|  |
| --- |
| Trackit-V3 |
| Standard Operation Procedure |
| By Andreas Wulff-Abramsson |
|  |
| 14/11/2023 |

Content

[What is TrackIt 3](#_Toc155179650)

[Prerequisite 3](#_Toc155179651)

[How to install 3](#_Toc155179652)

[Menu items 4](#_Toc155179653)

[Home menu 4](#_Toc155179654)

[Configuration 4](#_Toc155179655)

[Show Base Configuration 5](#_Toc155179656)

[Show Game Configuration 8](#_Toc155179657)

[Show Serial Configuration 13](#_Toc155179658)

[How to Run Trackit 14](#_Toc155179659)

[Write your own sequence mode 17](#_Toc155179660)

[Random event Generation mode 20](#_Toc155179661)

[Configuring Trackit 24](#_Toc155179662)

[Trackit Datafiles 26](#_Toc155179663)

[Events 26](#_Toc155179664)

[InputData 27](#_Toc155179665)

[Statistics 27](#_Toc155179666)

[How to Run SVIPT 28](#_Toc155179667)

[Write your own sequence mode 28](#_Toc155179668)

[Random event Generation mode 32](#_Toc155179669)

[SVIPT Datafiles 35](#_Toc155179670)

[SVIPTEvents 36](#_Toc155179671)

[InputData 36](#_Toc155179672)

[SVIPTTrials 37](#_Toc155179673)

[References 37](#_Toc155179674)

# What is TrackIt

Trackit is a simple game, where the subject has to reach different rectangular targets as fast and precise as possible. The game can be customized in different ways to support the very nature of the experiment.

# Prerequisite

Before running Trackit for the first time you need to have the following software installed on the computer:

Python version 3.11.4

With PIP install the following packages to python.

* Pygame
* Dearpygui
* pyDAQmx
* numpy
* Pillow
* Desktopmagic
* Pywin32
* Pyserial

With these software installations, you are able to run Trackit\_v3

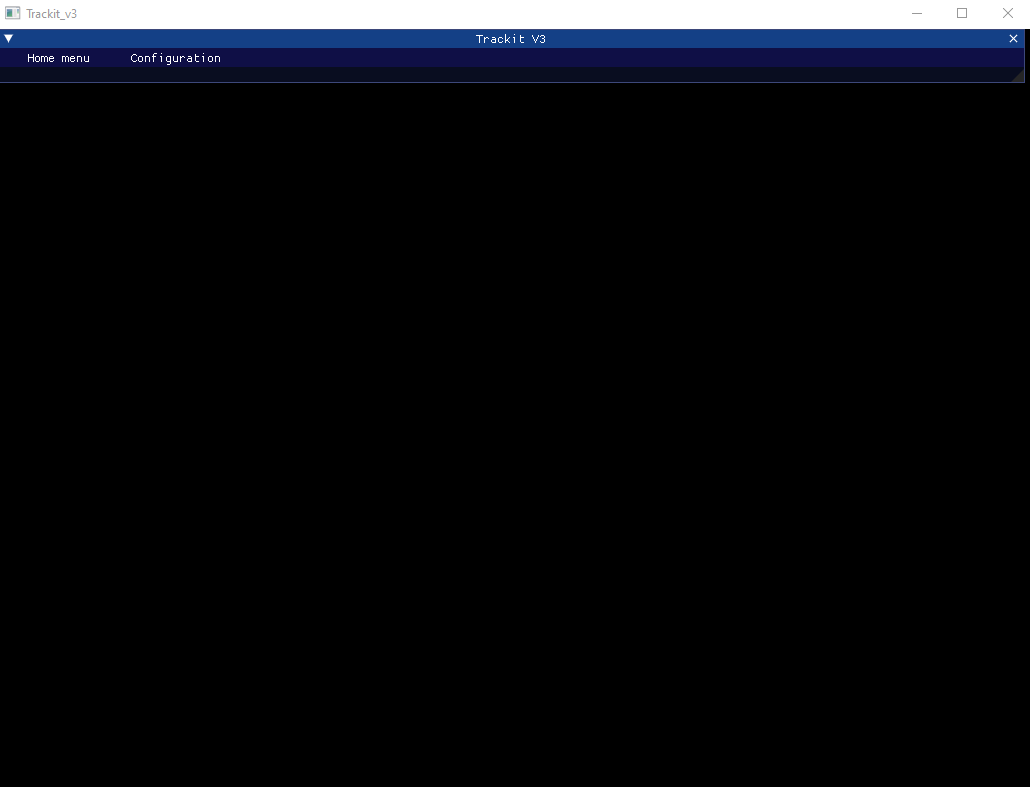
**Optional -**  if you want to connect a sensor to Trackit and control it through it connect a NiDAQ board from national instruments or a microcontroller board configured by UCPH Electronics workshop like a STM32.

