Digital Semiology Manual

Short overview

Digital Semiology (DS) is a Python based research tool for interpreting, encoding and reporting epilepsy ictal semiology, recorded by video. DS cannot be used for detecting seizures and it assumes that seizures were detected by other means: manually or by algorithms. DS does not take into account EEG and, therefore, cannot be used solely for seizure classification. However, DS, if combined with EEG interpretation, can assist to classify seizures. The output of DS is a report saved in binary file, which can be opened using Python based program: unpack_ictal_episode.

In addition to basic Python, DS requires installation of WxPython. The program unpack_ictal_episode requires installation of numpy and MatPlotLib modules.

Getting Started

Installation (Windows)

1. Install anaconda from https://www.anaconda.com
2. Install WxPython.

   Write in Search: *anaconda prompt*. Open the prompt. In the prompt write *conda install wxpython* and press Enter.

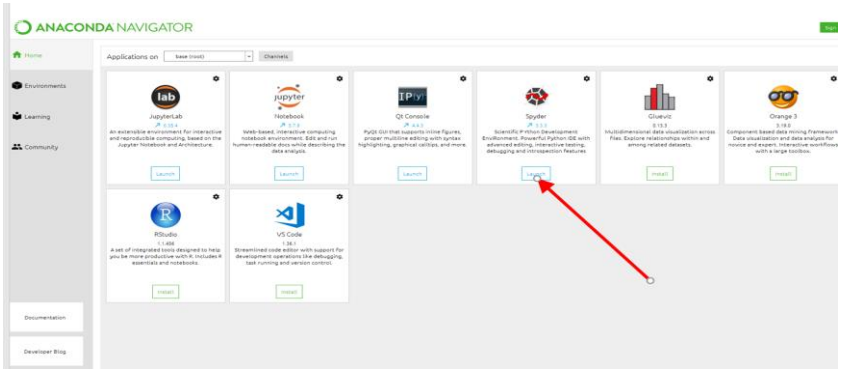
3. Installation Digital Semiology and unpack_ictal_episode.

   Place files Digital_Semiology.py and unpack_ictal_episode.py into folder:

   .spyder-py3. (usually: This PC > Windows(C:) > Users > name of the computer user > .spyder-py3)

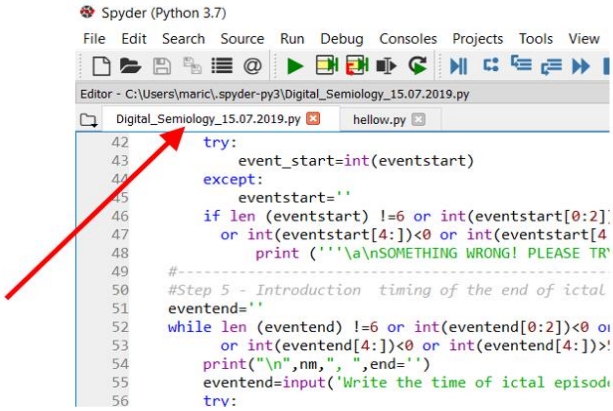
Start Digital Semiology

1. Write in Search: *anaconda navigator*.
2. Left double click on Anaconda Navigator App.
3. Left double click on the Launch button of Spyder – this will open Spyder environment.

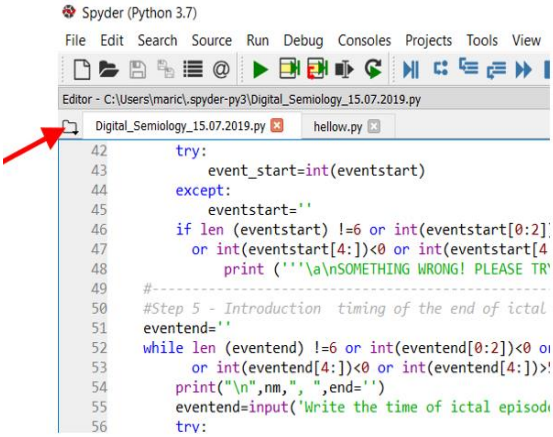


The details about Spyder can be found in https://www.spyder-ide.org

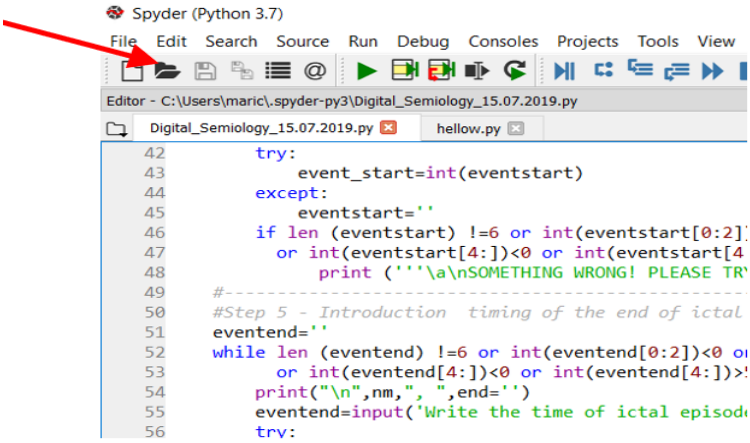
4. Check whether Digital_Semiology.py is listed in the Editor window.



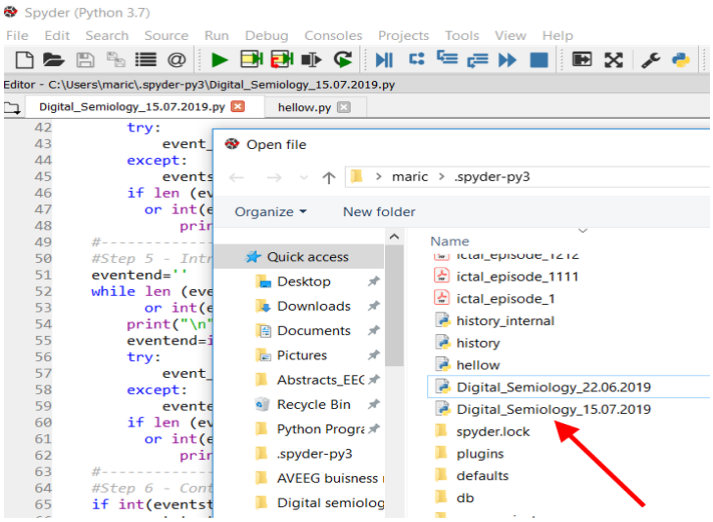
You can use the Browse tabs button in the Editor window – left single click.



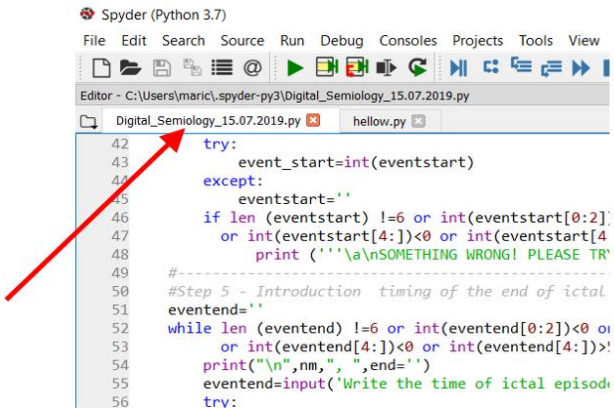
5. If not listed, left single click on the Open File button in the Spyder menu (black open folder in the left upper corner).



After the directory is open, left double click on the Digital_Semiolofy.py.

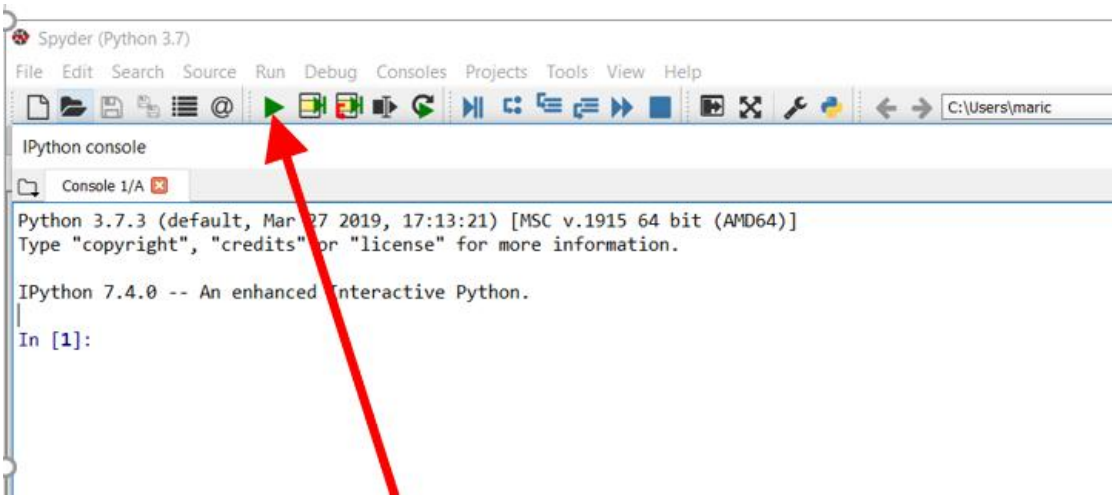


6. Left single click on the Digital_Semiology.py button in the upper part of the Editor window – the background of the button converted to white.

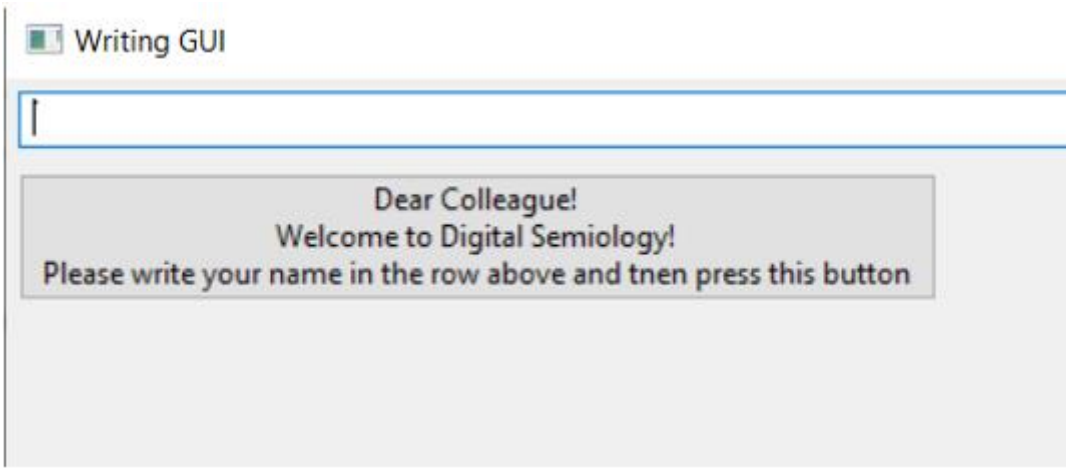


7. Since user can follow the DS performance through IPython console, we recommend enlarging IPython console by dragging its edges (simultaneously pressing the left mouse).

8. Start Digital Semiology by left single click on Run button (green triangle).



On the screen appears GUI:

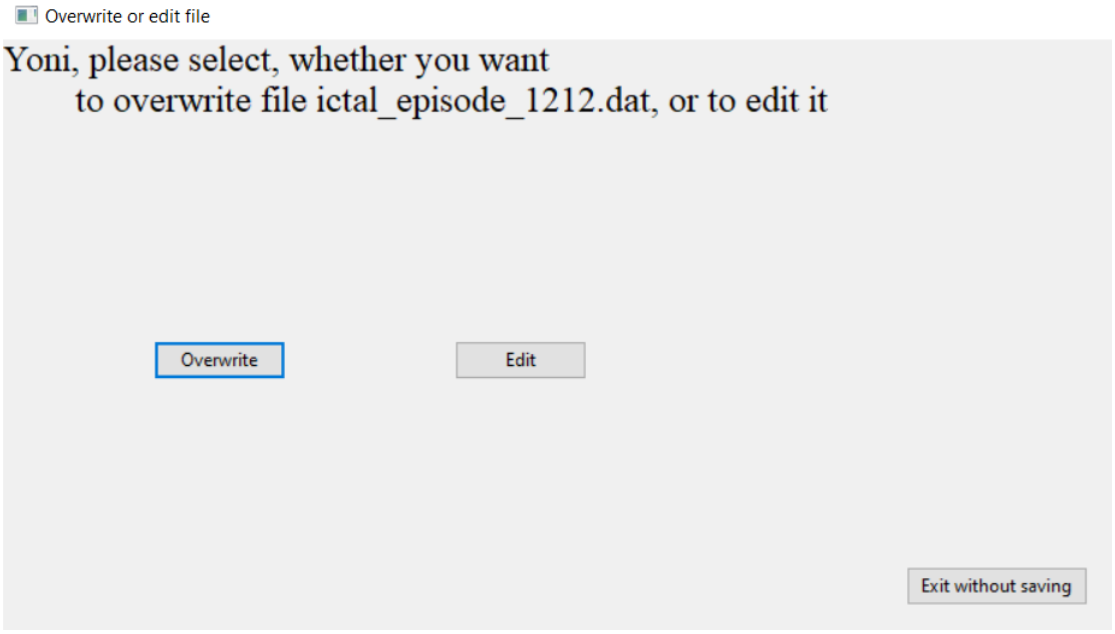


DIGITAL SEMIOLOGY: GENERAL ASPECTS

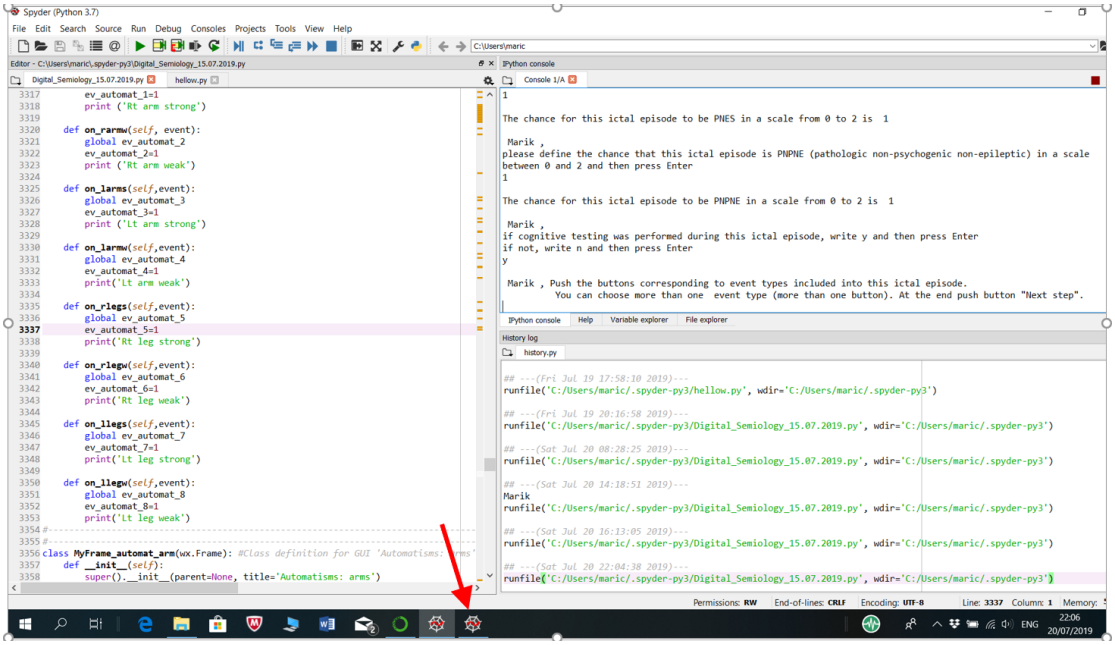
Digital semiology graphical user interface:

DS uses two types of GUI:

1) Writing GUI - an example you can see right above.
2) Selecting GUI – an example is right below:



When DS generates GUI, the second Spyder symbol appears on the lower part of the screen.



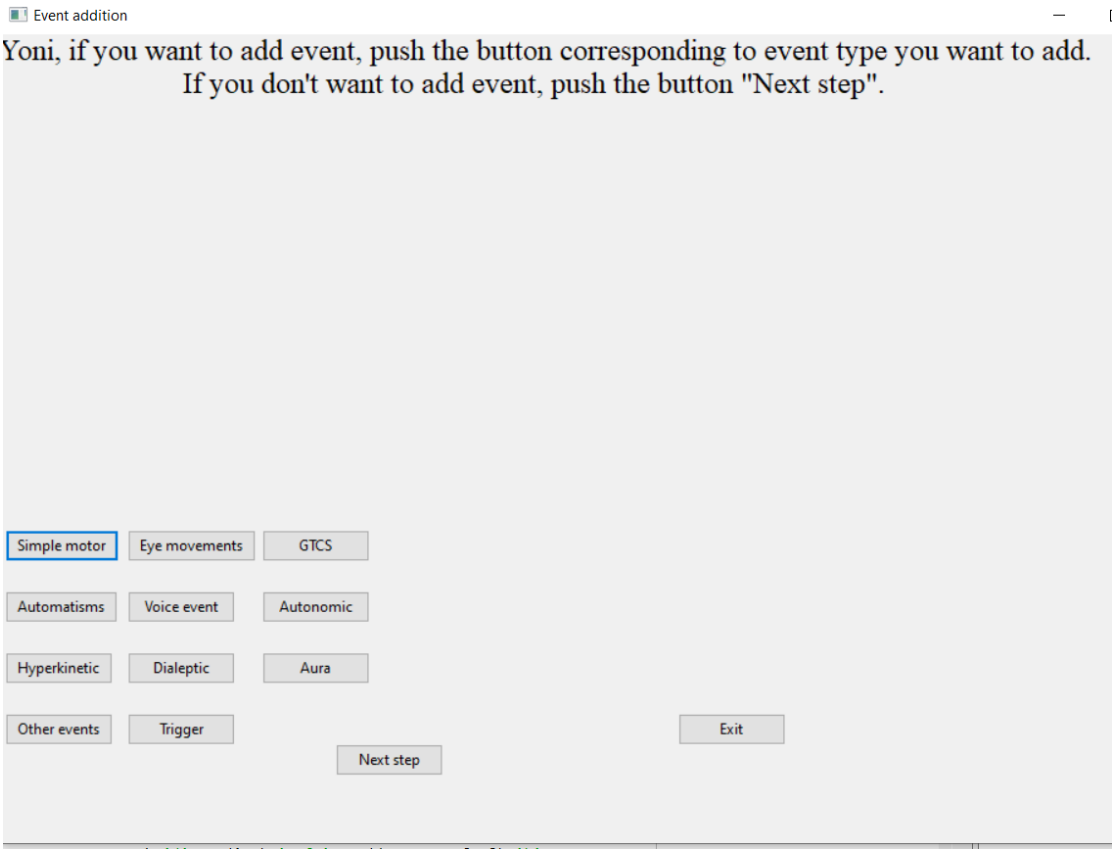
GUI either automatically appears on the screen or the user should click on the second Spyder symbol to make GUI visible.

Writing GUI

Writing GUI contains two parts: the row for writing and the button with instructions. User is expected to write something in the row (according to instructions on the button) and then press the button, which is located right below the row.

Selecting GUI

Selecting GUI presents several options, each corresponds to individual button. User can see appropriate instructions in the upper part of Selecting GUI. Often, Selecting GUI allows only one option selection and after pressing any button, the GUI disappears, and DS continues to the next step. In some cases, it is possible to press sequentially on several buttons, sometimes, however, not all combinations are allowed. Often the Selecting GUI has the button "Next step"; clicking on this button quits the GUI and DS continues to the next step.



With appearance of some Selecting GUIs, the short explanations of terminology appear in IPython Console (see the paragraph "Terminology explanation").

The role of IPython console in DS

In DS, IPython console serves for 5 purposes:

1. Displaying the user-software (DS) communication history during the session (see the paragraph "User-software communication history").
2. Presenting some explanations regarding the terminology used in GUIs (see the paragraph "Terminology explanation").
3. In the case when DS opens existing file, to see its content (see the paragraph "Files").
4. To view the incomplete (or complete) report during or at the end of the process of the ictal episode description (see the paragraph "DS Report").
5. One of the ways to write the comments (see the paragraph "Writing comments").
6. Presenting error message.

User-software (DS) communication history

All steps performed by DS and the user during encoding are logged in realtime in the IPython console and later saved at the end of the report file. The steps performed by DS are labeled SOFTWARE: … , and the steps of the user are labeled USER: … All steps

are presented in the chronological order and the timings of execution are also presented in IPython console.

Here is an example of User/Software communication history:

```
21:58:17
Software:
Dear Colleague!
Welcome to Digital Semiology!
Please write your name in the row above and tnen press this button

21:58:34
User typed:
Yoni

21:58:34
Software:
Yoni, please, write the CODE of ictal episode in the window above and then press this button

21:58:42
User typed:
1212

21:58:42
Software:
Yoni, please select, whether you want
        to overwrite file ictal_episode_1212.dat, or to edit it
```

In addition to communication timings, the date of the session is displayed in IPython at the start of Digital Semiology.

```
Software:
Today's date: 2020-09-23
```

The User/Software communication is saved in the variable software_user_dialogue. This variable is saved in the binary file (see the paragraph "Files").

Presenting error message

DS presents error message in IPython console. The content of error messages can be different. Here is an example:

```
Software:
SOMETHING WRONG! PLEASE TRY AGAIN!
```

Files

DS creates, updates or reads several files.

Ictal_epusode_ictal-episode-code.dat file (binary)

The main fail, which is created or updated is .dat binary file. This file includes three variables: 1) ds_starter; 2ictus; 2) report; 3) software_user_dialogue.

ds_starter is a variable, which includes general information about ictal episode (not including details of events)

ictus is a variable, which includes numerically encoded information about events in ictal episode.

report is another variable, which verbally describes an ictal episode (More details in the paragraph DS report.

software_user_dialogue is a variable, which includes information about all steps that user and software took during interpretation of ictal episode, including the date of the interpretation and timings of every user's or software's step. If user interpreted an ictal episode in several sessions, each new DS start is labeled by the date.

The name of .dat binary file is created as: ictal_episode_ictal-episode-code.dat. For example, if ictal episode code is 1212, the filename will be ictal_episode_1212.dat.

If .dat binary file of the ictal episode (with the code of this ictal episode) doesn't exist in the same directory as DS, DS will create this file. If exists – the user can choose whether overwrite this file or to edit it. In the case, when editing option was selected, all three variables of the file are updated. If overwriting option is selected, ictus and report are overwritten, while software_user_dialogue is updated. The editing and overwriting don't be executed, if during the exit from the program the user selects option "do not save".

Ictal_epusode_ictal-episode-code_draft.dat file (binary)

This is a back-up file, which is updated at the end of every event interpreting. In the case of incorrect exit from the program (e.g. forced shutdown), this file remains in the directory of DS. In the case of correct exit from the program, this file is deleted from the directory.

ictal_episode-code.txt file

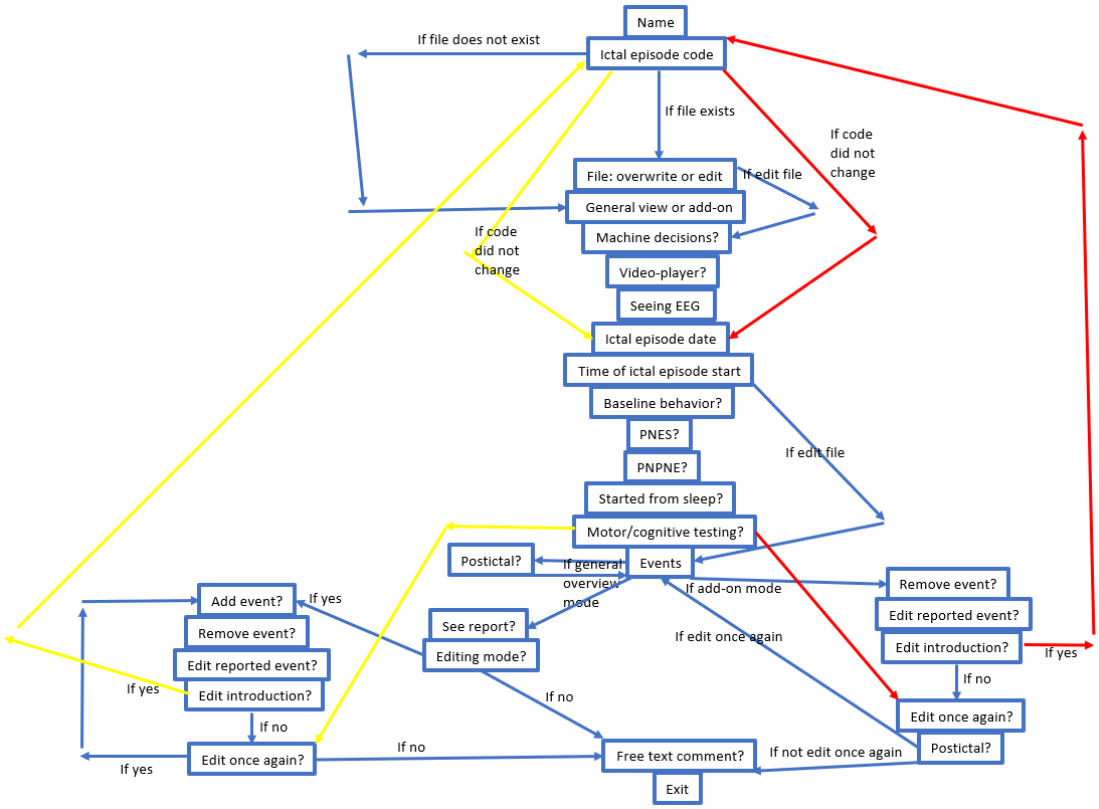
The content of .dat binary file can be written into .txt file using python code: unpack_ictal_episode (see separate paragraph about this code). The name of this .txt file is also composed as: ictal_episode_ictal-episode-code.txt. If ictal episode code is 1212, then the name of this file will be ictal_episode_1212.txt. We recommend placing the DS code and unpack_ictal_episode code in the same directory. In that case also .dat binary file and .txt file will be in the same directory.

video-player_go_no_go.txt and **intercode.txt**

These files serve for interactions between DS code and video-player code. They are created by DS and are in the same directory.

All files associated with DS are in the same directory with the DS code, except of the files, associated with video-player, which are discussed in the paragraph "Videoplayer".

DS workflow



It is practical to Divide DS workflow to three parts:

1. Introduction – everything before Events
2. Events
3. Editing ictal episode – everything after Events (not to miss with editing existing file).

OPERATING VIDEO PLAYER

Video Player Overview

The important function of DS is timing labelling of events during ictal episode. DS allows to do this in one of two ways: 1) manually and 2) using a video player interacting with DS. Video player has also the audio features, so it is more correctly to name it video-audio player.

The code supporting the video player is a separate phyton code: playing_a_video.py, which should be placed to the same directory with DS code. The communication between these two codes (Digital Semiology.py and playin_a_video.py) is done by writing and reading two text files by these codes:

1. video_player_go_no_go.txt
2. intercode.txt

Video player uses video files of the type .wmv. These files should not nessesarly be in the same file with DS. The path to .wmv-file should be pasted to appropriate place in the playing_a_video.py code.

Installing Video Player

Installing new packages in anaconda environment

1. Open Anaconda.

2. Select Environments in the left-hand pane (below home).

3. Just to the right of where you selected and below the "search environments" bar, you should see "base(root)" - Click on it.

4. A triangle pointing right should appear, click on it and then select "open terminal".

5. Use the regular pip install command here. There is no need to point to an environment/ path.

What to install

1. Install wxPython: pip install wxPython==4.0.7 (if not installed previously)

2. Install MplayerCtrl: pip install MplayerCtrl

3. Download MPlayer zip file, unzip after downloading, save in same file as other python codes.

OR: go to site: http://www.mplayerhq.hu/design7/news.html, download: MPlayer 1.4 HTTP (xz, 15 MB), unzip after downloading, save in same file as other python codes.

4. Place code playing_a_video in the same directory as other python codes.

Starting Video Player

Setting the path

At first the user should paste the path to .wmv-file in the code playing_a_video.py.

```
195 while read_video==  :
196     text_file=open("video-player_go_no_go.txt", "r", encoding='utf-8')
197     read_video=text_file.read(10)
198     text_file.close()
199     print('mobile perpetum')
200 if read_video=='0':
201     quit()
202 if read_video=='1':
203     # CHOOSE VIDEO HERE
204     myVideo=r"C:\Users\maric\Downloads\marik_newer\\"+ictal_episode_code+".wmv"
205 
206     app = wx.App(0)
207     frame = wx.Frame(None, size=(640, 480))
208     panel = TestPanel(frame)
209     frame.Show()
210     app.MainLoop()
211 
212 
```

The path should by pasted to line 204. In this example the path is:

C:\Users\maric\Downloads\marik_newer\\

In general form line 204 can be presented as:

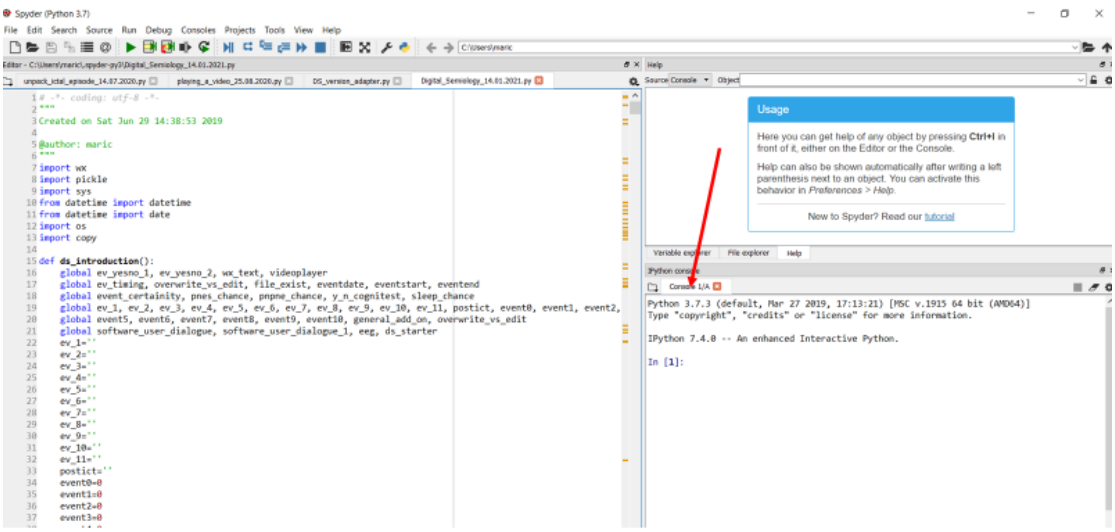
myVideo=r"path"+ictal_episode_code+".wmv"

In can be prudent to use all the time the same directory where the .wmv-files (video files) are placed, so that it will be no need to reset the path.

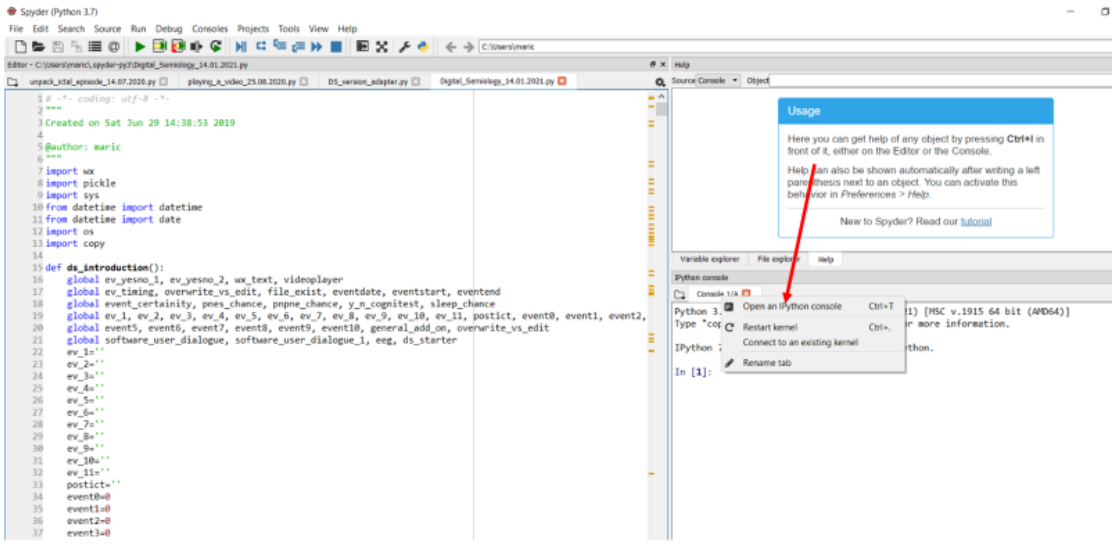
Opening video player

It is important to open video player in a terminal separate from DS code. It can be achieved by double click on the paying_a_video.py in the directory, where this file is located.

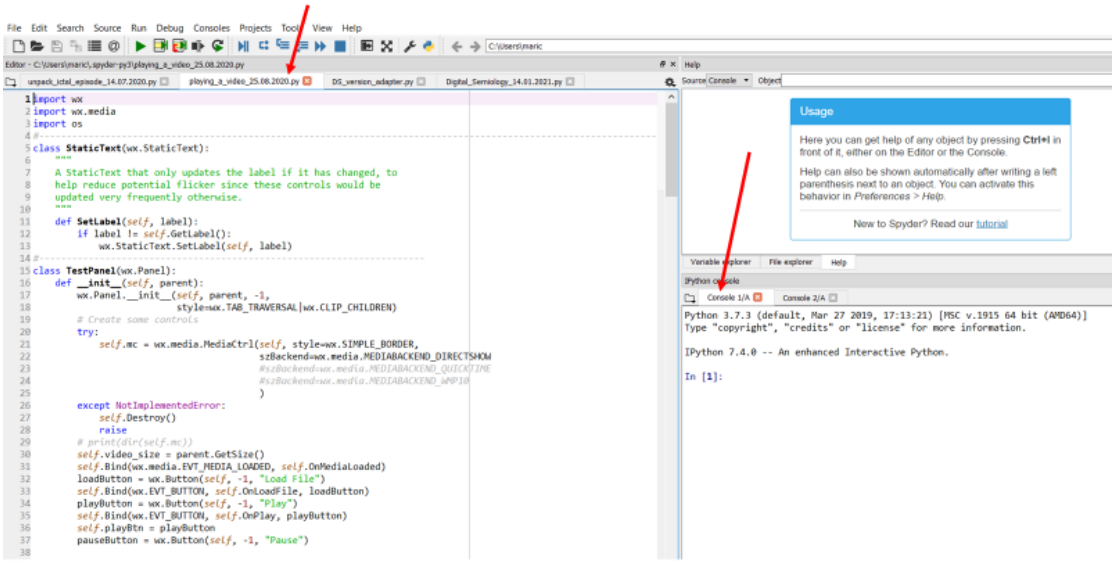
Another way is using user interface of Spyder. At first, the user should make single left click to the console window:



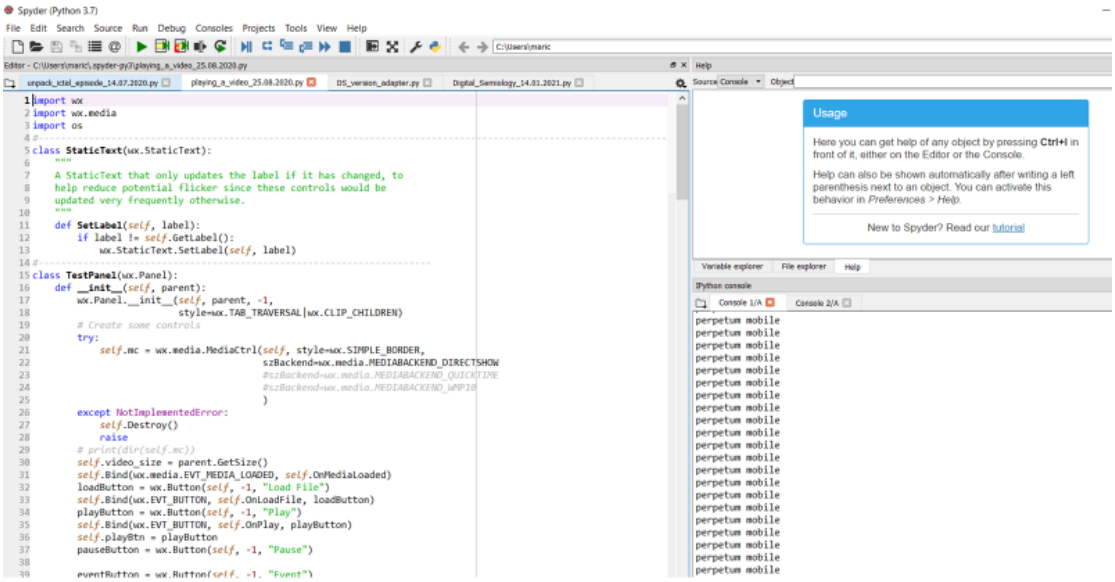
The small window will open. The user should select the uppermost option "open the IPhyton console":



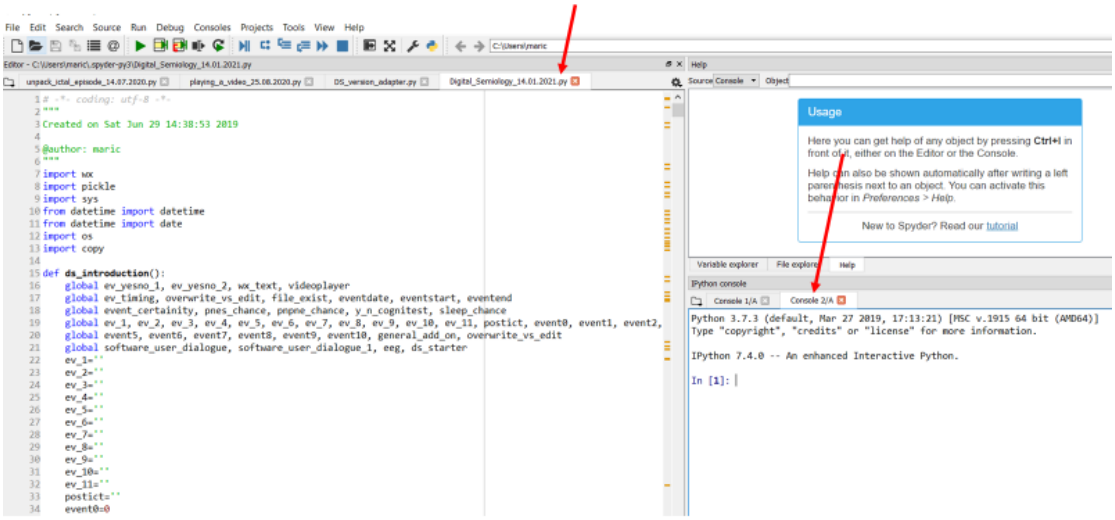
This will open the second console. User should make single click (left) on one of consoles window and to start run plaing_a_video code.



