number reference to the bitmap or sound file to be presented.

   The string array is filled with empty strings to fill up the array to have equal numbers of columns for each row.

   The up to 4 digits that follows an optional ¤ sign in the final part of a subitems will be used as local inter-stimulus duration. In addition, the up to 4 digits that follows an optional @ sign in the final part of a subitems (i.e., it will always have to be presented before the ¤ sign in a given subitem) will be used as local stimulus presentation duration. **This is only true if the global workspace variable VariableStimDur in PCP-Header-Template.pcl is set as true.**

Example of practice trial items in sentence presentation.

In PCP-Items-Template.pcl

```
# 1. ITEM TYPE  |  2. ITEM CODE  |  3. CORRECT RESPONSE BUTTON  |  4. NUMBER OF SUBITEMS  |  5. SUBITEM1  |  6. SUBITEM2  | ...
# PRACTICE CLASSIFICATION SENTENCES (KEY 1 = NOT ACCEPT; KEY 2 = ACCEPT)
array<string> PracticeItems[1][12] = {
  {"STARTERLEFT", "XXXXXXXXXXXXXXXXXXXXXXXX", "1", "8", "This", "is", "a", "template@200", "starter", "sentence,", "press", "button." }
};
```

Example of trial items in sentence presentation:

In PCP-Items-Template.pcl:

```
# 1. ITEM TYPE  |  2. ITEM CODE  |  3. CORRECT RESPONSE BUTTON  |  4. NUMBER OF SUBITEMS  |  5. SUBITEM1  |  6. SUBITEM2  | ...
# CLASSIFICATION SENTENCES (KEY 1 = NOT ACCEPT; KEY 2 = ACCEPT)
array<string> Items[2][12] = {
  { "T1001A" , "010012012012012012011XXX", "2", "8", "The", "girl" , "spread", "the", "bread@500", "with@", "butter.@"  , " @999¤999"},
  { "@100009", "040042042042042042041"   , "1", "7", "The", "robot", "spread", "the", "bread"    , "with" , "socks.¤999", " "        }
};
```

Example of trial items in sentence presentation in a 3D string array:

In PCP-Items-Template.pcl

```
# 1. ITEM TYPE  |  2. ITEM CODE  |  3. CORRECT RESPONSE BUTTON  |  4. NUMBER OF SUBITEMS  |  5. SUBITEM1  |  6. SUBITEM2  | ...
# CLASSIFICATION SENTENCES (KEY 1 = NOT ACCEPT; KEY 2 = ACCEPT)
array<string> ItemsBlock[2][12][2] = {

  { # Block1
    {"T1001A" ,"010012012012012012011XXX","2","8","The","girl" ,"spread","the","bread@500","with@","butter.@"  ," @999¤999"},
    {"@100009","040042042042042042041"   ,"1","7","The","robot","spread","the","bread"    ,"with" ,"socks.¤999"," "        }
  },

  { # Block2, Block3, ..., BlockN
    {"@200009","040042042042042042041"   ,"1","7","The","robot","spread","the","bread"    ,"with" ,"socks.¤999"," "        },
    {"T2001A" ,"010012012012012012011XXX","2","8","The","girl" ,"spread","the","bread@500","with@","butter.@"  ," @999¤999"}
  }

};
```

Example of trial items a bitmap presentation:

In PCP-Items-Template.pcl

```
# 1. ITEM TYPE  |  2. ITEM CODE  |  3. CORRECT RESPONSE BUTTON  |  4. NUMBER OF SUBITEMS  |  5. SUBITEM1  |  6. SUBITEM2  | ...
# BITMAPS
array<string> BitmapItems[2][8] = {
  { "Bananax4", "222222222222", "2", "4", "1", "1", "1", "1" },
  { "Bananax2", "221221"      , "1", "2", "1", "1", " ", " " }
};
```

Example of trial items in a sound presentation:

In PCP-Items-Template.pcl

```
# 1. ITEM TYPE  |  2. ITEM CODE  |  3. CORRECT RESPONSE BUTTON  |  4. NUMBER OF SUBITEMS  |  5. SUBITEM1  |  6. SUBITEM2  | ...
# SOUNDS
array<string> SoundItems[1][9] = {
  { "Sound", "XXXXXXXXXXXX", "1", "5", "111", "1", "1", "1", "111" }
};
```

In both the bitmap and sound trial items is the subitem interpreted as the file number of the bitmap or sound file, respectively. For example, “1” in the bitmap presentation mode (**Modality = "Bitmap”**) refers to the P1.bmp file in the stimuli folder while it refers to the S1.wav file if in sound presentation mode(**Modality = "Sound”**).

4.2 Global variables

Global variables to be used throughout the experiment.

Use template header for multiple experiment common setting, stored e.g. in PCP-Header-Template.pcl, and keep individual setting in secondary header files, e.g. PCP-Header-Experiment1.pcl, PCP-Header-Experiment2.pcl, ...

In PCP-Header-Template.pcl

```
#---GLOBAL WORKSPACE VARIABLES-----------------------------#
string LogFile              = "Logfile.res";          # Specific logfile file name
bool   SubStimulusStructure  = false;                 # Split each stimuli in its sub stimulus parts {true|false}
string Modality              = "Text";                # Presentation modality {"Text"|"Sound"|"Text+Sound"|"Bitmap"}
bool   SendCode              = true;                  # Send port code {true|false}. NB!! write_code must be true in Scenario!
bool   ContinuousSound       = false;                 # Play sound continuously in Text+Sound Modality {true|false}
bool   VariableStimDur       = true;                  # Variable stimulus presentation duration {true|false}
string RandomizeTrialOrder   = "No";                  # Randomized trial order {"All"|"Block"|"No"}
bool   FixationBars          = false;                 # Show centred fixation bars during presentation {true|false}
bool   AddPoint              = false;                 # Add point after last sentence item {true|false}
int    NumItemsPres          = 1;                     # Number of presentations of each test item
int    NumItemsBlocks        = 1;                     # Number of presentation blocks (evenly dividable with full trial set)
bool   AwaitResponse         = false;                 # Forced await response {true|false}
string ResponseType          = "StimFirstResponse";   # {"FixedDuration"|"FirstResponse"|"CorrectResponse"|"StimFirstResponse"|
                                                      #  "StimCorrectResponse"|"Retype"|"Selection"|"MultipleChoice"|"Completion"}
int    SelfPaced             = 0;                     # Await subject response (NextTrialButton) OR do inter-trial period (0)
bool   FeedbackTrialProgress = true;                  # Show progress feedback during self paced pause {true|false}
bool   DoQuestionnaire       = true;                  # Perform online questionnaire {true|false}
bool   Debug                 = false;                 # Print debug information to terminal command prompt {true|false}
```

In PCP-Header-Experiment1.pcl

```
...
include "PCP-Header-Template.pcl";                        # Include Header file
Modality = "Bitmap";                                      # Switch to bitmap item presentation
...
```

Explanation of the global variables:

• **LogFile**	Specific logfile file name (e.g. "Logfile.res"). This is an additional log file to the ordinary Presentation log file.
• **SubStimulusStructure**	Split each subitem in a trial in its sub stimulus parts.
= true	Present each stimulus character one-by-one, i.e. treat each sub-item in the trial items array as a stimulus complex consisting of a sequence of sub-stimulus.
= false	Present each stimulus as a whole, i.e. treat each sub-item in the trial items array as a complete stimulus.
• **Modality:**	Presentation modality.
= "Text"	Present the trial text specified in the trial items array.
= "Sound"	Present the sound file as specified in the trial item array. The sound must be part of the sound alphabet (see **SoundAlphabet**).
= "Text+Sound"	Same as **"Text"** but will also present a sound file specified in the sound alphabet (see **SoundAlphabet** and **ContinuousSound**).
= "Bitmap"	Present the bitmap picture as specified with a file number in the trial items array.
• **SendCode**	Send port code over output port. **NB! write_code must be set as true in the Scenario file!**

	= true	Translate and send the sub-item code over the output port
	= false	Don't send port code**.**
•	**ContinuousSound**	Play continuous sound over subsequent trials (see **SoundAlphabet**). Only functional for **Modality = "Text+Sound"**.
	= true	Play continuous sound that last over the complete sound file length, and over subsequent files depending on the length of the sound file. The sound played is picked from the first entry in the sound alphabet (see **SoundAlphabet**).
	= false	Play sound that only last over the trial independent of the length of the sound file. The sounds played are picked from the sound alphabet in a randomized order (see **SoundAlphabet**).
•	**VariableStimDur**	Use variable stimulus presentation duration as defined in **VarStimDur** (see **Stimulus Period**).

In addition will it activate a sub-item timing specificity for the sub-items in the trial items array. For example, the trial item sub-stimulus "ShowMe@5000" with **Modality ="Text"** and **SubStimulusStructure = false** would present the text ShowMe for 5 s. With **SubStimulusStructure = true** this will instead set a variable stimulus duration for each sub-stimulus (e.g. "ShowMe@5000" would be presented letter by letter for 5 s each letter). Similarly, will the trial item sub-stimulus "ShowMe¤5000" introduce a 5 s inter-stimulus duration. As a final example, will the trial item sub-stimulus "ShowMe@30¤1000" present the text ShowMe for 30 ms followed by a 1 s inter-stimulus duration.

	= true	Use variable stimulus presentation duration.
	= false	Do not use variable stimulus presentation duration.
•	**RandomizeTrialOrder**	Randomized trial order.
	= "All"	Randomize trials over complete trial items array.
	= "Block"	Randomize trials block-wise over the trial items array.
	= "No"	No randomization.
•	**FixationBars**	Centred fixation bars during presentation (see **Fixation Bar Parameters**).
	= true	Show fixation bars.
	= false	Do not show fixation bars.
•	**AddPoint**	Add point after last sentence item.
	= true	Add point.
	= false	Do no add point.
•	**NumItemsPres**	Number of presentations of trial items in trial items array.
	= 1	Present each trial in the trial items array ones.