**Optional –** If you want to send out triggers connect a NiDAQ board to your system it to distribute trigger signals

# How to install

When the prerequisite items has been for filled you are able to run Trackit\_v3 from the very folder you have placed it.in the Trackit\_v3 folder find “Trackit\_v3\_mainmenu.py” open it with IDLE, , visual code, or your favorite python IDE or run it through the terminal.

As a result, The following output will appear.



# Menu items

## Home menu

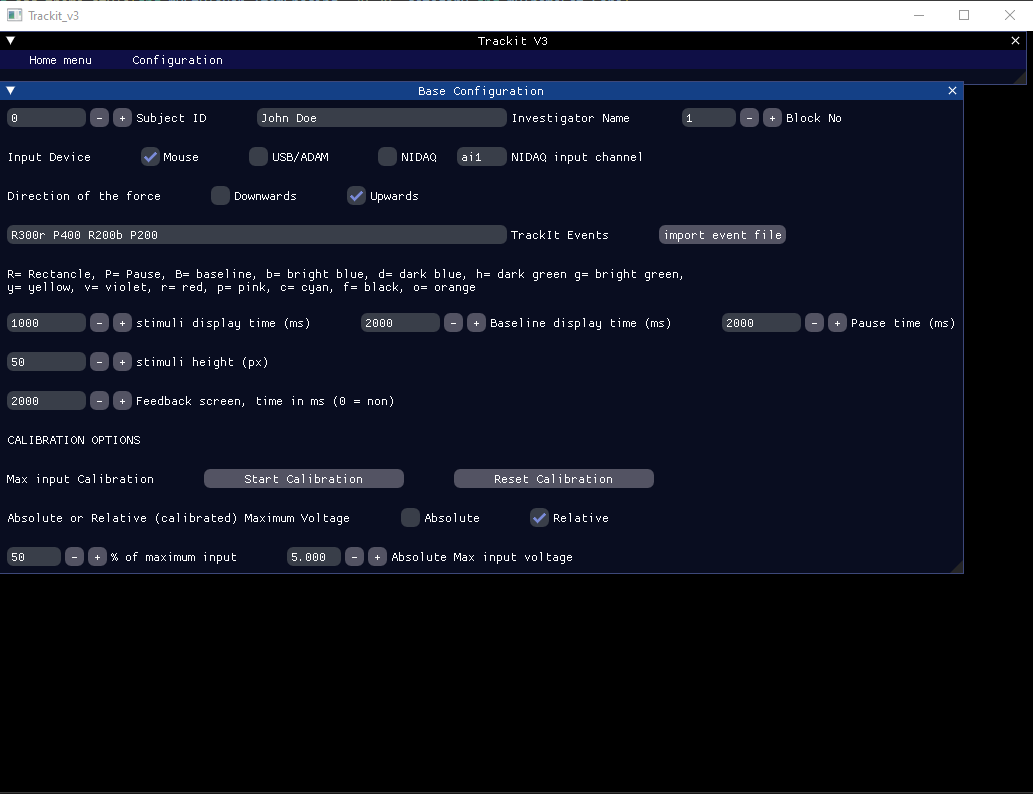
This menu item contains the main basic elements of any software. In here you will be able to start Trackit, Save your configuration, Load previous configuration, look at high scores and quit Trackit.

The most important elements in here are the save configuration and start Trackit. You will need to save your current trackit configurations before running trackit. The system will look for that configuration file after you have played and save it together with the data files created by the play through.

## Configuration

Configuring Trackit requires you to be aware of the different configuration panels. Base, Game and Serial, especially the two former are important for most of your experiments.