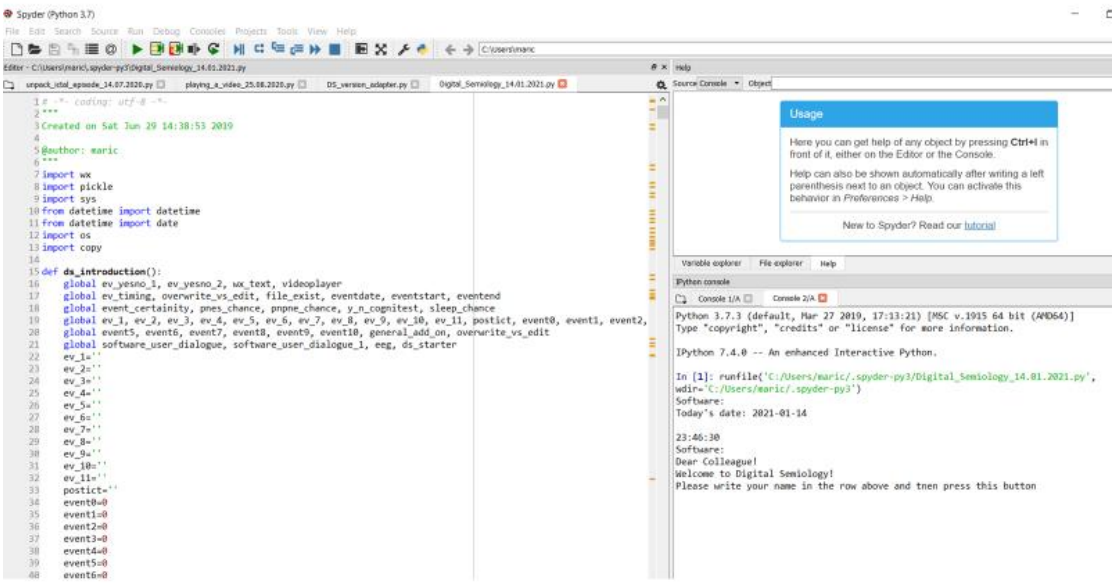
In the console running row "perpetum mobile" will appear.



Next, user should make single left click on the second consoles window and run digital semiology code:



The following text will appear in the IPhyton console:



At this step, both playing_a_video and Digital Semiology codes are running in the two separate terminals.

Setting the ictal episode code

Now user can open DS and proceed to the step of ictal episode code. In order to direct video player to correct video-file, the ictal episode code should be name of the video-file without the dilename extension (see paragraph Ictal Episode Code in the chapter INTRODUCTION).

Video player use confirmation

When user runs DS and approaches step of the decision to use video-payer or not, selecting option "Yes" will open video player and selecting option "No" will close the terminal of video player (see the paragraph Video Player: Yes or No in the chapter INTRODUCTION).

So, opening video player depends on following conditions:

1. Video player code should be in the same directory as DS code.
2. Correct path to video-file should be set in appropriate line in the video player code
3. The video player code should start run in the separate terminal.
4. The ictal episode code should be the name of video file (without extension).
5. When DS approaches the step of Video Player: Yes or No, user should select option "Yes".

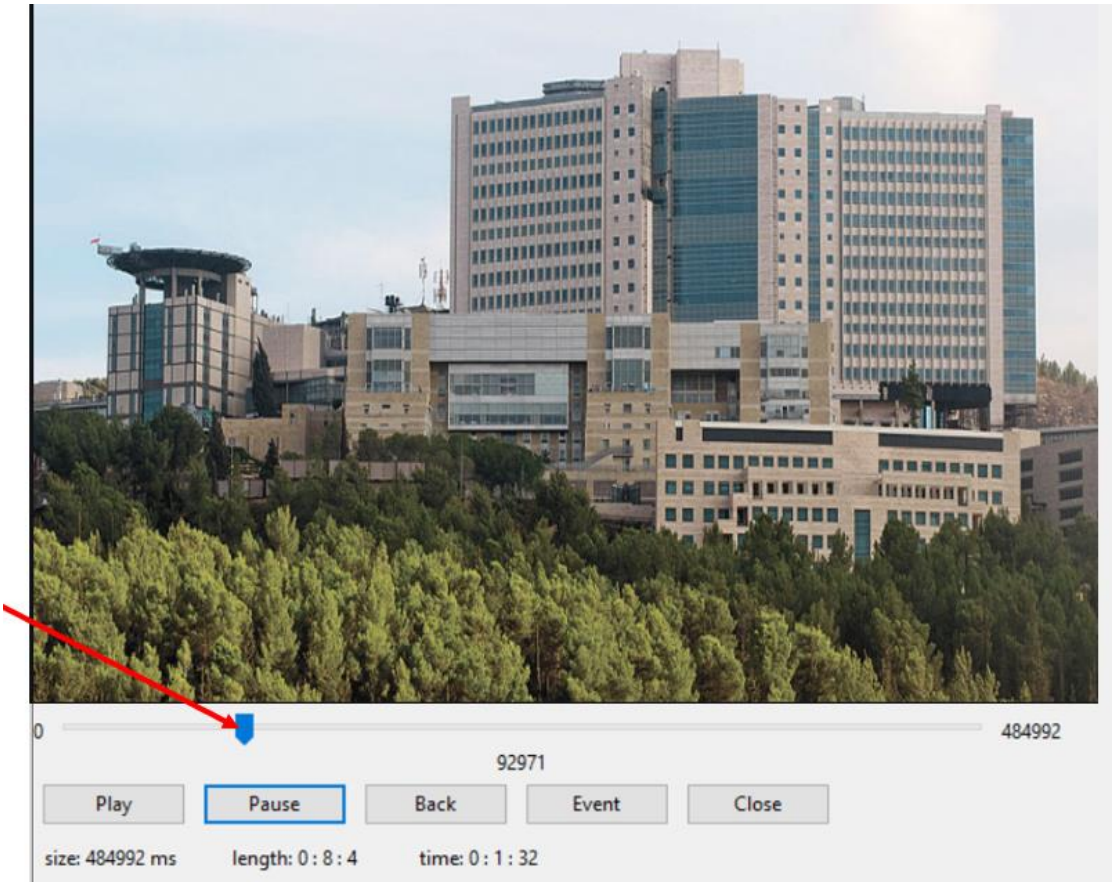
Using Video Player

To run video, user should press Play button, to stop video – Pause button. Pause button also serves for timing indication. Another way to indicate timing is to press Event button – in that case, video will not stop.

Pressing Back button returns video to beginning.

Pressing close button will close the video player and its terminal.

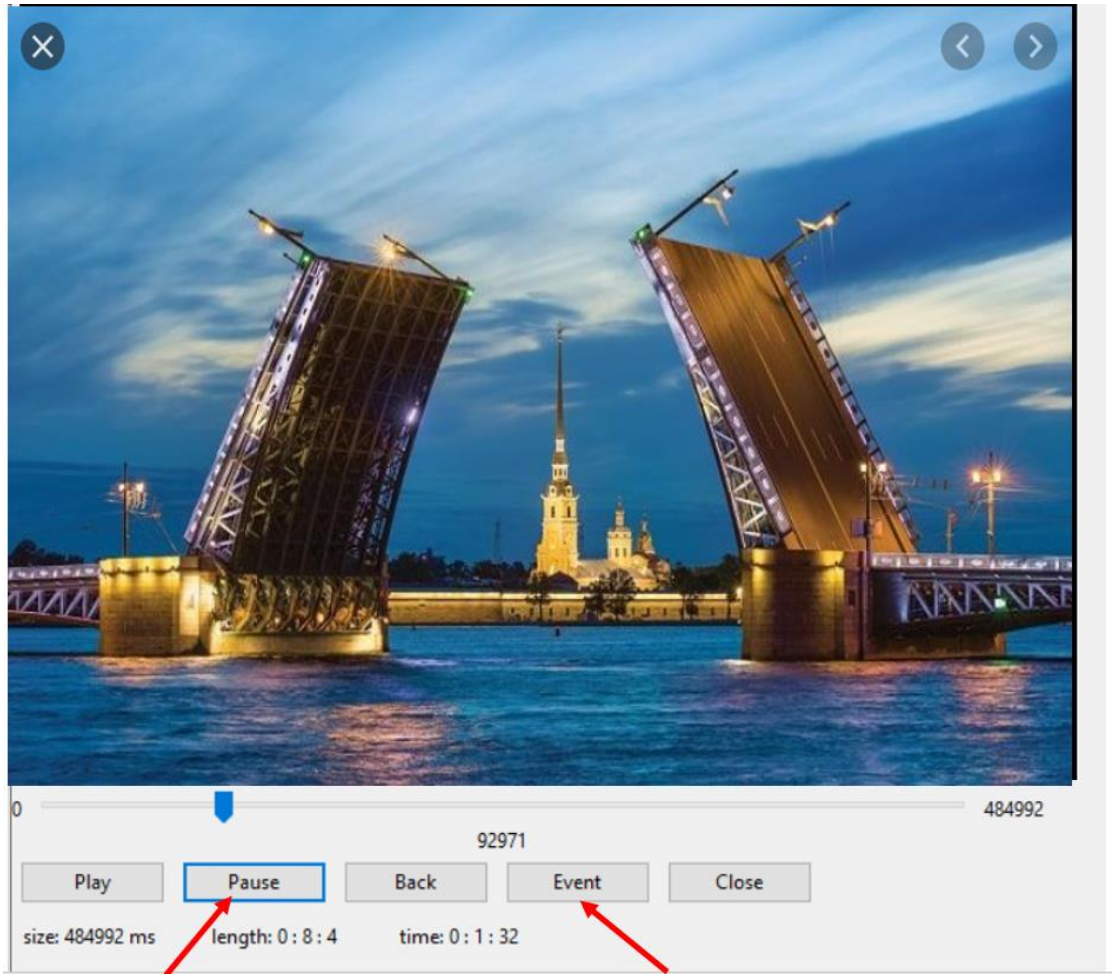
The running bar (labeled by red arrow) is moving when video runs. This running bar can be moved also manually, **the user, however, is asked to press Pause button before manually moving running bar. If running bar will be moved manually while video is running it will temporarily prevent video player functioning**.



The timescale of video player is started from 0. *Size* is number of milliseconds in the video (the same value is also displayed at the right end of running bar row). *Length* is hours : minutes : seconds in the video. *Time* is the timing of the running bar in hours : minutes : seconds (the number above the interval between Back and Event buttons is *time* in milliseconds).

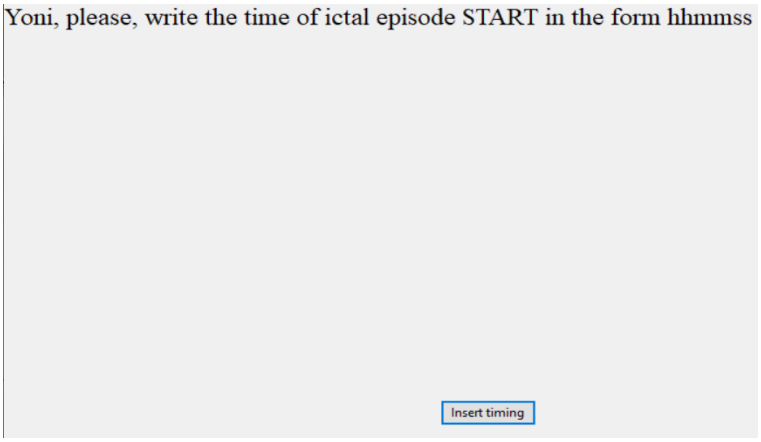
Timing Indication

In order to indicate timing, the user should press Pause button or Event button. If user presses Pause, the video will stop, if – Event – will not stop.

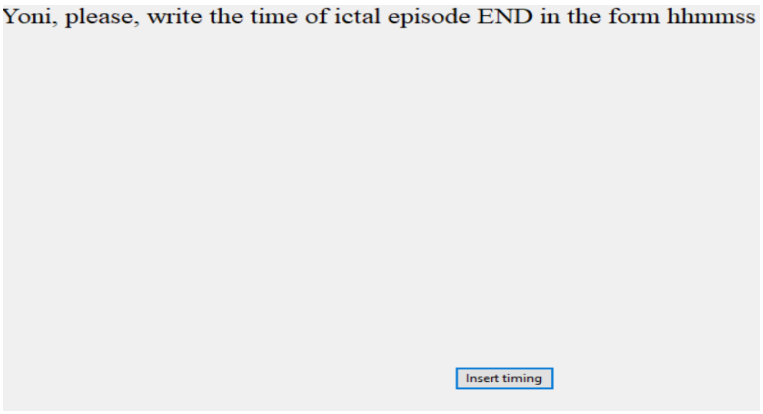


The timing indication by video player should be done when DS presents GUI asking to do that. In most cases DS asks to indicate timing of start or end of event or ictal episode. The exception is when DS asks to indicate the timing of generalized tonic clonic seizure transformation from tonic phase to clonic (see details in the paragraph Generalized Tonic Clonic Seizures in the chapter EVENTS).

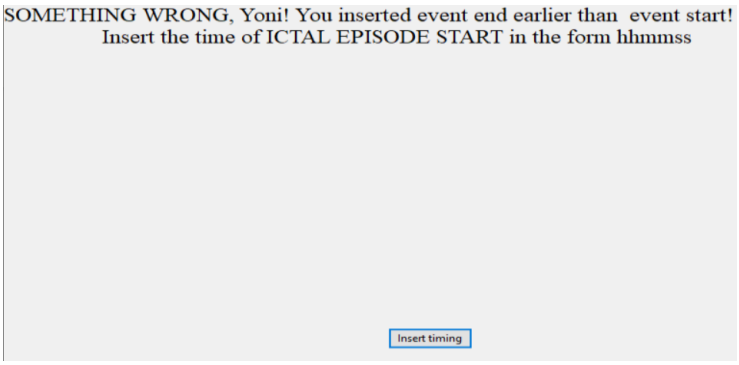
Here is the GUI asking user to indicate the timing of ictal episode start. User should first press Pause or Event on video player and then press Insert timing button on DS GUI.

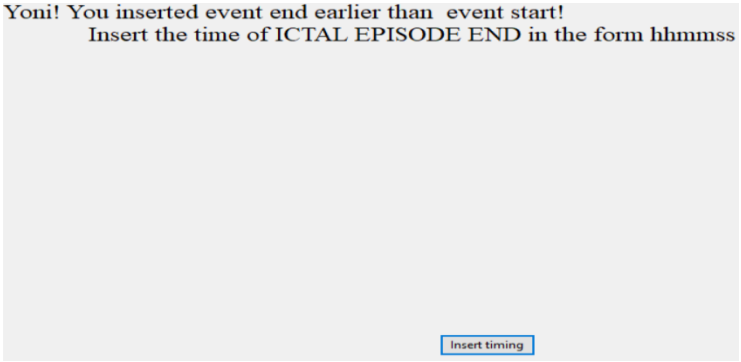


Here is the GUI asking to indicate ictal episode end. The principle is the same as for ictal episode start.



If ictal episode (or event) start is indicated later than end, DS will again ask user to re-indicate both start and end.



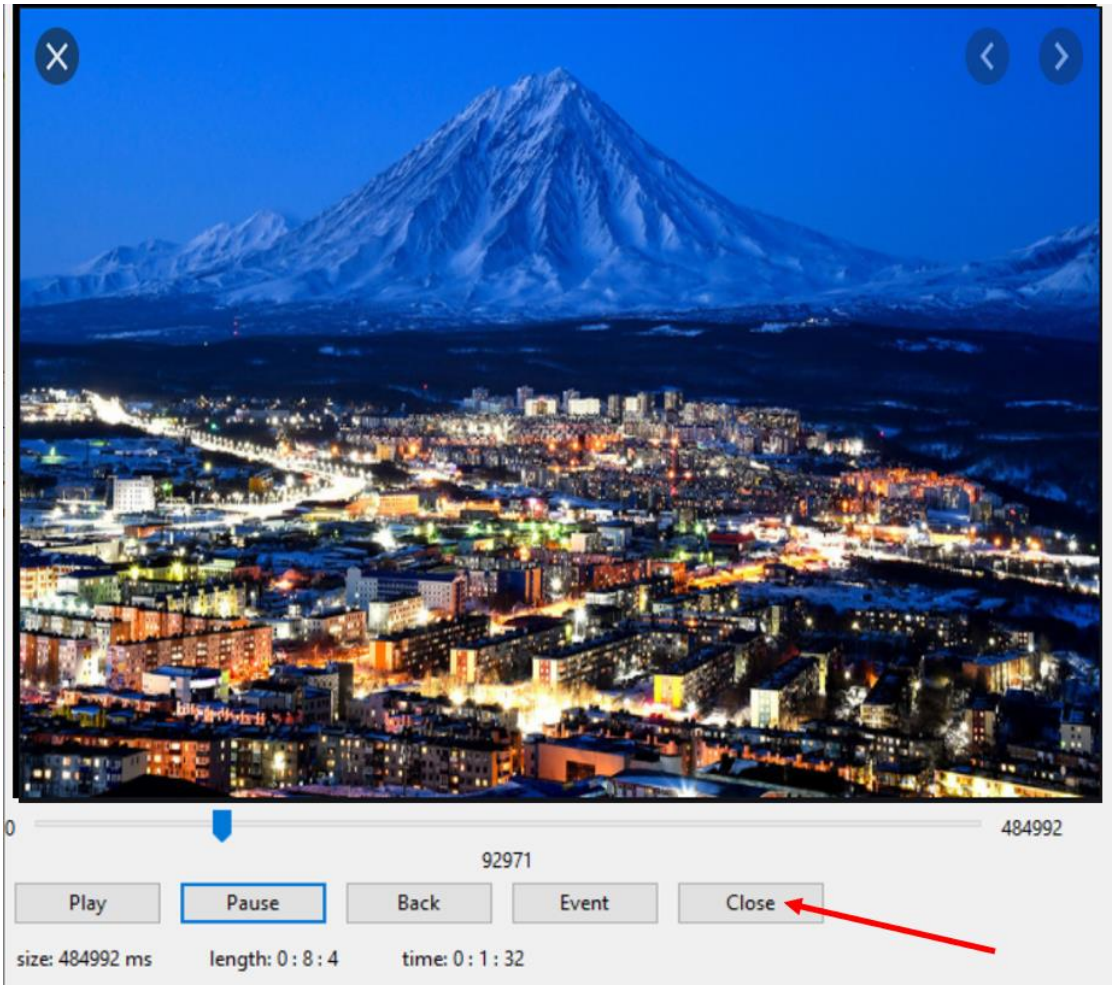


So, timing indication using video player has two requirements:

1. The timing should be indicated by pressing Pause or Event in video player at appropriate time and after that...
2. In the DS GUI the button Insert timing should be pressed.

Closing Video Player

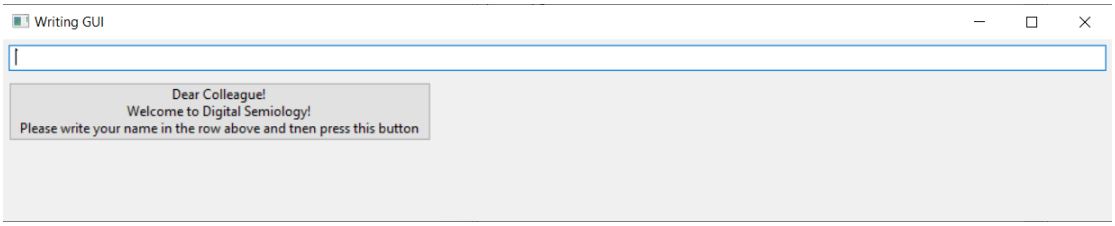
In order to close the video player, user should press the Close button – this will close both the video player and its terminal.



INTRODUCTION PART OF DS

Name of the User

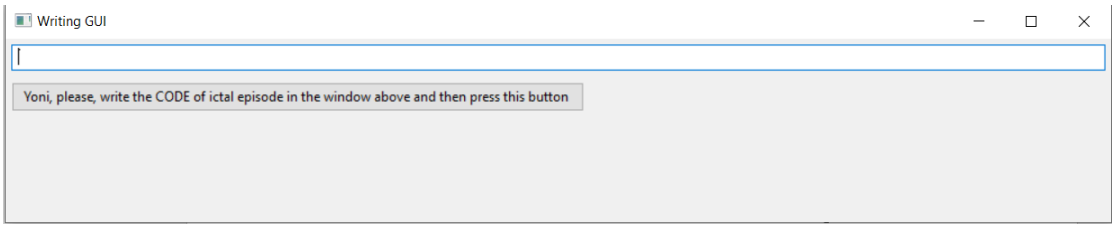
With opening DS, software asks the user to write her/his name in a writing GUI.



Letters, numbers, command punctuation characters (such as commas) can be used, but not empty spaces. Specifying user name is mandatory and progression to the next step will only be made after the user enters her/his name. When the name is written and writing GUI button is pressed, DS follows to the next step: writing the code of ictal episode.

Ictal Episode Code

At the next step, DS asks the user to write ictal episode code in another writing GUI.



Also, here, in principle, every letter(s), every number(s) or every sign will be accepted, except forempty spaces.

When using the video-player, the ictal episode code should be the same as the name of the video-file (without filename extension). For example, if the name of video-file is 1212.wmv, the ictal episode code should be 1212 and the user should write 1212 in the row of writing GUI and then to press button.

When a .dat binary file with the same name exists in the same directory with DS code, DS will proceed to the next step, which is Overwrite or Edit.

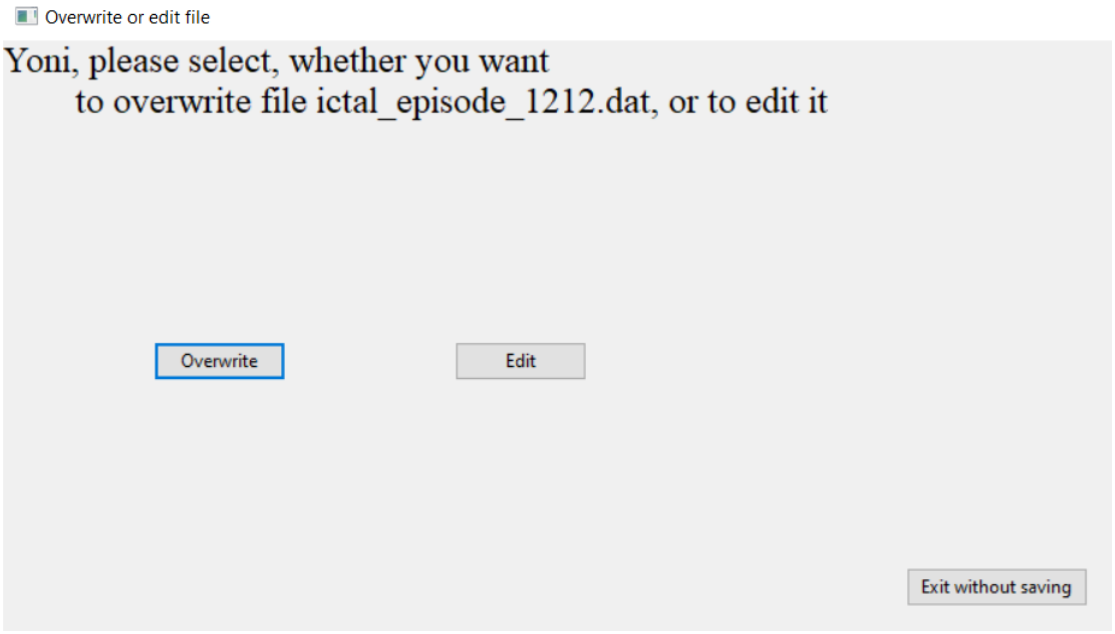
If the .dat binary file with the same name does not exist in the same directory with DS code, the software will write in IPython console the following:

"The file ictal_episode_1212.dat does not exist"

In this case DS will skip the step of Overwrite or Edit and will directly proceed to the the next step: General Overview or Add-on Mode.

Overwrite or Edit

This step opens only if the .dat binary file with the same name is present in the same directory as DS code. Otherwise, DS bypassing this step. At this step selecting GUI is opening.



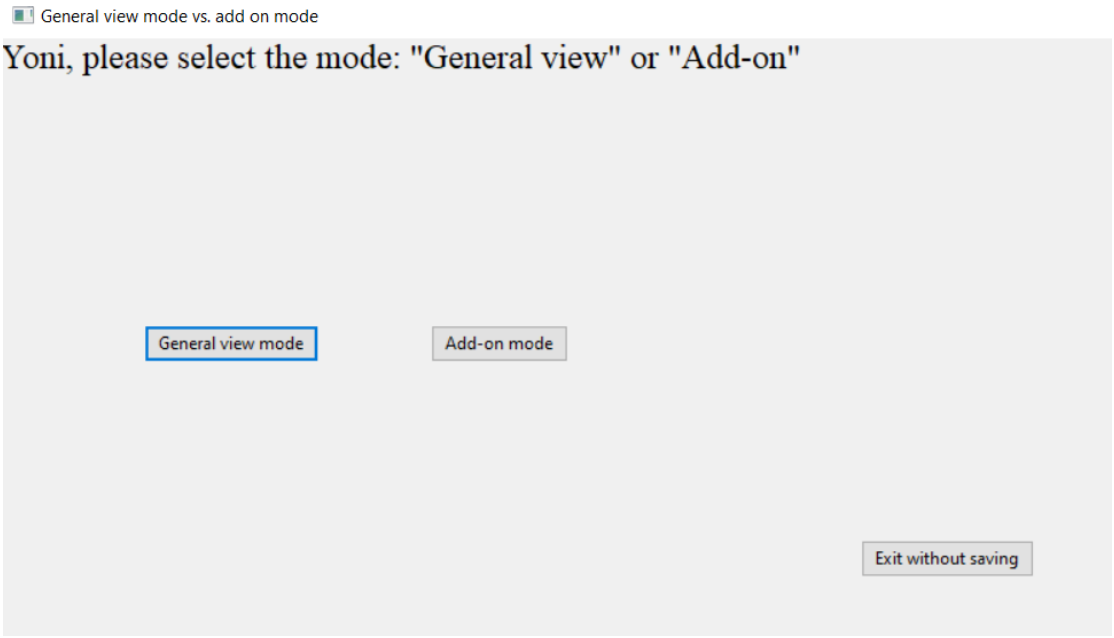
The user is asked to choose whether she/he wants to overwrite or to edit the existing .dat binary file. In the case of selection of Overwrite option, the ictus and report will be rewritten and the old information in these variables (ictus and report) will be replaced by new information. The software_user_dialogue variable will be not rewritten, rather updated and will contain both old and new information. If Edit option is selected, no variable in the .dat binary file will be overwrite – all three variables (ictus, report and software_user_dialogue) will be updated and will contain both old and new information.

If Overwrite is selected, then DS will subsequently go through all steps of Introduction, whereas, if Edit is selected, DS will bypass part of Introduction (see DS workflow).

If Overwrite option is selected, DS will continue to the next step: General View or Add-on Mode. If Edit option is selected, DS will bypass General View or Add-on Mode and will proceed directly to Machine Decisions step and the software will enter the Add-on mode.

General View or Add-on Mode

DS opens this step if .dat binary file (with the same name) doesn't exist or if the file exist, but the user selected Overwrite option. If file exists and user selected Edit option at previous stage, DS enters Add-on mode and doesn't provide opportunity to select General View mode. At this stage the selecting GUI appears.



The user is offered to choose one out of two operation modes: General Overview or Add-on.

The difference between these two modes is at first seen, when DS come to the step of Events and later during the Ictal Episode Editing. During Introduction there is no difference between these two modes.

General View mode allows user to describe events in two stages. At first, the user define what types of events are recorded during whole ictal episode and how many events of each type were recorded. Next, DS opens to user each event in sequence for the detailed description.

Add-on mode allows user open separate event, describe it (without listening all events), then open next one and so on, until the user describes that all events are already described.

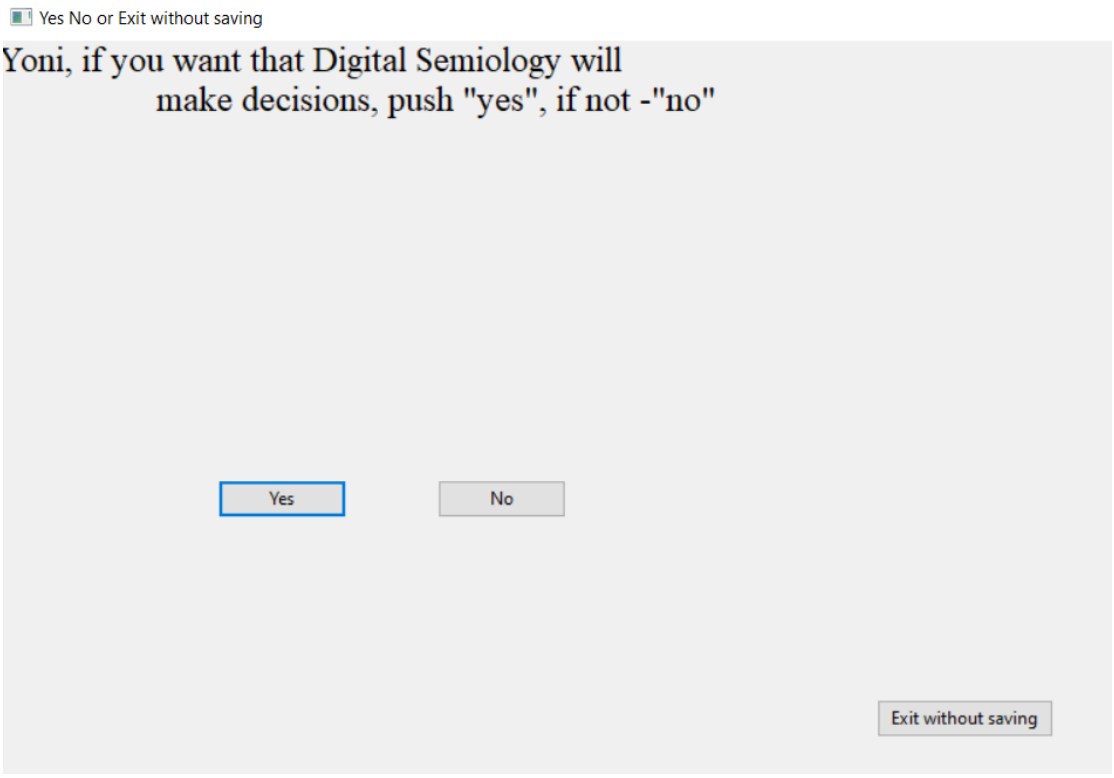
According the authors experience with DS, in most situations, Add-on mode is more practical and is generally recommended as a main operating mode.

More details about these two modes of events description can be found in the chapter EVENTS.

After the operating mode is selected by user, DS proceeds to the next step, which is Machine Decisions: yes or no?

Machine Decisions: Yes or No?

This is an obligatory step which is included in all pathways in DS. Here DS presents to user the selecting GUI dealing with machine decisions.



If user selects Yes at this step, DS will make and report decisions regarding focality of the ictal episode, presence of motor Jacsonian march and PNES features. If the selection is No – DS will not make such decisions. Then DS pass to next step which is Video Player: yes or no?

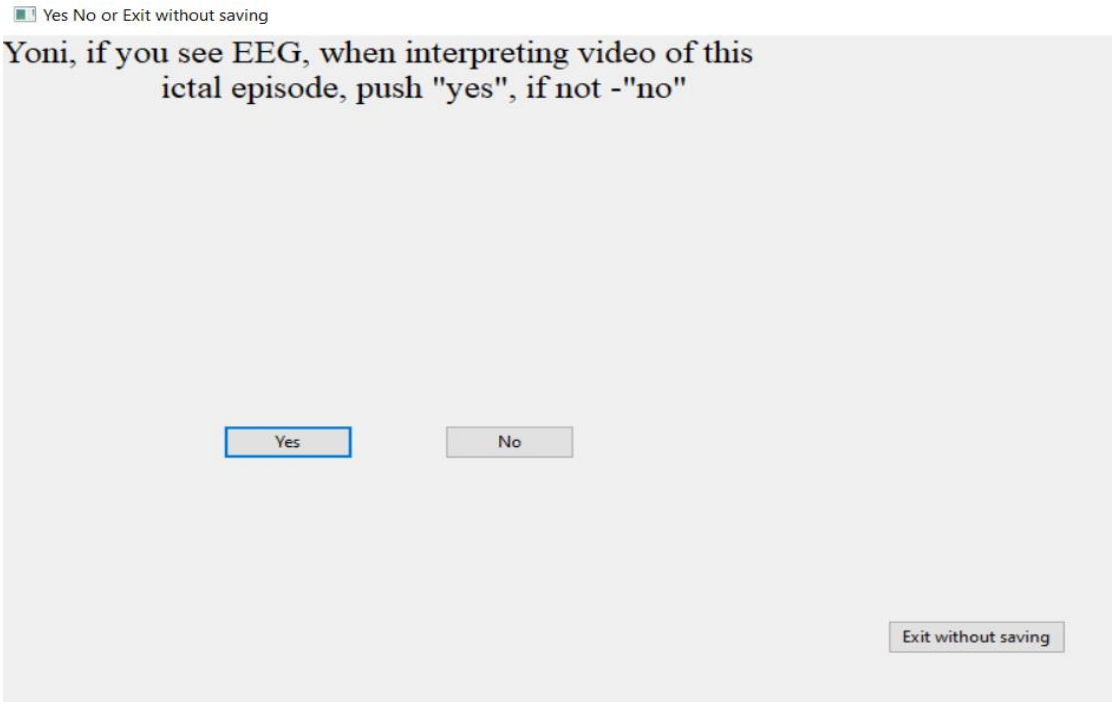
Video Player: Yes or No?