	= 2, ..., N	Present each trial in the trial items array 2, ..., N times.
•	**NumItemsBlocks**	Number of presentation blocks of the trial items array. Must be evenly dividable with full trial set, i.e. number of trial items * number of presentation blocks (**NumItemPres**).
	= 1	Present trials from the trial items array in one block.
	= 2, ..., N	Present trials from the trial items array in 2, ..., N blocks.
•	**AwaitResponse**	Forced await response during the response period before continuing with next trial.
	= true	Await response.
	= false	Continue independent of response.
•	**ResponseType**	Response key or response type.
	= "FixedDuration"	Keep fixed duration independent of response.
	= "FirstResponse"	Wait for first response.
	= "CorrectResponse"	Wait for correct response.
	= "StimFirstResponse"	Wait for first response already during the stimulus presentation period.
	= "StimCorrectResponse"	Wait for correct response already during the stimulus presentation period.
	= "Retype"	Get keyboard input (works with the button keys ordered as {left, right, A-Z, space, enter}, see

	PCP-Experiment-30Keys-Template.exp.
= "Selection"	Get mouse selection input. NB! The mouse button should be one of the response buttons.
= "MultipleChoice"	Get mouse selection input from multiple choices. NB1! The mouse button should be one of the response buttons. NB2! The list of choices are to be inserted in the trial item array with number of choices on column 6 and the choices themselves on position 7 and on.
= "Completion"	Get keyboard input while presenting stimulus.
• **SelfPaced**	Self paced subject response.
= 0	Do not wait for button press but instead execute inter-trial interval.
= 1,2,...	Wait for button press (1,2,...) before continue to the next trial.
• **FeedbackTrialProgress**	Show trial number progress feedback during self paced pause. Only functional for **SelfPaced = 0**.
= true	Show trial number progress feedback.
= false	Do not show feedback.
• **DoQuestionnaire**	Perform online questionnaire from the files specified in **QuestionnaireFileName**.
= true	Perform online questionnaire.
= false	Do not perform questionnaire.

• **Debug**	Print debug information to terminal command prompt.
= true	Show debug information.
= false	Do not show debug information.

4.3 Instructions

Update instructions to be presented throughout the experiment, e.g. translate into non-english language.

PCP-Header-Template.pcl

```
#---INSTRUCTIONS-----------------------------------------------#
array<string> Instrs[13] = {                                   # Instrs
  "Press a button for sequence "                             , #     1: Pause text 1
  "\nwhen you are ready to continue."                        , #     2: Pause text 2
  "Pause...\nPress a button to continue."                    , #     3: Pause text 3
  "Press left to continue."                                  , #     4: Left to continue
  "Press right to accept."                                   , #     5: Right to accept
  "Start experiment!"                                        , #     6: Start experiment!
  "Retype the sequence."                                     , #     7: Response instruction
  "End of block.\n\n\nPress a button to continue."           , #     8: End of block
  "End of experiment."                                       , #     9: End of experiment
  "Different?                              Same?"            , #    10: Different? Same?
  "Not pleasent?                       Pleasent?"            , #    11: Not pleasent? Pleasent?
  "Non-grammatical?                 Grammatical?"            , #    12: Non-grammatical? Grammatical?
  "Incorrect?                           Correct?"            , #    13: Incorrect? Correct?
};                                                             #
```

4.4 Stimulus timing

As described earlier, the main trial period is divided into several trial sub-periods. Information on sub-trial period lengths as well as information on actual screen presentations are defined for each period.

```
#---TIMING PERIOD SETTINGS-------------------------------------#
#DURATION|JITTER| MARKER|FONTSIZE|XPOS| YPOS|EVENTCODE|PORTCODE#
array<string> Timings[11][8] = {                               #
  {"1000",  "NA",    "+",    "20","NA",  "0",    "Pre", "241"},#  [1] PRE-STIMULUS
  { "300",  "NA",    " ",    "30","NA",  "0",     "NA",  "NA"},#  [2] STIMULUS
  { "300",  "NA",   "NA",    "NA","NA", "NA",     "NA",  "NA"},#  [3] INTER-SUBSTIMULUS
  { "300",  "NA",    " ",    "20","NA",  "0",    "ISI", "242"},#  [4] INTER-STIMULUS
  {"1000", "500",    " ",    "20","NA",  "0",   "Post", "243"},#  [5] POST-STIMULUS
  {"2000",  "NA",    "+",    "20","NA",  "0",   "Resp", "244"},#  [6] RESPONSE
  {  "NA",  "NA",    " ",    "20","NA","150",     "NA",  "NA"},#  [7] RESPONSE INSTRUCTION
  {"1000", "250","* * *",    "20","NA", "-5",    "ITI", "245"},#  [8] INTER-TRIAL
  {"1000",  "NA",    " ",    "20","NA","150",  "Pause", "246"},#  [9] PAUSE
  {"3000",  "NA",    " ",    "20","NA",  "0",    "End", "248"} # [10] END-OF-EXPERIMENT
};                                                             #
Timings[7][3]  = Instrs[7];                                    # INTRUCTION TEXT : RESPONSE INSTRUCTION
Timings[9][3]  = Instrs[3];                                    # INTRUCTION TEXT : PAUSE
Timings[10][3] = Instrs[9];                                    # INTRUCTION TEXT : END-OF-EXPERIMENT
# !NB. Sound stimuli longer than Stimulus + InterStimulus duration might be clipped!
```

(Array position values NA is not yet defined in any PCP functions.)

- **[1] PRE-STIMULUS** Pre-stimulus period before stimulus presentation.

  Present a fixation cross in font size 20 for 1000 ms in centre position, write Pre in log file, and send value 241 over port.

- **[2] STIMULUS** Stimulus/Sub-stimulus presentation period (presentation period for each stimuli/sub-stimuli).

  Present stimulus for 300 ms in centre position (see e.g. trial items array Items for stimulus value, event code value, and port code value).

- **[3] INTER-SUBSTIMULUS** Inter-sub-stimulus period (period in-between each sub-stimulus).

  Present nothing for 300 ms in-between sub-stimulus (no log written to file nor sent over port).

- **[4] INTER-STIMULUS** Inter-stimulus period (period in-between each trial subitem stimulus).

  Present nothing in font size 20 for 300 ms in centre position, write ISI in log file, and send value 242 over port.

- **[5] POST-STIMULUS** Post-stimulus period with optional time jitter.

  Present nothing for 1000 ± 500 ms after last stimulus presentation, write Post in log file, and send value 243 over port.

  PostStimDur – PostStimJitter < **Post-stimulus period** < PostStimDur + PostStimJitter

- **[6] RESPONSE** Response period with possibility to add instructions for response.

  Present a fixation cross for 2000 ms in font size 20 in centre position, write Resp in log file, and send value 244 over port.

- **[7] RESPONSE INSTRUCTION**

  Present during the response period the response instruction in Instrs[7] in font size 20 in position (0,150) (no additional log written to file nor sent over port).

- **[8] INTER-TRIAL** Inter-trial period with optional time jitter.

  Present three stars for 1000 ± 250 ms in font size 20 in position (0,-5), write ITI in log file, and send value 245 over port.

  InterTrialDur – InterTrialJitter < **Inter-trial period** < InterTrialDur + InterTrialJitter

- **[9] PAUSE**

  Present the pause instruction in Instrs[3] for 1000 ms in font size 20 in position (0,150), write Pause in log file, and send value 246 over port.

- **[10] END-OF-EXPERIMENT**

  Present the end-of-experiment instruction in Instrs[9] for 3000 ms in font size 20 in centre position, write End in log file, and send value 248 over port.

4.5 Colour and font setting

Option to use variable background and text colour, as well as text fonts.

```
array<int> BackgroundColor[3]= { 255 , 255 , 255 };            # Default background colour
array<int> TextColor[3]      = { 0 , 0 , 0 };                  # Default colour for text objects
string DefaultFont           = "Tahoma";                       # Default font for text object
string StimFont              = "Tahoma";                       # Font for stimulus text object
string TextFont              = DefaultFont;                    # Holder for text object font
```

4.6 Variable stimulus timings

Variable stimulus timings are in use when **VariableStimDur = true** (see above).

```
# For Variable Stimulus Duration (VariableStimDur) == true     #
array<string> VarStimDur[7][2] = {                             # Variable stimulus duration times (ms)
     {"Base"     ,"187"},                                      # Base stimulus duration time (ms)
     {"Target"   ,"350"},                                      # Target letter(s) stimulus duration time (ms)
     {"PerLetter","27" },                                      # Per letter stimulus duration time (ms)
     {"MaxLength","10" },                                      # Max word length (letters)
     {"Max"      ,"450"},                                      # Max word length stimulus duration time (ms)
     {"PerComma" ,"200"},                                      # Per comma stimulus duration time (ms)
     {"PerPoint" ,"293"}};                                     # Per point stimulus duration time (ms)
```

4.7 Questionnaire parameters

Questionnaire is run when **DoQuestionnaire = true**.

```
#---QUESTIONNAIRE PARAMETERS-----------------------------------#
int    QuestionDelay         = 1000;                           # Delay time (ms) for requested questionnaire response
array<string> QuestionnaireFileName[2];                        # Text array QuestionnaireFileName of [number of files] length
QuestionnaireFileName[1]       = "PCP-Question1.txt";          # Questionnaire file name
QuestionnaireFileName[2]       = "PCP-Question2.txt";          # Questionnaire file name
```

4.8 Bitmap parameters

```
#---BITMAP PARAMETERS------------------------------------------#
bool   PreLoadedBitmaps      = false;                          # Bitmap objects preloaded in scenario {true|false}
int    NumBitmapFiles        = 0;                              # Number of bitmap files
```

- **PreLoadedBitmaps** Bitmap objects preloaded in scenario.

= true	Bitmap objects are preloaded in scenario.
= false	Bitmap objects are not preloaded in scenario.