### Show Base Configuration



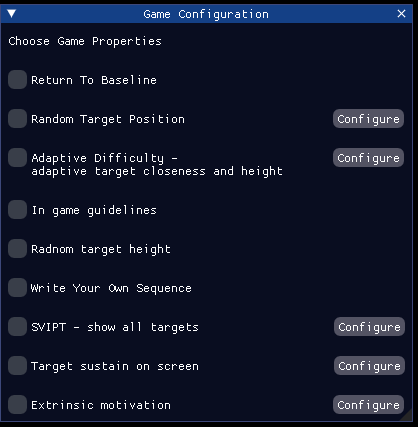
This menu will configure the core parameters of the Trackit experience. It concerns who the subject is and who the investigator is. The main input and output parameters. Lastly, the menu concerns calibration and how to use the calibration. From the top let’s look at the different elements

1. **Subject ID** – This is the identifier of the subject currently being investigated
2. **Investigator Name** – The name of the researcher conducting the experiment this can also contain project acronym for better data overview.
3. **Block No** – The number of the block currently being investigated

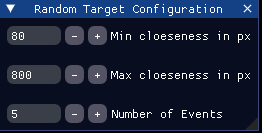
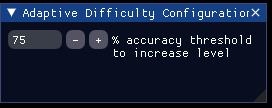
It is important that you keep track of the three upper item as these will be used to name the configuration and data files exported from the program after a successful Trackit run

1. **Input Device** – Here you choose whether you are working with the mouse, a sensor integrated into a microcontroller configured by UCPH Workshop (USB/ADAM) or a sensor connected to a NIDAQ national instruments board (NIDAQ). It is likely that the two latter will be for your experiment while the first option is primarily for testing and ´sitting up the experience.
2. **NIDAQ input channel** – this is the physical analog input port you have connected the sensor to the NIDAQ board, thus you are able to change sensor if there are more than one connected to the board.
3. **Direction of the force –** This item denoted the direction of the player when force is applied on the sensor, This is handy if you have an exercise where you press your foot down you will expect the cursor on the screen to move downwards. Likewise, if you raise your foot in order to adjust the angle you expect the cursor to move upwards.
4. **TrackIt Events –**  In this field you are able to write your very own Trackit sequence. There are three types of events you are able to create Baseline (B), Rectangle (R), and Pause (P). Baseline will always appear at either the bottom or the top of the screen dependent on the direction of the force **(6)**. When writing a baseline event you write e.g “B2000r”, where “B” is for baseline, “2000” is the amount of ms baseline is present on the screen, “r” is the color of the baseline. In this case, the color is red. You can see in the configuration menu a number of colors to choose from. Rectangle (R) is a bit different. It is written in the same way “R600g”, here “600” denotes the position on the screen. Pause (P) is the same syntax as a baseline. The only difference is that the color notation is irrelevant as the Pause is invisible and is a break between events. It shall be noted that all events have to be separated by space.
5. **Import Event File –** This button allows you to import Trackit events written in a Text file. Formatted in the same way as described in **(7)** e.g “B1500r P1000r R600g P1000r R400v” Note that all events are separated by spaces.
6. **Stimuli Display Time (ms) –** This information is used to tell how long time each of the Rectangle events denoted in (**8**) is shown on the screen
7. **Baseline Display Time (ms) –** This information is only used when “Return to Baseline” game mode is used, as it will create the baselines with the given time noted in this input box.
8. **Pause Time (ms) –** This time information is used when the researchers uses the game modes called “Return to baseline” and “Random Target Position”. In these games modes this variable denotes the lengths of the time the Pause is present between Rectangles and Baselines.
9. **Stimuli height (px) –** The size of the targets and the baseline, use this parameter as one way to adjust the difficulty level of your TrackIt game.
10. **Feedback screen, time in ms (0=non) –** denotes the amount of time feedback is shown to the subject after they have successfully completed a Trackit run.
11. **Start Calibration –** This button opens up a simple game where the subject has to apply the highest input into the system they are able to. The system provides direct feedback with a raw input number coming from the sensor of choice. It takes roughly 8.5 seconds to run the calibration sequence
12. **Reset Calibration –** If you are not satisfied with the calibration measure This button will reset all the recorded numbers and allow the subject to provide you with another calibration run.
13. **Absolute or Relative (calibrated) Maximum Voltage –** Here you are able to select whether the input from the subject will be compared up against the relative (calibrated) highest possible input or against an absolute input boundary, which is provided to the program.
14. **% of maximum input –** From either the calibrated maximum or the absolute value a certain % is used as the boundary used in the game, such that the maximum effort is newer used, but at max X % of the relative or absolute value.
15. **Absolute Max input voltage –** Here you denote an absolute value for the max value the subject have to perform in order to reach the toughest targets.