This step is also obligatory (included in all pathways of DS) as a previous one. At this step user selects whether she/he will use DS-integrated video player during ictal episode analysis. DS presents at this stage appropriate selecting GUI.

If user selects No, video player does not open, and the user should label timings of events manually. If user selects Yes, DS opens video player, but this requires several conditions (see in chapter OPERATING VIDEO PLAYER).

Looking at EEG, when interpreting video

At this step the DS user is asked: whether she/he see EEG during video interpreting.

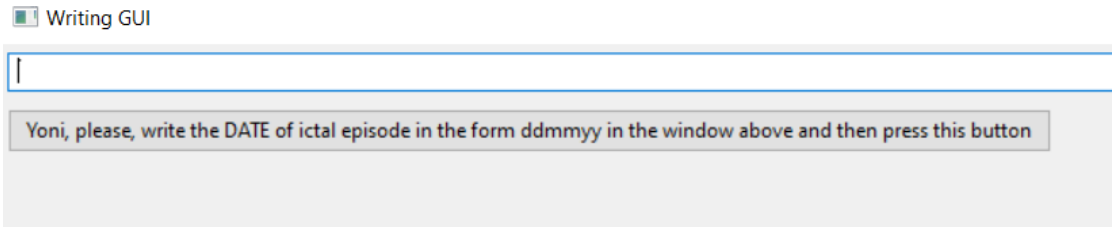


The next steps of DS depend on two conditions: 1. Whether the .dat file is new and 2. What was the user selected at one of the previous steps: Overwrite or Edit.

After the user selected Yes or No at this step, if it is new .dat binary file, or if option Overwrite (not Edit) was selected, DS continues to the next step, which is Ictal Episode Date. If it is an existing .dat binary file and option Edit was selected, DS from this point will bypass several steps of Introduction and will continue directly to Events.

Ictal Episode Date

At this step, DS opens writing GUI, asking user to write the date of ictal episode in the form ddmmyy.



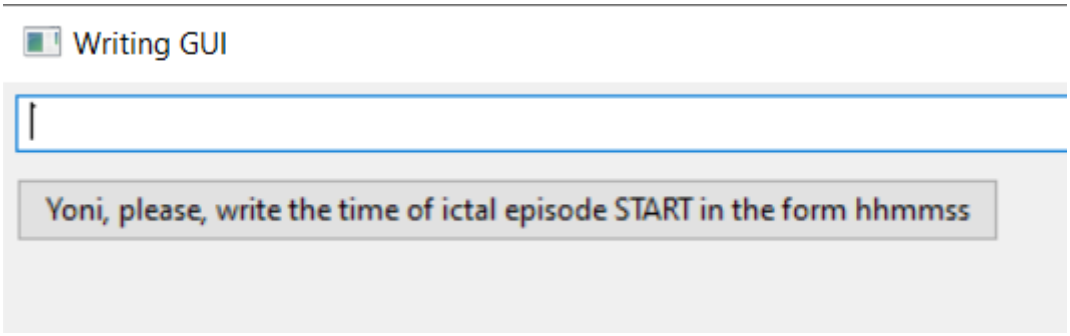
Later, in report the date will be converted to dd.mm.20yy. For example, if user write 300920, DS will generate date 30.09.2020 in the report.

If user will write the combination of numbers, which cannot be converted to the date, such as 340811 (34.08.2011) or 122520 (12.25.2020), DS will not continue to further step, rather write in IPhyton console the error message and will again present to user the same writing GUI until reasonable date will be written.

After this step is correctly completed, DS proceeds to next two sequential steps which are Timings of Ictal Episode Start and End.

Timings of Ictal Episode Start

This step is different with and without video player. When user doesn't employ video player, she/he should manually write the timing in writing GUI in the form ssmmhh. Then DS converts this to ss:mm:hh. For example, if user wrote 013305, DS will write in report 01:33:05.



Using video player, it is no need to write timing (for details see chapter OPERATING VIDEO PLAYER).

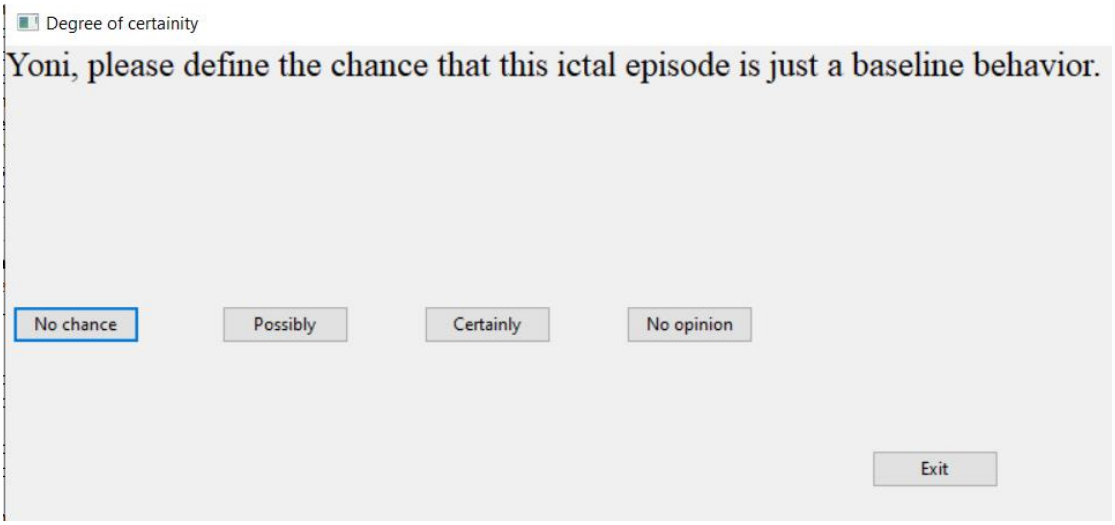
If user will write number combination, which cannot be converted to reasonable timing, error message will appear in IPython console and DS will again open the writing GUI, asking to write the reasonable timing.

After completing these steps, DS will ask user several questions regarding the general characteristics of ictal episode.

Baseline Behavior?

DS asks the user, whether this ictal episode can be just a baseline behavior.

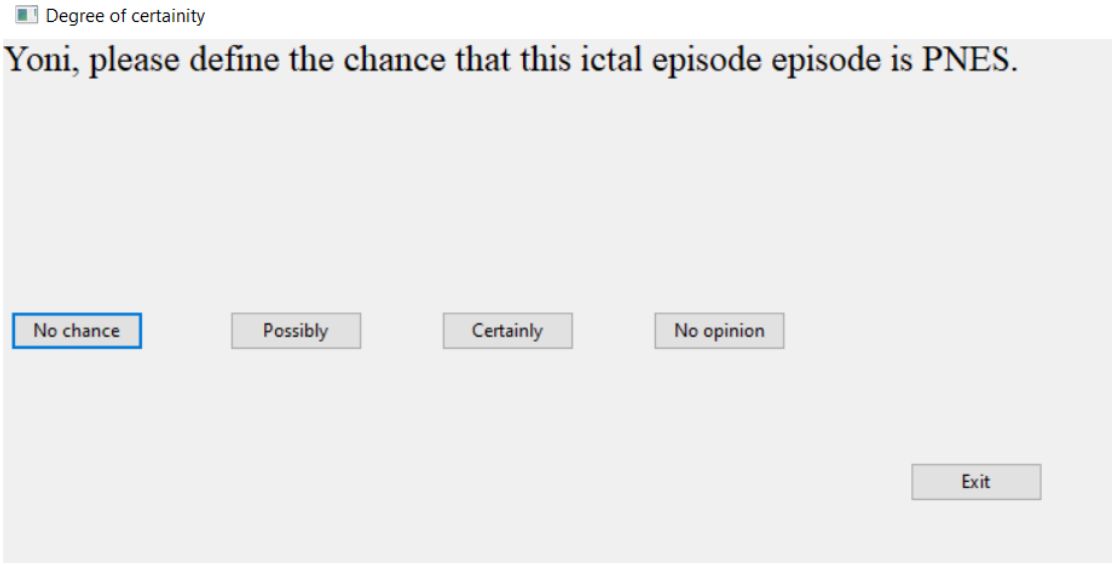
DS opens a selecting GUI and gives four options: "No chance", "Possibly", "Certainly" and "No opinion".



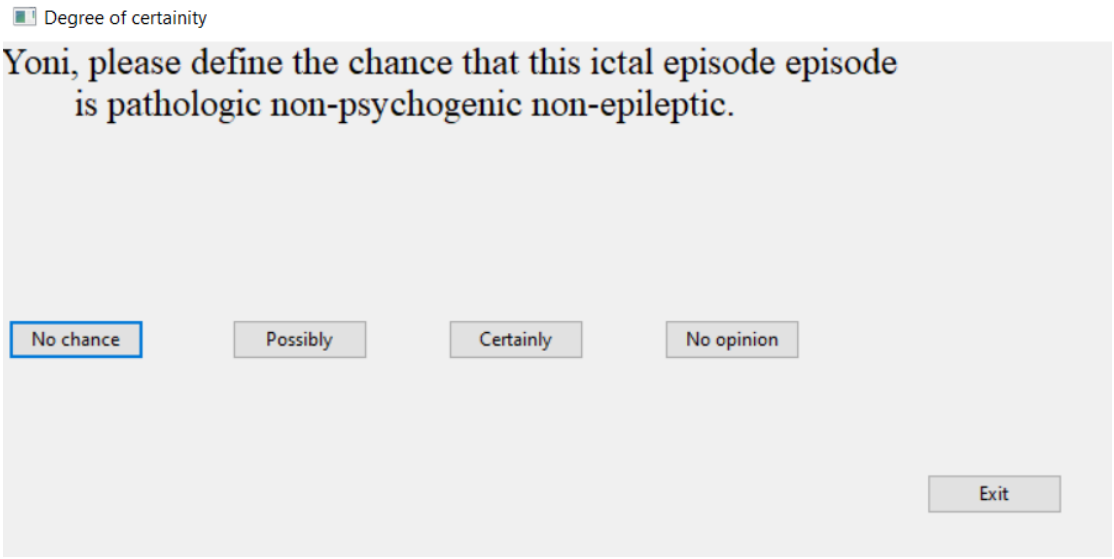
After user answered the question, DS continues to the next question: PNES?

PNES?

PNES is psychogenic non-epileptic seizure. This step is similar to previous one. Here is the GUI:



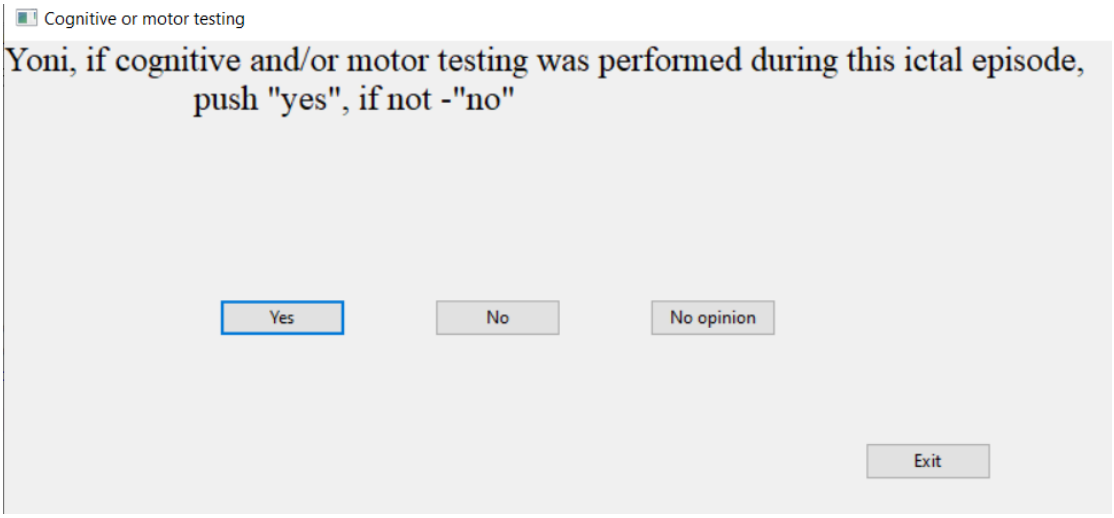
Pathologic non-psychogenic non-epileptic?



Whether ictal episode started from sleep?

Cognitive/motor Testing During Ictal Episode?

In contrast to previous steps, the user is offered to answer this question in the form Yes or No (or No opinion).



After this step, DS proceeds to Events. However, here we will discuss now how to operate video player and thereafter we will continue to Events.

Hierarchical description of ictal semiology

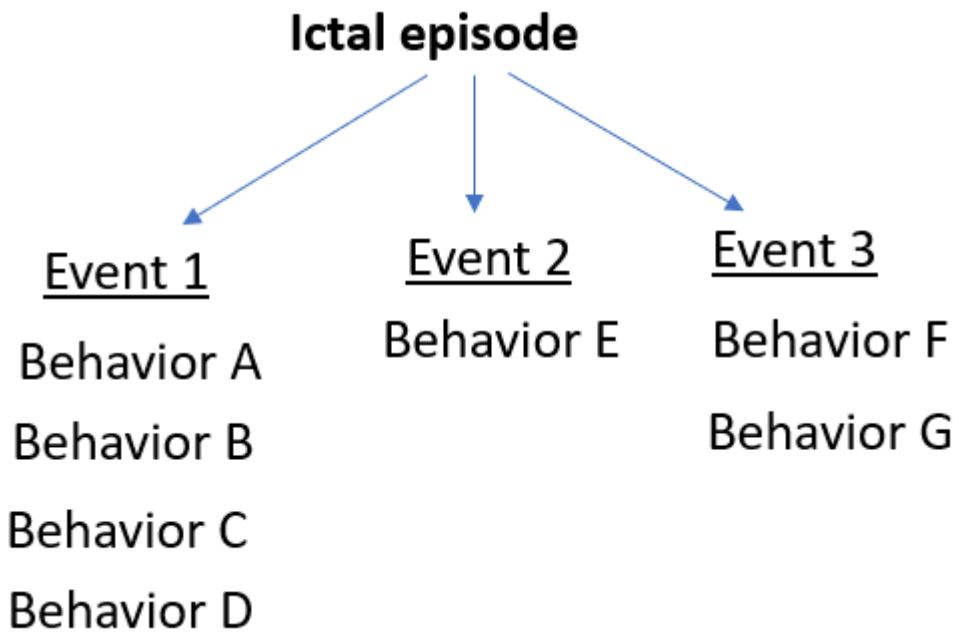
DS describes ictal episode in hierarchical way. There are 3 hierarchical categories:

1. ictal episode
2. ictal events
3. ictal behaviors

The first hierarchical category **- ictal episode** itself is characterized by the timings of its beginning and the end, separating it from baseline behaviors and postictal phenomena.

The second hierarchical category **- ictal events** are discreet behavioral elements of ictal episode. The ictal events are categorized as simple motions, automatisms, hyper-motor phenomena and others. Every ictal event has its own start and end timings.

The third (and the lowest) hierarchical category is **ictal behaviors**. Ictal behaviors are the elements of ictal events and have the same start and end timings as the ictal event. In other words, ictal behaviors are elementary semiology entities, which occur simultaneously and belong to the same ictal event type. For example, the automatic ictal event can include simultaneously occurring oral automatic behavior and hand automatic behavior.



EVENTS

Events: general aspects

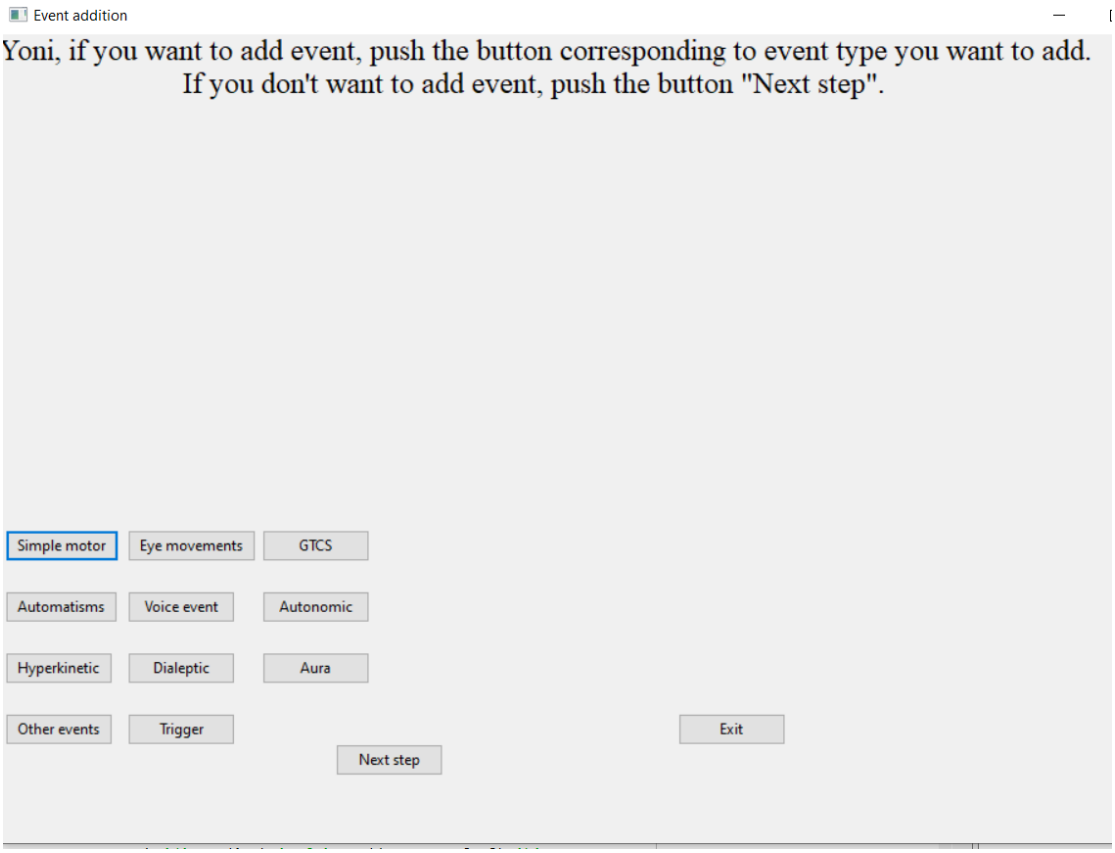
Events are elements of ictal episode, which have its own timing. DS uses following classification of events:

1. Simple motor events
2. Automatisms
3. Hypermotor events
4. Eye movement events
5. Voice events
6. Dialeptic events
7. Autonomic events
8. Generalized Tonic Clonic events
9. Aura reporting events
10. Other events
11. Trigger events

In addition to its own timing, events can have their characteristics: behaviors. For example, simple motor event may include behaviors: tonic position of left arm and clonic movement of right arm. DS has different ways to describe events: one is add-on mode, and another is general view mode.

Events: add-on mode

In add-on mode, user should choose only one event, describe it and then chose another event (in contrast to general view mode, when user choose several events at once). In add-on mode the events' GUI includes buttons, which correspond to different types of events, Next Step button and Exit button.



Pressing one of the event type buttons leads to opening GUI of timing indication (event start and after that – event end) and next – GUI of behaviors related to selected event type. After user finalized describe the event, DS returns to events' GUI.

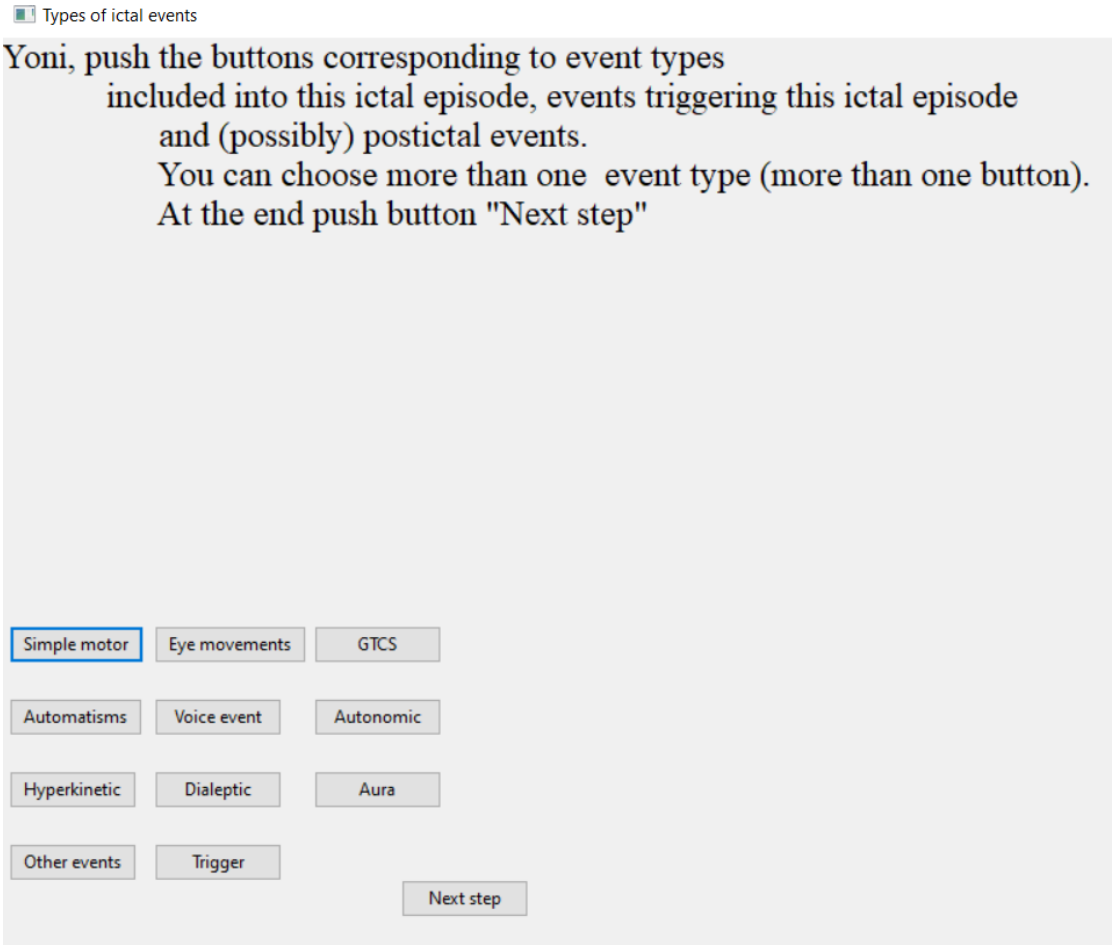
Pressing Next Step button leads to proceeding to editing ictal episode part of DS.

Pressing Exit button initiates the process of exit from DS.

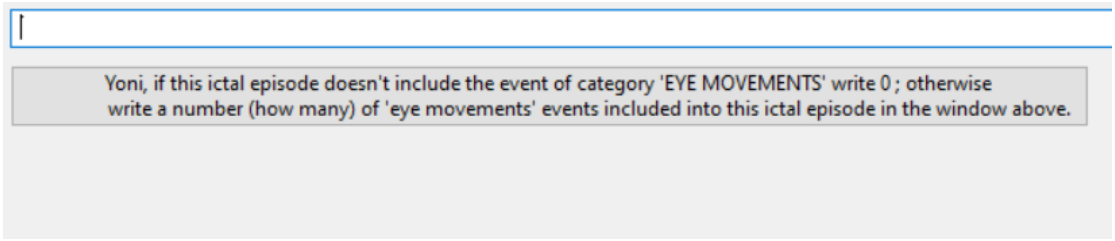
Events: general view mode

In general view mode, user can first select several events, then indicate how many events of a given type are included into the ictal episode, and next to describe each event separately.

The events' GUI in general view mode is similar to one in the add-on mode except of absence of Exit button.



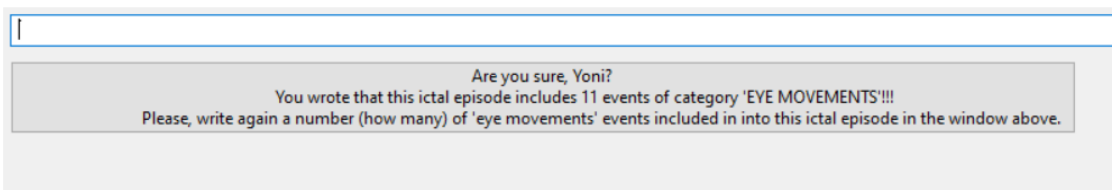
If user selects event(s) and press Next step button, DS will open writing GUI asking user to write how many events of given type are included in this ictal episode.



DS will continue to open this GUI for all types of events that user selected.

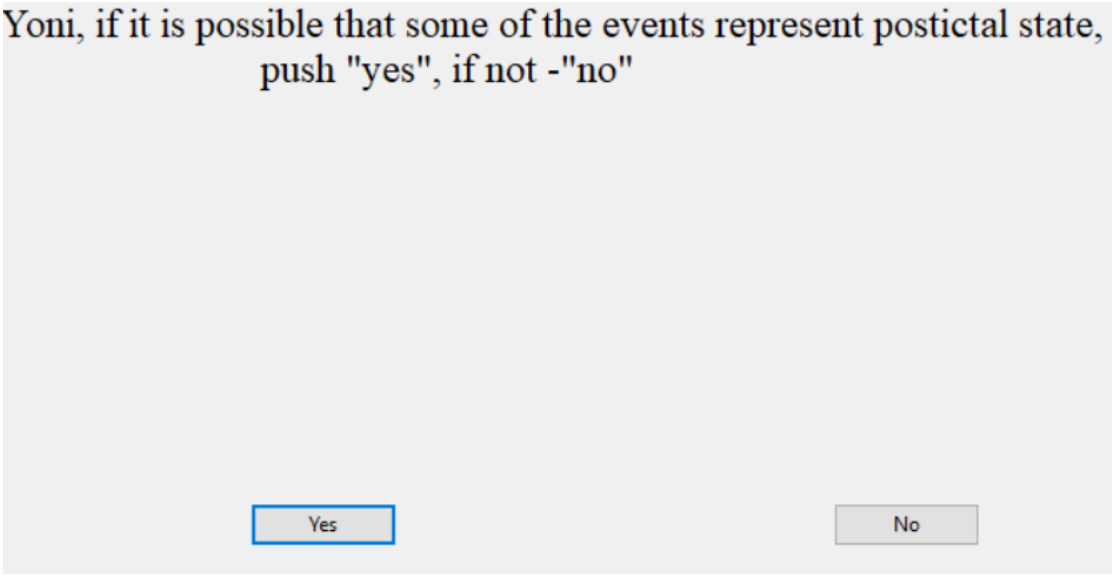
If user writes 0, DS will not open this type of event for description.

If user writes more than 10 events, then DS will warn her/him that he indicated to many events and will ask her/him to write again the number of events of given type.



Then DS will accept any number of events that user will write.

Next DS asks the user, whether it is possible that some events are postictal.



After user selected the answer Yes or No, DS opens sequentially GUIs of all events that user previously defined.

Abbereviations and terms used in DS

With the appearance of some Selecting GUIs, DS presents some terminology/abbreviations clarifications in IPython console.

DS uses several abbreviations:

PNES – psychogenic non-epileptic seizure (line 477).

PNPNE – pathologic non-psychogenic non-epileptic (line 503).

Polit-fist – politician's fist (line 1630 and line 1741).

Foreign language – the patient speaks, but the interpreter does not understand the language (line 5345).

PMLS – periodic limb movements of sleep (line 7755).

RLS – restless leg syndrome (line 7758).

BEHAVIORS

Behaviors are the elements of DS report, which are next lower hierarchical category after events. While events have their time characteristics: start and end, behaviors have not their own time characteristics and accept timings of event. In other words, behaviors are characteristics of event. Behaviors can be described in five dimensions:

1. Type of behavior (e.g. tonic or clonic movements).
2. Body parts involved in a behavior (e.g. neck or right arm).

DS uses also terms *distal* or *proximal* extremity. The mining is distal arm: elbow and more distally; and distal leg: knee and more distally.

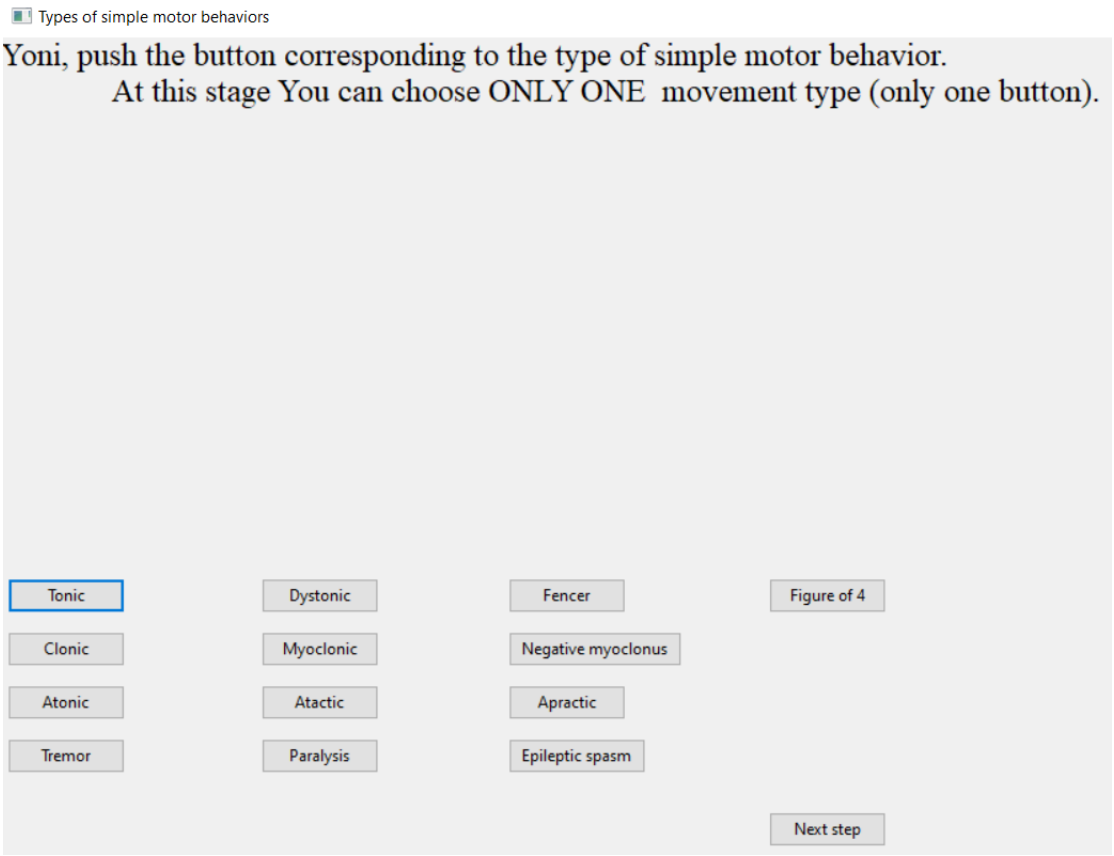
3. Power of the behavior (e.g. strong or weak)
4. Completeness of behavior, especially regarding posturing (e.g. complete or partial)
5. Details of movement or posturing (e.g. flexion [movement] or "politicians' fist" [posturing]).

Not all behaviors are described in all five dimensions.

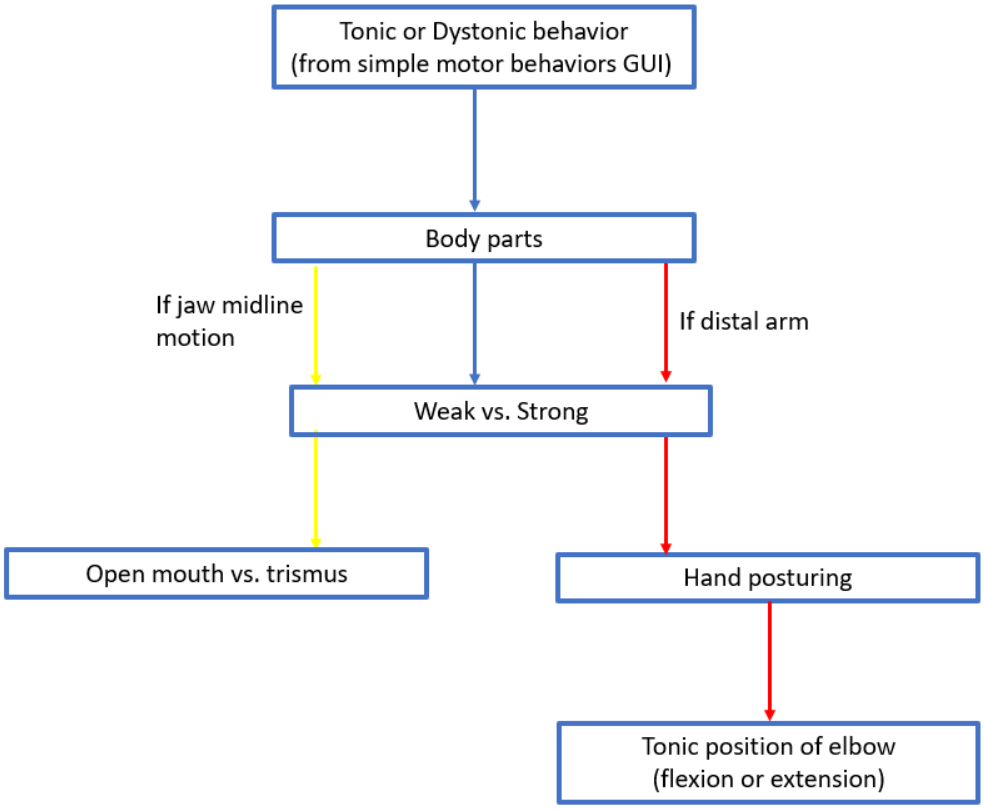
SIMPLE MOTOR BEHAVIORS

Simple motor behaviors are motor behaviors that cannot be part of normal movement repertoire of the humane (in contrast to automatisms) – in other words they are always pathologic behaviors, which can be part of epileptic seizure (or postictal phenomena) or part of nonepileptic event.

After choosing simple motor event on the events' GUI and inserting event start and end timings, DS presets to user selecting GUI with different types of simple motor behaviors.