4.9 Sound parameters

```
#---SOUND PARAMETERS--------------------------------------------#
int    SoundAlphabetSize     = 5;                               # Size of sound alphabet (= total number of wavefiles)
array<string> SoundAlphabet[SoundAlphabetSize] =                # Sound alphabet
                            {"M","S","V","R","X"};              #
```

4.10 Selection response parameters

For **ResponseType = "Selection"**

```
#---SELECTION RESPONSE PARAMETERS----------------------------#
int    SelectionFontSize         = 50;                      # Selection response font size
int    SelectionJitter           = 20;                      # Add position jitter to selection response positions (== 0 for no jitter)
bool   SelectionActiveCentre     = true;                    # Use centre position to mark skipped selection reponses {true|false}
bool   SelectionResponseFeedback = false;                   # Show selection response feedback {true|false}
bool   SelectionPerformanceFeedback = false;                # Show performance feedback (hit/miss = green/red background) {true|false}
int    SelectionHalfWidth        = 25;                      # Selection response box halfwidth
int    SelectionNumPositions     = 10;                      # Number of selection response positions
array<int> SelectionPositions[SelectionNumPositions][2] = {  # Selection response positions ([x y]-coordinates)
  {    0,  200 },                                           #   POS1  {X, Y}
  {  118,  162 },                                           #   POS2  {X, Y}
  {  190,   62 },                                           #   POS3  {X, Y}
  {  190,  -62 },                                           #   POS4  {X, Y}
  {  118, -162 },                                           #   POS5  {X, Y}
  {    0, -200 },                                           #   POS6  {X, Y}
  { -118, -162 },                                           #   POS7  {X, Y}
  { -190,  -62 },                                           #   POS8  {X, Y}
  { -190,   62 },                                           #   POS9  {X, Y}
  { -118,  162 }                                            #   POS10 {X, Y}
};                                                          #
array<string> SelectionAlphabet[SelectionNumPositions] = {   # Selection response alphabet (same size as SelectionNumPositions)
  "0","1","2","3","4","5","6","7","8","9"};                 #
```

4.11 Keyboard input response parameters

```
#---KEYBOARD INPUT RESPONSE PARAMETERS-------------------------#
array<int> ButtonCodes[4] = {1,2,29,30};                      # Button codes used in the template experiments {Left,Right,Erase,Enter}
```

4.12 Fixation bar parameters

```
#---FIXATION BAR PARAMETERS------------------------------------#
int    BarUpPos              = 95;                              # Text object BarUp position on Y-axis
int    BarDownPos            = -15;                             # Text object BarDown position on Y-axis
if                                                              # Set fixation bar
  Modality == "Text" &&                                         #
  FixationBars == true                                          #
then                                                            # Trim fixation bars for visual presentations
  string FixationBar = "__";                                    #
  if                                                            #
    SubStimulusStructure == false                               #
  then                                                          #
    FixationBar = "__________";                                 #
  end;                                                          #
  Bar.set_caption(FixationBar);                                 #
  Bar.set_font_size(60);                                        #
  Bar.redraw();                                                 #
  default.add_part( Bar , 0 , BarUpPos   );                     # Add fixation bars to picture object default
  default.add_part( Bar , 0 , BarDownPos );                     #
  Picture.add_part( Bar , 0 , BarUpPos   );                     # Add fixation bars to picture object Picture
  Picture.add_part( Bar , 0 , BarDownPos );                     #
end;                                                            #
```

5. Function library

The library of functions is found in $Experiment-Template/Lib/ and is loaded in the PCP-Header-Template.pcl. See the appendix for detailed description and syntax for the implemented functions.

```
################################################################
# SET LIBRARY
################################################################
include "../Lib/gettext.pcl";                                  #
include "../Lib/getpicture.pcl";                               #
include "../Lib/gettrial.pcl";                                 #
                                                               #
include "../Lib/trialorderrandomization.pcl";                  #
include "../Lib/getsubarray.pcl";                              #
                                                               #
include "../Lib/readfile.pcl";                                 #
include "../Lib/showfile.pcl";                                 #
include "../Lib/showerror.pcl";                                #
include "../Lib/logdata.pcl";                                  #
include "../Lib/testitemsarray.pcl";                           #
include "../Lib/pre4string2int.pcl";                           #
                                                               #
include "../Lib/setresponse.pcl";                              #
                                                               #
include "../Lib/dummyfmri.pcl";                                #
                                                               #
include "../Lib/getkeyboardinput.pcl";                         #
                                                               #
include "../Lib/resetselection.pcl";                           #
include "../Lib/getselection.pcl";                             #
include "../Lib/getmouseinput.pcl";                            #
                                                               #
include "../Lib/loadbitmap.pcl";                               #
include "../Lib/loadsound.pcl";                                #
                                                               #
include "../Lib/presentdefault.pcl";                           #
include "../Lib/presenttrial.pcl";                             #
include "../Lib/presentsound.pcl";                             #
include "../Lib/presentbitmap.pcl";                            #
include "../Lib/presentitem.pcl";                              #
                                                               #
include "../Lib/presentquestion.pcl";                          #
include "../Lib/runquestionnaire.pcl";                         #
                                                               #
include "../Lib/runexperiment.pcl";                            #
```

6. Experiment Lego

Experiment building block examples:

6.1 SAVE TEXT TO LOG FILE

```
#---SAVE TEXT TO LOG FILE--------------------------------------#
InputData = "Log text\n";                                      #
FileName  = "Log.res";                                         #
logdata (InputData,FileName,true);                             # OVERWRITE FILE CONTENT
logdata (InputData,FileName,false);                            # APPEND TEXT TO FILE
```

6.2 TRIAL ORDER RANDOMIZATION

```
#---TRIAL ORDER RANDOMIZATION----------------------------------#
int NumBlock   = 4;                                            # Number of blocks in experiment
int NumMax     = 30;                                           # Maximum number of trials
array<int> Out = trialorderrandomization (NumBlock,NumMax);    #
```

6.3 TRIAL ORDER RANDOMIZATION

```
#---GET SUBARRAY-----------------------------------------------#
array<string> In[2][3] = {{"1","2","3"},{"4","5","6"}};        # Full array
int Row1 = 1;                                                  # Sub array row start index
int Row2 = 2;                                                  # Sub array row end index
int Col1 = 2;                                                  # Sub array col start index
int Col2 = 3;                                                  # Sub array col end index
array<string> Out[2][2] = getsubarray(Row1,Row2,Col1,Col2);    #
```

6.4 PRESENT INSTRUCTIONS

```
#---PRESENT INSTRUCTIONS----------------------------------------#
string FileName = "PCP-Instruction-Template.txt";               #
int FontSize    = 15;                                           #
showfile (FileName,FontSize);                                   #
```

6.5 SYNCHRONIZE WITH FMRI SCANNER

```
#---SYNCHRONIZE WITH FMRI SCANNER - WAIT FOR 10 PULSES---------#
int NumPulses = 10;                                             #
dummyfmri (NumPulses);                                          #
```

6.6 GET INPUT FROM KEYBOARD

```
#---GET INPUT FROM KEYBOARD------------------------------------#
#                                                               #
# NB! Requires 30 response buttons                              #
#     in Experiment Input Devices = {LEFT,RIGHT,A:Z,ERASE,STOP}#
#     in Scenario file                                          #
#        active_buttons = 30;                                   #
#        button_codes = 1,2,3,...,28,29,30;                     #
#                                                               #
string Instr      = "Retype text";                              #
int InstrFontSize = 30;                                         #
int InstrPos      = 150;                                        #
int StimFontSize  = 60;                                         #
int StimPos       = 0;                                          #
array<string> ButtonCodes = {29,30};                            #
getkeyboardinput (Instr,InstrFontSize,InstrPos,                 #
                  StimFontSize,StimPos,ButtonCodes);            #
```

6.7 LOAD/UNLOAD BITMAP FILES

```
#---LOAD/UNLOAD BITMAP FILES-----------------------------------#
loadbitmap(true,"All");                                        # LOAD ALL BITMAP FILES
                                                               #
int trialnum = 1;                                              # Set trial number in Items trial array
loop                                                           # UNLOAD BLOCK OF BITMAP FILES
  int Pos = 1                                                  #
until                                                          # LOOP OVER STIMULI IN TRIAL ARRAY ITEM 1
  Pos > Items[trialnum][4]                                     #
begin                                                          #
  loadbitmap(false,Items[trialnum][4 + Pos]);                  #
  Pos = Pos + 1;                                               #
end;                                                           #
```

6.8 LOAD/UNLOAD SOUND FILES

```
#---LOAD/UNLOAD SOUND FILES------------------------------------#
bool          Load  = true;                                    # Load/unload sound file
string        Sound = "M";                                     # Sound in SoundAlphabet
array<string> SoundAlphabet[5] = {"M","S","V","R","X"};        # Sound alphabet (ordered as in sound object Sounds)
loadsound (Load,Sound,SoundAlphabet);                          #
```

6.9 SET RESPONSE

```
#---SET RESPONSE-----------------------------------------------#
Key       = 3;                                                 # Set button 3 as target buttom key
StartTime = 0;                                                 # Start time (ms)
StopTime  = 0;                                                 # No stop time
setresponse (Key,StartTime,StopTime)                           #
```

6.10 PRESENT TRIAL

6.10.1 FIXED DURATION

```
#---PRESENT TRIAL - FIXED DURATION-----------------------------#
Duration     = 1000;                                           # Set duration to 1000 ms
ResponseType = "FixedDuration";                                # Collect but ignore responses
TextStr      = "FixedDuration";                                # Set text string
FontSize     = 20;                                             # Set font size to 20
yPosition    = 150;                                            # Set object position y-coordinate to 150
EventCode    = "Log";                                          # Set event log code to "Log"
PortCode     = "20";                                           # Set port code to 20
SendCode     = true                                            # Send code
presenttrial (Duration,ResponseType,TextStr,FontSize,          #
              yPosition,EventCode,PortCode,SendCode)           # PRESENT TRIAL WITH FIXED DURATION
```

6.10.2 AWAIT ANY RESPONSE

```
#---PRESENT TRIAL - AWAIT ANY RESPONSE-------------------------#
Duration     = 1000;                                           # Set duration to 1000 ms
ResponseType = "FirstResponse";                                # Collect but ignore responses
TextStr      = "FirstResponse";                                # Set text string
FontSize     = 20;                                             # Set font size to 20
yPosition    = 150;                                            # Set object position y-coordinate to 150
EventCode    = "Log";                                          # Set event log code to "Log"
PortCode     = "20";                                           # Set port code to 20
SendCode     = true                                            # Send code
presenttrial (Duration,ResponseType,TextStr,FontSize,          #
              yPosition,EventCode,PortCode,SendCode)           # PRESENT TRIAL AND AWAIT ANY RESPONSE
```

6.10.3 AWAIT CORRECT RESPONSE

```
#---PRESENT TRIAL - AWAIT CORRECT RESPONSE---------------------#
Duration     = 1000;                                           # Set duration to 1000 ms
ResponseType = "CorrectResponse";                              # Collect but ignore responses
TextStr      = "CorrectResponse";                              # Set text string
FontSize     = 20;                                             # Set font size to 20
yPosition    = 150;                                            # Set object position y-coordinate to 150
EventCode    = "Log";                                          # Set event log code to "Log"
PortCode     = "20";                                           # Set port code to 20
SendCode     = true                                            # Send code
setresponse(3,0,0);                                            # Set target button 3 as correct response (SEE SETRESPONSE LEGO PART)
presenttrial (Duration,ResponseType,TextStr,FontSize,          #
              yPosition,EventCode,PortCode,SendCode)           # PRESENT TRIAL AND AWAIT CORRECT RESPONSE
```

