```
/*
Auditory P300 Task

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%% Version History %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
- 06-10-2019; created by Chris Gill: created the backbone of the experiment
- 10-19-2019; edited by Alex He: changed the structure of the task, improved the iteration code block
- 03-13-2020; edited by Alex He: added proper response logging and volume comments
- 11-09-2020; edited by Alex He: added triggers, added version history, added 22/23 start/end triggers, updated header parameters, enabled proper logfile saving
- 11-11-2020; edited by Alex He: removed unnecssary screenshots, made random ISI from 1500-2000ms to 1500-2500ms, added 200ms delay during example tone presentataion,
                                 fixed the switched order of standard and oddball tones
- 01-19-2021; edited by Alex He: updated logfile naming and added auditory tone repeat during instruction
10-05-2022; edited by Anthony Edgar: changed task to be compatible with c-pod.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

*IMPORTANT Note: Before beginning task set volume on Presentation Laptop to 70

*/
########################################################################################################################
# Header
scenario = "Auditory_P300";
active_buttons = 1;
button_codes = 11; #1 = spacebar
default_font = "Helvetica";
default_background_color = 127, 127, 127;
default_font_size = 40;
write_codes = true;
response_port_output = true;
default_output_port = 1;
pulse_width = 5;
########################################################################################################################
#SDL
begin;

picture {} default;

#rest box
box { height = 1080; width = 1920; color = 127,127,127;} rest_box;

text { caption = "+"; font_size = 96; font_color = 255,255,255; } Crosshair_text;

#Sounds- these serve as placeholders, the actual sounds used are generated below
array {
sound {wavefile { filename = "standard_stereo.wav";};} standard_beep;
sound {wavefile { filename = "deviant_stereo.wav";};} oddball_beep;
} sounds;


picture {
   text {
   caption = "INSTRUCTIONS:";
   font_size = 44;
   };
   x = 0;y = 400;

```

```
   text {
   caption =
   "In this task you will hear a series of tones.
   There are two different tones.

   Every time you hear the following tone, press the space bar.";
   font_size = 44;
   text_align = align_left;
   };
   x = 0;y = 20;

   text {
   caption = "Press the space bar to hear the tone.";
   font_size = 32;
   };
   x = 0;y = -400;
} instructions1;

picture {
   text {
   caption = "INSTRUCTIONS:";
   font_size = 44;
   };
   x = 0;y = 400;

   text {
   caption =
   "Every time you hear the following tone, don't press the space bar.
   You don't need to make a response.";
   font_size = 44;
   text_align = align_left;
   };
   x = 0;y = 20;

   text {
   caption = "Press the space bar to hear the tone.";
   font_size = 32;
   };
   x = 0;y = -400;
} instructions2;

picture {
   text {
   caption = "INSTRUCTIONS:";
   font_size = 44;
   };
   x = 0;y = 400;

   text {
   caption =
   "Now we will hear both tones again.
   Pay attention and decide which tone you need to respond to:
   is it the first tone or the second tone?";
   font_size = 44;
   text_align = align_left;
   };
   x = 0;y = 20;

   text {
   caption = "Press the space bar to hear the tones.";
   font_size = 32;
   };
   x = 0;y = -400;
} instructions4;
```

```
121
122   picture {
123      text {
124      caption = "INSTRUCTIONS:";
125      font_size = 44;
126      };
127      x = 0;y = 400;
128
129      text {
130      caption =
131      "Would you respond to the first tone or the second tone?";
132      font_size = 44;
133      text_align = align_left;
134      };
135      x = 0;y = 20;
136
137      text {
138      caption = "Press the space bar to continue.";
139      font_size = 32;
140      };
141      x = 0;y = -400;
142   } instructions5;
143
144   picture {
145      text {
146      caption = "INSTRUCTIONS:";
147      font_size = 44;
148      };
149      x = 0;y = 400;
150
151      text {
152      caption =
153      "Please perform this task with your eyes closed and try to respond
154      as quickly and accurately as possible.