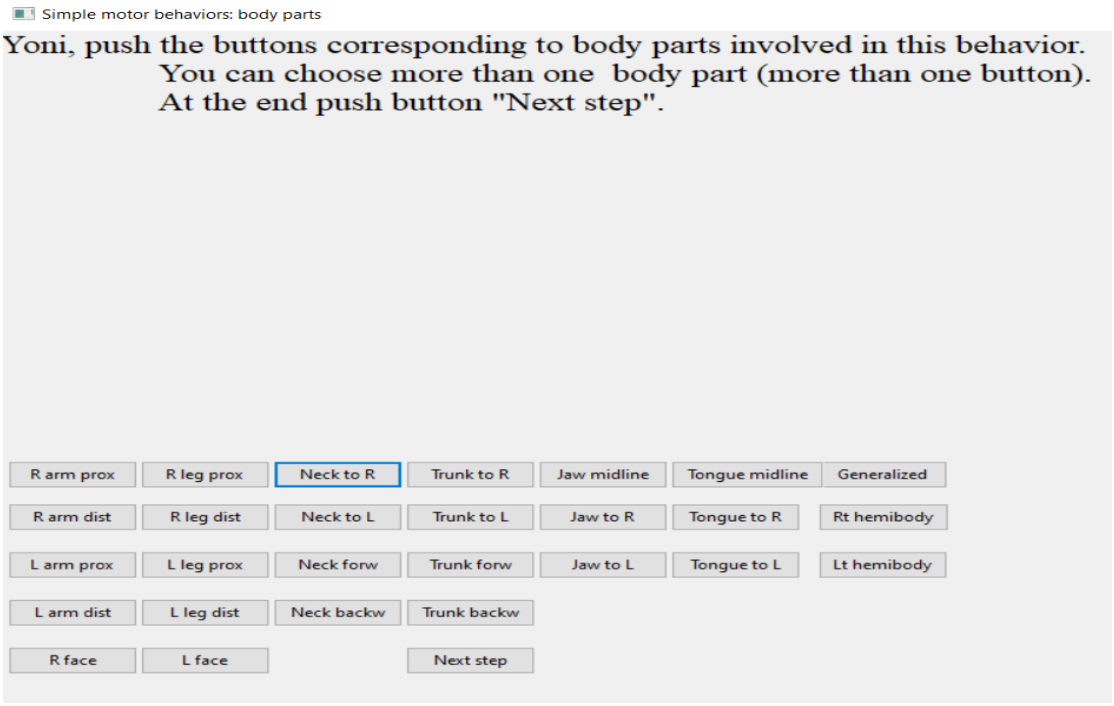


Tonic and dystonic behaviors

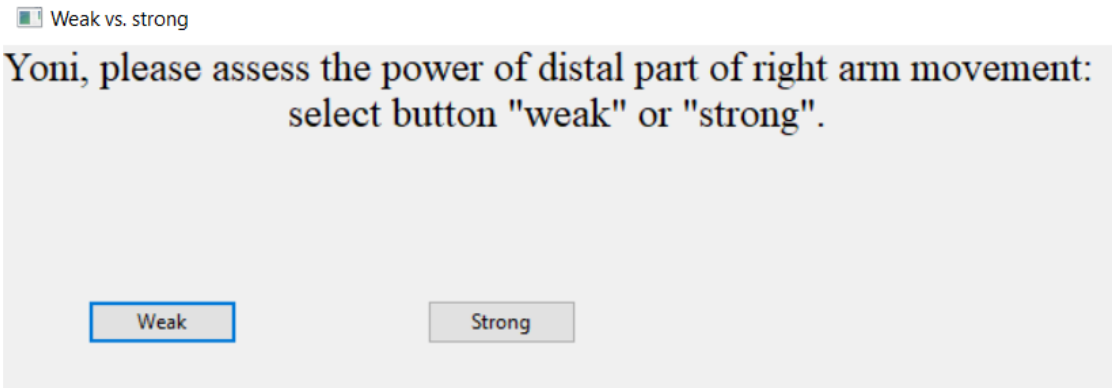


These behaviors are associated with increased muscle tone. Regarding the difference between tonic and dystonic behaviors, the authors recommend chousing dystonic option, when rotatory component is present.

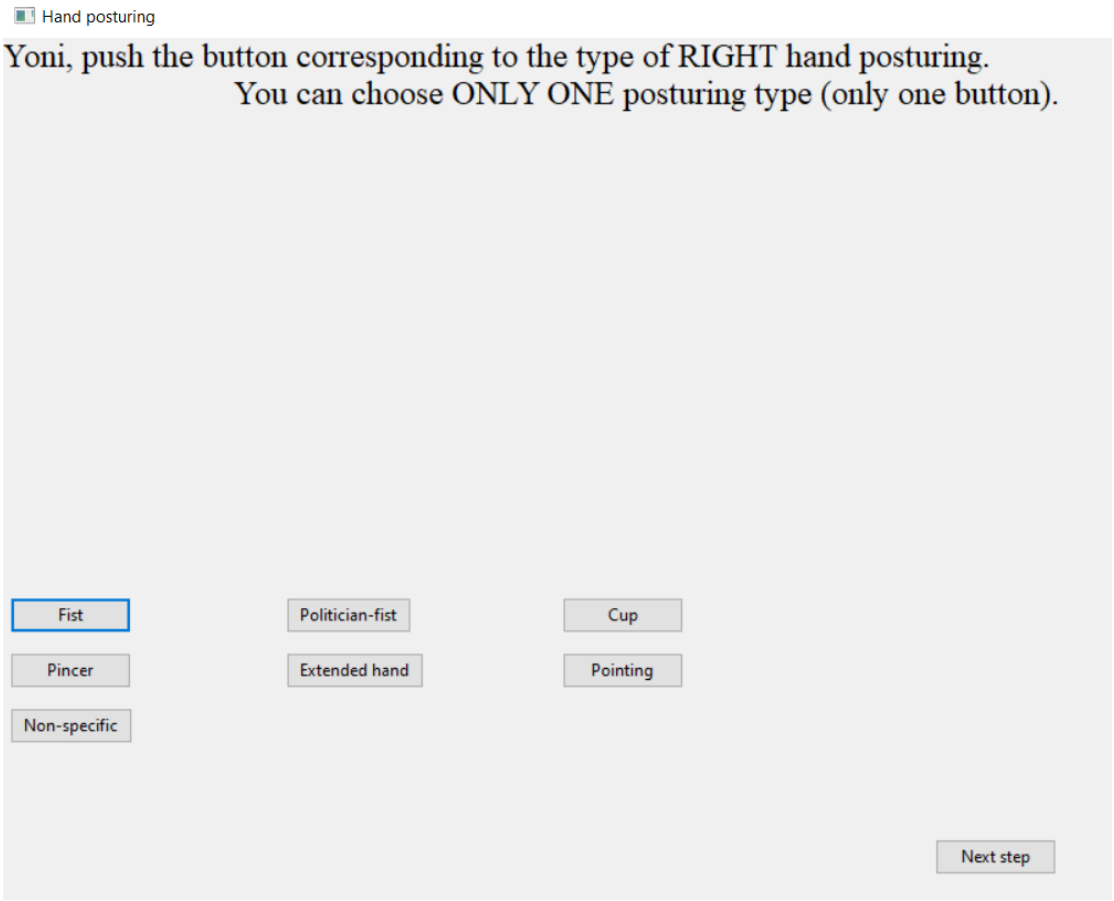
After choosing tonic or dystonic behavior in events GUI, DS opens a body parts GUI:



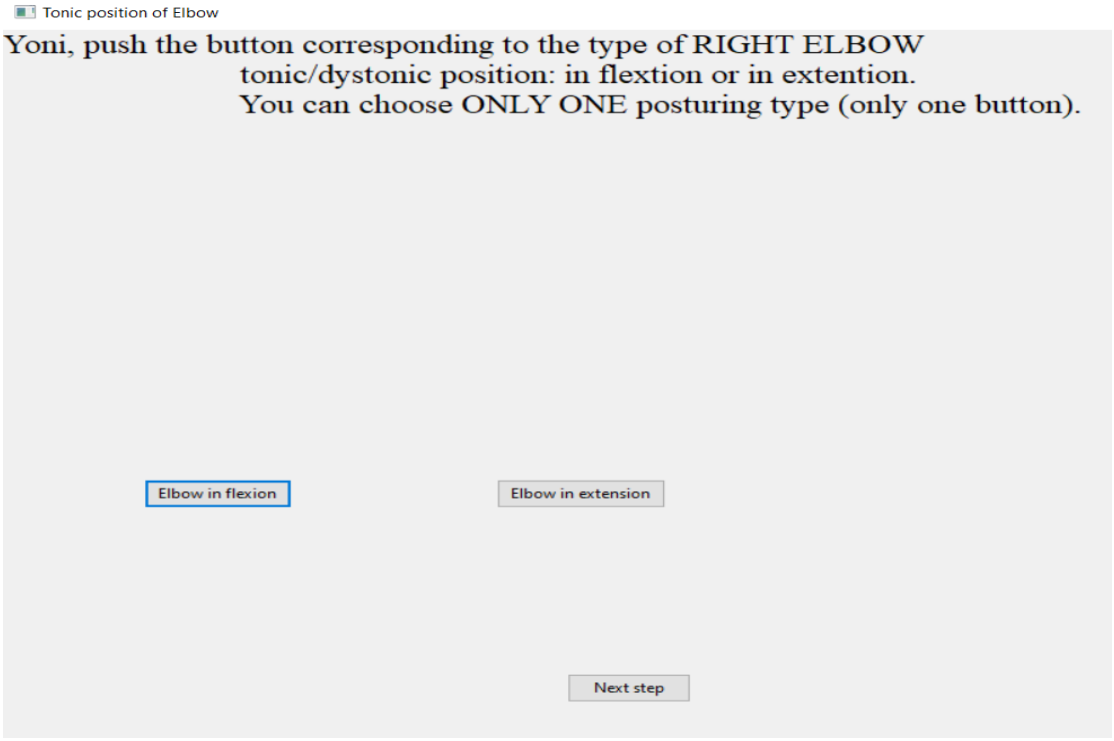
At the next step, DS ask to select, whether the movement was weak or strong.



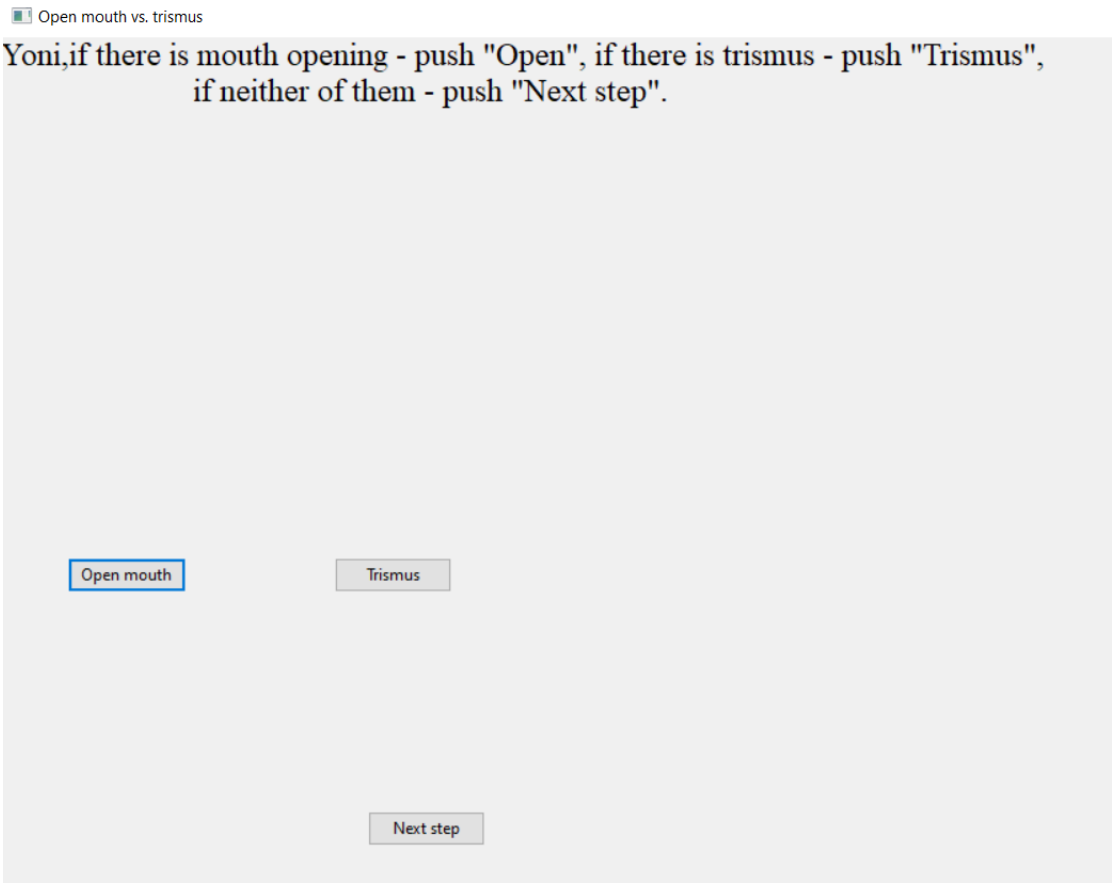
If <u>distal arm</u> was selected, DS will next open the following GUI:



And next - the following GUI:

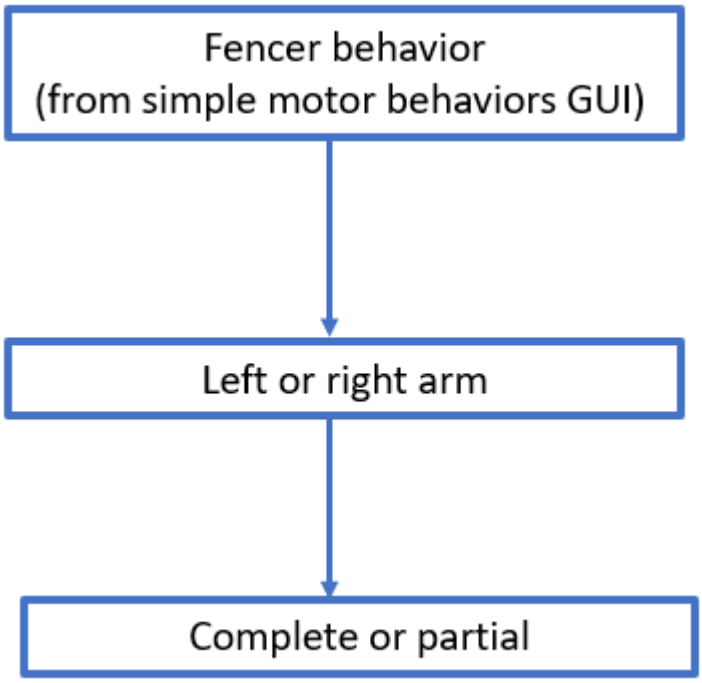


If user selected jaw midline movement in body parts GUI, the user is asked first to select whether the movement is weak or strong and later DS ask her/him to select what is the jaw movement: mouth opening or trismus:

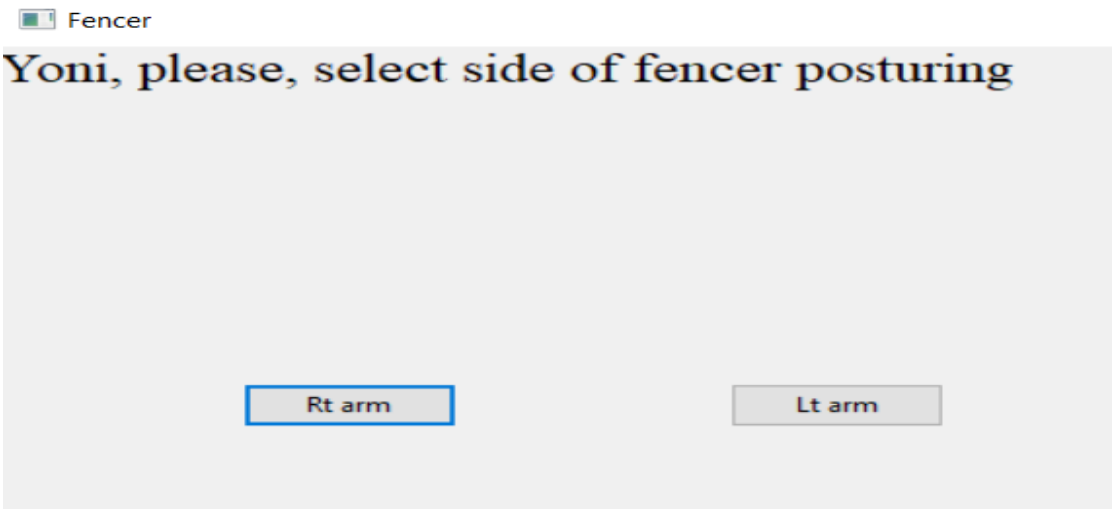


Fencer posturing

Fencer posturing and figure-of-four posturing are both tonic movements, however, DS offers separate buttons for these entities in simple motor behaviors GUI.



If user selected fencer from simple motor behaviors GUI, DS first ask the user to select, whether fencer posturing involves right arm or left arm.



At the next step, DS asks the user, whether fencer posturing is complete or partial.

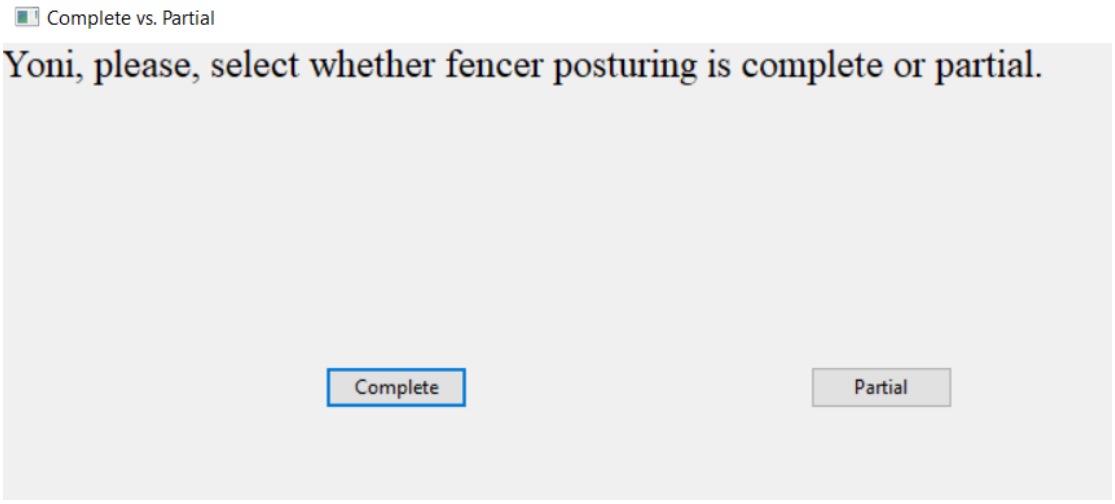
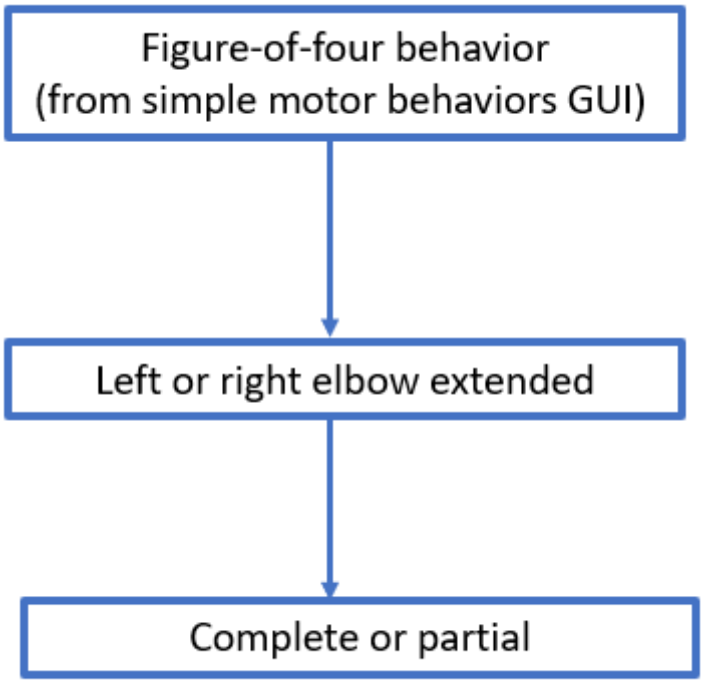
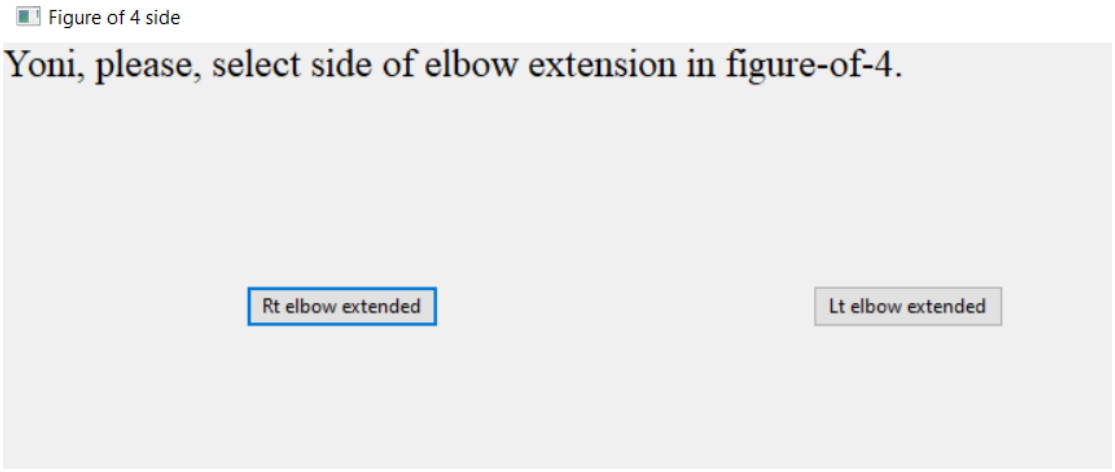


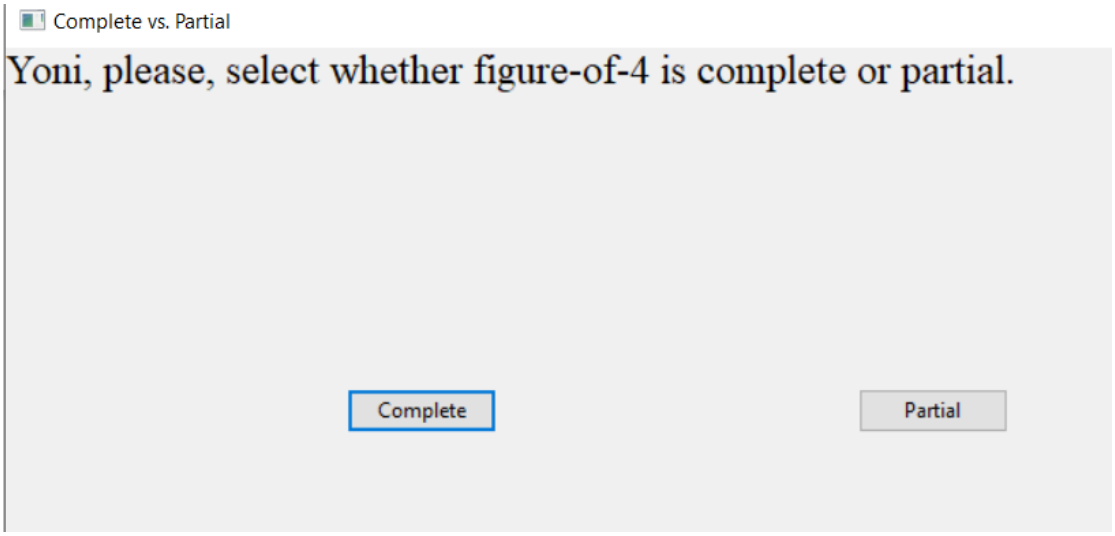
Figure-of-four posturing



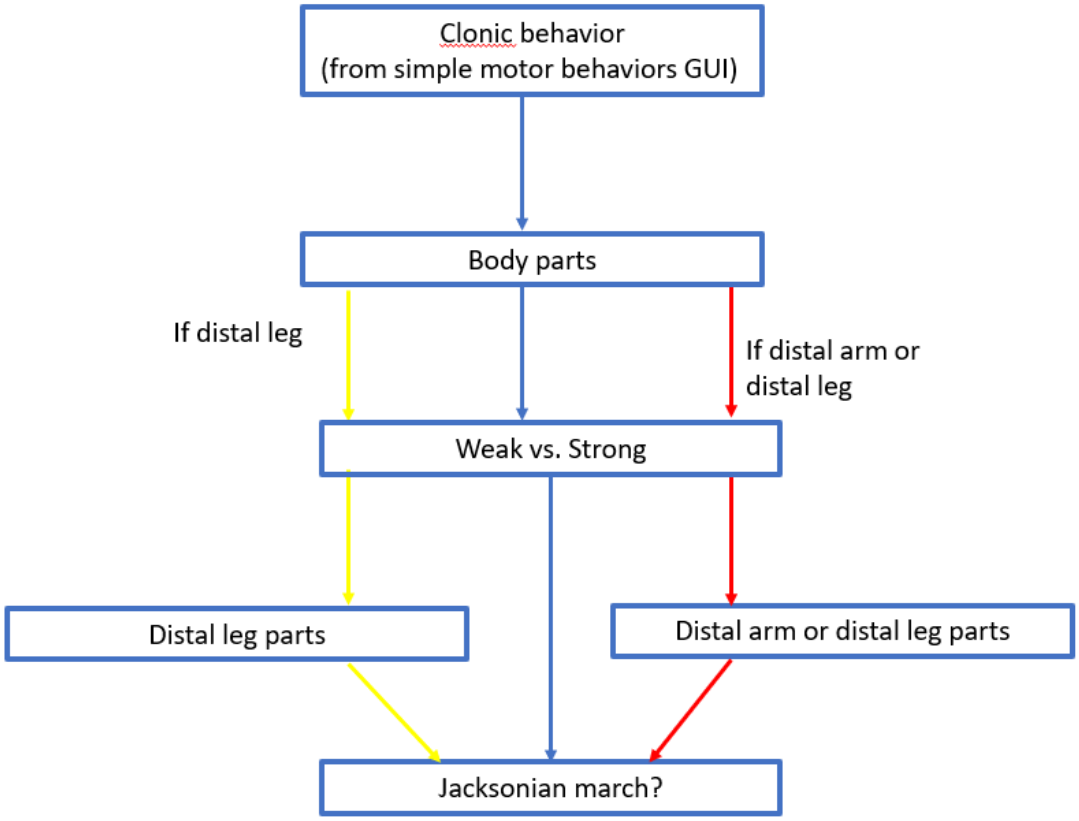
If user selected figure-of-four from simple motor behaviors GUI, DS first ask the user to select, what is the side of extended elbow.



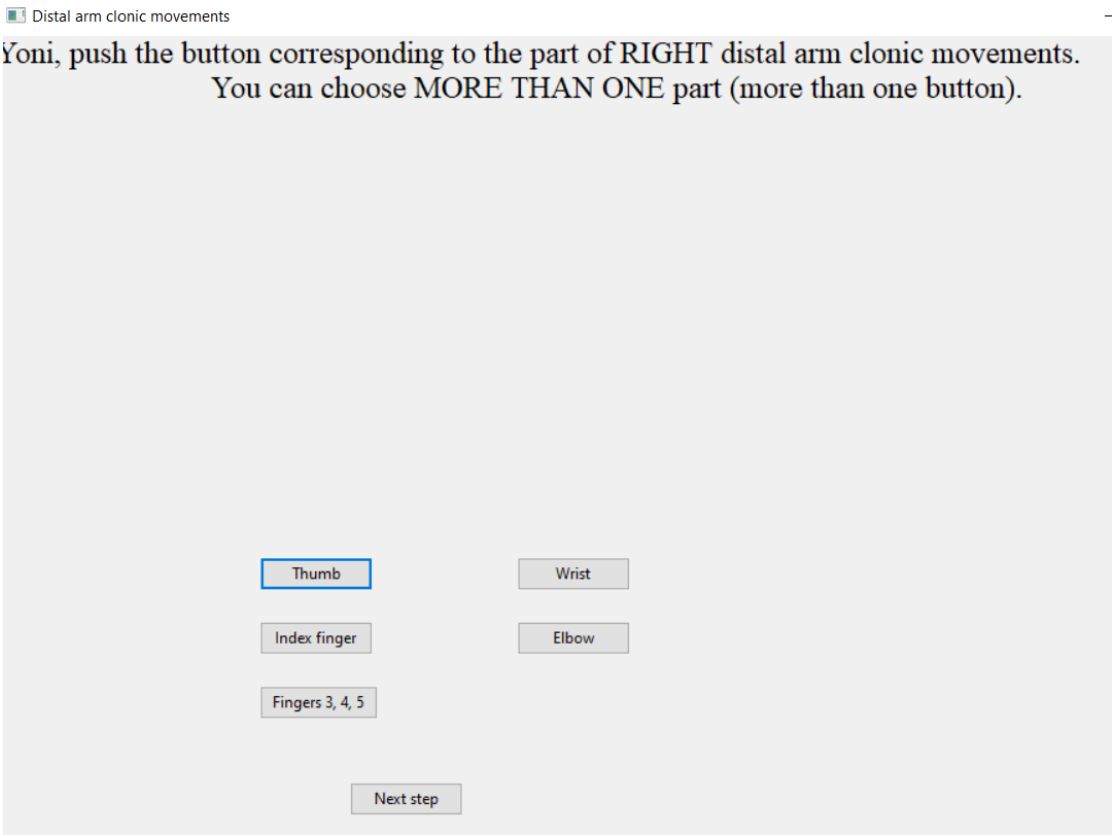
At the next step, DS asks the user, whether figure-of-four posturing is complete or partial.



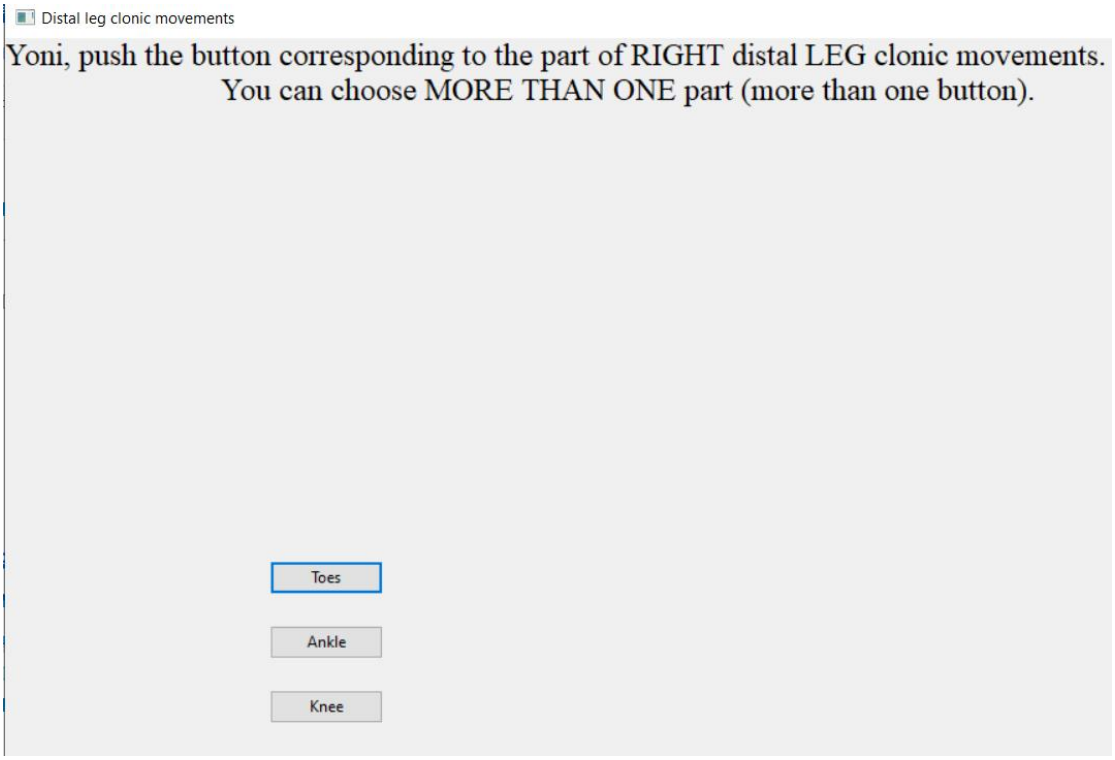
Clonic behaviors



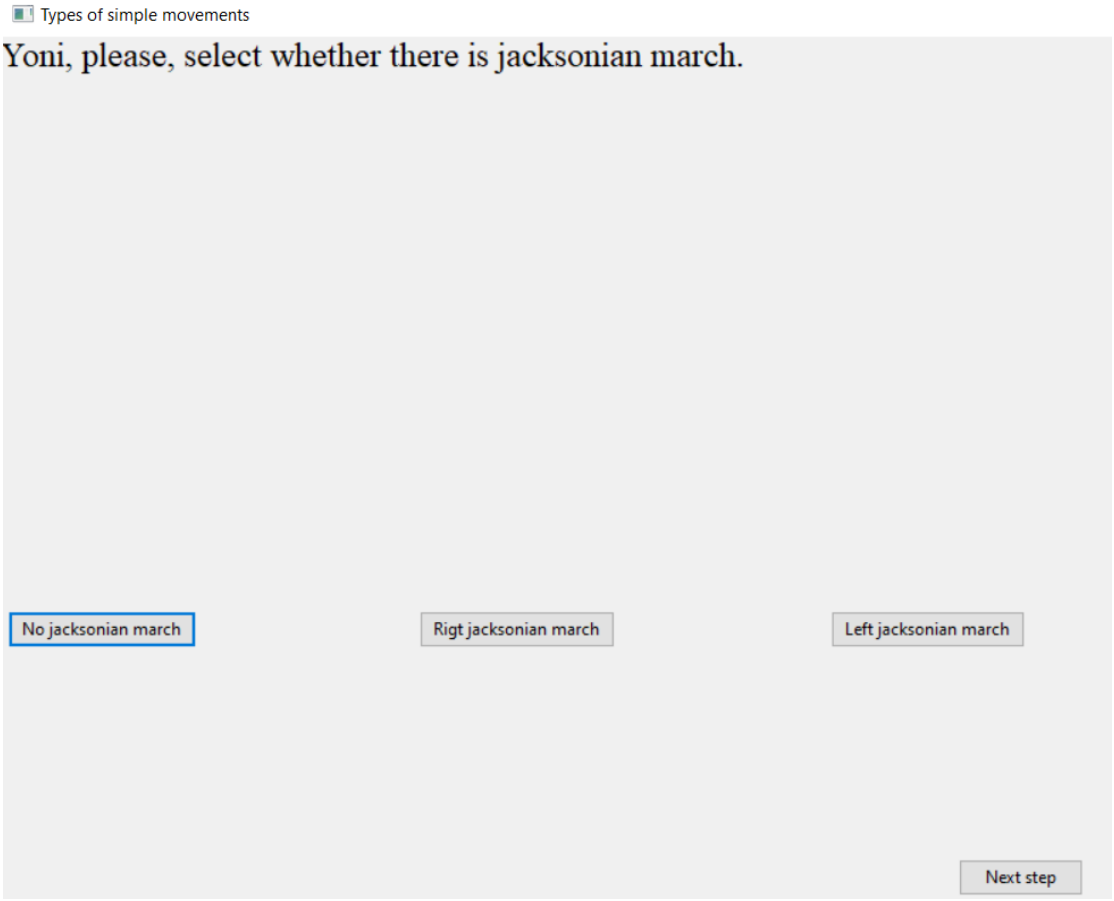
If user selected clonic behavior from simple motor behaviors GUI, DS presents first the body parts GUI and then Weak vs. Strong GUI. If distal arm was chosen, the following GUI will appear:



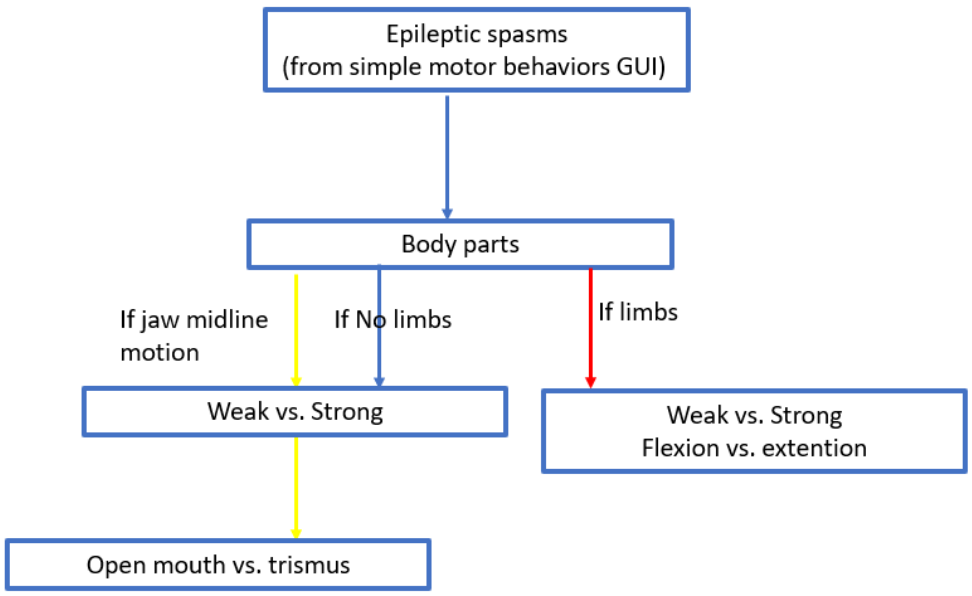
If distal leg was selected, then DS opens the following GUI:



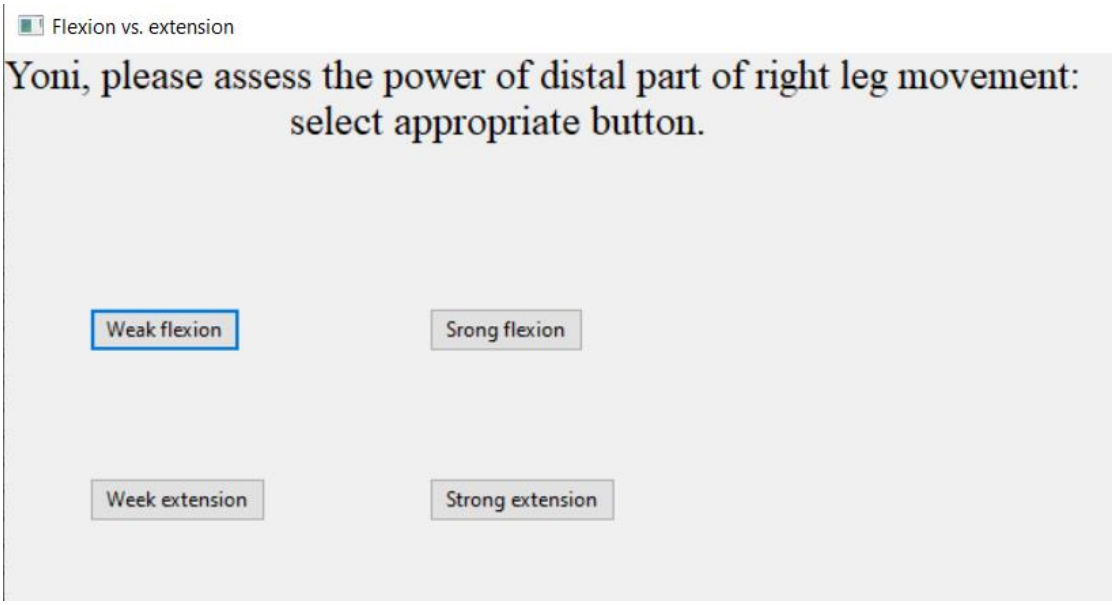
Next, DS asks the user, whether or not the patient demonstrated jacksonian march



Epileptic spasm



If user selected epileptic spasm from simple motor behaviors GUI, DS the body parts GUI. If user selected limbs, DS asks whether it is strong or weak flection or extension.