### Show Game Configuration



This menu adjust which kind of experience you will expose the subjects to. All of these checkmarks will have a major impact on the experience.

1. **Return To Baseline** – This option will take the sequence you have written and add a baseline and a Pause event after every second event. To use this option properly either use it together with “Random Target Position” or together with “Write Your Own Sequence”. If you write your own you have to follow this syntax R1 P1 R2 P2 R3 P3 ….. Rn Pn.
2. **Random Target Position –** This option allows the system to create a random sequence. For it to do it properly you have to configure it. You can tell the system how far or close apart the different targets are by adjusting min and max closeness. Between each generated target the next target cannot be closer than the minimum value nor further away than the maximum from the previous generated target. Number of Events determines how many events will be generated to run this version of Trackit. This option will use **(9, 11 and 12)** from Base configuration screen.
3. **Adaptive Difficulty – adaptive target closeness and height –** Dependent on how good the subject performs the height of targets will be shallower and the targets will be further apart. As a default setting, the subject has to have an accuracy of 75% in order for the user to progress and reach a higher difficulty level. This can be adjusted to a different number if needed. Accuracy is simply the amount of time on target versus the total time the target is visible. Each subject starts at level 1. Behind the curtains the algorithm for these five adjusted parameters looks like the following

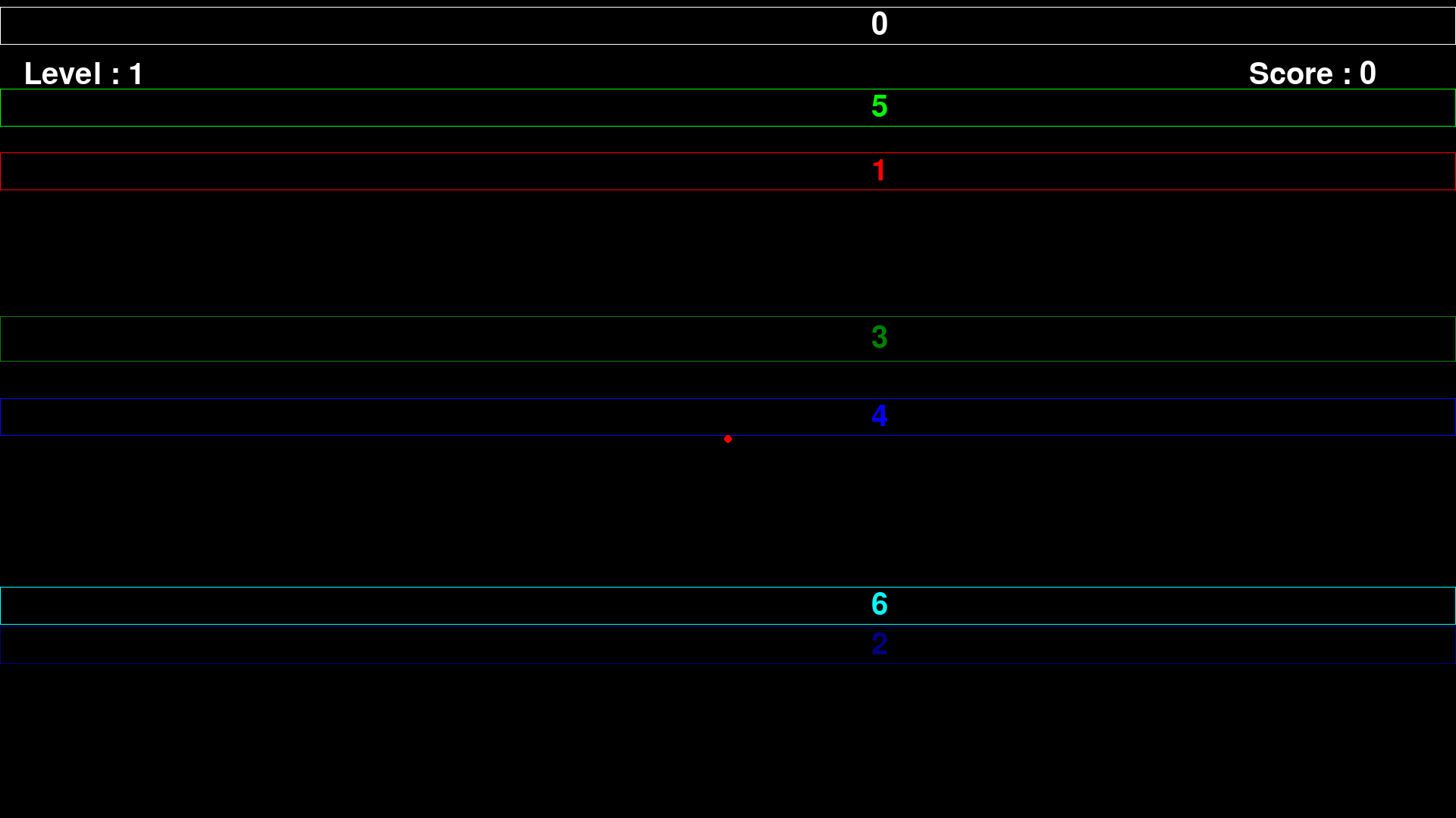
Minimum Closeness = Minimum Closeness + (level\*5)\*2