6.10.4 END OF EXPERIMENT

```
#---END OF EXPERIMENT------------------------------------------#
Duration     = 10000;                                          # Set duration to 10000 ms
ResponseType = "FixedDuration";                                # Collect but ignore responses
TextStr      = Instrs[9];                                      # Set text string
FontSize     = 20;                                             # Set font size to 20
yPosition    = 0;                                              # Set object position y-coordinate to centre position
EventCode    = "End";                                          # Set event log code to "Log"
PortCode     = "200";                                          # Set port code to 200
SendCode     = true                                            # Send code
presenttrial (Duration,ResponseType,TextStr,FontSize,          #
              yPosition,EventCode,PortCode,SendCode)           #
```

6.11 RUN QUESTIONNAIRE

```
#---RUN QUESTIONNAIRE------------------------------------------#
array<string> FileNames[3]   = {"Q1.txt","Q2.txt","Q3.txt"};   # File name string array
int           Delay          = 1000;                           # Delay response time (ms)
array<int>    ButtonCodes[2] = {1,2};                          # Button codes {Left,Right}
runquestionnaire (FileNames,Delay,ButtonCodes);                #
```

6.12 PRESENT BITMAP FILE

```
#---PRESENT BITMAP FILE----------------------------------------#
int    Bitmap          = 1;                                    # Bitmaps bitmap object array position
int    Duration        = 1000;                                 # Duration (ms)
bool   PreLoadedBitmaps = false;                               # Preloaded bitmap objects {true|false}
string EventCode       = "TestBitmapPresentation";             # Event log code
string PortCode        = "000";                                # Port code
bool   SendCode        = false;                                # Send port code {true|false}
presentbitmap(Bitmap,Duration,PreLoadedBitmaps,                #
              EventCode,PortCode,SendCode)                     #
```

6.13 PRESENT SOUND FILE

```
#---PRESENT SOUND FILE-----------------------------------------#
string        Sound       = "M";                               # Sound item
array<string> Alphabet[5] = {"M","S","V","R","X"};             # Sound alphabet (ordered as in sound object Sounds)
int           Duration    = 1000;                              # Duration (ms)
string        EventCode   = "LogThis";                         # Event log code
string        PortCode    = "000";                             # Port code
bool          SendCode    = false;                             # Send code {true|false}
presentsound (Sound,Alphabet,Duration,                         #
              EventCode,PortCode,SendCode);                    #
```

6.14 PRESENT ITEM FILE

```
#---PRESENT ITEM FILE------------------------------------------#
int           StimDur               = 300;                     # Stimulus/Substimulus duration (ms)
int           InterStimDur          = 300;                     # Inter-substimulus duration (ms)
int           InterSubItemDur       = 300;                     # Inter-Stimulus duration (ms)
bool          VariableStimDur       = true;                    # Variable stimulus duration {true|false}
array<string> VarStimDur[7][2]      = VarStimDur;              # Variable stimulus times (ms)
array<string> Str[Items[1].count()] = Items[1];                # Trial items array
int           yPosition             = StimPos;                 # Stimulus y-coordinate position
int           FontSize              = StimFontSize;            # Font size
string        Modality              = "Bitmap";                # Modality mode
bool          SubStimulusStructure  = true;                    # Split each stimuli in substimulus {true|false}
bool          AddPoint              = AddPoint;                # Add point after last substimuli {true|false}
bool          PreLoadedBitmaps      = PreLoadedBitmaps;        # Preloaded bitmap objects {true|false}
array<string> Alphabet[5]           = SoundAlphabet;           # Sound alphabet
string        LogFile               = LogFile;                 # Logfile file name
bool          SendCode              = SendCode;                # Send code {true|false}
presentitem(StimDur,InterStimDur,InterSubItemDur,              #
  VariableStimDur,VarStimDur,Str,yPosition,FontSize,           #
  Modality,SubStimulusStructure,AddPoint,PreLoadedBitmaps,     #
  Alphabet,LogFile,SendCode);                                  #
```

6.15 GET MOUSE SELECTION RESPONSE

```
#---GET MOUSE SELECTION RESPONSE-------------------------------#
#     NB!                                                      #
#     THIS FUNCTION USE THE FIRST RESPONSE BUTTON              #
#     EVEN IF THIS WOULD NOT BE THE MOUSE BUTTON!              #
int           NumResponsesTmp = 5;                             # Number of mouse click to collect
int           NumPositionsTmp = 10;                            # Number of selection positions
int           JitterTmp       = 20;                            # Selection position jitter
int           HalfwidthTmp    = 25;                            # Halfwidth of selection area
bool          ActiveCentreTmp = true;                          # Active centre position to jump reponse
string        CentreTextTmp   = "?";                           # Centre text
array<string> AlphabetTmp[NumPositionsTmp] = {                 # Selection alphabet
  "0","1","2","3","4","5","6","7","8","9"};                    #
bool          FeedbackResponseTmp = false;                     # Response feedback
array<int>    PositionsTmp[NumPositionsTmp][2] = {              # Selection positions x,y-coordinates
  {   0, 200 },{ 118, 162 },{ 190,  62 },{ 190,-62 },          #   POS1-4  {X, Y}
  { 118,-162 },{   0,-200 },{-118,-162 },{-190,-62 },          #   POS5-8  {X, Y}
  {-190,  62 },{-118, 162 } };                                 #   POS9-10 {X, Y}
int           FontSizeTmp     = 50;                            # Font size
string InputTmp = getmouseinput(NumResponsesTmp,               #
  NumPositionsTmp,JitterTmp,HalfwidthTmp,                      #
  ActiveCentreTmp,CentreTextTmp,AlphabetTmp,                   #
  FeedbackResponseTmp,PositionsTmp,FontSizeTmp);               #
```

6.16 RUN EXPERIMENT

```
#---RUN EXPERIMENT---------------------------------------------#
array<string> Trials[2][5] =                                   # Trial item array
  {"T1","010020","1","1","Ergo"," "},                          #
  {"T2","030040","2","2","cogito","sum"}                       #
};                                                             #
runexperiment(Trials);                                         #
```

6.17 RUN PRACTICE EXPERIMENT ITEMS

```
#---RUN PRACTICE EXPERIMENT ITEMS FROM 2D STRING ARRAY---------#
NumItemsPres     = 1;                                           # 1:N. Number of presentations of each test item
AwaitResponse    = true;                                        # Await response
int OrigResponseDur = ResponseDur;                              # Store original response duration time
ResponseDur      = 0;                                           # 0 ms response duration time
ResponseType     = "CorrectResponse";                           # "FirstResponse"|"FixedDuration"|"CorrectResponse"
presenttrial(0,"FirstResponse",                                 # Begin practice screen (await response)
  "Press any button to begin practice",PeriodFontSize,          #
  RespInstrPos,"Await"," ",SendCode);                           #
runexperiment(PracticeItems);                                   # Run practice items
ResponseDur = OrigResponseDur;                                  # Reset to original response duration time
```

6.18 RUN MAIN EXPERIMENT ITEMS

```
#---RUN MAIN EXPERIMENT ITEMS FROM 2D STRING ARRAY-------------#
NumItemsPres     = 1;                                           # 1:N. Number of presentations of each test item
AwaitResponse    = true;                                        # Reset to not await response
ResponseDur      = 2000;                                        # Reset to original response duration time
ResponseType     = "FixedDuration";                             # "FirstResponse"|"FixedDuration"|"CorrectResponse"
setresponse(3,0,0);                                             # Set target button
presenttrial(0,"CorrectResponse",                               # Begin experiment screen (await response)
  "Press space bar to start experiment",PeriodFontSize,         #
  RespInstrPos,"Await"," ",SendCode);                           #
presenttrial(2000,"FixedDuration",Asterix,AsterixFontSize,      # Present blink fixation for 2 s
  AsterixPos,"Begin","200",true);                               #
runexperiment(Items);                                           # Run main experiment
```

7. Example experiment

```
################################################################
# EXPERIMENT
################################################################
```

Logdata, show instructions, and run sentence practice session:

```
#---LOG DATA---------------------------------------------------#
logdata("Item\tResponse\tTime\n",LogFile,true);                # LOG: ONE PRESENTATION
#--------------------------------------------------------------#

#---SHOW INSTRUCTIONS------------------------------------------#
showfile("PCP-Instruction-Template.txt",15);                   # Present instruction from file
#--------------------------------------------------------------#

#---RUN PRACTICE ITEMS-----------------------------------------#
DoQuestionnaire  = false;                                      # Perform online questionnaire {true|false}
NumItemsPres     = 1;                                          # 1:N. Number of presentations of each test item
Modality         = "Text";                                     # Presentation modality {"Text"|"Sound"|"Text+Sound"|"Bitmap"}
AwaitResponse    = true;                                       # Await response
ResponseDur      = 0;                                          # 0 ms response duration time
ResponseType     = "FixedDuration";                            # "FirstResponse"|"FixedDuration"|"CorrectResponse"
presenttrial(0,"FirstResponse",                                # Begin practice screen (await response)
  "Press any button to begin practice",20,RespInstrPos,        #
  "Start"," ",SendCode);                                       #
runexperiment(PracticeItems);                                  # Run practice items
#--------------------------------------------------------------#
```

Run bitmap presentation session:

```
#---RUN BITMAP PRESENTATION SESSION----------------------------#
DoQuestionnaire  = false;                                      # Perform online questionnaire {true|false}
NumItemsPres     = 1;                                          # 1:N. Number of presentations of each test item
Modality         = "Bitmap";                                   # Presentation modality {"Text"|"Sound"|"Text+Sound"|"Bitmap"}
AwaitResponse    = false;                                      # Reset to not await response
ResponseDur      = 2000;                                       # Reset to original response duration time
ResponseType     = "FixedDuration";                            # "FirstResponse"|"FixedDuration"|"CorrectResponse"
setresponse(3,0,0);                                            # Set target button
presenttrial(0,"CorrectResponse",                              # Begin experiment screen (await response)
  "Press space bar to start bitmap presentation experiment",   #
  20,RespInstrPos,"Start"," ",SendCode);                       #
runexperiment(BitmapItems);                                    # Run main experiment
```