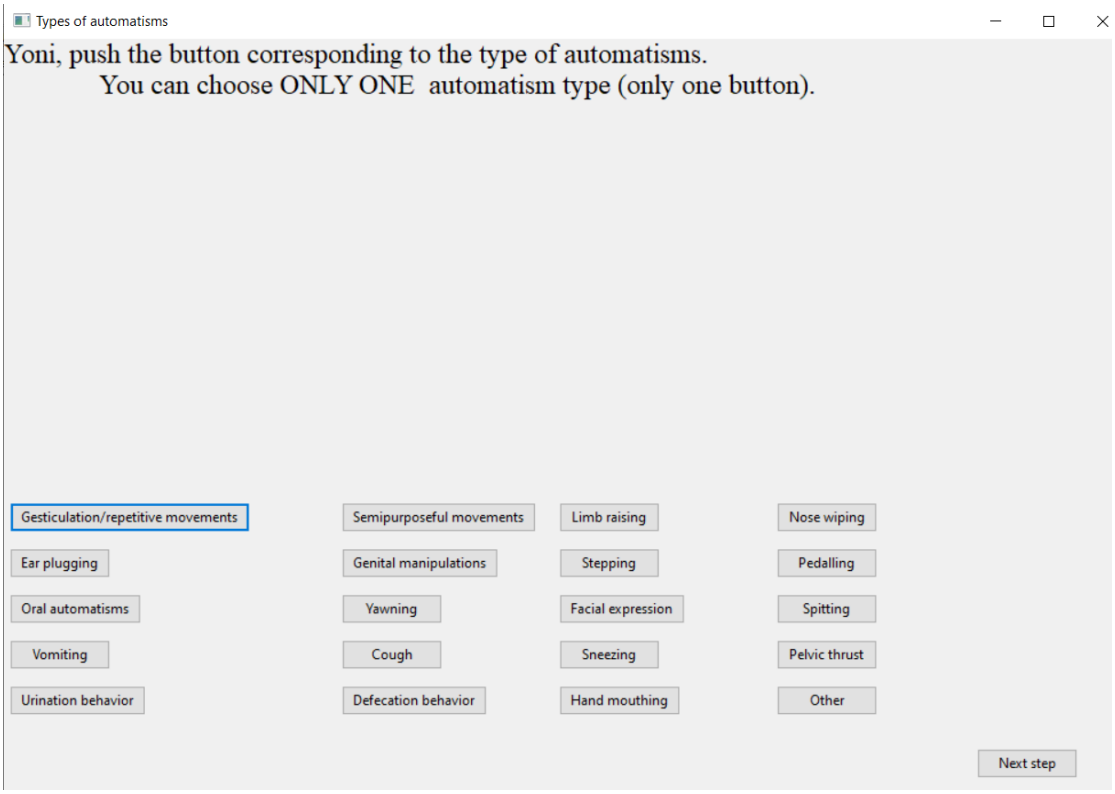
If user selected other (no limbs) body parts, DS opens Weak vs. Strong GUI.

If user selected jaw midline movement in body parts GUI, the user is asked first to select whether the movement is weak or strong and later DS ask her/him to select what is the jaw movement: mouth opening or trismus (as with tonic behaviors).

AUTOMATISMS

Automatisms are movements, which under other circumstances can be part of normal movement repertoire. However, in the context of epileptic seizures, they are pathologic. Automatisms are often are repetitive and inappropriate to the situation.

If user selected Automatism in events GUI and inserted event start and end timings, DS opens types of automatisms GUI:

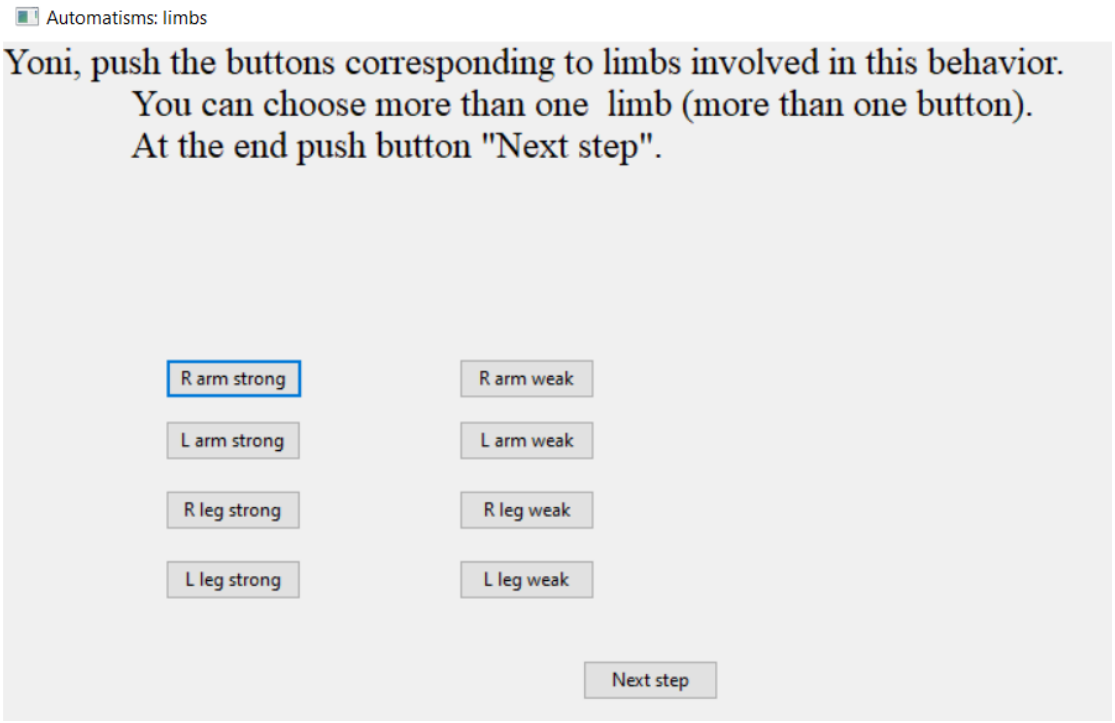


It should be noted that DS allows labeling pelvic thrusting both as automatism and as "other" behavior, their it is named unspecified pelvic thrusting. The authors suggest using pelvic thrusting as automatism if user believes that it is part of epileptic seizure and as "other" behavior if she/he believes that it is PNES.

If user selected:

- Gesticulation/repetitive movements or
- Semipurposeful movements or
- Limb raising,

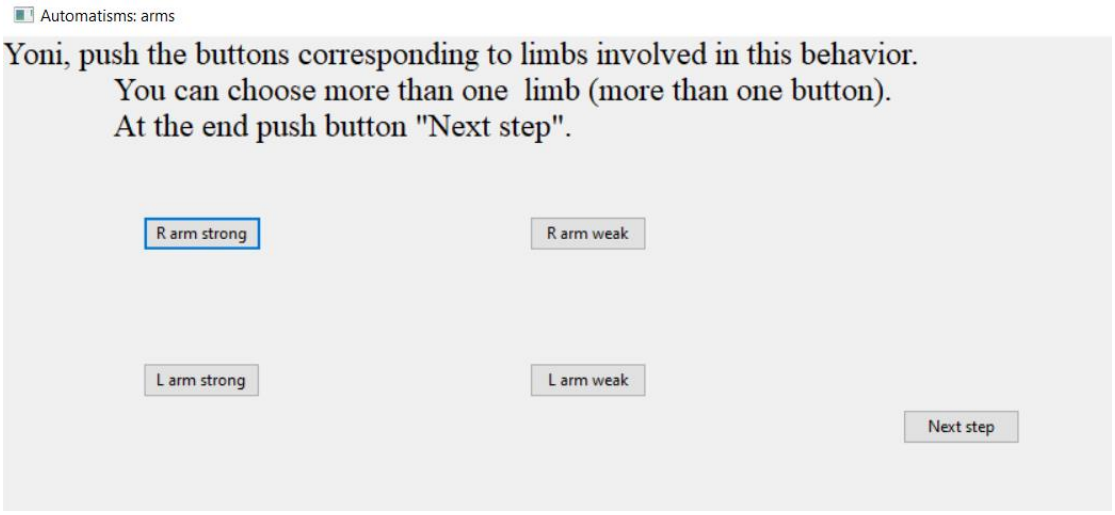
DS will ask user to label the limbs (right or left arm or leg) that are involved and whether the involvement is strong or weak:



If user selected:

- Nose wiping or
- Ear plugging or
- Genital manipulations or
- Hand mouthing,

DS will ask user to label the arms (right or left) that are involved and whether the involvement is strong or weak.

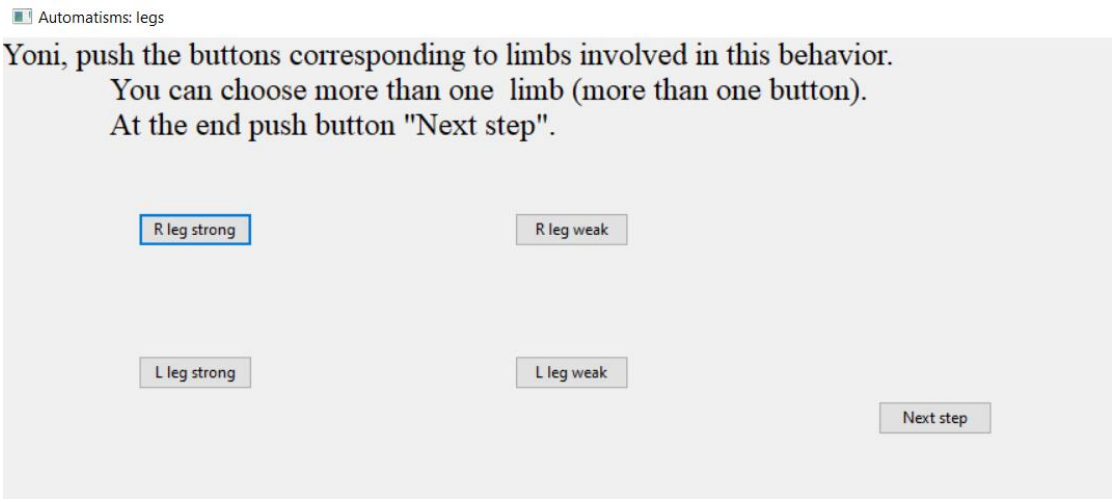


If user selected:

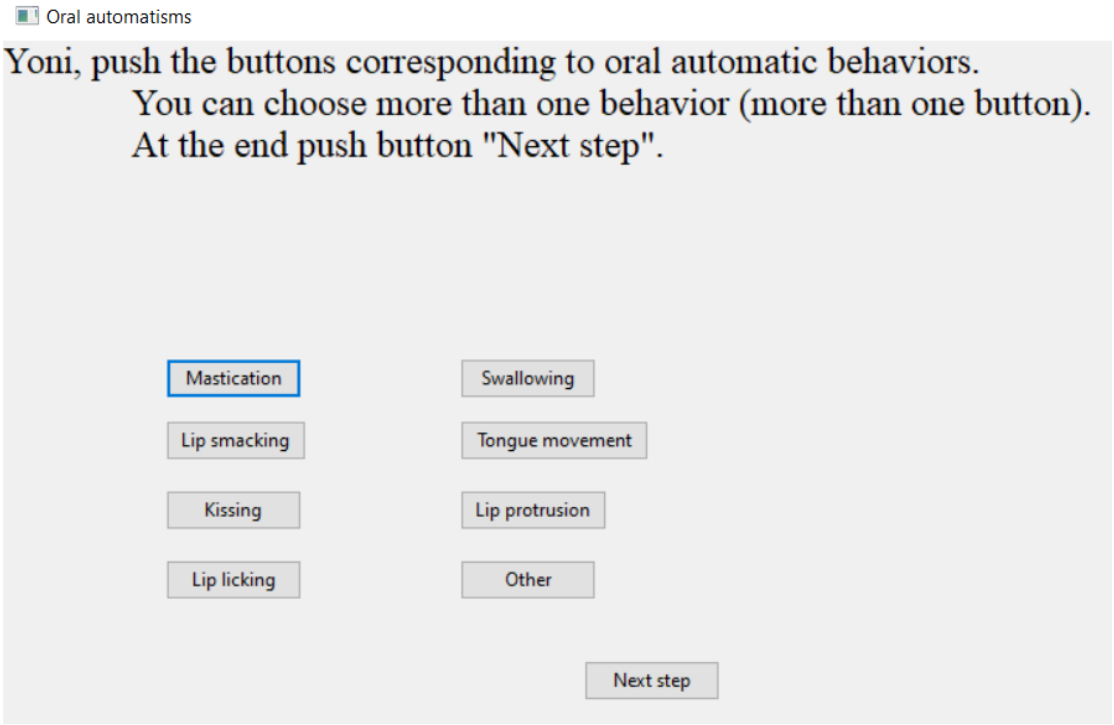
- Nose wiping or
- Ear plugging or
- Genital manipulations or

- Hand mouthing,

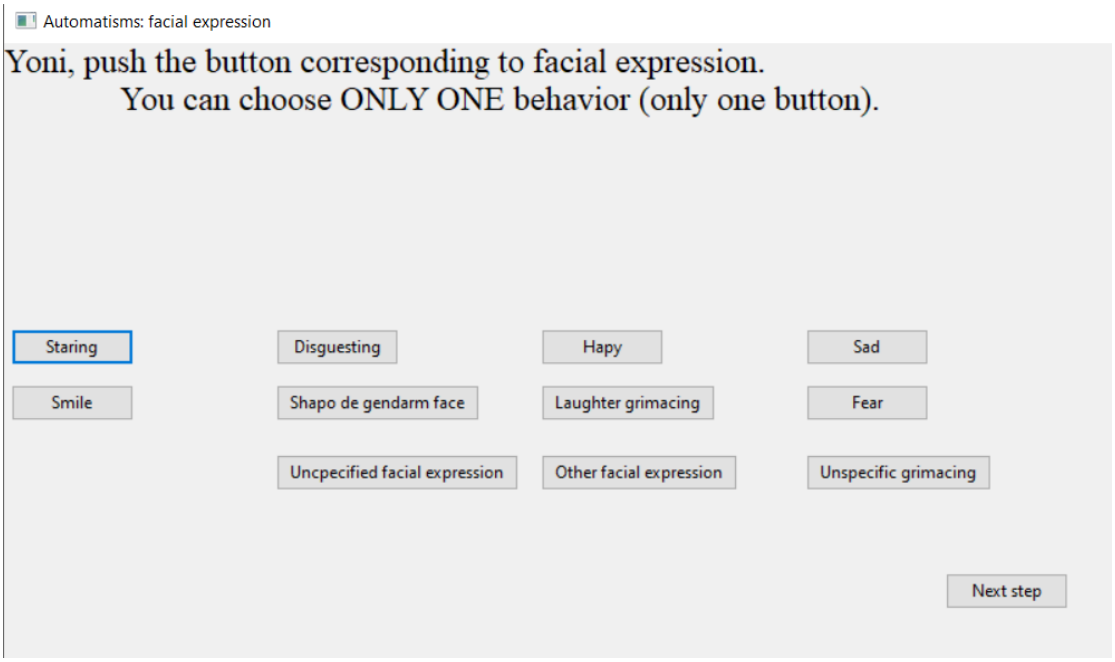
DS will ask user to label the legs (right or left) that are involved and whether the involvement is strong or weak:



If user selected oral automatism in types of Automatisms GUI, DS opens oral automatisms GUI:

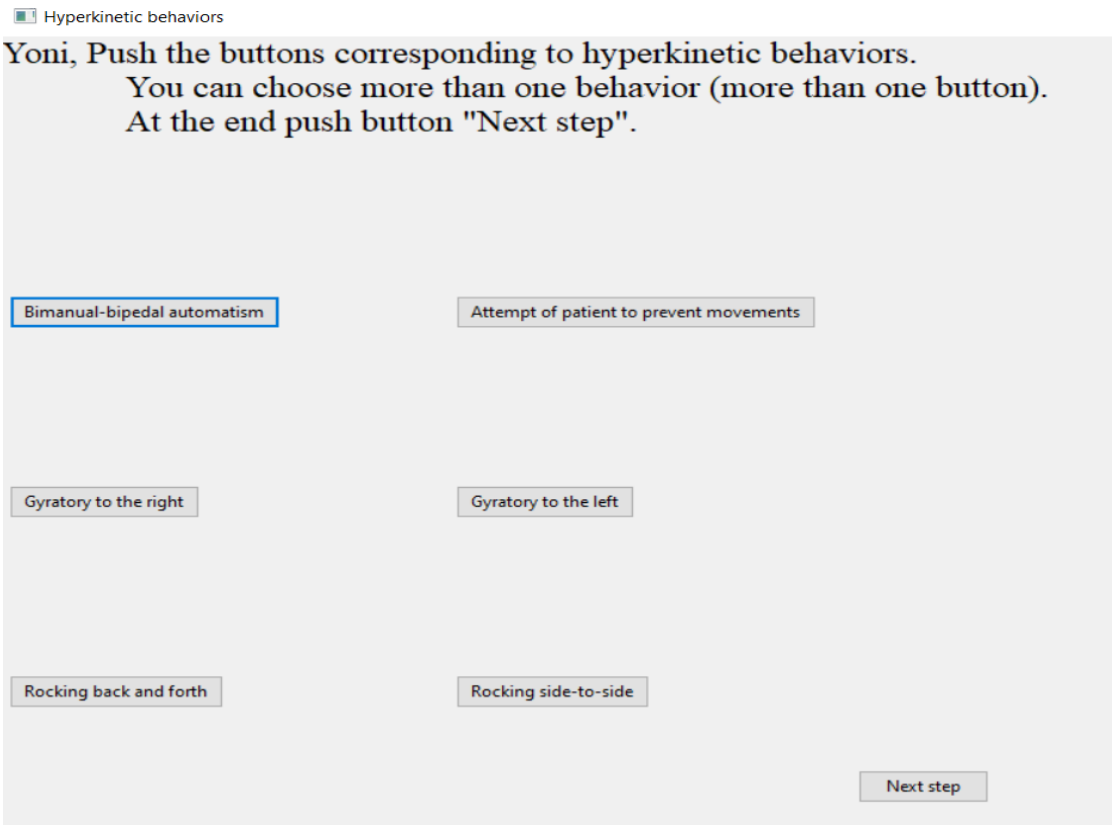


If user selected facial expression in types of Automatisms GUI, DS opens Automatisms: facial expression GUI:

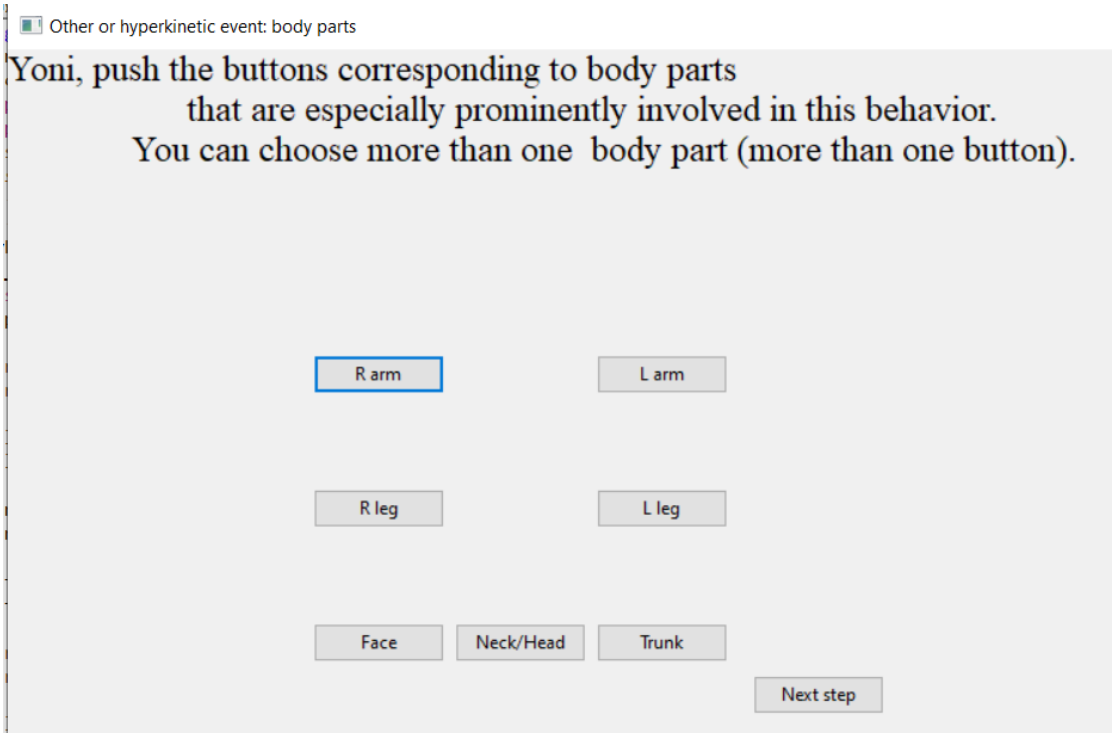


HYPERKINETIC BEHAVIORS

When user selects hyperkinetic behaviors from event GUI and inserts event start and end timings, DS opens the hyperkinetic behaviors GUI.



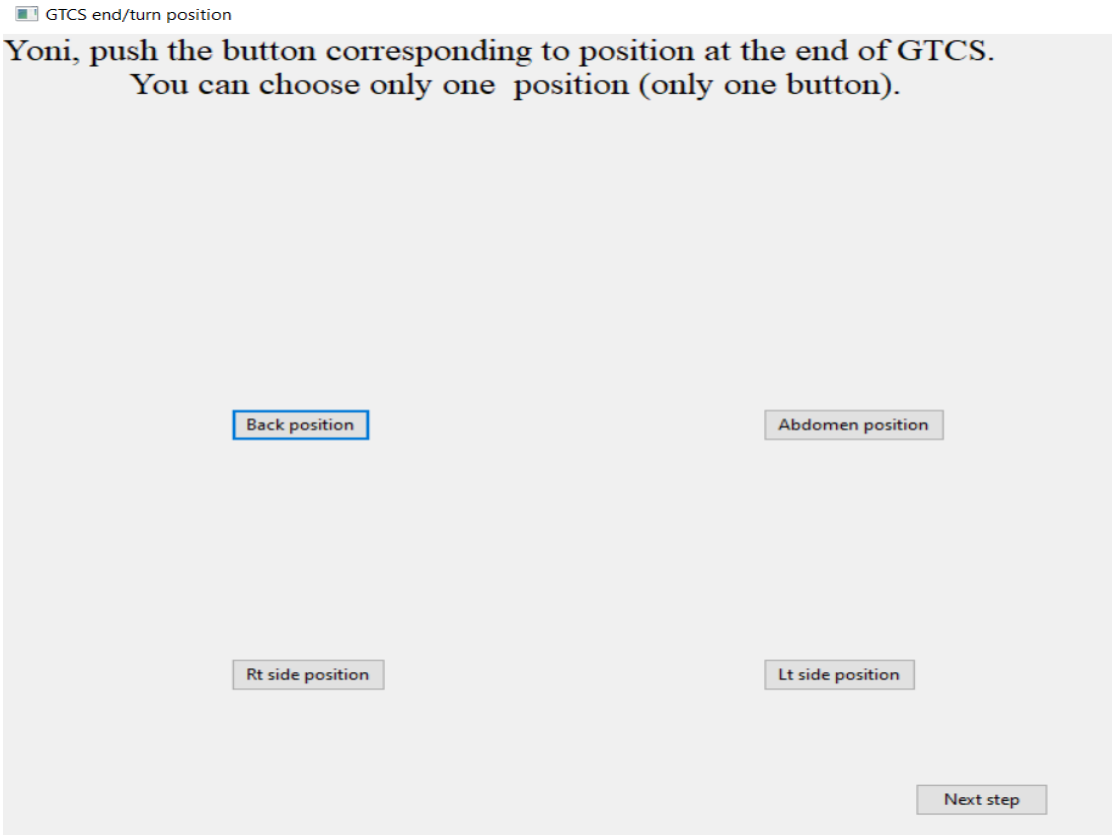
Next, DS asks user, what are the body parts, which are involved especially prominently in hyperkinetic behavior.



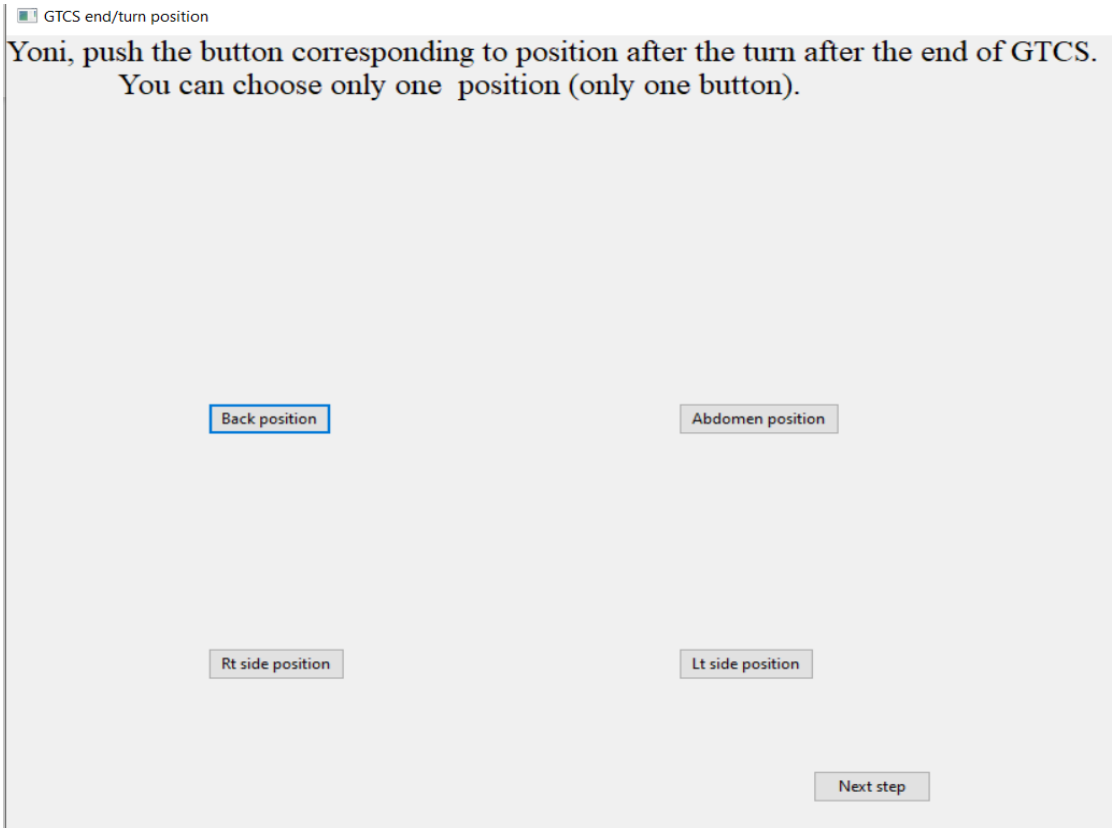
GENERALIZED TONIC CLONIC SEIZURES

If user selected GTCS from event GUI and inserted event start and end timings, DS will ask her/him to inset the time of transition from tonic phase to clonic. GTCS is only event type, which requires inserting three timings and not two (event start and event end).

Next DS will ask user to select the body position at the end of seizure:



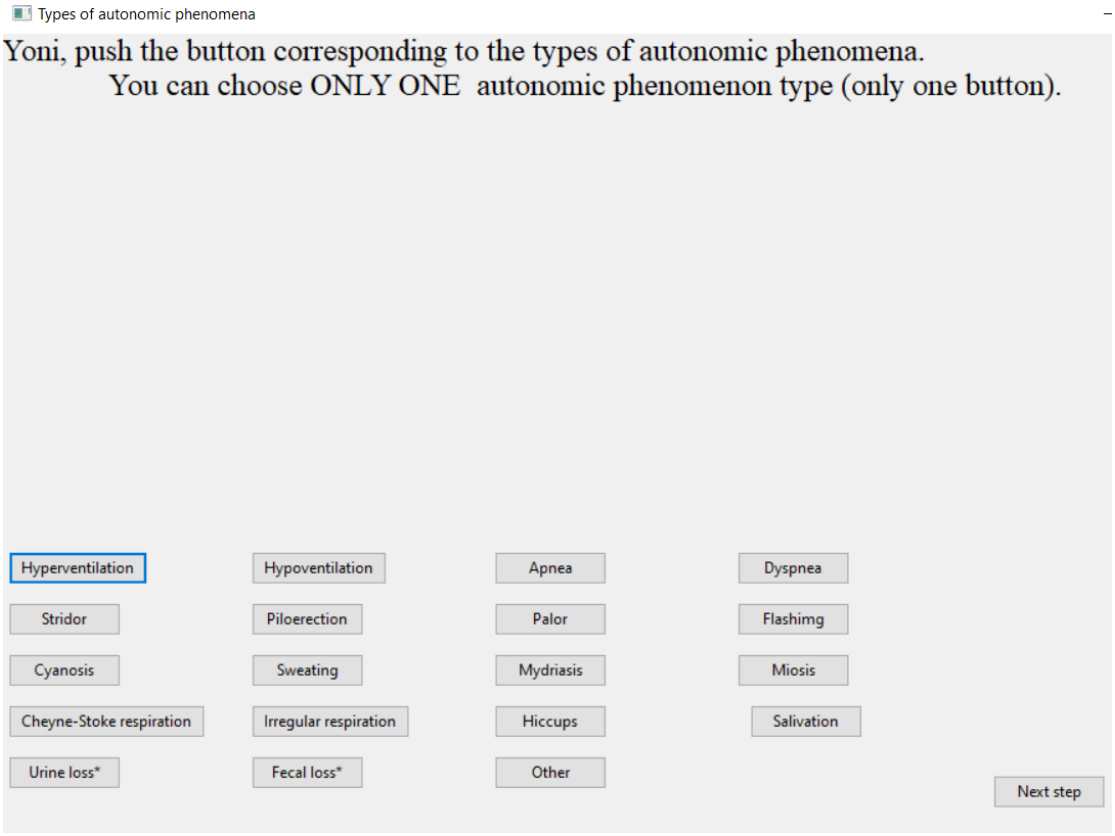
Next, DS asks to select the body position after turn the patient after the end of seizure. If patient did not turn after the seizure, the authors suggest selecting the same body position as in the previous GUI.



These two last questions are related to risk of SUDEP.

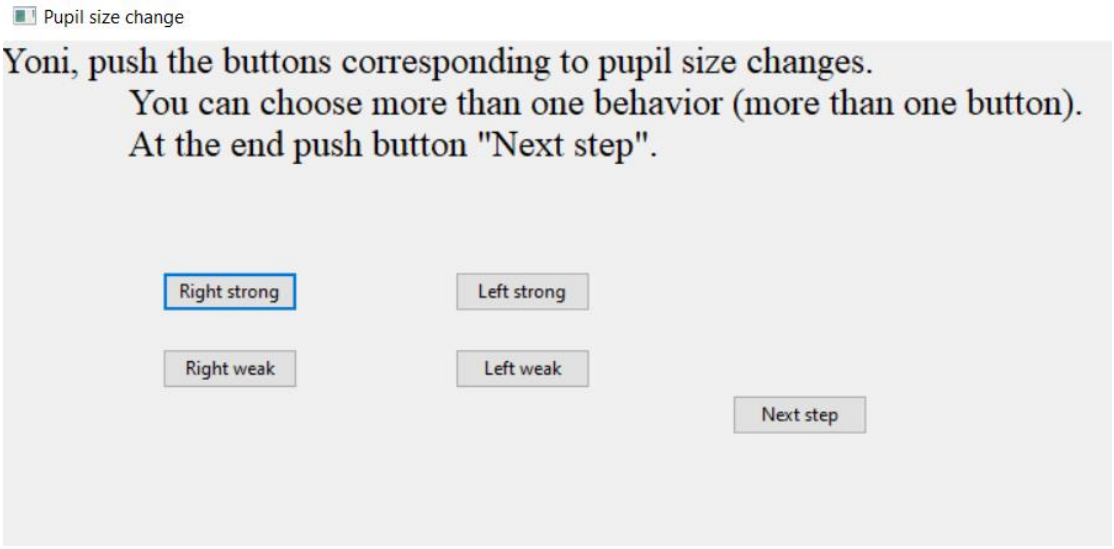
AUTONOMIC BEHAVIORS

If user selected "autonomic" from event GUI and inserted event start and end timings, DS will open the types of autonomic phenomena GUI:



It is important to note that the meaning of urine loss and fecal loss is different that urinary behavior or defecation behavior in automatisms GUI. The authors suggest considering urine loss or fecal loss as autonomic phenomena, when they are not associated with urination or defecation behavior, otherwise they should be considered as automatisms.