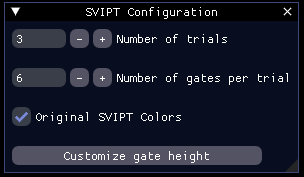
Maximum Closeness = Maximum Closeness + (level\*5)\*2

Random minimum target height = Random minimum target height - (level\*5)/2

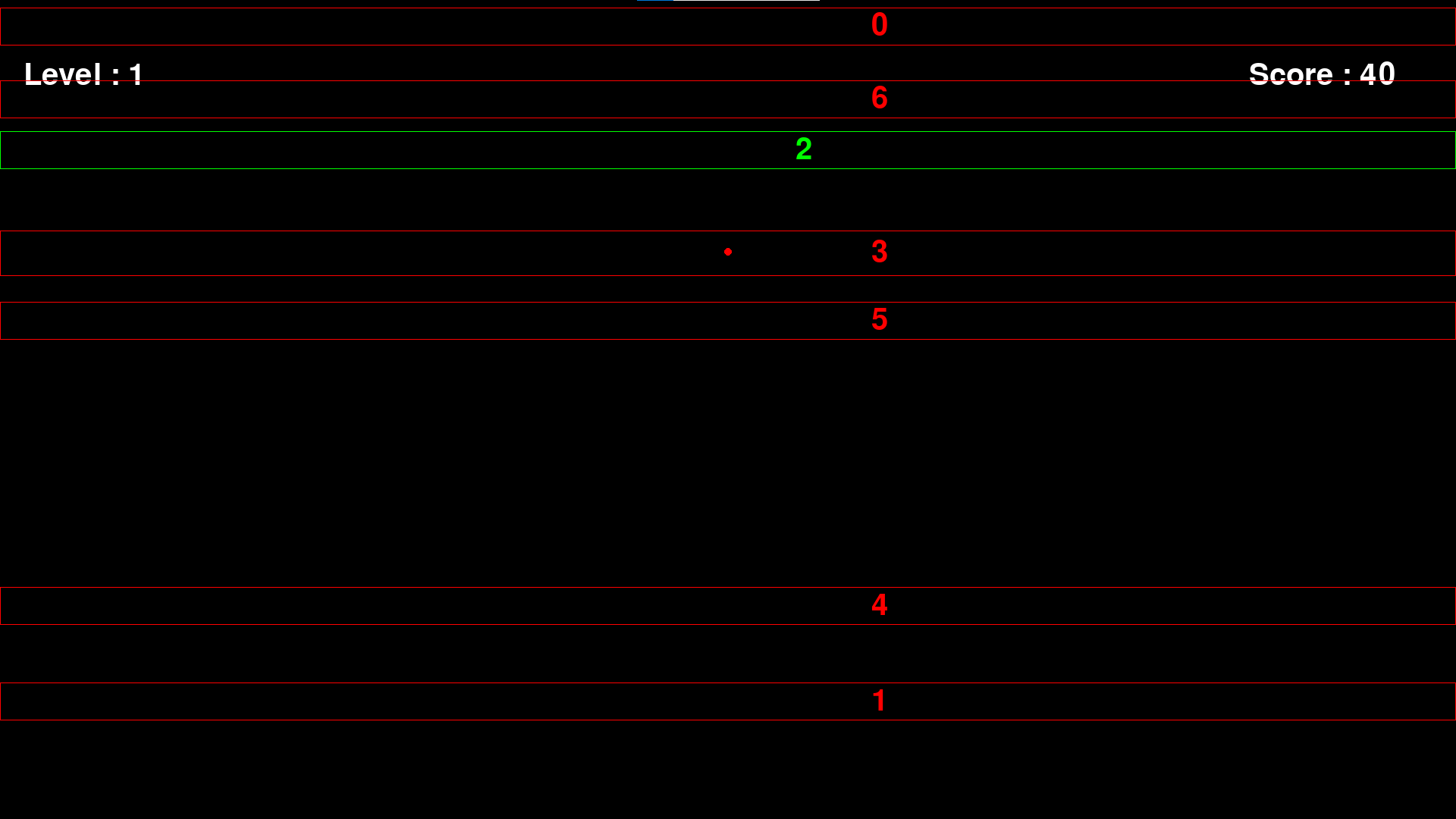
Random maximum target height = Random maximum target height - (level\*5)