Run sound presentation session and end experiment session:

```
#---RUN SOUND PRESENTATION SESSION-----------------------------#
NumItemsPres         = 1;                                      # 1:N. Number of presentations of each test item
DoQuestionnaire      = true;                                   # Perform online questionnaire {true|false}
SubStimulusStructure = true;                                   # Split each stimuli in its sub stimulus parts {true|false}
StimDur              = 150;                                    #
InterSubStimDur      = 0;                                      #
InterStimDur         = 300;                                    # Inter-item duration time (ms) for acquisition-classify
SoundAlphabet[1]     = "1";                                    # Sound alphabet
Modality             = "Sound";                                # Presentation modality {"Text"|"Sound"|"Text+Sound"|"Bitmap"}
AwaitResponse        = false;                                  # Reset to not await response
ResponseDur          = 0;                                      # Reset to original response duration time
ResponseType         = "FixedDuration";                        # "FirstResponse"|"FixedDuration"|"CorrectResponse"
setresponse(3,0,0);                                            # Set target button
presenttrial(0,"CorrectResponse",                              # Begin experiment screen (await response)
  "Press space bar to start sound presentation experiment",    #
  20,RespInstrPos,"Start"," ",SendCode);                       #
runexperiment(SoundItems);                                     # Run main experiment
#--------------------------------------------------------------#

#---END OF EXPERIMENT------------------------------------------#
presenttrial(EndDur,"FixedDuration",EndMarker,EndFontSize,     # End
             EndPos,"End",EndCode,false);                      #
#--------------------------------------------------------------#
```

8. Appendix – PCP function library

Detailed description and syntax for the implemented functions.

8.1 DUMMYFMRI

```
#===============================================================#
# DUMMYFMRI                                                     #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Synchronize with FMRI scanner pulse                         #
#                                                               #
# SYNTAX:                                                       #
#                                                               #
#   dummyfmri (NumPulses);                                      #
#                                                               #
# OPTION:                                                       #
#                                                               #
#   NumPulses = Number of FMRI pulses to wait for               #
#                                                               #
# EXAMPLE:                                                      #
#                                                               #
#   int NumPulses = 10;                                         #
#   dummyfmri (NumPulses);                                      #
#                                                               #
# DEPENDENCIES:                                                 #
#                                                               #
#   No dependencies                                             #
#                                                               #
# OBJECTS:                                                      #
#                                                               #
#   No objects                                                  #
#===============================================================#
```

8.2 GETKEYBOARDINPUT

```
#===============================================================#
# GETKEYBOARDINPUT                                              #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Collect input from keyboard                                 #
#                                                               #
# SYNTAX:                                                       #
#                                                               #
#   getkeyboardinput (Instr,InstrFontSize,InstrPos,             #
#                     StimFontSize,StimPos,ButtonCodes);        #
#                                                               #
# OPTION:                                                       #
#                                                               #
#   Instr         = Instruction text                            #
#   InstrFontSize = Instruction font size                       #
#   InstrPos      = Instruction y-position coordinate           #
#   StimFontSize  = Stimuli font size                           #
#   StimPos       = Stimuli y-position coordinate               #
#   ButtonCodes   = Button codes { ... , ERASE KEY, STOP KEY}   #
#                                                               #
# EXAMPLE:                                                      #
#                                                               #
#   NB! Requires 30 response buttons                            #
#       Experiment Input Devices: {LEFT,RIGHT,A:Z,ERASE,STOP}   #
#       Scenario file:                                          #
#         active_buttons = 30;                                  #
#         button_codes   = 1,2,3,...,28,29,30;                  #
#                                                               #
#   string Instr      = "Retype text";                          #
#   int InstrFontSize = 30;                                     #
#   int InstrPos      = 150;                                    #
#   int StimFontSize  = 60;                                     #
#   int StimPos       = 0;                                      #
#   array<string> ButtonCodes = {29,30};                        #
#   getkeyboardinput (Instr,InstrFontSize,InstrPos,             #
#                     StimFontSize,StimPos,ButtonCodes);        #
#                                                               #
# DEPENDENCIES:                                                 #
#                                                               #
#     GETTEXT                                                   #
#     GETPICTURE                                                #
#     GETTRIAL                                                  #
#                                                               #
# OBJECTS:                                                      #
#                                                               #
#     Text object Text                                          #
#     Picture object Picture                                    #
#     Trial object Trial                                        #
#===============================================================#
```

8.3 GETMOUSEINPUT

```
#================================================================#
# GETMOUSEINPUT                                                  #
#================================================================#
# DESCRIPTION:                                                   #
#                                                                #
#   Get and classify mouse input                                 #
#     NB! THIS FUNCTION USE THE FIRST RESPONSE BUTTON            #
#     EVEN IF THIS WOULD NOT BE THE MOUSE BUTTON!                #
#                                                                #
# SYNTAX:                                                        #
#                                                                #
#   string Out = getmouseinput (NumResponses,NumItems,Jitter,   #
#                               HalfWidth,ActiveCentre,          #
#                               CentreText,Alphabet,             #
#                               FeedbackResponse,                #
#                               FeedbackPerformance,             #
#                               Positions,FontSize);             #
#                                                                #
# OPTION:                                                        #
#                                                                #
#   NumResponses     = Number of mouse click to collect          #
#   NumPositions     = Number of selection positions             #
#   Jitter           = Selection position jitter                 #
#   HalfWidth        = Halfwidth of selection area               #
#   ActiveCentre     = Active centre position to jump reponse    #
#   CentreText       = Centre text                               #
#   Alphabet         = Selection alphabet                        #
#   ShuffleAlphabet  = Shuffle alphabet {true,false}             #
#   FeedbackResponse = Response feedback                         #
#   FeedbackPerformance = Performance feedback                   #
#   Positions        = Selection positions x,y-coordinates       #
#   FontSize         = Font size                                 #
#                                                                #
# EXAMPLE:                                                       #
#                                                                #
#   int           NumResponses = 5;                              # Number of mouse click to collect
#   int           NumPositions = 10;                             # Number of selection positions
#   int           Jitter       = 20;                             # Selection position jitter
#   int           Halfwidth    = 25;                             # Halfwidth of selection area
#   bool          ActiveCentre = true;                           # Active centre position to jump reponse
#   string        CentreText   = "?";                            # Centre text
#   array<string> Alphabet[NumPositions] = {                     # Selection alphabet
#     "0","1","2","3","4","5","6","7","8","9"};                  #
#   bool          ShuffleAlphabet = true;                     # Shuffle alphabet
#   bool          FeedbackResponse = false;                   # Response feedback
#   bool          FeedbackPerformance = false                 # Performance feedback
#   array<int>    Positions[NumPositions][2] = {              # Selection positions x,y-coordinates
#     {   0, 200 },{ 118, 162 },{ 190,  62 },{ 190,-62 },     #   POS1-4  {X, Y}
#     { 118,-162 },{   0,-200 },{-118,-162 },{-190,-62 },     #   POS5-8  {X, Y}
#     {-190,  62 },{-118, 162 } };                            #   POS9-10 {X, Y}
#   int           FontSizeTmp     = 50;                       # Font size
#   string Input = getmouseinput(NumResponses,NumPositions,   #
#                               Jitter,Halfwidth,ActiveCentre,#
#                               CentreText,Alphabet,          #
#                               FeedbackResponse,             #
#                               FeedbackPerformance,          #
#                               Positions,FontSize);          #
#                                                             #
# DEPENDENCIES:                                               #
#                                                             #
#   One axis device                                           #
#   SETRESPONSE                                               #
#   RESETSELECTION                                            #
#   GETSELECTION                                              #
#   GETTEXT                                                   #
#   GETBOX                                                    #
#                                                             #
# OBJECTS:                                                    #
#                                                             #
#   Text object Text                                          #
#   Box object Box                                            #
#   Pictuer object Picture                                    #
#   Trial object Trial                                        #
#=============================================================#
```

8.4 GETPICTURE

```
#=============================================================#
# GETPICTURE                                                  #
#=============================================================#
# DESCRIPTION:                                                #
#                                                             #
#   Get Picture object                                        #
#                                                             #
# SYNTAX:                                                     #
#                                                             #
#   picture Out = getpicture (Picture)                        #
#                                                             #
# OPTION:                                                     #
#                                                               #
#   picture Picture = Picture object Picture                    #
#                                                               #
# EXAMPLE:                                                      #
#                                                               #
#   picture Out = getpicture (Picture)                          # Copy picture object Picture to Out
#                                                               #
# DEPENDENCIES:                                                 #
#                                                               #
#   No dependencies                                             #
#                                                               #
# OBJECTS:                                                      #
#                                                               #
#   Text object Bar                                             #
#   Picture object Picture                                      #
#   Global variable Modality                                    #
#   Global variable FixationBars                                #
#===============================================================#
```

8.5 GETSELECTION

```
#===============================================================#
# GETSELECTION                                                  #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Get and classify selection input                            #
#                                                               #
# SYNTAX:                                                       #
#                                                               #
#   int Out = getselection (Halfwidth,NumLocations,Positions,  #
#                           ActiveCentre,xMouse,yMouse);        #
#                                                               #
# OPTION:                                                       #
#                                                               #
#   HalfWidth    = Halfwidth of selection area                  #
#   NumPositions = Number of selection positions                #
#   Positions    = Selection positions x,y-coordinates          #
#   ActiveCentre = Active centre position to jump reponse       #
#   xMouse       = Mouse x-coordinate                           #
#   yMouse       = Mouse y-coordinate                           #
#                                                               #
# EXAMPLE:                                                      #
#                                                               #
#   int  Halfwidth    = 25;                                     # Halfwidth of selection area
#   int  NumPositions = 10;                                     # Number of selection positions
#   array<int> Positions[NumPositions][2] = {                   # Selection positions x,y-coordinates
#     {   0, 200 },{ 118, 162 },{ 190,  62 },{ 190,-62 },       #   POS1-4  {X, Y}
#     { 118,-162 },{   0,-200 },{-118,-162 },{-190,-62 },       #   POS5-8  {X, Y}
#     {-190,  62 },{-118, 162 }                                  #   POS9-10 {X, Y}
#   };                                                          #
#   bool ActiveCentre = true;                                   # Active centre position to jump reponse
#   int  xMouse       = 20;                                     # Mouse x-coordinate
#   int  yMouse       = 40;                                     # Mouse y-coordinate
#   int Out = getselection(Halfwidth,NumPositions,Positions,    #
#                          ActiveCentre,xMouse,yMouse);         #
#                                                               #
# DEPENDENCIES:                                                 #
#                                                               #
#   No dependencies                                             #
#                                                               #
# OBJECTS:                                                      #
#                                                               #
#   No objects                                                  #
#===============================================================#
```