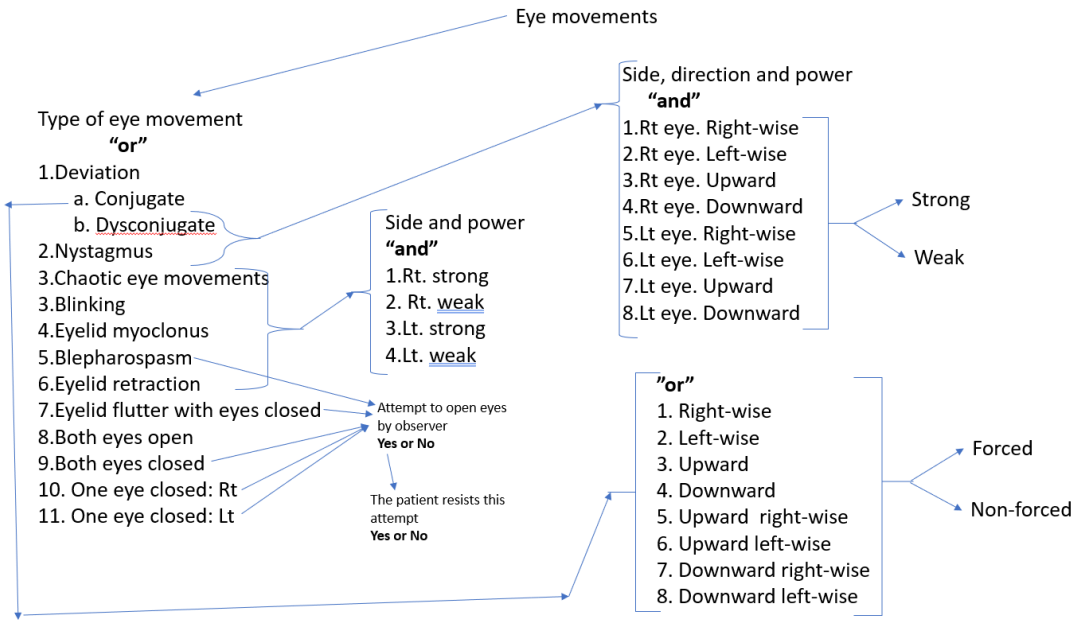
If user selected mydriasis or miosis, DS will open the following GUI:



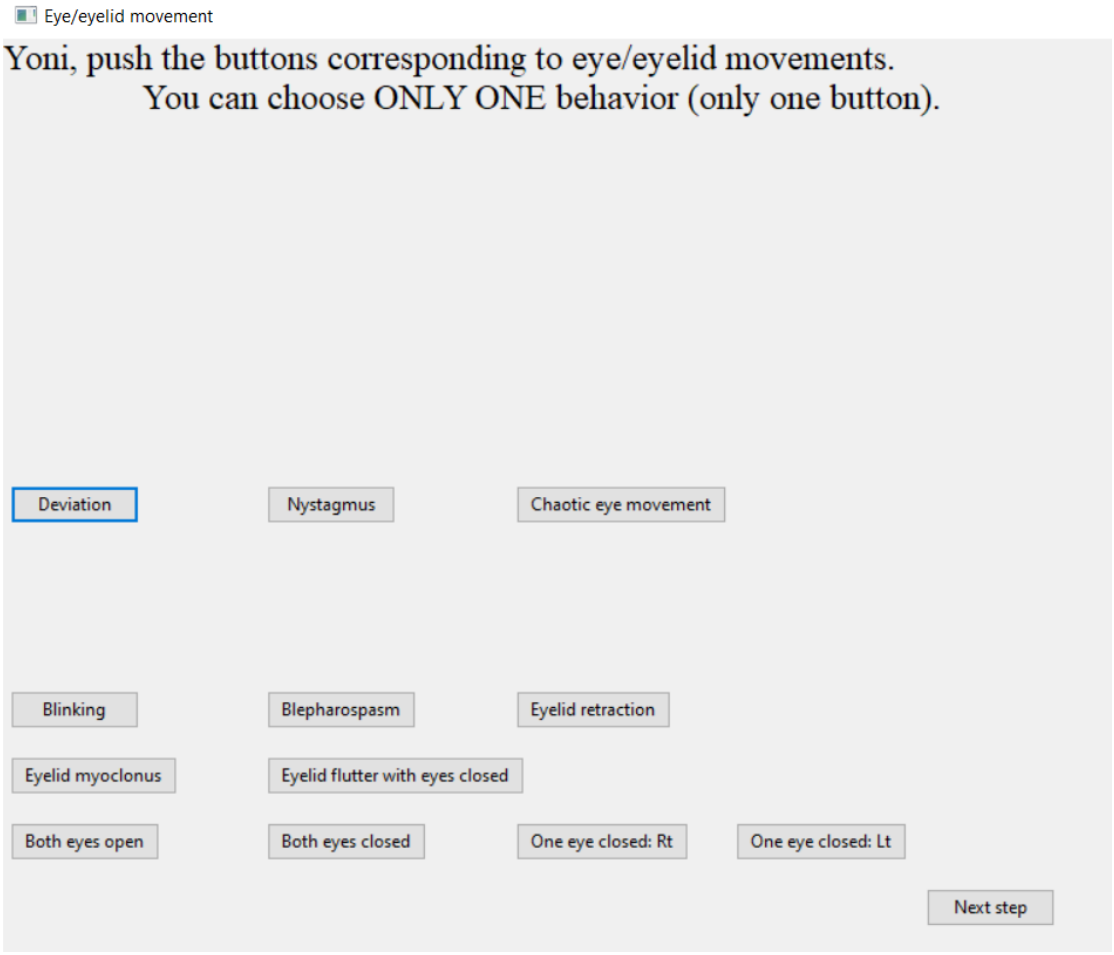
If user selects the combination of Right strong and Right weak or Left strong and Left weak, DS will not accept such combination and will return this GUI to the user, while in the IPython console appears the following statement:

```
Software:
YOU CAN'T PRESS BOTH 'STRONG' AND 'WEAK' OF THE SAME PUPIL!
 PLEASE TRY AGAIN!
```

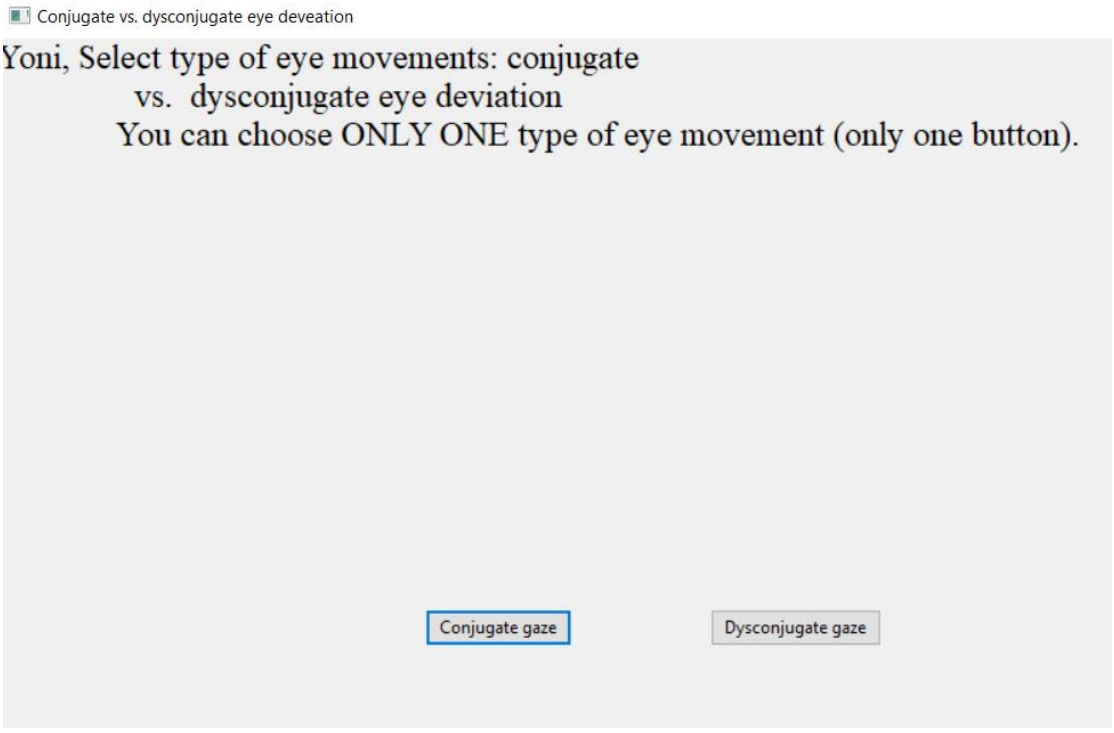
EYE/EYELID MOVEMENT BEHAVIORS



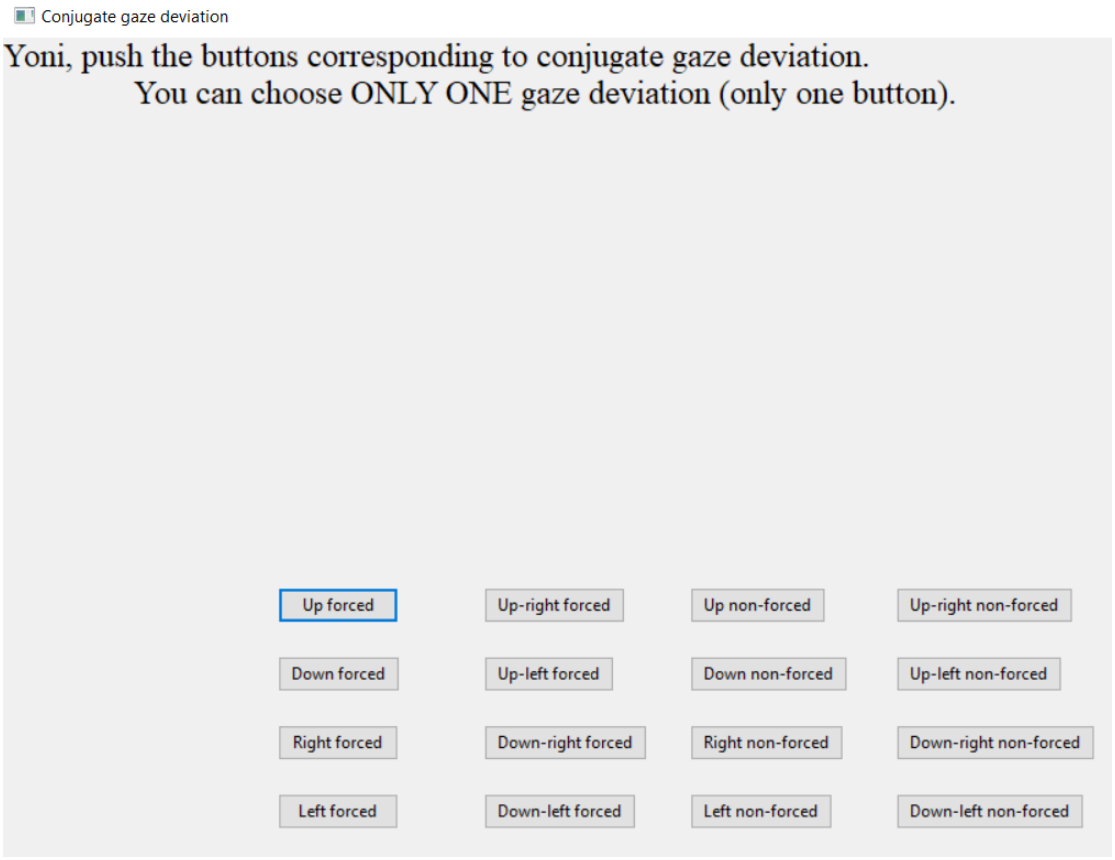
After user selected eye movements from the event GUI and inserted event start and end timings, DS opens Eye/eyelid movement GUI. In this GUI the upper row of buttons corresponds to eye movements and the lower three rows – to eyelid movements.



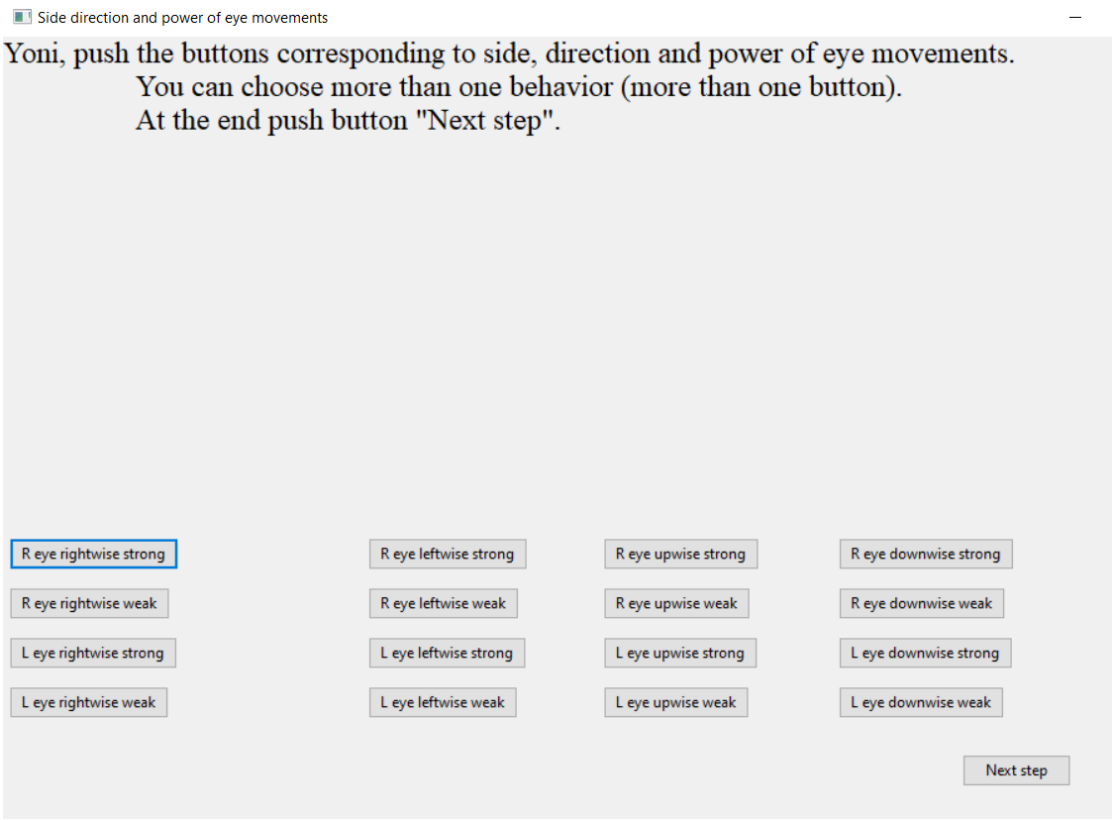
If user selects Deviation, DS presents the following GUI:



If user selects conjugate gaze, DS opens the following GUI:



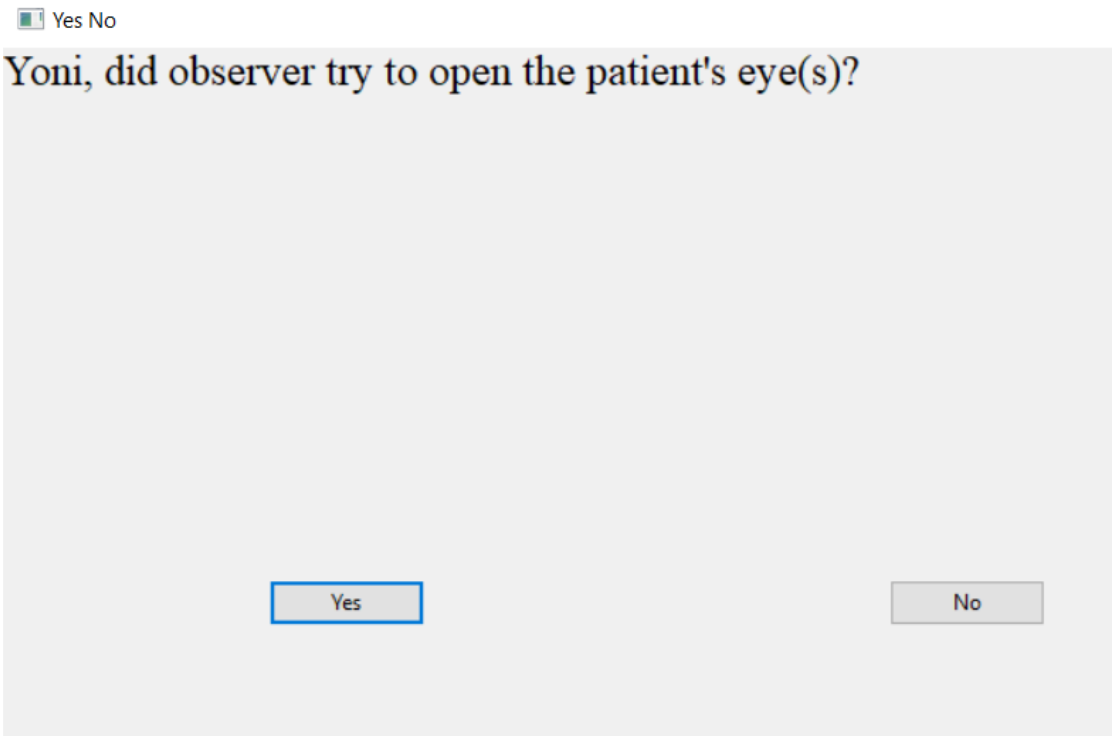
If user selects nystagmus or dysconjugated gaze deviation, DS opens the Side direction and power of eye movements GUI:



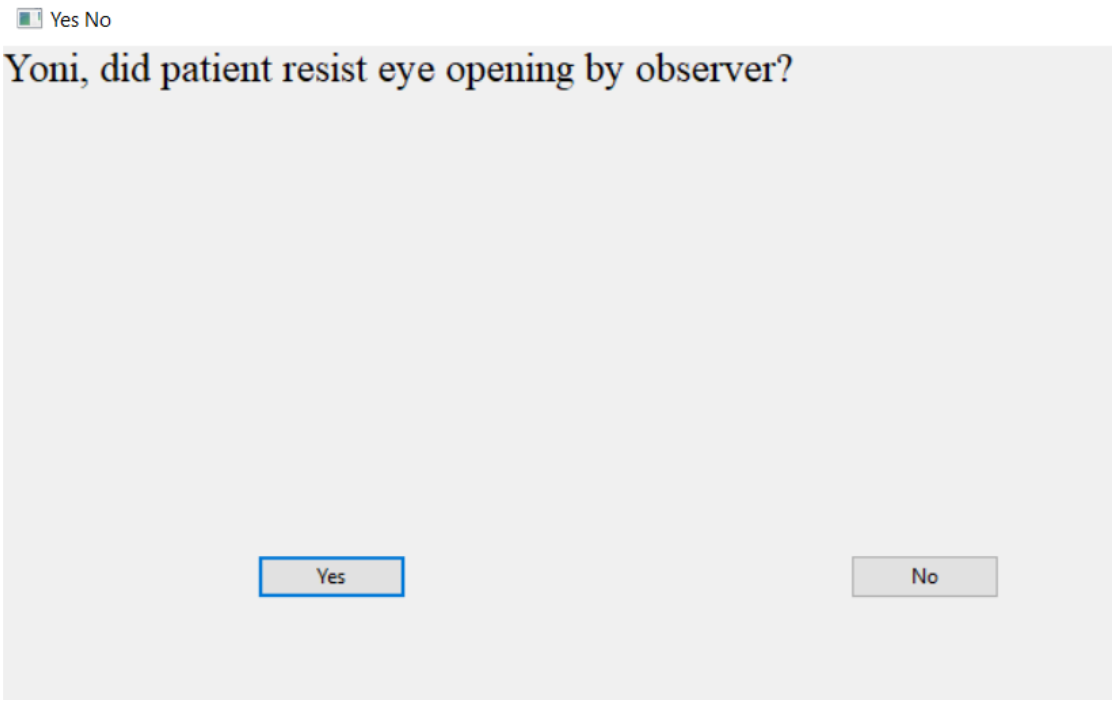
If user selects impossible eye movements combination (the movement of the same eye in opposite direction or both weak and strong movement of the same eye in the same direction), DS returns this GUI and the following statement appears in IPython console:

```
Software:
YOU SELECTED IMPOSSIBLE COMBINATION!
 PLEASE TRY AGAIN!
```

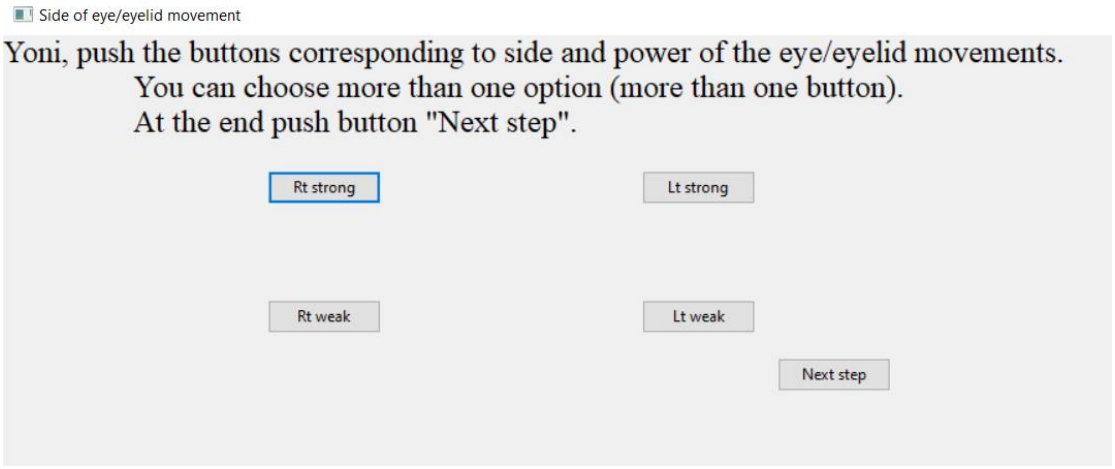
If user selected blepharospasm, eyelid flutter with eyes closed, both eyes closed, one eye closed: Rt or one eye closed: Lt, DS presents the following GUI:



In this GUI, if user selects Yes, DS will open the following GUI:



If user selects chaotic eye movements, blinking, eyelid myoclonus, blepharospasm or eyelid retraction, DS opens Side of eye/eyelid movement GUI:



If user will select both strong and weak of the same side, DS will not accept it, will return the GUI and will write in IPython console:

```
Software:
YOU CAN'T PRESS BOTH 'STRONG' AND 'WEAK' OF THE SAME EYE/EYELID!

                                        PLEASE TRY AGAIN
```

VOICE BEHAVIORS

Voice phenomena

Type of voice phenomena

“or”

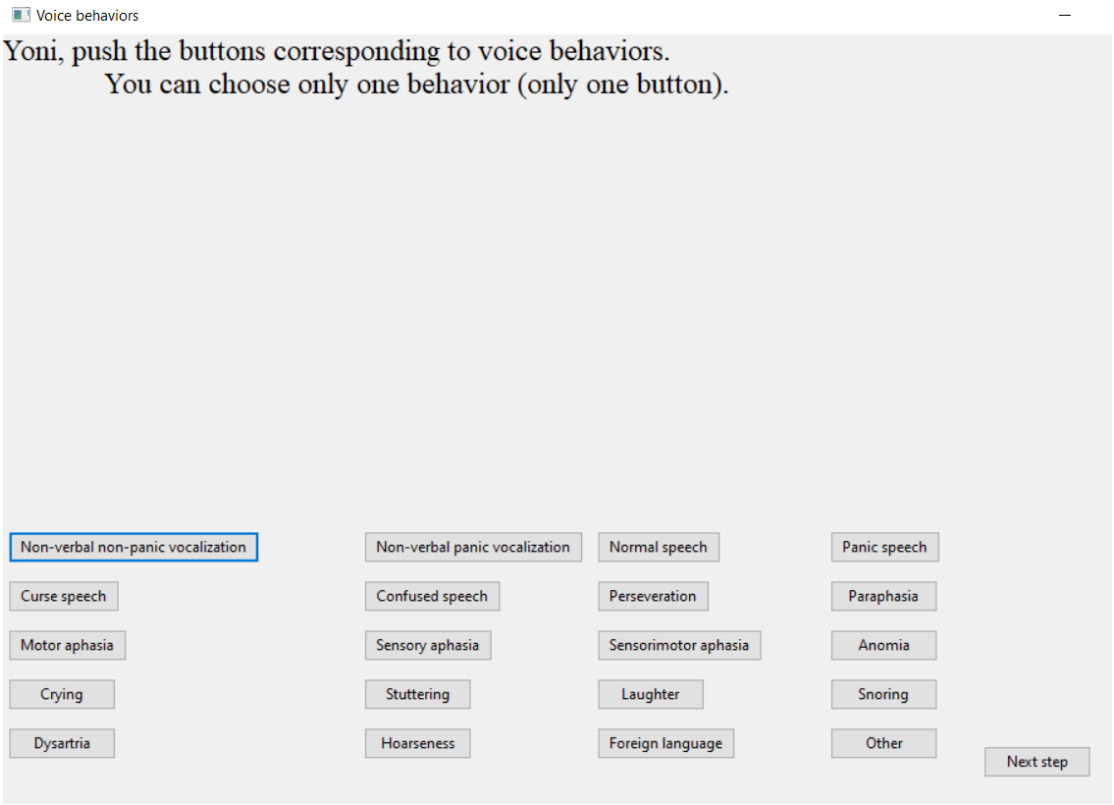
1. Non-verbal vocalization without panic behavior
2. Non-verbal vocalization with panic behavior
3. Coherent purposeful adequate speech during ictal episode
4. Panic speech
5. Curse speech
6. Confused speech
7. Perseverations
8. Paraphasia
9. Motor aphasia: patient understands verbal instructions but cannot speak
10. Sensory aphasia: Patient does not understand verbal instruction but can communicate non-verbally
11. Sensorimotor aphasia
12. Patient cannot find the name of the object
13. Laughter
14. Crying
15. Stuttering
16. Patient speaks but the interpreter does not understand the language
17. Snorring
18. Dysartria
19. Hoarsness
20. Other

“or”

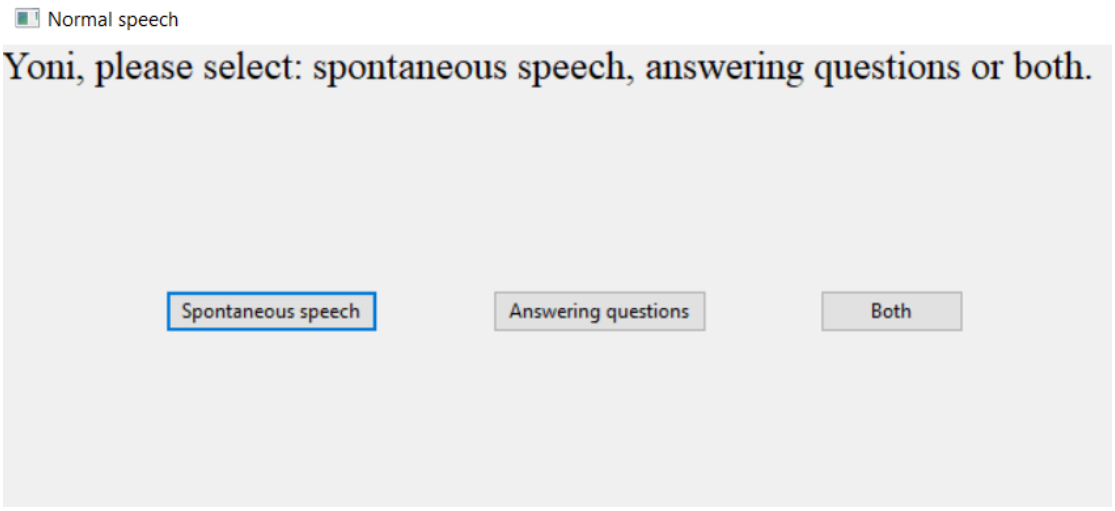
1.Spontaneous
2.Answering questions
3.Spontaneous+Answering questions

1.Partial
“or”
2.Coplete

After user selected "voice event" from the event GUI and inserted event start and end timings, DS opens voice behaviors GUI:

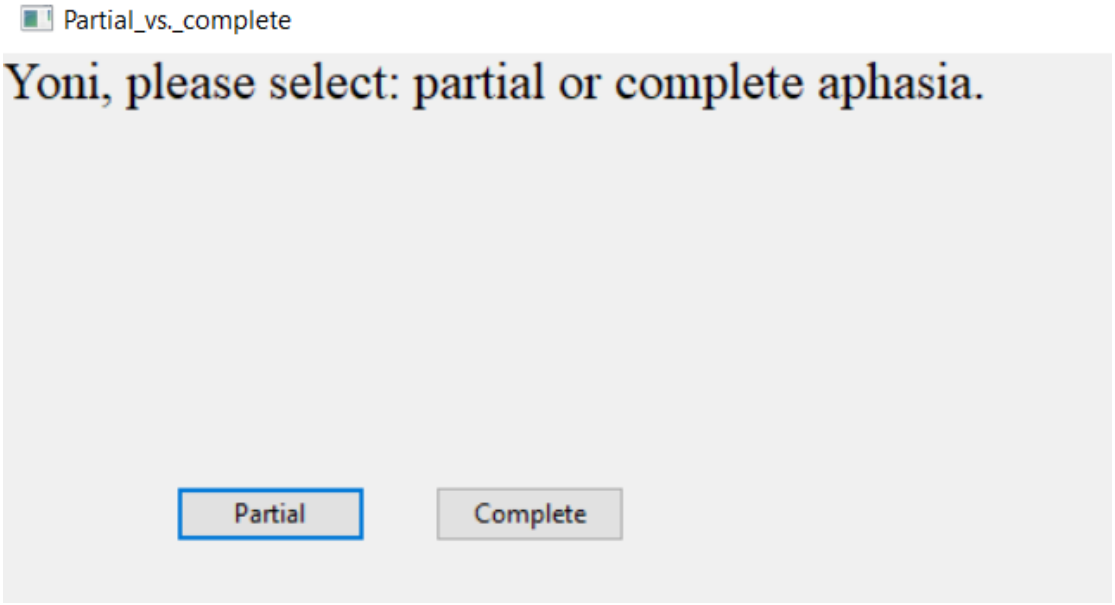


If user selects "Normal speech", DS will ask her/him whether it is spontaneous speech, answering questions or both:



In the report DS will describe normal speech as: "coherent purposeful adequate speech".

If user selected "Motor aphasia", "Sensory aphasia" or "Sensorimotor aphasia", DS will ask her/him, whether aphasia is partial or complete:

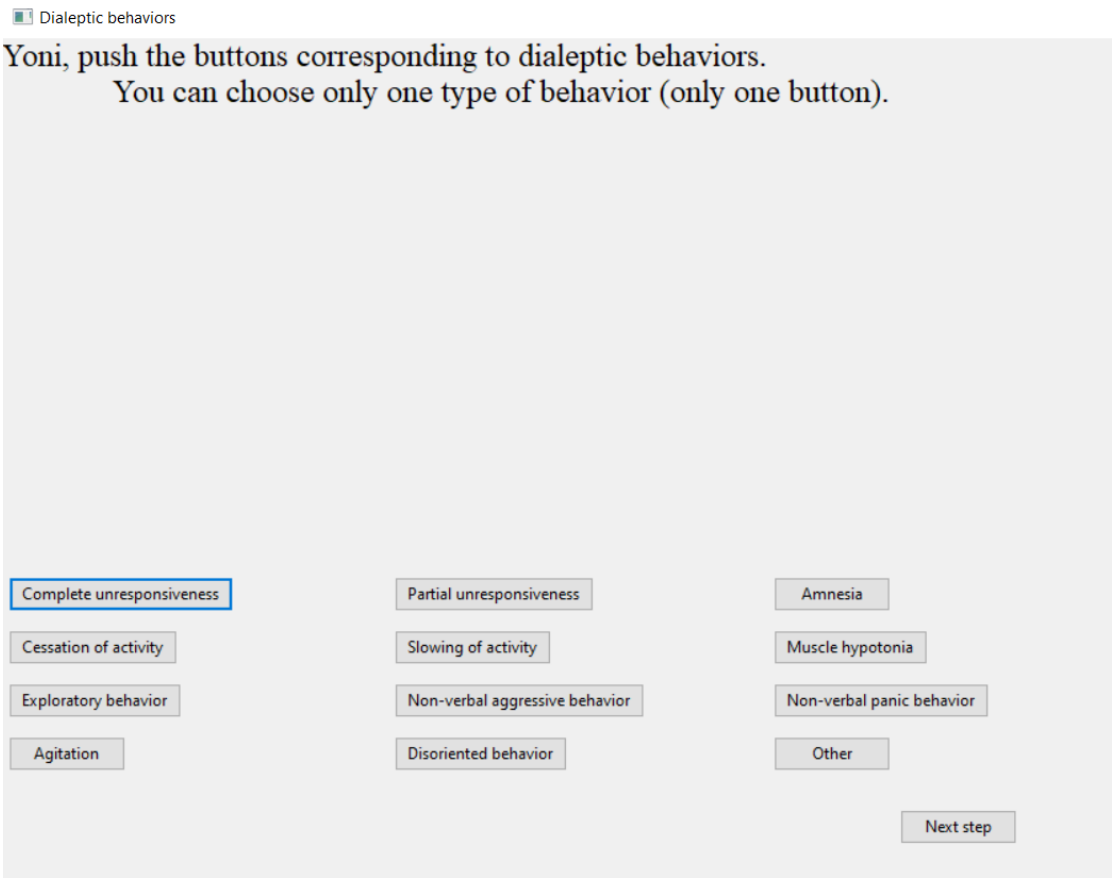


If the user selects the button "Foreign language" from the voice behaviors GUI, DS will include into report following statement:

"the patient speaks, but the interpreter does not understand the language".

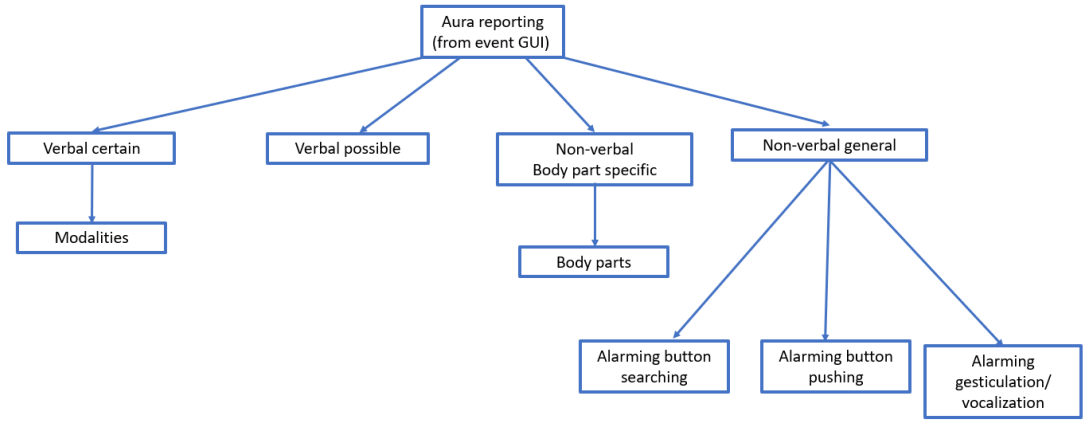
DIALEPTIC BEHAVIORS

After user selected eye movements from the event GUI and inserted event start and end timings, DS opens dialeptic behaviors GUI:

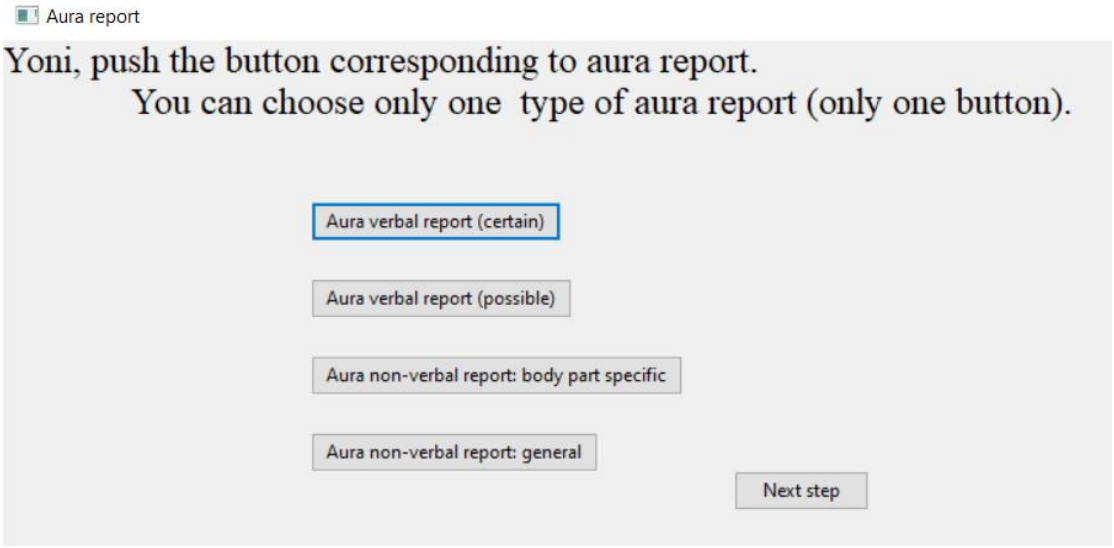


AURA REPORTING BEHAVIORS

While aura can provide important clinical information, it is a filling, and therefore cannot be seen by itself on video, the patient however can report about aura during video/audio recording, therefore aura reporting is included in DS.