155
156      At the halfway point of the task there will be a rest period. At
157      that time the experimenter will notify you that you can open your
158      eyes and take a short break.";
159      font_size = 44;
160      text_align = align_left;
161      };
162      x = 0;y = 20;
163
164      text {
165      caption = "Press the space bar to begin the task.";
166      font_size = 32;
167      };
168      x = 0;y = -400;
169   } instructions3;
170
171
172   picture {
173      box rest_box;
174      x = 0; y = 0;
175
176      text Crosshair_text;
177      x = 0; y = 0;
178   } background_pic;
179
180   ##############################################################################
      ########################
181   #Trials
182
183   #Instructions
```

```
trial {
   trial_type = specific_response;
   trial_duration = forever;
   terminator_button = 1;
      stimulus_event{
         picture instructions1;
         code = "Instruction 1";
      };
}instructions_trial1;

trial {
   trial_type = specific_response;
   trial_duration = forever;
   terminator_button = 1;
      stimulus_event{
         picture instructions2;
         code = "Instruction 2";
      };
}instructions_trial2;

trial {
   trial_type = specific_response;
   trial_duration = forever;
   terminator_button = 1;
      stimulus_event{
         picture instructions3;
         code = "Instruction 3";
      };
}instructions_trial3;

trial {
   trial_type = specific_response;
   trial_duration = forever;
   terminator_button = 1;
      stimulus_event{
         picture instructions4;
         code = "Instruction 4";
      };
}instructions_trial4;

trial {
   trial_type = specific_response;
   trial_duration = forever;
   terminator_button = 1;
      stimulus_event{
         picture instructions5;
         code = "Instruction 5";
      };
}instructions_trial5;

trial {
   trial_type = fixed;
   trial_duration = 2000;
   picture background_pic;
   stimulus_event {
      sound oddball_beep;
      code = "Oddball example";
   } dev_event2;
} oddball_trial_example;

trial {
   trial_type = fixed;
   trial_duration = 2000;
   picture background_pic;
```

```
248      stimulus_event {
249         sound standard_beep;
250         code = "Standard example";
251      } std_event2;
252   } standard_trial_example;
253
254   trial {
255      trial_type = fixed;
256      trial_duration = 2000;
257      picture background_pic;
258      stimulus_event {
259         sound standard_beep;
260         target_button = 1;
261         port_code = 2;
262         code = "Standard Tone";
263      } std_event;
264   } standard_trial;
265
266   trial {
267      trial_type = fixed;
268      trial_duration = 2000;
269      picture background_pic;
270      stimulus_event {
271         sound oddball_beep;
272         target_button = 1;
273         port_code = 3;
274         code = "Oddball Tone";
275      } dev_event;
276   } oddball_trial;
277
278   trial {
279      trial_type = fixed;
280      trial_duration = 2000;
281      stimulus_event {
282         picture background_pic;
283         code = "2s Blank";
284      } blank_event;
285   } blank_trial;
286
287   trial {
288      trial_type = correct_response;
289      trial_duration = forever;
290      stimulus_event {
291         picture {
292            box rest_box;
293            x = 0; y = 0;
294
295            text {
296            caption = "Rest";
297            font_size = 44;
298            };
299            x = 0; y = 120;
300
301            text {
302            caption = "Press the space bar to begin Block 2.";
303            font_size = 32;
304            };
305            x = 0; y = -300;
306
307         }rest_pic;
308         target_button = 1;
309         port_code = 4;
310         code = "Rest";
311      } rest_event;
```

```
} rest_trial;

################################################################################
########################
#Begin PCL
begin_pcl;

#specifying output file
string logpath = logfile_directory;
string fn = logpath +logfile.subject()+"_P300.txt";
logfile.set_filename(logpath +logfile.subject() + "_P300_logfile.log");
output_file ofile1 = new output_file;
ofile1.open_append(fn);

###Variables###
int num_blocks = 2;
int stim_per_block = 100;
int num_trials = 10;
int stim_per_trial = 10;
int STD_IDX = 1;
int DEV_IDX = 2;
int oddball;
int Trial_duration;

response_data my_response;
stimulus_data my_data;
int resp_button;
int RT;
int stim_onset;
int Correct_Resp;

##Make Sounds##
array<sound> stim_snds[2];
# Initialize some values
double ramp_up_time = parameter_manager.get_double( "Generated Sound Rise Time" );
double ramp_down_time = parameter_manager.get_double( "Generated Sound Fall Time"
);

# Make the rise/fall ramps
asg::line ramp_down = new asg::line( ramp_down_time, 1.0, 0.0 );
asg::line ramp_up = new asg::line( ramp_up_time, 0.0, 1.0 );

# Make the waveform data
double std_duration = parameter_manager.get_double( "Generated Sound Standard
Duration" );
double dev_duration = parameter_manager.get_double( "Generated Sound Deviant
Duration" );