8.6 GETSUBARRAY

```
#===============================================================#
# GETSUBARRAY                                                   #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Get sub array                                               #
#                                                               #
# SYNTAX:                                                       #
#                                                               #
#   array<string> Out = getsubarray (In,Row1,Row2,Col1,Col2);   #
#                                                               #
# OPTION:                                                       #
#                                                               #
#   In   = Full array                                           #
#   Row1 = Sub array row start index                            #
#   Row2 = Sub array row end index                              #
#   Col1 = Sub array col start index                            #
#   Col2 = Sub array col end index                              #
#                                                               #
# EXAMPLE:                                                      #
#                                                               #
#   array<string> In[2][3] = {{"1","2","3"},{"4","5","6"}};     # Full array
#   int Row1 = 1;                                               # Sub array row start index
#   int Row2 = 2;                                               # Sub array row end index
#   int Col1 = 2;                                              # Sub array col start index
#   int Col2 = 3;                                              # Sub array col end index
#   array<string> Out[2][2] = getsubarray(Row1,Row2,Col1,Col2);#
#                                                              #
# DEPENDENCIES:                                                #
#                                                              #
#   No dependencies                                            #
#                                                              #
# OBJECTS:                                                     #
#                                                              #
#   No objects                                                 #
#==============================================================#
```

8.7 GETTEXT

```
#==============================================================#
# GETTEXT                                                      #
#==============================================================#
# DESCRIPTION:                                                 #
#                                                              #
#   Get text object Text                                       #
#                                                              #
# SYNTAX:                                                      #
#                                                              #
#   text Out = gettext (Text);                                 #
#                                                              #
# OPTION:                                                      #
#                                                              #
#   Text = Text object Text                                    #
#                                                              #
# EXAMPLE:                                                     #
#                                                              #
#   text Out = gettext (Text);                                 # Copy text object Text to Out
#                                                              #
# DEPENDENCIES:                                                #
#                                                              #
#   Text object font TextFont                                  #
#   Text object color TextColor                                #
#                                                              #
# OBJECTS:                                                     #
#                                                              #
#   Text object Text                                           #
#==============================================================#
```

8.8 GETTRIAL

```
#===============================================================#
# GETTRIAL                                                      #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Get Trial object                                            #
#                                                               #
# SYNTAX:                                                       #
#                                                               #
#   trial Out = gettrial (Trial);                               #
#                                                               #
# OPTION:                                                       #
#                                                               #
#   trial Trial = Trial object Trial                            #
#                                                               #
# EXAMPLE:                                                      #
#                                                               #
#   trial Out = gettrial (Trial);                               # Copy trial object Trial to Out
#                                                               #
# DEPENDENCIES:                                                 #
#                                                               #
#   No dependencies                                             #
#                                                               #
# OBJECTS:                                                      #
#                                                               #
#   Trial object Trial                                          #
#===============================================================#
```

8.9 LOADBITMAP

```
#===============================================================#
# LOADBITMAP                                                    #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Load or unload bitmap                                       #
#                                                               #
# SYNTAX:                                                       #
#                                                               #
#   loadbitmap (Load,Pos);                                      #
#                                                               #
# OPTION:                                                       #
#                                                                #
#   Load = Load/unload bitmap(s) {true|false}                    #
#   Pos  = Position in bitmap array Bitmaps or all ("All")       #
#                                                                #
# EXAMPLE:                                                       #
#                                                                #
#   loadbitmap(true,"All");                                      # LOAD ALL BITMAP FILES
#                                                                #
#   loop                                                         # LOAD/UNLOAD BLOCK OF BITMAP FILES
#     int Pos = 1                                                #
#   until                                                        # LOOP OVER STIMULI IN TRIAL ARRAY ITEM 1
#     Pos > Items[1][4]                                          #
#   begin                                                        #
#     loadbitmap(false,Items[1][4 + Pos]);                       #
#     Pos = Pos + 1;                                             #
#   end;                                                         #
#                                                                #
# DEPENDENCIES:                                                  #
#                                                                #
#   SHOWERROR                                                    #
#                                                                #
# OBJECTS:                                                       #
#                                                                #
#   Bitmap array Bitmaps                                         #
#================================================================#
```

8.10 LOADSOUND

```
#================================================================#
# LOADSOUND                                                      #
#================================================================#
# DESCRIPTION:                                                   #
#                                                                #
#   Load or unload sound in sound array Sounds                   #
#                                                                #
# SYNTAX:                                                        #
#                                                                #
#   loadsound (Load,Sound,SoundAlphabet);                        #
#                                                                #
# OPTION:                                                        #
#                                                                #
#   Load          = Load/unload sound file                       #
#   Sound         = Sound in SoundAlphabet                       #
#   SoundAlphabet = Sound alphabet                               #
#                                                                #
# EXAMPLE:                                                       #
#                                                                #
#   bool          Load  = true;                                  # Load/unload sound file
#   string        Sound = "M";                                   # Sound in SoundAlphabet
#   array<string> SoundAlphabet[5] = {"M","S","V","R","X"};      # Sound alphabet (ordered as in sound object Sounds)
#   loadsound (Load,Sound,SoundAlphabet);                        #
#                                                                #
# DEPENDENCIES:                                                  #
#                                                                #
#   No dependencies                                              #
#                                                                #
# OBJECTS:                                                       #
#                                                                #
#   Sound array Sounds                                           #
#================================================================#
```

8.11 LOGDATA

```
#================================================================#
# LOGDATA                                                        #
#================================================================#
# DESCRIPTION:                                                   #
#                                                                #
#   Save text to log file                                        #
#                                                                #
# SYNTAX:                                                        #
#                                                                #
#   logdata (InputData,FileName,Overwrite);                      #
#                                                                #
# OPTION:                                                        #
#                                                                #
#   InputData = Text to write to file                            #
#   FileName  = Log file file name                               #
#   Overwrite = Option true to overwrite file.                   #
#               Option false to append text to file.             #
#                                                                #
# EXAMPLE:                                                       #
#                                                                #
#   InputData = "Log text\n";                                    #
#   FileName  = "Log.res";                                       #
#   logdata (InputData,FileName,true);                           # OVERWRITE FILE CONTENT
#   logdata (InputData,FileName,false);                          # APPEND TEXT TO FILE
#                                                                #
# DEPENDENCIES:                                                  #
#                                                                #
#   No dependencies                                              #
#                                                                #
# OBJECTS:                                                      #
#                                                               #
#   No objects                                                  #
#===============================================================#
```

8.12 PRE4STRING2INT

```
#===============================================================#
# PRE4STRING2INT                                                #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Transform text string to an integer string by first         #
#   removing any initial zeroes in the string                   #
#                                                               #
# SYNTAX:                                                       #
#                                                               #
#   string IntegerString = pre4string2int (IntegerString);      #
#                                                               #
# OPTION:                                                       #
#                                                               #
#   IntegerString = Integer string                              #
#                                                               #
# EXAMPLE:                                                      #
#                                                               #
#   string IntegerString = "0001";                              #
#   string IntegerString = pre4string2int (IntegerString);      #
#                                                               #
# DEPENDENCIES:                                                 #
#                                                               #
#   SHOWERROR                                                   #
#                                                               #
# OBJECTS:                                                      #
#                                                               #
#   No objects                                                  #
#===============================================================#
```

8.13 PRESENTBITMAP

```
#===============================================================#
# PRESENTBITMAP                                                 #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Present bitmaps                                             #
#                                                               #
# SYNTAX:                                                       #
#                                                               #
#   presentbitmap (Bitmap,Duration,PreLoadedBitmaps,            #
#                  EventCode,PortCode,SendCode);                #
#                                                               #
# OPTION:                                                       #
#                                                               #
#   Bitmap           = Bitmaps bitmap object array position     #
#   Duration         = Duration (ms)                            #
#   PreLoadedBitmaps = Preloaded bitmap objects {true|false}    #
#   EventCode        = Event log code                           #
#   PortCode         = Port code                                #
#   SendCode         = Send port code {true|false}              #
#                                                               #
# EXAMPLE:                                                      #
#                                                               #
#   int    Bitmap           = 1;                                # Bitmaps bitmap object array position
#   int    Duration         = 1000;                             # Duration (ms)
#   bool   PreLoadedBitmaps = false;                            # Preloaded bitmap objects {true|false}
#   string EventCode        = "TestBitmapPresentation";         # Event log code
#   string PortCode         = "000";                            # Port code
#   bool   SendCode         = false;                            # Send port code {true|false}
#   presentbitmap(Bitmap,Duration,PreLoadedBitmaps,             #
#                 EventCode,PortCode,SendCode)                  #
#                                                               #
# DEPENDENCIES:                                                 #
#                                                               #
#   GETTRIAL                                                    #
#   GETPICURE                                                   #
#                                                               #
# OBJECTS:                                                      #
#                                                               #
#   Trial object Trial                                          #
#   Picure object Picture                                       #
#   Bitmap object array Bitmaps                                 #
#===============================================================#
```

8.14 PRESENTDEFAULT

```
#===============================================================#
# PRESENTDEFAULT                                                #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Present default trial screen                                #
#                                                              #
# SYNTAX:                                                      #
#                                                              #
#   presentdefault (Duration);                                 #
#                                                              #
# OPTION:                                                      #
#                                                              #
#   Duration = Duration (ms)                                   #
#                                                              #
# EXAMPLE:                                                     #
#                                                              #
#   int Duration = 1000;                                       # Duration (ms)
#   presentdefault (Duration);                                 #
#                                                              #
# DEPENDENCIES:                                                #
#                                                              #
#   GETPICTURE                                                 #
#   GETTRIAL                                                   #
#                                                              #
# OBJECTS:                                                     #
#                                                              #
#   Picture object default                                     #
#   Trial object Trial                                         #
#==============================================================#
```