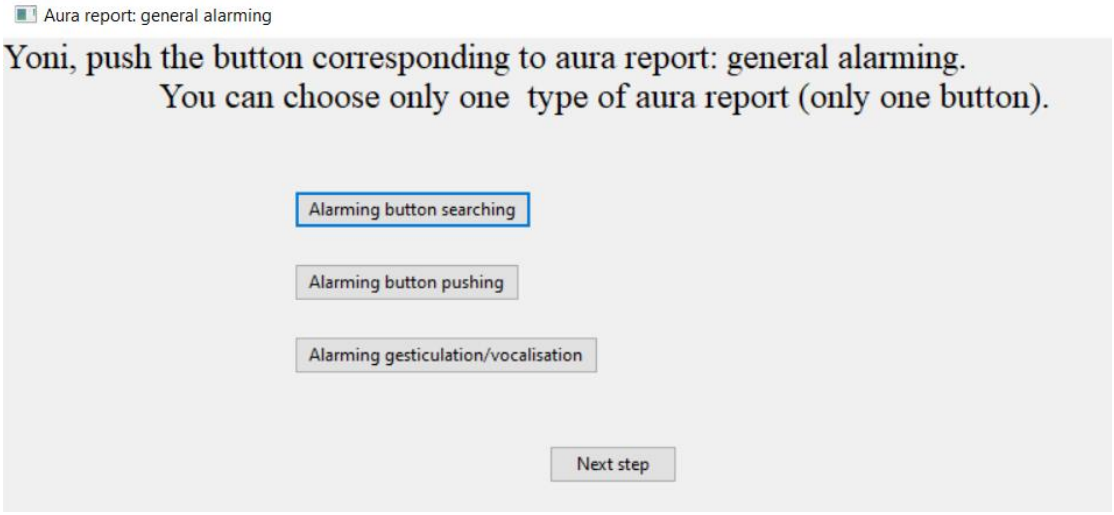


After user selected aura from event GUI, DS opens the Aura report GUI:



When the patient clearly reports the aura by comprehensible words, the user is offered to select "Aura verbal report (certain)". If the patient tries to report aura, but it is not clear what patient says before seizure start, the user is offered to select "Aura verbal report (possible)". If the patient tries non-verbally report aura and indicates some body part, here the user is expected to select "Aura non-verbal report: body part specific". If patient report aura non-verbally by changing behavior without indicating body part, the user is offered to select "Aura non-verbal report: general".

If user selects "Aura non-verbal report: general", DS opens Aura report: general alarming GUI:

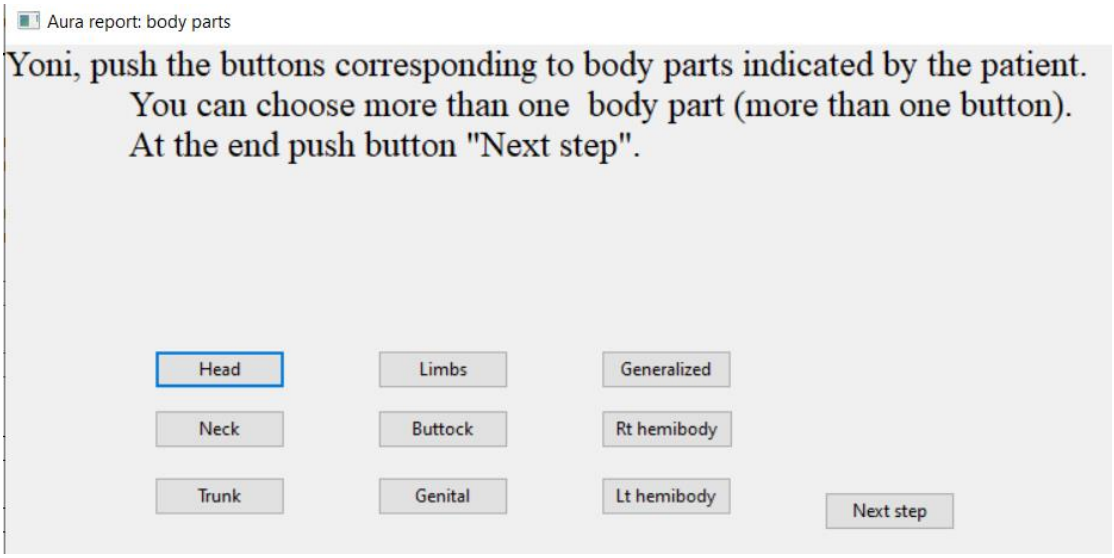


Body parts in aura reporting behaviors

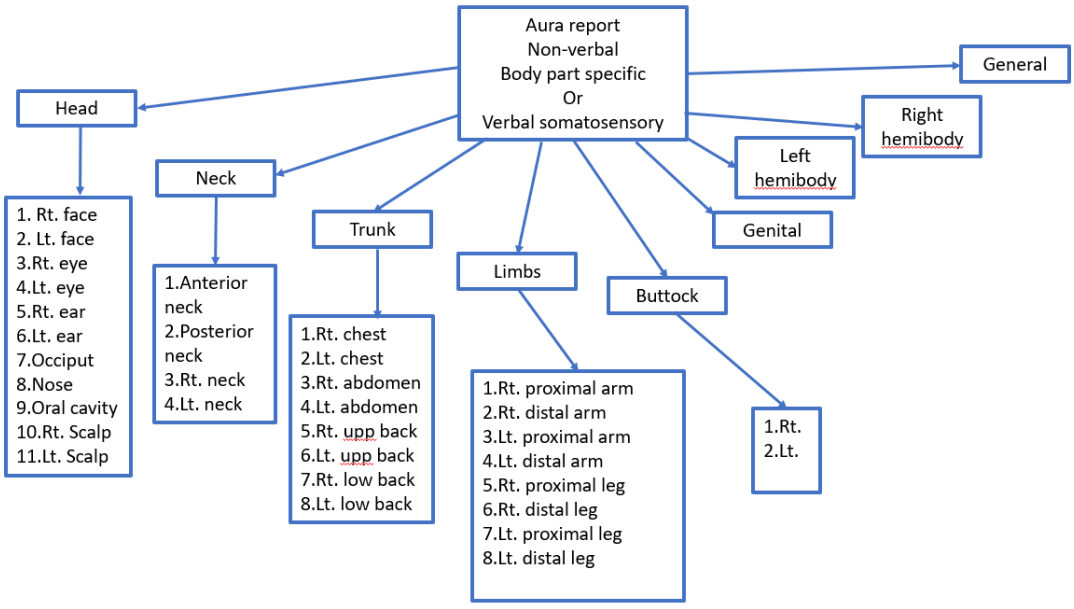
In relation to aura reporting behavior DS refers to body parts in two situations:

1. Aura non-verbal report: body part specific
2. Aura verbal report (certain) – somatosensory aura.

If user selects one of these two options, DS opens the following GUI:

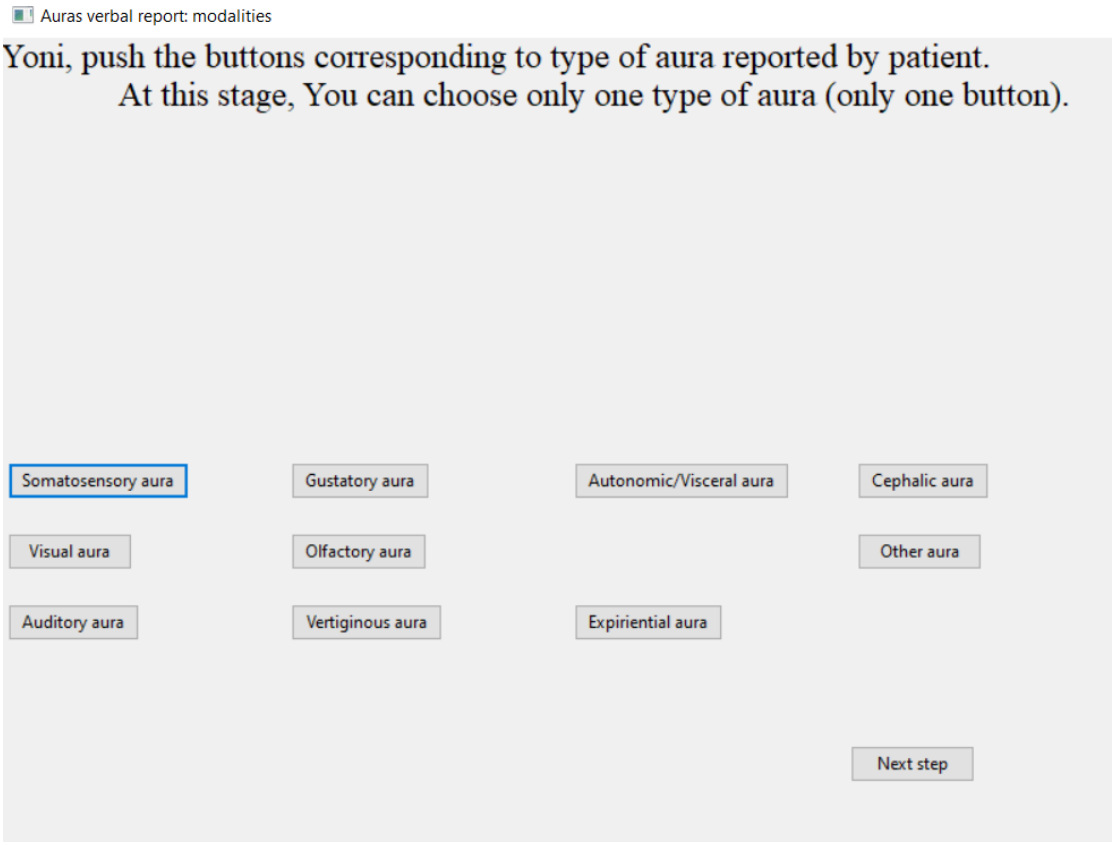


Here is the schematic view of the body parts in aura reporting behaviors:

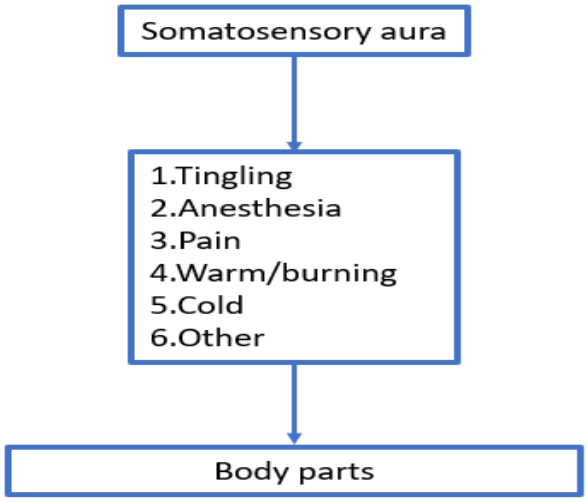


Aura verbal report (certain)

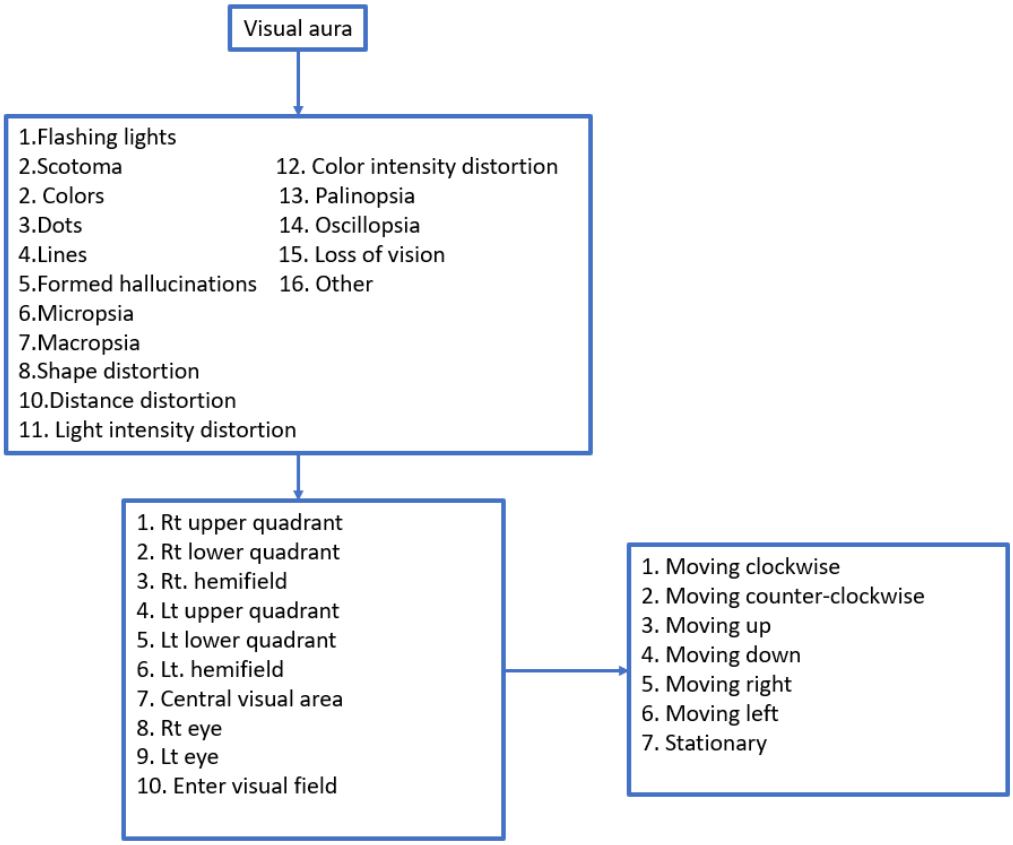
If user selects "Aura verbal report (certain)", DS opens following GUI:



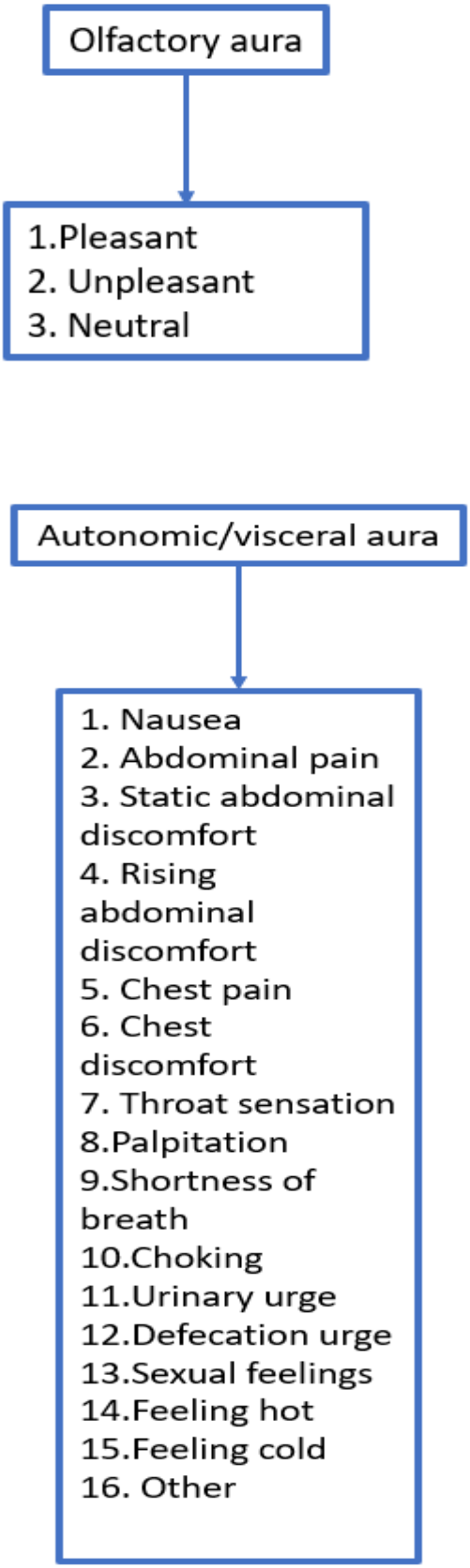
Here are the schematic presentations of aura modalities in DS:

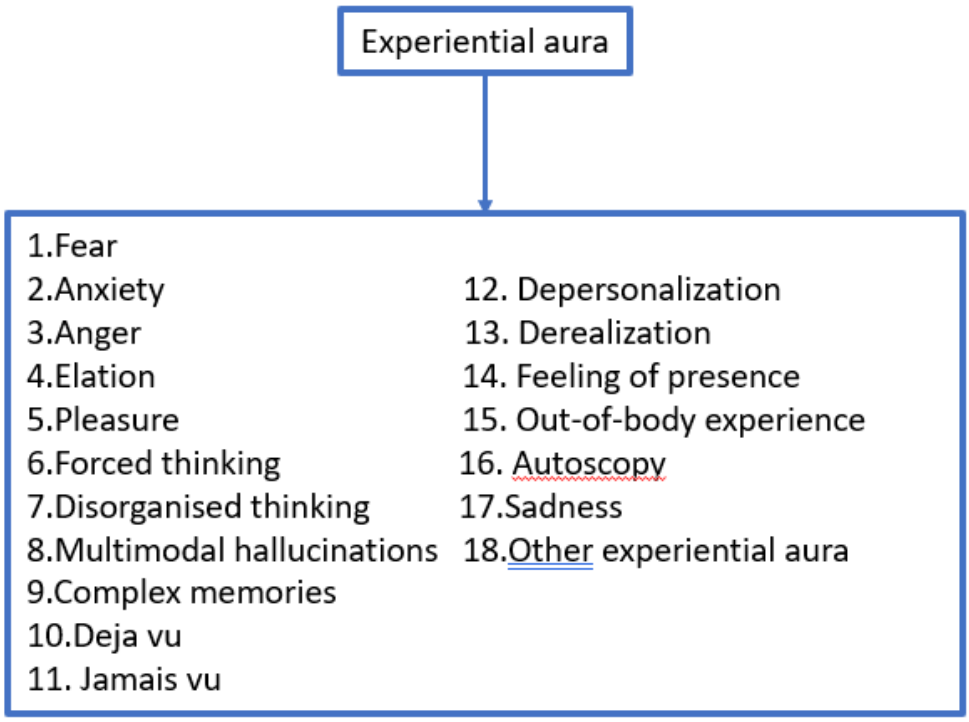


:


Visual aura
1.Flashing lights
2.Scotoma
2. Colors
3.Dots
4.Lines
5.Formed hallucinations
6.Micropsia
7.Macropsia
8.Shape distortion
10.Distance distortion
11. Light intensity distortion
12. Color intensity distortion
13. Palinopsia
14. Oscillopsia
15. Loss of vision
16. Other
1. Rt upper quadrant
2. Rt lower quadrant
3. Rt. hemifield
4. Lt upper quadrant
5. Lt lower quadrant
6. Lt. hemifield
7. Central visual area
8. Rt eye
9. Lt eye
10. Enter visual field
1. Moving clockwise
2. Moving counter-clockwise
3. Moving up
4. Moving down
5. Moving right
6. Moving left
7. Stationary

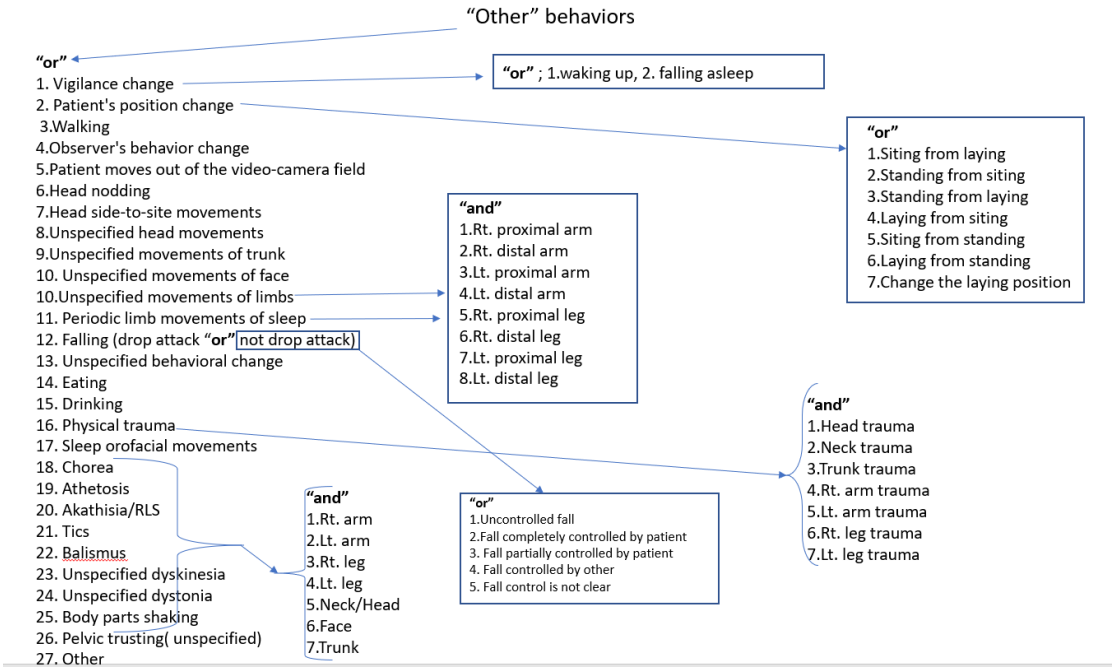

Auditory aura
1. Ringing
2. Buzzing
3. Roaring
4. Voices
5. Melodies
6. Sound distortion
7.Other
1. Right
2. Left
3. Non-lateralized


Olfactory aura
1.Pleasant
2. Unpleasant
3. Neutral
Autonomic/visceral aura
1. Nausea
2. Abdominal pain
3. Static abdominal discomfort
4. Rising abdominal discomfort
5. Chest pain
6. Chest discomfort
7. Throat sensation
8.Palpitation
9.Shortness of breath
10.Choking
11.Urinary urge
12.Defecation urge
13.Sexual feelings
14.Feeling hot
15.Feeling cold
16. Other

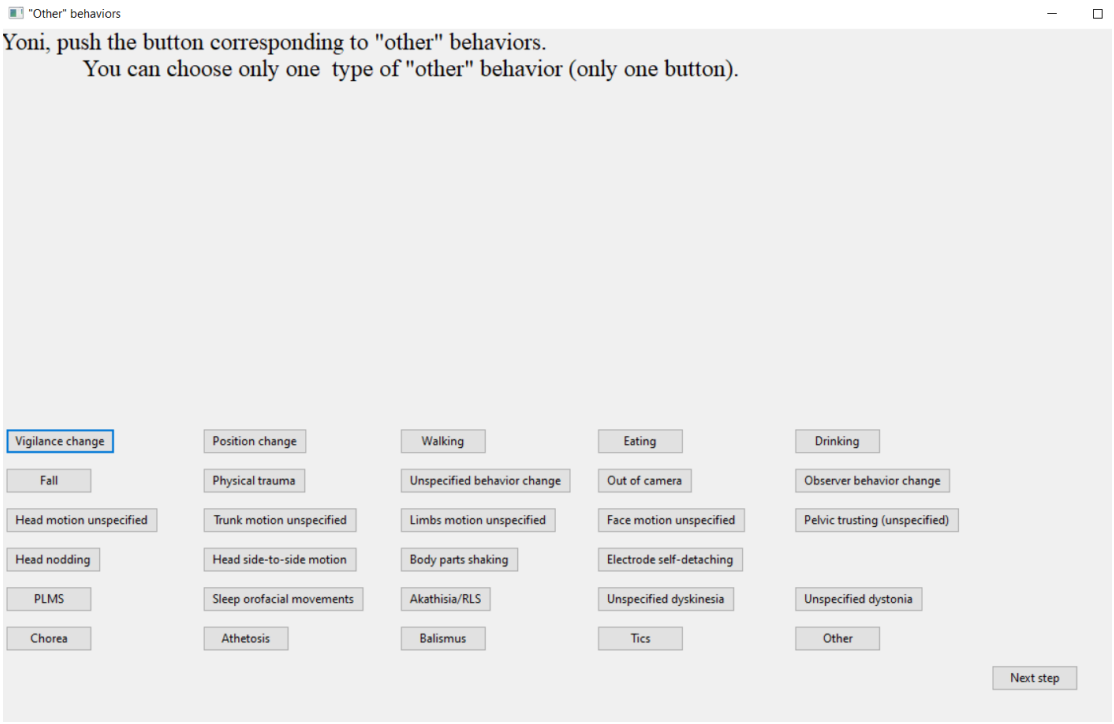


OTHER BEHAVIORS

"Other" behaviors are behaviors that are non-epileptic or non-specific or unclassified:

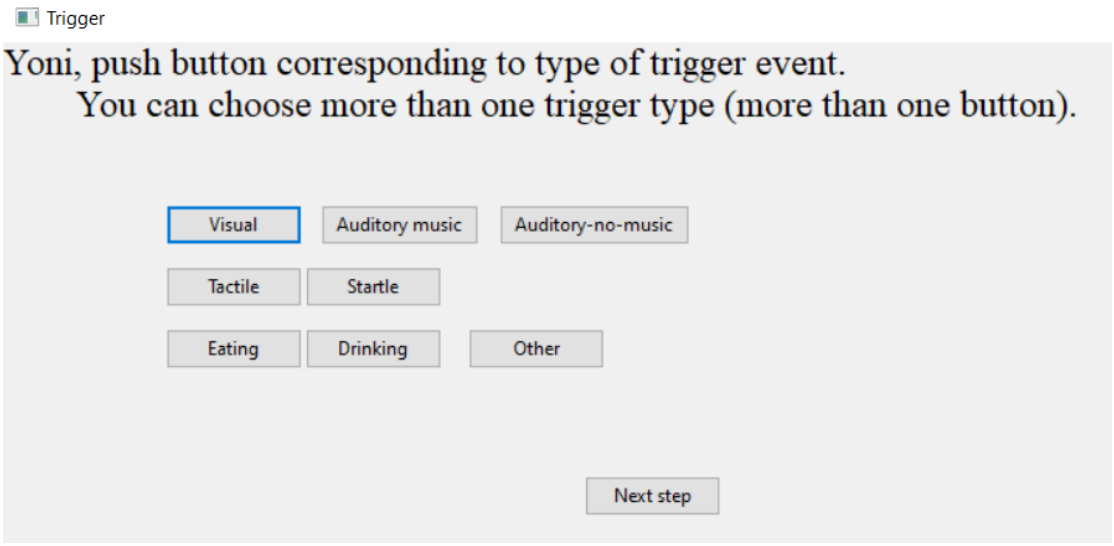


If user selects "Other" from the event GUI, DS opens the "Other" behaviors GUI:



TRIGGER

If user selects trigger from event GUI, DS opens the following GUI:



SEMIOLOGIC EXTRAPOLATION FUNCTIONS ("MACHINE DECISIONS")

DS analyses the data inserted by user and make decisions regarding four aspects:

1. Focality – whether the seizure has focal onset.
2. Motor jacksonian march
3. PNES
4. SUDEP risk

The semiologic exploration functions suggest the possibility that the features given entity (for example PNES) exist or they declare avoidance of decision. They do not suggest the absence of the given entity. In the case that these functions suggest the possibility of entity existence, they provide the reason of this suggestion.

The outcome of semiologic exploratory functions are seen in the Report.

These functions are activated only if in Introduction the user allows DS to make decisions.

Focality criteria

In Digital Semiology, focality is defined according to several criteria.

Unconditioned criteria:

1. Figure of 4 (Rt or Lt)
2. Fencer's posturing (Rt or Lt)
3. Forced head turn (Rt or Lt)
4. Forced gaze deviation (Rt or Lt)

Conditioned Criteria:

1. **Hypermotor** event (if started earlier than 5 sec before start of GTCS).
2. **Aura reporting** event (if started earlier than 5 sec before any other event, except of another aura reporting event or trigger event).
3. Unilateral **tonic** movement of arm or leg or side of face or hemibody (if started earlier than 5 second before start of contralateral tonic or dystonic or clonic movements or contralateral paralysis)
4. Unilateral **dystonic** movement of arm or leg or side of face or hemibody (if started earlier than 5 second before start of contralateral tonic or dystonic or clonic movements or contralateral paralysis)
5. Unilateral **clonic** movement of arm or leg or side of face or hemibody (if started earlier than 5 second before start of contralateral tonic or dystonic or clonic movements or contralateral paralysis)
6. Unilateral **paralysis** of arm or leg or side of face or hemibody (if started earlier than 5 second before start of contralateral paralysis).

Motor jacksonian march criteria

1. Jacksonian march is defined as spatio-temporal clonic movement propagation on the same body side between face, arm, and leg and/or different parts of distal extremities.

2. If between two epochs with clonic movements is an epoch without clonic movements, such sequences are not considered as Jacksonian march.

3. If at the same moment leg and face are simultaneously involved in clonic movements, further propagation is not considered as Jacksonian march.

4. Synchronous bilateral clonic movements (with the same start and end timings) are excluded. However, if not completely synchronous, clonic movements involving both body sides are not excluded, rather considered as to separate sequences.

PNES criteria

1. At least part of time the eyes of patient were closed during ictal episode.
2. The patient resisted the attempt to open eyes by observer.
3. The patient completely controlled her/his fall during ictal episode.
4. The duration of ictal episode was longer than 5 minutes.

SUDEP risk criteria

1. GTCS started from sleep
2. GTCS ended on abdomen
3. After GTCS the patient turned to abdomen
4. Apnea after GTCS
5. Cyanosis after GTCS
6. GTCS (It is important to differentiate between GTCS only and GTCS plus additional SUDEP risk factors.)

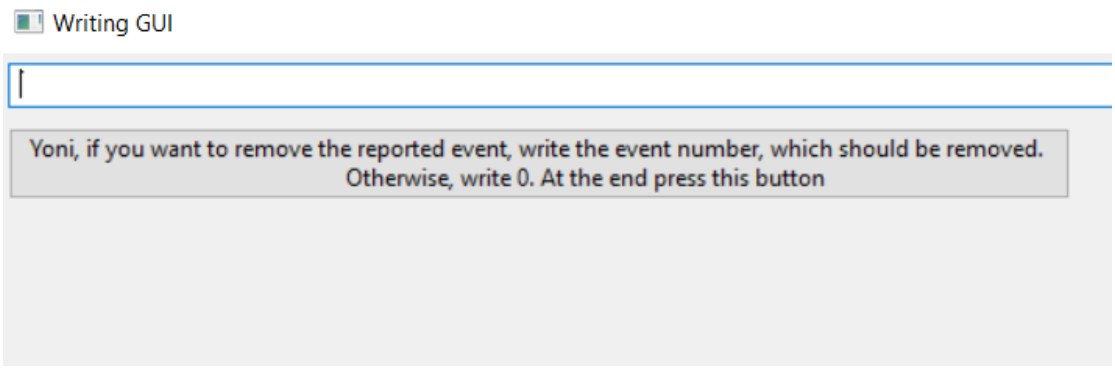
EDITING ICTAL EPISODE

After user finalized EVENTS part, DS continues to the next part, which is EDITING ICTAL EPISODE. The workflow of this part differs between General View Mode and Add-on Mode. Here we will first discuss the workflow of this part for Add-on Mode and then will enter some remarks regarding General View Mode. (User can see the DS workflow scheme.)

In Add-on Mode, after user described all events, she/he should press the Next step button, and DS will proceed to the next step, which is Event removal.

Event removal

At this step, DS presents to user writing GUI and asks user to write the number of event (according to report) that should be removed.



If User writes 0, DS proceeds to next step, which is Editing reported event.

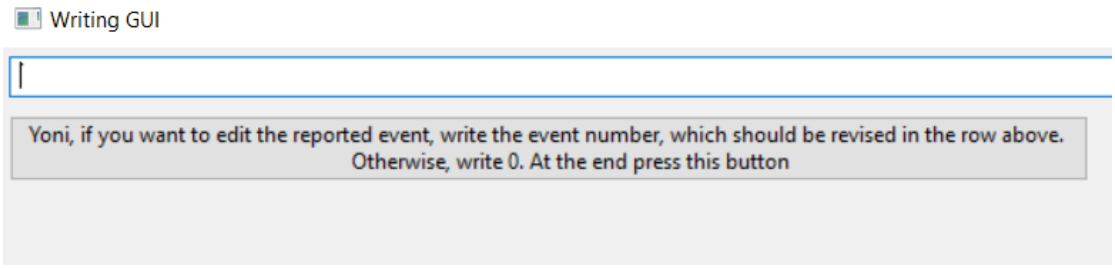
If user writes number that exceeds the number of events in the report, DS will generate error message and will return the same writing GUI.

If user writes existing event number, this event will be removed both from ictus and from the report and the events numeration in the report will be changed appropriately. Thereafter DS will again present the same GUI until user will write 0, then DS will proceed to the next step.

Editing reported event

Here user is asked first to select the event that should be removed, but its content should be changed.

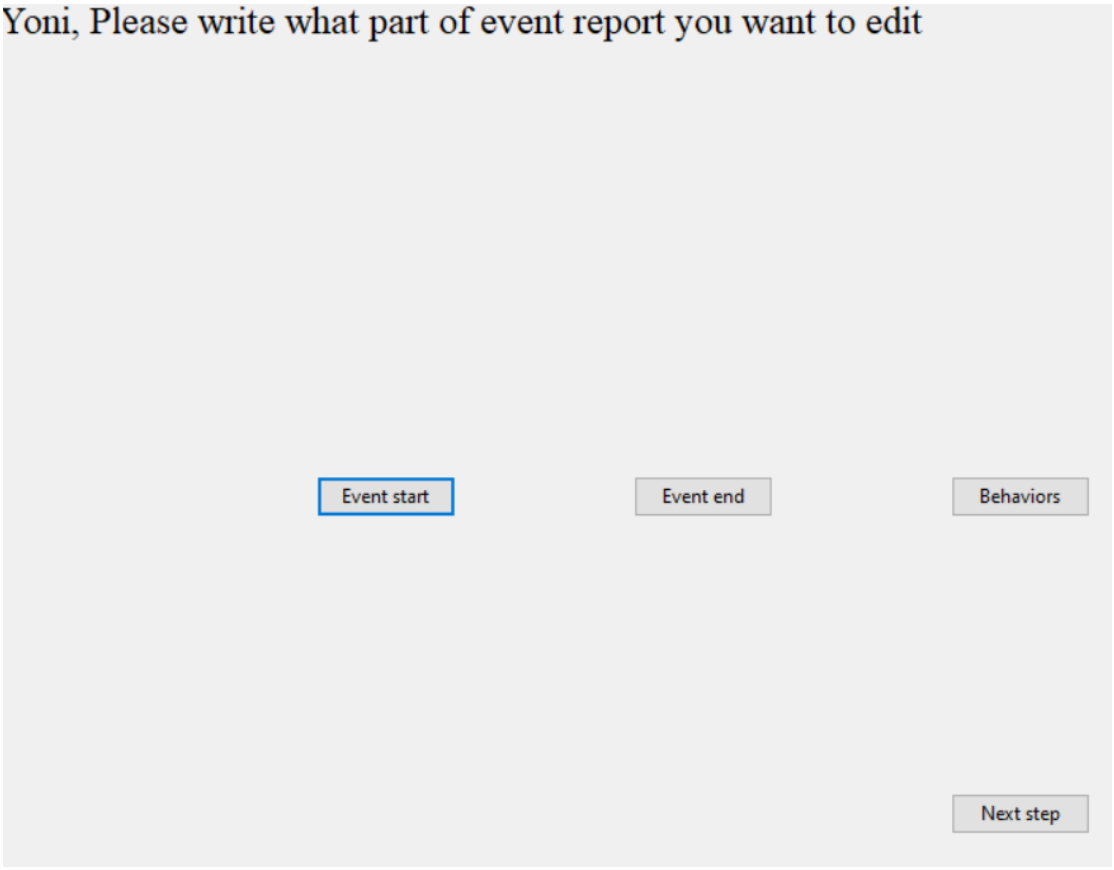
Which reported event should be edited if any?