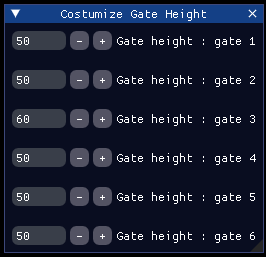
Target height = Target height - (level\*5)

1. **In game guidelines –** This is a simple text prompt telling the subject to be as accurate and fast as possible.
2. **Random target height –** This will provide your targets with random heights between 20 and 50. You are able to adjust it by visiting your configuration file and find the variables called minRandomHeight and maxRandomHeight. They are currently hidden from the interface. We have seen previously these two variables are adjusted by the adaptive difficulty option.
3. **Write Your Own Sequence –** This option enables you to write your own sequence. It takes its input from (**7**) in base configuration. This or “Random Target Position” has to be enabled in order for TrackIt to work.
4. **SVIPT – show all targets –** This is a completely different game mode than the usual TrackIt. While track it is all about reaching target fast and accurate as possible before they disappear SVIPT is a different beast entirely. In this game mode all targets are visible all the time, they are numbered and you have to return to baseline after each target. You have to go through the sequence as fast and accurate as possible. You only have to be within each target 150 ms before it counts as a visit. Each run through of a sequence of targets is called a Trial and targets are called gates. Two interesting parameters are given from each trial completion time and error. Completion time is the amount of time it took to go visit all the gates in the right order. Error is the amount of times the subjects misses the gate when they try to stop within the boundaries of the rectangles. Stopping before a gate also count as an error this is called an undershoot while stopping after a gate is called an overshoot.

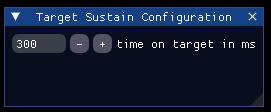
You are capable of adjusting different parameters regarding the SVIPT game.Apart from the already visited options this configuration menu is unique to SVIPT.

You van adjust the number of Trials and the number of gates per trial. You can choose to work with original SVIPT colors, which are adjustable colors, which in the random target case matches those of SVIPT articles [1]. While none original colors will provide the subject with more guidance towards which gate he has to visit.



The last element you are able to customize regarding the SVIPT game is the height of the gates. In case you want to have different heights and you want to have control to how different the height has to be you can adjust it in the configuration menu point.

When choosing SVIPT you have to either choose **Random Target Position** or **Write Your Own Sequence** as well. If you choose Random target position the computer will compute the position of the gates. It can fail a couple of times, but do not be discourage and try again. If you choose Write Your Own Sequence you have to use the input field found in the previous configuration menu **(7)**. The syntax however is a bit different**.** Writing a SVIPT sequence only includes gates and baseline, but only the baseline once. E.g. B1000r R300g R200b R600v …

1. **Target sustain on screen** – This item changes the TrackIt game entirely. With this option the target will remain on the screen until the subject has been within the target for a certain amount of time. Thereby changing the game to be more focused on tonic muscle control as they now have to be within the target for a certain amount of time instead of hitting it and stay within it before it disappears. The associated configuration menu can adjust the time.
2.  **Extrinsic motivation –** This option enables you to add game elements into the Trackit or SVIPT experience.   
   Levels is simply providing the user with a visual representation of their level, which if adaptive difficulty is enabled, will change with their performance.