8.15 PRESENTITEM

```
#==============================================================#
# PRESENTITEM                                                  #
#==============================================================#
# DESCRIPTION:                                                 #
#                                                              #
#   Present item                                               #
#                                                              #
# SYNTAX:                                                      #
#                                                              #
#   presentitem (StimDur,InterStimDur,InterSubStimDur,         #
#                VariableStimDur,VarStimDur,Str,               #
#                InterStimMarker,StimPos,InterStimPos,         #
#                StimFontSize,InterStimFontSize,               #
#                Modality,SubStimulusStructure,AddPoint,       #
#                PreLoadedBitmaps,Alphabet,InterStimCode,      #
#                LogFile,SendCode);                            #
#                                                              #
# OPTION:                                                      #
#                                                              #
#   StimDurF             = Stimulus/Substimulus duration (ms)  #
#   InterStimDur         = Inter-stimulus duration (ms)        #
#   InterSubStimDur      = Inter-subtimulus duration (ms)      #
#   VariableStimDur      = Variable stimulus duration          #
#   VarStimDur           = Variable stimulus times (ms)        #
#   Str                  = Trial items array                   #
#   InterStimMarker      = Inter-stumulus marker               #
#   StimPos              = Stimulus y-coordinate position      #
#   InterStimPos         = Inter-stimulus y-coord. position    #
#   StimFontSize         = Stimulus font size                  #
#   InterStimFontSize    = Inter-stimulus font size            #
#   Modality             = Modality mode                       #
#   SubStimulusStructure = Split each stimuli in substimulus   #
#   AddPoint             = Add point after last substimuli     #
#   PreLoadedBitmaps     = Preloaded bitmap objects            #
#   Alphabet             = Sound alphabet                      #
#   InterStimCode        = Inter-stimulus port code            #
#   LogFile              = Logfile file name                   #
#   SendCode             = Send code                           #
#                                                              #
# EXAMPLE:                                                     #
#                                                              #
#   int           StimDur              = 300;                  # Stimulus/Substimulus duration (ms)
#   int           InterStimDur         = 300;                  # Inter-stimulus duration (ms)
#   int           InterSubStimDur      = 300;                  # Inter-substimulus duration (ms)
#   bool          VariableStimDur      = true;                 # Variable stimulus duration {true|false}
#   array<string> VarStimDur[7][2]     = VarStimDur;           # Variable stimulus times (ms)
#   array<string> Str[Items[1].count()] = Items[1];            # Trial items array
#   string        InterStimMarker      = InterStimMarker;      # Inter-stimulus marker
#   int           StimPos              = StimPos;              # Stimulus y-coordinate position
#   int           InterStimPos         = InterStimPos;         # Inter-stimulus y-coordinate position
#   int           StimFontSize         = StimFontSize;         # Stimulus font size
#   int           InterStimFontSize    = InterStimFontSize;    # Inter-stimulus font size
#   string        Modality             = "Bitmap";             # Modality mode
#   bool          SubStimulusStructure  = true;                # Split each stimuli in substimulus {true|false}
#   bool          AddPoint             = AddPoint;             # Add point after last substimuli {true|false}
#   bool          PreLoadedBitmaps     = PreLoadedBitmaps;     # Preloaded bitmap objects {true|false}
#   array<string> Alphabet[5]          = SoundAlphabet;        # Sound alphabet
#   string        InterStimCode        = InterStimCode;        # Inter-stimulus event code
#   string        LogFile              = LogFile;              # Logfile file name
#   bool          SendCode             = SendCode;             # Send code {true|false}
#   presentitem(StimDur,InterStimDur,InterSubStimDur,          #
#               VariableStimDur,VarStimDur,Str,                #
#               InterStimMarker,StimPos,InterStimPos,          #
#               StimFontSize,InterStimFontSize,Modality,       #
#               SubStimulusStructure,AddPoint,                 #
#               PreLoadedBitmaps,Alphabet,InterStimCode,       #
#               LogFile,SendCode);                             #
#                                                              #
# DEPENDENCIES:                                                #
#                                                              #
#   LOGDATA                                                    #
#   PRESENTDEFAULT                                             #
#   PRESENTBITMAP                                              #
#   PRESENTSOUND                                               #
#   PRESENTTRIAL                                               #
#                                                              #
# OBJECTS:                                                     #
#                                                              #
#   No objects                                                 #
#==============================================================#
```

8.16 PRESENTQUESTION

```
#==============================================================#
# PRESENTQUESTION                                              #
#==============================================================#
# DESCRIPTION:                                                 #
#                                                              #
#   Present online questions                                   #
#                                                              #
# SYNTAX:                                                      #
#                                                              #
#   presentquestion(FileName,Delay,ButtonCodes)                #
#                                                              #
# OPTION:                                                      #
#                                                              #
#   FileName = FileName file name                              #
#   DelayF = Delay time (ms) for requested questionair response#
#   ButtonCodes = Button codes                                 #
#                                                              #
# EXAMPLE:                                                     #
#                                                              #
#   string     FileName = "Q1.txt";                            # File name
#   int        Delay = 1000;                                   # Delay response time (ms)
#   array<int> ButtonCodes[2] = {1,2};                         # Button codes {Left,Right}
#   presentquestion(FileName,Delay,ButtonCodes);               #
#                                                              #
# DEPENDENCIES:                                                #
#                                                              #
#   GETPICURE                                                  #
#   GETTRIAL                                                   #
#   GETTEXT                                                     #
#   LOGDATA                                                     #
#                                                               #
# OBJECTS:                                                      #
#                                                               #
#   10 text objects in text objects array Texts                 #
#   Trial object Trial                                          #
#   Picure object Picture                                       #
#   Text object Text                                            #
#===============================================================#
```

8.17 PRESENTSOUND

```
#===============================================================#
# PRESENTSOUND                                                  #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Present sound item                                          #
#                                                               #
# SYNTAX:                                                       #
#                                                               #
#   presentsound (Sound,Alphabet,Duration,                      #
#                 EventCode,PortCode,SendCode);                 #
#                                                               #
# OPTION:                                                       #
#                                                               #
#   Sound     = Sound item                                      #
#   Alphabet  = Sound alphabet                                  #
#   Duration  = Duration (ms)                                   #
#   EventCode = Event log code                                  #
#   PortCode  = Port code                                       #
#   SendCode  = Send code {true|false}                          #
#                                                               #
# EXAMPLE:                                                      #
#                                                               #
#   string        Sound       = "M";                            # Sound item
#   array<string> Alphabet[5] = {"M","S","V","R","X"};          # Sound alphabet (ordered as in sound object Sounds)
#   int           Duration    = 1000;                           # Duration (ms)
#   string        EventCode   = "LogThis";                      # Event log code
#   string        PortCode    = "000";                          # Port code
#   bool          SendCode    = false;                          # Send code {true|false}
#   presentsound (Sound,Alphabet,Duration,                      #
#                 EventCode,PortCode,SendCode);                 #
#                                                               #
# DEPENDENCIES:                                                 #
#                                                               #
#     GETTRIAL                                                  #
#                                                               #
# OBJECTS:                                                      #
#                                                               #
#     Trial object Trial                                        #
#     Sounds                                                    #
#===============================================================#
```

8.18 PRESENTTRIAL

```
#===============================================================#
# PRESENTTRIAL                                                  #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Present trial screen                                        #
#                                                               #
# SYNTAX:                                                       #
#                                                               #
#   presenttrial (Duration,ResponseType,TextStr,FontSize,      #
#                 yPosition,EventCode,PortCode,SendCodeF)      #
#                                                               #
# OPTION:                                                       #
#                                                               #
#   Duration     = Duration (ms)                                #
#   ResponseType = Response type {"FirstResponse"|              #
#         "FixedDuration"|"CorrectResponse"|"SpecificResponse"}#
#   TextStr      = Text string or nothing ("Keep")              #
#   FontSize     = Font size                                    #
#   yPosition    = Object position y-coordinate                 #
#   EventCode    = Event log code                               #
#   PortCode     = Port code                                    #
#   SendCode     = Send code {true|false}                       #
#                                                               #
# EXAMPLE:                                                      #
#                                                               #
#   Duration     = 1000;                                        # Set duration to 1000 ms
#   ResponseType = "FixedDuration";                             # Collect but ignore responses
#   TextStr      = "FixedDuration";                             # Set text string
#   FontSize     = 20;                                          # Set font size to 20
#   yPosition    = 150;                                         # Set object position y-coordinate to 150
#   EventCode    = "Log";                                       # Set event log code to "Log"
#   PortCode     = "20";                                        # Set port code to 20
#   SendCode     = true                                         # Send code
#   presenttrial (Duration,ResponseType,TextStr,FontSize,      #
#                 yPosition,EventCode,PortCode,SendCode)       # PRESENT TRIAL WITH FIXED DURATION
#                                                              #
#   ResponseType = "FirstResponse";                            # Collect but ignore responses
#   TextStr      = "FirstResponse";                            # Set text string
#   presenttrial (Duration,ResponseType,TextStr,FontSize,      #
#                 yPosition,EventCode,PortCode,SendCode)       # PRESENT TRIAL AND AWAIT ANY RESPONSE
#                                                              #
#   ResponseType = "CorrectResponse";                          # Collect but ignore responses
#   TextStr      = "CorrectResponse";                          # Set text string
#   setresponse(3,0,0);                                        # Set target button 3 as correct response
#   presenttrial (Duration,ResponseType,TextStr,FontSize,      #
#                 yPosition,EventCode,PortCode,SendCode)       # PRESENT TRIAL AND AWAIT CORRECT RESPONSE
#                                                              #
# DEPENDENCIES:                                                #
#                                                              #
#   GETTEXT                                                    #
#   GETPICTURE                                                 #
#   GETTRIAL                                                   #
#                                                              #
# OBJECTS:                                                     #
#   Text object Text                                           #
#   Picture object Picture                                     #
#   Trial object Trial                                         #
#==============================================================#
```

8.19 READFILE

```
#==============================================================#
# READFILE                                                     #
#==============================================================#
# DESCRIPTION:                                                 #
#                                                              #
#   Open and read a text file                                  #
#                                                              #
# SYNTAX:                                                      #
#                                                              #
#   string Out = readfile (FileName);                          #
#                                                              #
# OPTION:                                                      #
#                                                              #
#   FileName = File name                                       #
#                                                              #
# EXAMPLE:                                                     #
#                                                              #
#   string FileName = "File.txt";                              # File name
#   string Out = readfile (FileName);                          #
#                                                               #
# DEPENDENCIES:                                                 #
#                                                               #
#   No dependencies                                             #
#                                                               #
# OBJECTS:                                                      #
#                                                               #
#   No objects                                                  #
#===============================================================#
```