Here the user is asking to write which reported event should be edited. If user writes 0, no reported event will be edited, and DS will continue to the next step – Edit introduction.

If user writes number that exceeds the number of events in the report, DS will generate error message and will return the same writing GUI.

If user writes existing event number, DS will ask her/him, which part of reported event she/he wants to edit: timing of event start, timing of event end and/or behaviors, the user can select more than one option.



If user selects Event start and /or Event end, DS will open appropriate GUIs. If user will indicate timing of Event start later than Event end, DS will ask user to redefine timings.

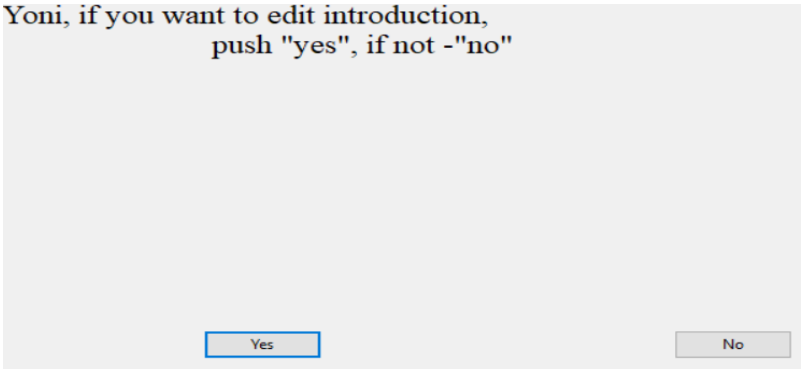
If user selects Behaviors, then DS will open GUIs of behaviors.

If user will press Next step button on the GUI presented just above or in the GUI of behaviors, DS will present the writing GUI asking to write the event number that should be edited or 0.

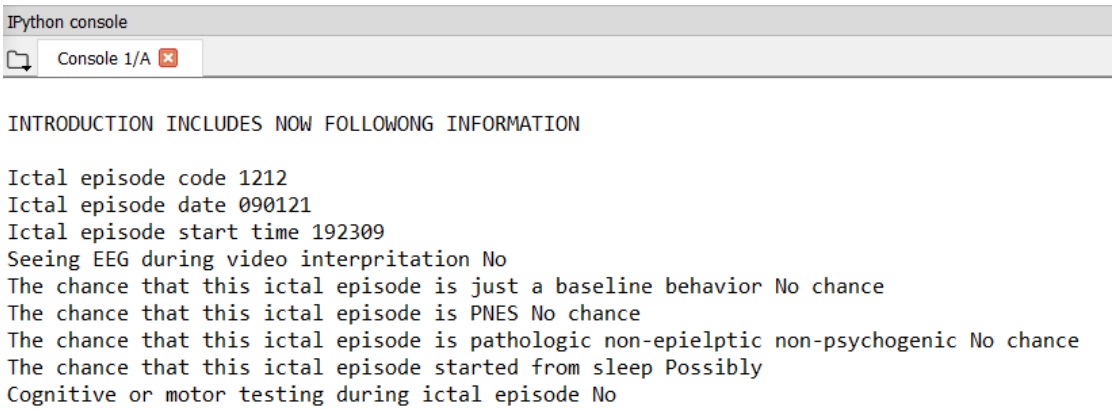
If user selects 0, DS will proceed to the next step.

Editing introduction

At this step, DS opens the selectin GUI asking user whether she/he wants to edit introduction.



At the same time In IPython Console appears the information, which was saved in INTRODUCTION. For example:



This information should help to user to decide whether or not to edit INTRODUCTION.

If user selects Yes, DS moves back to beginning of INTRODUCTION – to the question about ictal episode code, then DS bypasses parts of INTRODUCTION and come to ictal episode date (if ictal episode code did not change) and next DS pass through steps of INTRODUCTION to Motor/cognitive testing. Thereafter DS "jumps" to the next step after Editing introduction – Edit once again. (See DS workflow.)

During the INTRODUCTION editing, DS presents in IPython Console, how the given parameter was defined before editing. For example:

```
The ictal episode code was defined as 1212
```

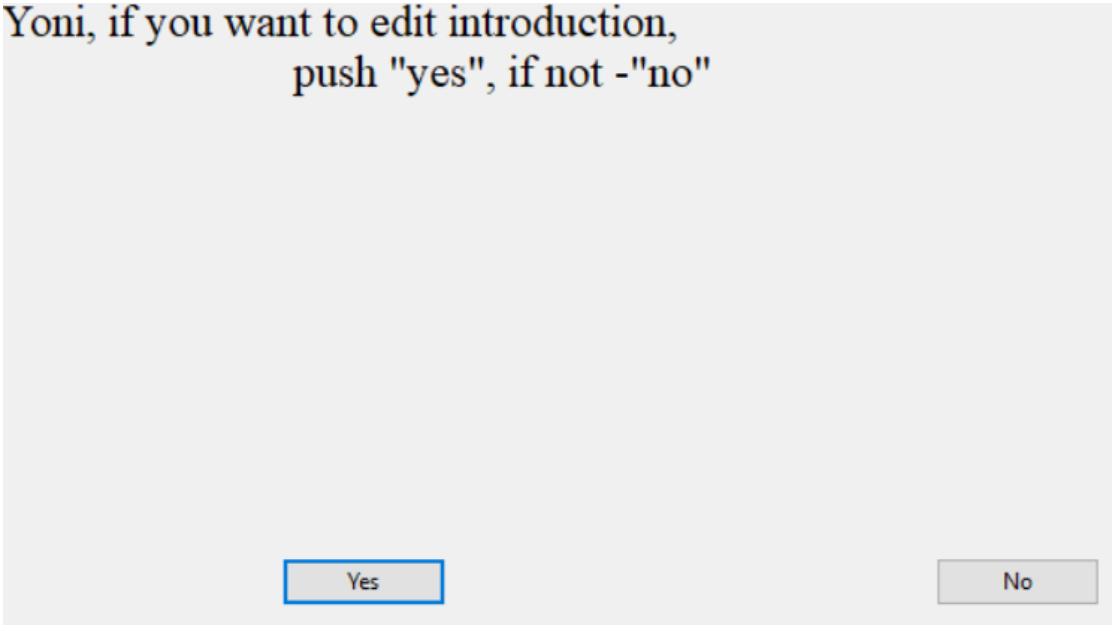
or

```
The ictal episode date was defined as 090121
```

If user selects No in the GUI of Edit introduction, DS continues to the next step – Edit once again.

Edit once again

Here DS presents selecting GUI, asking user to select Yes or No regarding the editing the report once again.



Whatever answer user will select, DS will continue to the next step – Postictal. However, the selection at this step will influence the workflow after Postictal step: if user selected No, DS will continue to offering user to write the final comment; if user selected Yes – DS will come back to Events and will go through whole EDITING ICTAL EPISODE part to Edit once again step.

The next step is Postictal.

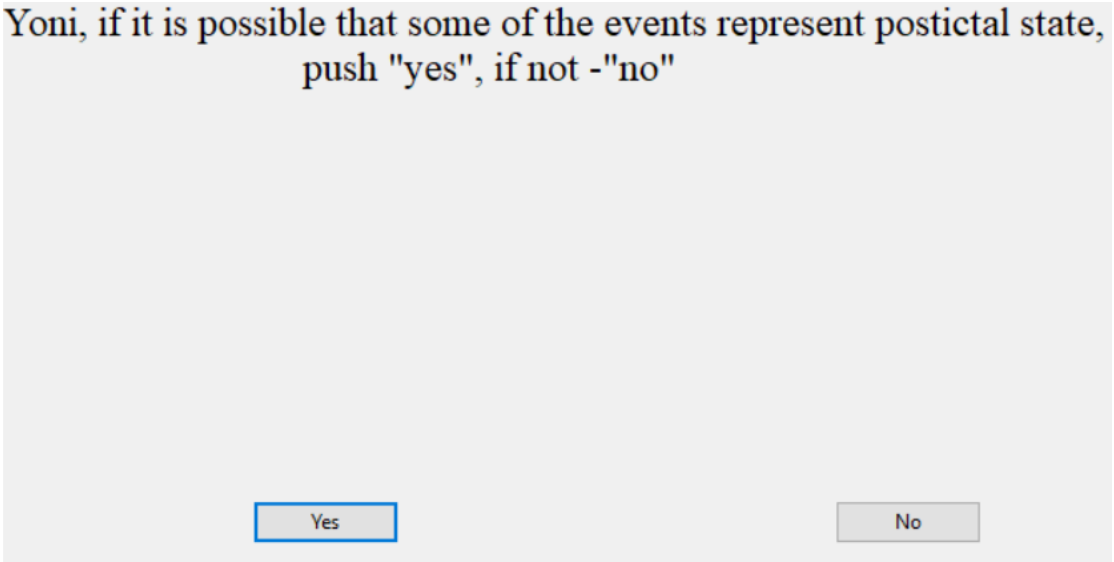
Postictal

This step consists out of two stages:

The first stage is the question whether some of reported events can be postictal.

The second stage is relevant if user select Yes at the first one and it is – which event can be labeled as postictal.

At the first stage DS opens selecting GUI:

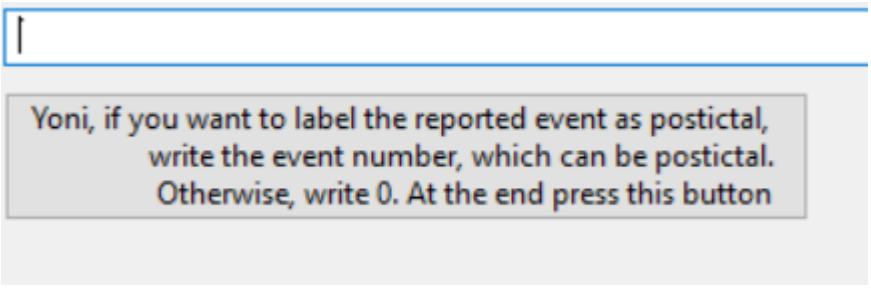


If user selects No, DS continues to the next step – Writing the final comment.

If user selects Yes, the following sentence is written to the report:

```
It is possible that last event\s represent(s) postictal state. This, however, can not be defined
        without taking into account EEG
```

and DS continues to the second stage of Postictal step – opens the writing GUI:



If user writes 0, DS continues to the next step – Writing the final comment.

If user writes the number, which excides the number of events, DS will return the GUI.

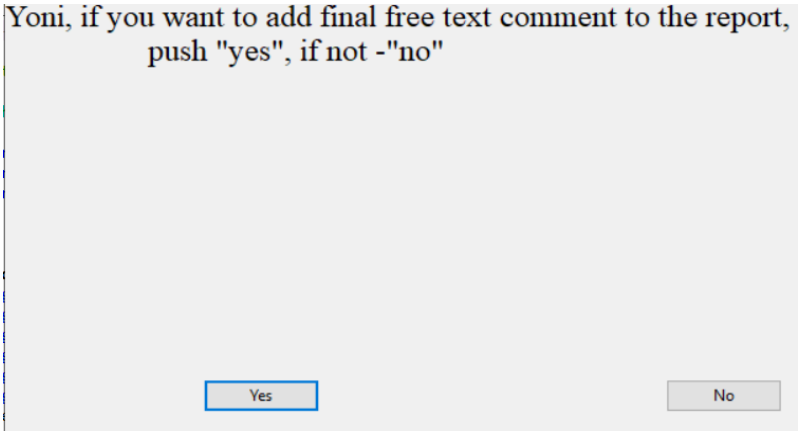
If user writes the existing event number, DS will write following sentence to the report (if for example, event number 1 was selected):

```
The event number 1 was labeled
as possible postictal event.
```

Then DS will again present the same GUI, until the user will write 0.

Writing final comment

This step follows Postictal step.DS presents selecting GUI, asking whether the user wonts to write final free text comment.



If user selects No, DS will continue to the next step – DS exit.

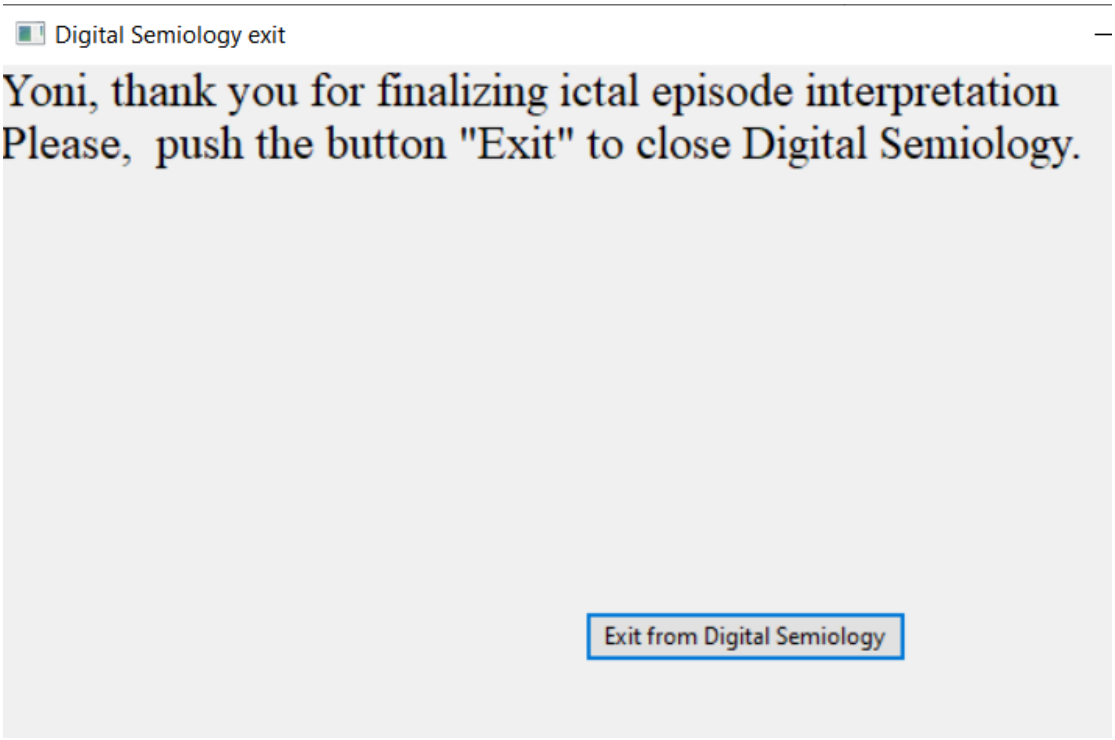
If user selects Yes, DS generates the following sentence in IPython consol:

```
Software:
Yoni, please, write final free text comment and, at the end press Enter
```

The user can now write her/his comments in the IPython console and this comment will be included into report. After user will press enter, DS will continue to the next step DS exit. The new file or changes to the old file will be saved.

DS exit

Here the user is asked to press button on the GUI and exit DS.



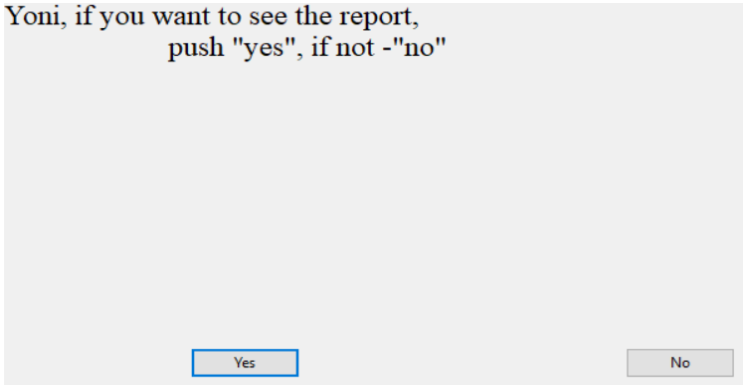
The nuances of ictal episode editing in General view mode

In previous sections we discussed the ictal episode editing in Add-on mode, the same process in General view mode is similar, but there are some differences. User can

see the DS workflow scheme. Most of the steps of ictal episode editing in General view mode are the same as in Add-on mode, however the three initial stems are different.

1. *See the report*

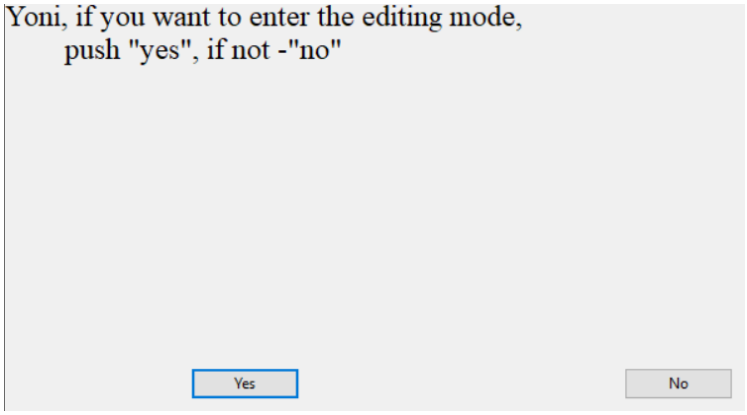
   After the user finalized describe the events, DS opens the selecting GUI. The user is offered to select, whether she/he wants to see the report.



   With any answer, DS will continue to the next step – Editing mode.
   If user selected Yes, DS will present report in the IPython console, if No – the report will be not generated at this step (but will be generated further).

2. *Editing mode*

   Here DS asks user whether she/he wants to edit the ictal episode



   If user answers No, DS bypasses the steps of editing ictal episode and come directly to Writing final comment step.
   If user answers Yes, DS proceeds to Event addition step.

3. *Event addition*

   This step is identical to EVENTS part of Add-on mode.

DIGITAL SEMIOLOGY REPORT

DS Report: Overview

DS report describes ictal events and behaviors. The parts of introduction (except of ictal episode date) are not included into report. These parts, however, are encoded in ictus.

The events in the report are listed and numerated in chronological order according to time of event start.

The timings in the report are defined according to the time scale of video-player, which starts as 0 at the beginning of video.

DS report includes several parts:

1. Short description of ictal episode.
2. Detailed description of ictal episode.
3. Statements regarding postictal state.
4. Final comment.

Short description of ictal episode

This part of report includes ictal episode date and list of events including their timings.

For example:

```
                    DIGITAL SEMIOLOGY REPORT


    The date of this ictal episode is 11.11.2011.

This ictal episode includes following events:
Event number 1 00:36:31-00:37:20 dialeptic behavior event
Event number 2 00:36:48-00:37:25 motor automatism event
Event number 3 00:37:42-00:38:41 generalized tonic-clonic event
Event number 4 00:38:41-00:39:51 dialeptic behavior event
```

Detailed description of ictal episode

This part of the report includes list of events and behaviors.

For example:

```
The detailed descrption of the ictal episode:

EVENT NUMBER 1 00:36:31-00:37:20 dialeptic behavior event
This event includes following behaviors:

1)Patrtial unresponsiveness.


EVENT NUMBER 2 00:36:48-00:37:25 motor automatism event
This event includes following behaviors:

1)semipurposeful movements involving:
         left  arm (mild)


EVENT NUMBER 3 00:37:42-00:38:41 generalized tonic-clonic event
This event includes following behaviors:

1) GTCS ended in the position on the back.
 After GTCS the patient turned to position on the back.
The time of trasition from tonic to cloic phaze of GTCS is 00:38:13.

EVENT NUMBER 4 00:38:41-00:39:51 dialeptic behavior event
This event includes following behaviors:

1)Complete unresponsiveness.
```

Statements regarding postictal state

Here are the statements: a) whether it is possible that some of reported events are postictal and b) which events user labeled as possible postictal.

For example:

```
It is possible that last event\s represent(s) postictal state. This, however, can not be defined
       without taking into account EEG
-----------------------------------------------------------------------------------------------
The event number 4 was labeled
as possible postictal event.
```

Final comment

If user wrote the final comment, it will be presented in this part of report.

For example:

```
------------------------------------------------

                  FINAL FREE TEXT COMMENT
It is hard to make a conclusion without EEG!
```

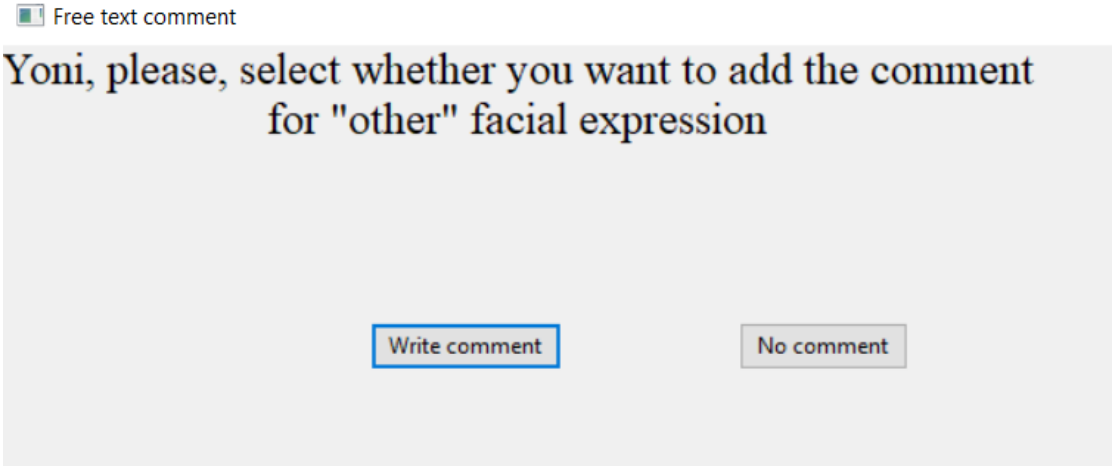
Writing free text comments in DS

Final comment is only one of the ways to write free text comments in DS. Another opportunity to write comment is when user defines the behavior as "other". In DS it

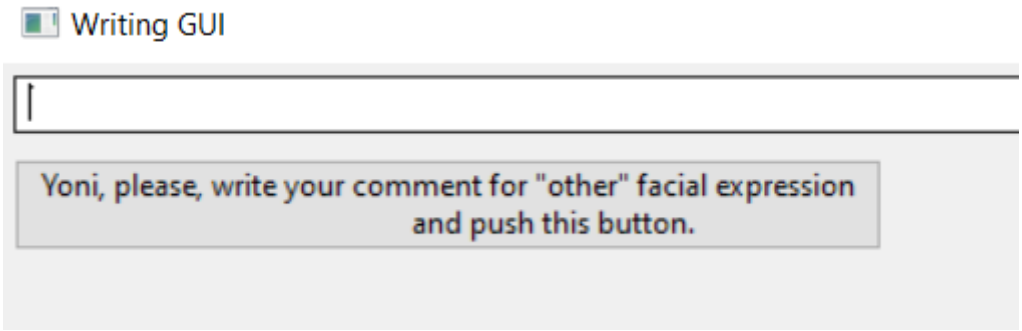
is a difference between "other" behavior and "unspecified" behavior. "Other" behavior is one that can be describe in specific terms, but these terms do not exist in the present version of digital semiology. In contrast, "unspecified" behavior is difficult to describe in specific terms. When user selects "unspecified" behavior, DS does not open possibility to write comment.

For example, if user wants to describe the face expression of patient as "thinking' but this specific term is absent in DS, then user can select "other" face expression and add the comment "thinking". In another case, when user can not specify the face expression in existing terms of DS but cannot imagine any specific term to describe this behavior, she/he can select "unspecified" face expression – then DS will not open the possibility to write comment.

When user selects "other" behavior, DS will open selecting GUI presenting three choices, for example:



If user selects "Write comment ", DS will open the writing GUI:



After user wrote the comment, DS will continue to the next step.

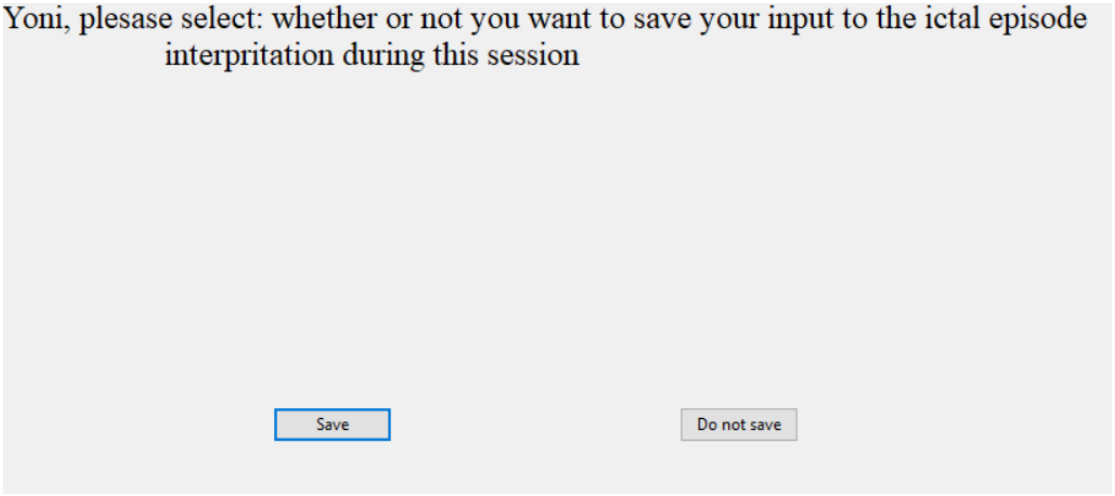
If user selects "No comment", DS will continue to the next step, without opening possibility to write comment.

The final comment before exit from the program DS allows to write in the console only.

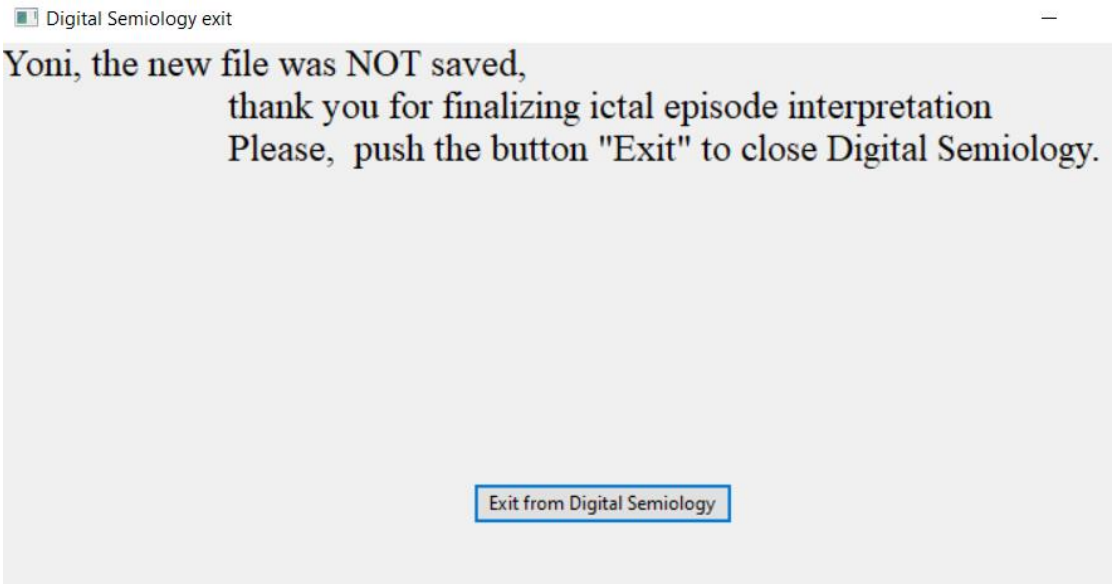
EXIT FROM DS

After finalizing ictal episode encoding, DS bring the user to the point when she/he should exit DS (see paragraph DS exit). In this case all changes will be saved.

It is possible also to exit from DS without going through all the way to the end. In the GUI of Events in Add-on mode (the same GUI in Add event step in General view mode). In this GUI if user selects Exit button, DS will ask the user whether she/he wants to save the changes – DS will open the GUI:



If user selects "Do not save", the changes of the session will not be saved, and DS will present the following GUI:



If user selects "Save", the changes will be saved. In this case DS opens the following GUI:

Presenting words "new file" DS means here either the changes to old file or the new file created during the session.

In addition, it is the possibility to exit from DS using GUIs in INTRODUCTION. Some GUIs have button: "Exit without saving" and other GUIs – Exit (In this case DS will open the GUI asking the user, whether to save the new file).

THE PROGRAM UNPACK_ICTAL_EPISODE

DS creates .dat binary file. To create .txt file user should employ another python-based program – unpack_ictal_episode (UIE).

UIE creates both .txt file and graphical presentation of ictal episode.

UIE code should be placed in the same directory with DS code. It is possible to operate UIE both in Spyder and without Spyder (just by double click on the UIE icon).

When UIE is open, it writes in the console the following statement:

```
Please, write the code of this ictal episode and then press Enter
```

After the user writes (or pastes) the code of ictal episode into console and presses Enter, UIE reads the .dat binary file of the ictal episode (according to code, which is a part of .dat binary file name), UIE presents in the console the report and ictus (encoded values) of this ictal episode.

UIE also sometimes displays some warning massage from MatPlotLib module,

For example:

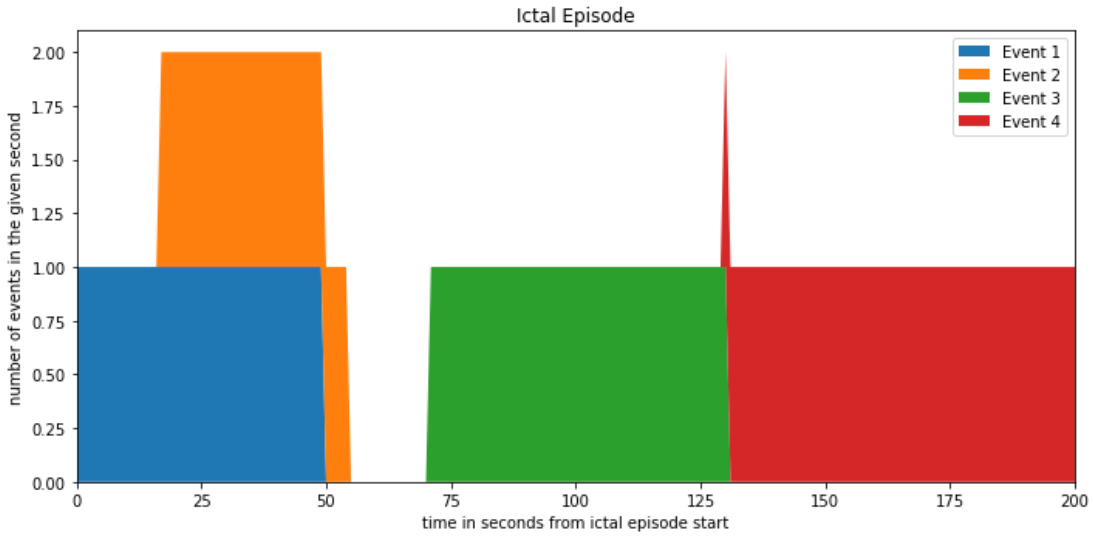
```
Unrecognized location "". Falling back on "best"; valid locations are
        best
        upper right
        upper left
        lower left
        lower right
        right
        center left
        center right
        lower center
        upper center
        center

  % (loc, '\n\t'.join(self.codes)))
```

We recommend ignoring this massage.

At the end, UIE displays in the console following statement:

```
Press Enter to exit
```

When user press Enter, the UIE is closed and the graphical presentation of ictal episode (GPIE) appears in the console, for example:



GPIE is created by MatPlotLib module. Different events are shown by different colors. The correspondence between colors and the event numbers are shown in right upper angle of the GPIE. The details about events user can find in the report.

The x-axis of GPIE is a time in seconds from the beginning of video (the same timescale as in the report). If the events overlap in time, then one of the events is shown on the top of another during the overlapping epoch. The y-axis is number of events in a given timepoint (in a given second). We recommend ignoring values that are not integers in the y-scale.

To save the GPIE we recommend using Snipping tool:

https://support.microsoft.com/en-us/windows/use-snipping-tool-to-capture-screenshots-00246869-1843-655f-f220-97299b865f6b

The .text file describing the ictal episode will be found in the same directory as UIE after exit from UIE.

THE PROGRAM DS_VERSION_ADAPTER

User can modify the code of DS. For example, she/he can add behaviors to the events. This can enlarge the lists of behaviors in ictus. Therefore, new versions can be inappropriate for editing of the files created by old versions.

To avoid this problem, user can apply DS_version_adapter, which is Python code, able to convert the .dat files created by old DS version to be readable by new version.

The authors recommend updating DS_version_adapter, so that the length of list of behaviors correspond to new DS version.

```
#This code adapts .dat files to version 07.02.2021 of Digital Semiology
import pickle
ictal_episode_code=input('Please, write the code of this ictal episode a
DS_version=input('Please, write the version of Digital Semiology and the
filename='ictal_episode_' + ictal_episode_code +'.dat'
f=open(filename, 'rb')
ds_starter = pickle.load(f)
ictus=pickle.load(f)
report=pickle.load(f)
software_user_dialogue=pickle.load(f)
f.close()
name_changer=0
for i in range (0,len(ictus)):
    for j in range (4,len(ictus[i])):
        if ictus[i][0]==0: #Simple motor
            if len(ictus[i][j])<52:
                ictus[i][j]=ictus[i][j]+['']*(52-len(ictus[i][j]))
                name_changer=1
        if ictus[i][0]==1: #Automatisms
            if len(ictus[i][j])<12:
                ictus[i][j]=ictus[i][j]+['']*(12-len(ictus[i][j]))
                name_changer=1
        if ictus[i][0]==2: #Autonomic
            if len(ictus[i][j])<4:
                ictus[i][j]=ictus[i][j]+['']*(4-len(ictus[i][j]))
                name_changer=1
        if ictus[i][0]==3: #Eye movements
            if len(ictus[i][j])<23:
                ictus[i][j]=ictus[i][j]+['']*(23-len(ictus[i][j]))
                name_changer=1
        if ictus[i][0]==4: #Hyperkinetic
            if len(ictus[i][j])<12:
                ictus[i][j]=ictus[i][j]+['']*(12-len(ictus[i][j]))
```

The red arrows show the numbers that can be changed if in the new DS version new behaviors are added.

After the code is activated by pressing Run button, the program will present to user the following instruction in IPython console:

```
Please, write the code of this ictal episode and then press Enter
```

After user wrote ictal episode code and pressed enter, the program will present the second instruction in IPython console:

```
Please, write the code of this ictal episode and then press Enter
```

After that, if needed, the program will modify ictus variable in the .dat file, adapting it to the current DS version.

If ictus variable in .dat file was changed, the program will rename the .dat file. For example, if the old file was ictal_epilsode_1212.dat, the new file will be

ictal_episode_1212_vers07.02.2021.dat (if .dat file was adapted to the version 07.02.2021 of DS).

If the .dat file corresponds to the current DS version, then DS_version_adapter will generate in IPython console the following statement:

```
ictal_episode_1212.dat doesn't require adaptation to the current Digital Semiology version
```

If user indicated ictal episode code of non-existing .dat file, DS will generate the following error message:

```
FileNotFoundError: [Errno 2] No such file or directory: 'ictal_episode_12.dat'
```