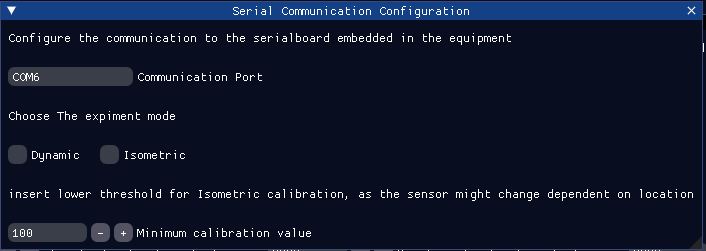
Visible Score is the current score of the TRackit run or SVIPT Trial.It is a number, which is counting upwards every time the cursor is within a rectangle**.**

If Highscore is enabled the score from before will be added to the highscore list, if the score is high enough.

Sound reward is a simple sound that happens every time a gate has been successfully visited in SVIPT or the accuracy within a target is high enough in a Trackit run.

Coin reward is a visible animated coin appearing in a TrackIt run if the accuracy is high enough for any given target.

### Show Serial Configuration



This configuration menu is only to be used when using ADAM / a microcontroller embedded into an equipment in which a dynamic or isometric mode can be used.

1. **Communication Port –** This is the registered Com port for the computer to communicate with the microcontroller. If you do not know it or have chosen the wrong one the system will tell you and guide you towards the right port.
2. **Choose the experiment mode –** To be able to change between sensors and thus the very exercise type the researchers have to choose if it is a dynamic or isometric exercise.
3. **Minimum calibration value –** When using ADAM in isometric mode it happens that the initial value of the sensor is not 0 or close to 0, which means that any calibration will fail and TrackIt / SVIPT will be unplayable. Try to run the calibration without touching the equipment. The calibration program tells you the initial value at the top of the program. Remember that number and insert a close approximation of it in the field above.

# How to Run Trackit

|  |  |
| --- | --- |
| Steps: | Screen shots |
| 1. Open Trackit application |  |
| 1. Press Configuration 2. Choose Base Configuration |  |
| 1. Insert Subject ID |  |
| 1. Insert researcher acronym and project acronym |  |
| 1. Indicate block No |  |
| 1. Choose input device, and configure NIDAQ input channel if needed |  |
| 1. Press start calibration and follow the instructions on the screen |  |
| 1. Press “Relative” |  |
| 1. Find and press “Show Game Configuration” in the Configuration menu |  |
| 1. Mark Write Your Own Sequence“ from the appeared menu |  |
| 1. Press Save Configuration from the Home Menu. Trackit will look for this file when Trackit creates the associated datafiles |  |
| 1. Now you are ready to run the first TrackIt game. Press Start Trackit |  |

## Write your own sequence mode

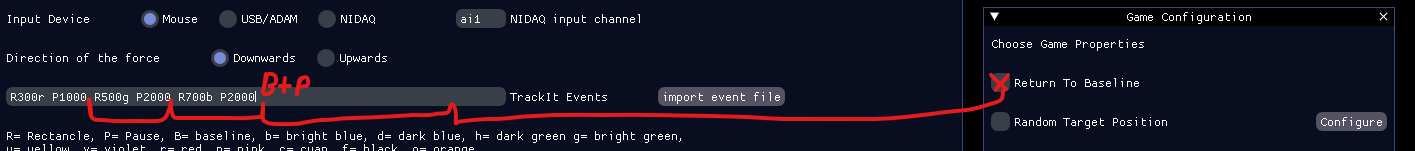
|  |  |
| --- | --- |
| Steps: | Screen shots |
| 1. Open Trackit application |  |
| 1. Press Configuration 2. Choose Base Configuration |  |
| 1. Insert Subject ID |  |
| 1. Insert researcher acronym and project acronym |  |
| 1. Indicate block No |  |
| 1. Choose input device, and configure NIDAQ input channel if needed |  |
| 1. Write the sequence you want or import it from .txt file | |
|  | |
| 1. Adjust the time you want targets to be visible |  |
| 1. Adjust the height of the targets |  |
| 1. Adjust the lengths of the feedback screen if you want to have any |  |
| 1. Press start calibration and follow the instructions on the screen |  |
| 1. Choose Absolute or Relative Maximum voltage | |
|  | |
| 1. Adjust the maximum percentage of max voltage the subject need to use |  |
| 1. If absolute is chosen in 13 adjust the maximum absolute voltage |  |
| 1. Find and press “Show Game Configuration” in the Configuration menu |  |
| 1. Mark Write Your Own Sequence“ from the appeared menu |  |
| 1. Press Save Configuration from the Home Menu. Trackit will look for this file when Trackit creates the associated datafiles |  |
| 1. Now you are ready to run the first TrackIt game. Press Start Trackit |  |