8.20 RESETSELECTION

```
#===============================================================#
# RESETSELECTION                                                #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Reset click selection                                       #
#                                                               #
# SYNTAX:                                                       #
#                                                               #
#   picture Out = resetselection (NumPositions,Alphabet,        #
#                                 Positions,FontSize,           #
#                                 ActiveCentre,CentreText);     #
# OPTION:                                                       #
#                                                               #
#   NumPositions = Number of selection positions                #
#   Alphabet     = Selection alphabet                           #
#   Positions    = Selection positions x,y-coordinates          #
#   FontSize     = Font size                                    #
#   ActiveCentre = Active centre position to jump reponse       #
#   CentreText   = Centre text                                  #
#                                                               #
# EXAMPLE:                                                      #
#                                                               #
#   int    NumPositions = 10;                                   # Number of selection positions
#   array<string> Alphabet[NumPositions] = {                    # Selection alphabet
#     "0","1","2","3","4","5","6","7","8","9"};                 #
#   array<int> Positions[NumPositions][2] = {                   # Selection positions x,y-coordinates
#     {   0, 200 },{ 118, 162 },{ 190,  62 },{ 190,-62 },       #   POS1-4  {X, Y}
#     { 118,-162 },{   0,-200 },{-118,-162 },{-190,-62 },       #   POS5-8  {X, Y}
#     {-190,  62 },{-118, 162 }                                 #   POS9-10 {X, Y}
#   };                                                          #
#   int    FontSize     = 50;                                   # Font size
#   bool   ActiveCentre = true;                                 # Active centre position to jump reponse
#   string CentreText   = "?";                                  # Centre text
#   picture Picture = resetselection(NumPositions,Alphabet,    #
#                                    Positions,FontSize,       #
#                                    ActiveCentre,CentreText); #
#                                                              #
# DEPENDENCIES:                                                #
#                                                              #
#   GETTEXTARRAY                                               #
#   GETPICTURE                                                 #
#   GETBOX                                                     #
#                                                              #
# OBJECTS:                                                     #
#                                                              #
#   Text object Text                                           #
#   Picture object Picture                                     #
#   Box object Box                                             #
#   TextFont                                                   #
#==============================================================#
```

8.21 RUNEXPERIMENT

```
#==============================================================#
# RUNEXPERIMENT                                                #
#==============================================================#
# DESCRIPTION:                                                 #
#                                                              #
#   Run the main experiment                                    #
#                                                              #
# SYNTAX:                                                      #
#                                                              #
#   runexperiment(Trials);                                     # String array Trials for trials
#                                                              #
# OPTION:                                                      #
#                                                              #
#   Trials = String array Trials for trials                    #
#                                                              #
# EXAMPLE:                                                     #
#                                                              #
#   array<string> Trials[2][5] =                               # Trial item array
#     {"T1","010020","1","1","Ergo"," "},                      #
#     {"T2","030040","2","2","cogito","sum"}                   #
#   };                                                         #
#   runexperiment(Trials);                                     #
#                                                              #
# DEPENDENCIES:                                                #
#                                                              #
#   GETKEYBOARDINPUT                                           #
#   GETMOUSEINPUT                                                #
#   LOGDATA                                                      #
#   PRESENTITEM                                                  #
#   PRESENTSOUND                                                 #
#   PRESENTTRIAL                                                 #
#   SETRESPONSE                                                  #
#   TESTITEMSARRAY                                               #
#                                                                #
# OBJECTS:                                                       #
#                                                                #
#   Trial object StopSoundTrial                                  #
#================================================================#
```

8.22 RUNQUESTIONNAIRE

```
#================================================================#
# RUNQUESTIONNAIRE                                               #
#================================================================#
# DESCRIPTION:                                                   #
#                                                                #
#   Display multiple online questions                            #
#                                                                #
# SYNTAX:                                                        #
#                                                                #
#   runquestionnaire (FileName,Delay,ButtonCodes)                #
#                                                                #
# OPTION:                                                        #
#                                                                #
#   FileNames   = File name string array                         #
#   DelayF      = Delay time (ms) for questionnaire response     #
#   ButtonCodes = Button codes {Left,Right}                      #
#                                                                #
# EXAMPLE:                                                       #
#                                                                #
#   array<string> FileNames[3] = {"Q1.txt","Q2.txt","Q3.txt"}; # File name string array
#   int           Delay        = 1000;                         # Delay response time (ms)
#   array<int>    ButtonCodes[2] = {1,2};                      # Button codes {Left,Right}
#   runquestionnaire (FileNames,Delay,ButtonCodes);              #
#                                                                #
# DEPENDENCIES:                                                  #
#                                                                #
#   PRESENTQUESTION                                              #
#                                                                #
# OBJECTS:                                                       #
#                                                                #
#   No objects                                                   #
```

```
#===============================================================#
```

8.23 SETRESPONSE

```
#===============================================================#
# SETRESPONSE                                                   #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Update correct response key in stimulus event object        #
#     StimulusEvent in trial object Trial                       #
#                                                               #
# SYNTAX:                                                       #
#                                                               #
#   setresponse (Key,StartTime,StopTime)                        #
#                                                               #
# OPTION:                                                       #
#                                                               #
#   Key       = Target buttom key                               #
#   StartTime = Start time (ms)                                 #
#   StopTime  = Stop time (ms, 0 == no stop time)               #
#                                                               #
# EXAMPLE:                                                      #
#                                                               #
#   int Key       = 3;                                          # Set button 3 as target buttom key
#   int StartTime = 0;                                          # Start time (ms)
#   int StopTime  = 0;                                          # No stop time
#   setresponse (Key,StartTime,StopTime)                        #
#                                                               #
# DEPENDENCIES:                                                 #
#                                                               #
#   No dependencies                                             #
#                                                               #
# OBJECTS:                                                      #
#                                                               #
#   Stimulus event object StimulusEvent                         #
#===============================================================#
```

8.24 SHOWERROR

```
#===============================================================#
# SHOWERROR                                                     #
#===============================================================#
# DESCRIPTION:                                                  #
#                                                               #
#   Display error message                                         #
#                                                                 #
# SYNTAX:                                                         #
#                                                                 #
#   showerror (ErrorMessage);                                     #
#                                                                 #
# OPTION:                                                         #
#                                                                 #
#   ErrorMessage = Error message                                  #
#                                                                 #
# EXAMPLE:                                                        #
#                                                                 #
#   ErrorMessage = "Error this or that";                          # Error message
#   showerror (ErrorMessage);                                     #
#                                                                 #
# DEPENDENCIES:                                                   #
#                                                                 #
#   GETTEXT                                                       #
#   GETPICTURE                                                    #
#   GETTRIAL                                                      #
#                                                                 #
# OBJECTS:                                                        #
#                                                                 #
#   Text object Text                                              #
#   Picture object Picture                                        #
#   Trial object Trial                                            #
#=================================================================#
```

8.25 SHOWFILE

```
#=================================================================#
# SHOWFILE                                                        #
#=================================================================#
# DESCRIPTION:                                                    #
#                                                                 #
#   Display instructions                                          #
#                                                                 #
# SYNTAX:                                                         #
#                                                                 #
#   showfile (FileName,FontSize);                                 #
#                                                                 #
# OPTION:                                                         #
#                                                                 #
#   FileName = Text file to present on screen                     #
#   FontSize = Text font size                                     #
#                                                                 #
# EXAMPLE:                                                    #
#                                                             #
#   string FileName = "PCP-Instruction-Template.txt";         #
#   int FontSize    = 15;                                     #
#   showfile (FileName,FontSize);                             #
#                                                             #
# DEPENDENCIES:                                               #
#                                                             #
#   GETTEXT                                                   #
#   GETPICTURE                                                #
#   GETTRIAL                                                  #
#                                                             #
# OBJECTS:                                                    #
#                                                             #
#   Text object Text                                          #
#   Picture object Picture                                    #
#   Trial object Trial                                        #
#=============================================================#
```

8.26 TESTITEMSARRAY

```
#=============================================================#
# TESTITEMSARRAY                                              #
#=============================================================#
# DESCRIPTION:                                                #
#                                                             #
#   Test consitencies in the trial items array                #
#                                                             #
# SYNTAX:                                                     #
#                                                             #
#   testitemsarray (NumItemPres,NumBlocks,Str);               #
#                                                             #
# OPTION:                                                     #
#                                                             #
#   NumItemPres = Number of presentations                     #
#   NumBlocks   = Number of presentation blocks               #
#   Trials      = Trial item array                            #
#                                                             #
# EXAMPLE:                                                    #
#                                                             #
#   int  NumItemPres  = 3;                                    # Number of presentations
#   int  NumBlocks    = 2;                                    # Number of presentation blocks
#   array<string> Trials[2][5] =                              # Trial item array
#     {"T1","010020","1","1","Ergo"," "},                     #
#     {"T2","030040","2","2","cogito","sum"} };               #
#   testitemsarray (NumItemPres,NumBlocks,Str);               #
#                                                              #
# DEPENDENCIES:                                                #
#                                                              #
#   No dependencies                                            #
#                                                              #
# OBJECTS:                                                     #
#                                                              #
#   No objects                                                 #
#==============================================================#
```

8.27 TRIALORDERRANDOMIZATION

```
#==============================================================#
# TRIALORDERRANDOMIZATION                                      #
#==============================================================#
# DESCRIPTION:                                                 #
#                                                              #
#   Generate randomized trial order array                      #
#                                                              #
# SYNTAX:                                                      #
#                                                              #
#   array<int> Out = trialorderrandomization (NumBlock,NumMax);#
#                                                              #
# OPTION:                                                      #
#                                                              #
#   int NumBlock = Number of blocks in experiment              #
#   int NumMax   = Maximum number of trials                    #
#                                                              #
# EXAMPLE:                                                     #
#                                                              #
#   int NumBlock   = 4;                                        # Number of blocks in experiment
#   int NumMax     = 30;                                       # Maximum number of trials
#   array<int> Out = trialorderrandomization (NumBlock,NumMax);#
#                                                              #
# DEPENDENCIES:                                                #
#                                                              #
#   No dependencies                                            #
#                                                              #
# OBJECTS:                                                     #
#                                                              #
#   No objects                                                 #
#==============================================================#
```