asg::sine std_data = new asg::sine( std_duration, parameter_manager.get_double(
"Generated Sound Standard Frequency" ), 0.0 );
asg::sine dev_data = new asg::sine( dev_duration, parameter_manager.get_double(
"Generated Sound Deviant Frequency" ), 0.0 );
asg::waveform_data std_wf = new asg::waveform_data( std_data );
asg::waveform_data dev_wf = new asg::waveform_data( dev_data );

# Multiply by the rise/fall times
std_wf.multiply_segment( ramp_up, 0.0 );
std_wf.multiply_segment( ramp_down, std_wf.duration() - ramp_down_time );
dev_wf.multiply_segment( ramp_up, 0.0 );
dev_wf.multiply_segment( ramp_down, dev_wf.duration() - ramp_down_time );

# Now make the sound objects
stim_snds[STD_IDX] = new sound( new wavefile( std_wf, std_wf ) );
stim_snds[DEV_IDX] = new sound( new wavefile( dev_wf, dev_wf ) );

```

```
370   # Set the attenuation on the sounds
371   begin
372      double std_atten = 1.0 - ( double( parameter_manager.get_int( "Standard Volume"
       ) ) / 100.0 );
373      double dev_atten = 1.0 - ( double( parameter_manager.get_int( "Deviant Volume"
      ) ) / 100.0 );
374      stim_snds[STD_IDX].set_attenuation( std_atten );
375      stim_snds[DEV_IDX].set_attenuation( dev_atten );
376   end;
377
378   std_event.set_stimulus(stim_snds[STD_IDX]);
379   dev_event.set_stimulus(stim_snds[DEV_IDX]);
380   std_event2.set_stimulus(stim_snds[STD_IDX]);
381   dev_event2.set_stimulus(stim_snds[DEV_IDX]);
382   ##############################################################################
      ##########################################################################
383   #response_manager.set_port_output( false );
384   instructions_trial1.present();
385   #display_device.screenshot("1.png");
386   wait_interval(200);
387   oddball_trial_example.present();
388   instructions_trial2.present();
389   #display_device.screenshot("2.png");
390   wait_interval(200);
391   standard_trial_example.present();
392   instructions_trial4.present();
393   wait_interval(200);
394   oddball_trial_example.present();
395   standard_trial_example.present();
396   instructions_trial5.present();
397   instructions_trial3.present();
398   #display_device.screenshot("3.png");
399   #response_manager.set_port_output( true );
400
401   output_port port = output_port_manager.get_port( 1 );
402
403   #Setting column headers for output file
404   ofile1.print("Block\t" + "Stim\t" + "Onset\t" + "RT\t" + "Oddball\t" + "Correct\n"
      );
405
406   loop int block = 1; int total_stim_count = 1; until block > num_blocks begin
407
408      if (block > 1) then
409         #response_manager.set_port_output( false );
410         rest_trial.present();
411         #response_manager.set_port_output( true );
412      end;
413
414      blank_trial.present();
415      wait_interval (100);
416      port.send_code(22);
417      wait_interval (100);
418
419      loop int trial_count = 1; until trial_count > num_trials begin #10 trials with
     10 stimuli each (8 standard and 2 deviant/oddball)
420
421         array<int>stimuli_type[10] =
422         {0,0,0,0,0,0,0,0,1,1};
423         stimuli_type.shuffle();
424
425         loop int stim_count = 1; until stim_count > stim_per_trial begin
426            Trial_duration = random(1500,2500);
427            if (stimuli_type[stim_count] == 1) then
428               oddball_trial.set_duration(Trial_duration);
```

```
            oddball_trial.present();

            #Saving data
            my_data = stimulus_manager.last_stimulus_data();
            my_response = response_manager.last_response_data();
            RT = my_data.reaction_time();
            stim_onset = my_data.time();
            if (RT == 0) then
               Correct_Resp = 0;
            else
               Correct_Resp = 1;
            end;

            ofile1.print(string(block) + "\t");
            ofile1.print(string(total_stim_count) + "\t");
            ofile1.print(string(stim_onset) + "\t");
            ofile1.print(string(RT) + "\t");
            ofile1.print("1" + "\t");
            ofile1.print(string(Correct_Resp) + "\n");

         else
            standard_trial.set_duration(Trial_duration);
            standard_trial.present();

            #Saving data
            my_data = stimulus_manager.last_stimulus_data();
            my_response = response_manager.last_response_data();
            RT = my_data.reaction_time();
            stim_onset = my_data.time();
            if (RT > 0) then
               Correct_Resp = 0;
            else
               Correct_Resp = 1;
            end;

            ofile1.print(string(block) + "\t");
            ofile1.print(string(total_stim_count) + "\t");
            ofile1.print(string(stim_onset) + "\t");
            ofile1.print(string(RT) + "\t");
            ofile1.print("0" + "\t");
            ofile1.print(string(Correct_Resp) + "\n");
         end;
         stim_count = stim_count + 1;
         total_stim_count = total_stim_count + 1;
      end;
      trial_count = trial_count + 1;
   end;
   block = block + 1;

   wait_interval (100);
   port.send_code(23);
end;

#close output file
ofile1.close();
```