## Random event Generation mode

|  |  |
| --- | --- |
| Steps: | Screen shots |
| 1. Open Trackit application |  |
| 1. Press Configuration 2. Choose Base Configuration |  |
| 1. Insert Subject ID |  |
| 1. Insert researcher acronym and project acronym |  |
| 1. Indicate block No |  |
| 1. Choose input device, and configure NIDAQ input channel if needed |  |
| 1. Adjust the time you want targets to be visible |  |
| 1. Adjust Baseline Display time in case you will be using Baseline |  |
| 1. Adjust Pause time |  |
| 1. Adjust the height of the targets |  |
| 1. Adjust the lengths of the feedback screen if you want to have any |  |
| 1. Press start calibration and follow the instructions on the screen |  |
| 1. Choose Absolute or Relative Maximum voltage | |
|  | |
| 1. Adjust the maximum percentage of max voltage the subject need to use |  |
| 1. If absolute is chosen in 13 adjust the maximum absolute voltage |  |
| 1. Find and press “Show Game Configuration” in the Configuration menu |  |
| 1. Mark Return To Baseline if needed.  else simply mark Random Target Position |  |
| 1. Press the configure button |  |
| 1. Adjust how close and far apart the targets are to each other and adjust the amount of targets / events |  |
| 1. Press Save Configuration from the Home Menu. Trackit will look for this file when Trackit creates the associated datafiles |  |
| 1. Now you are ready to run the first TrackIt game. Press Start Trackit |  |

## Configuring Trackit

**Return to Baseline**



Return to baseline will add a Baseline and a Pause event after every second event

**Adaptive Difficulty Configuration**

****

This element does not change any of the configurations inside the Base configuration panel apart from adjusting the stimuli Height down dependent on the level of the subject

**In Game guidelines**

****

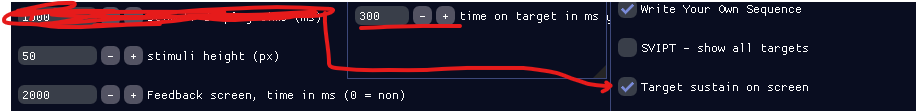
This adds a single prompt to the player telling them to be as accurate and fast as possible

**Random target height**

****

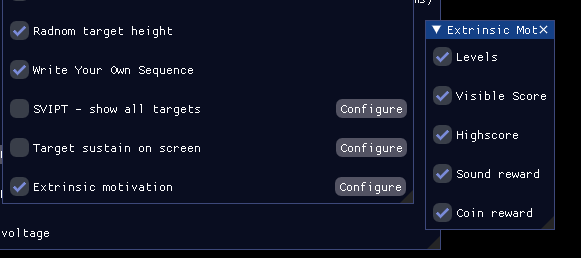
The height parameter denoted in the base configuration mode is no longer valid

**Target sustain on screen**



The Stimuli display time denoted in the base configuration window is no longer valid. Instead is the time on target shown in the middle panel. This is the time the subject needs to be within each target.

**Extrinsic motivation**

****

Described earlier these are different options to provide the subject with feedback upon the trackit game they are playing,

**Return to Baseline**, **Adaptive Difficulty**, **In Game guidelines**, **Random target height**, **Target Sustain on screen** and **Extrinsic motivation**, can all be chosen independent of each other. And can be added or removed as you like from your TrackIt Configuration.

You must however always choose to either have “**Random Target Position”** or “**Write Your Own Sequence**” activated as these are mutually exclusive

## Trackit Datafiles

After a successful Trackit run three datafiles are exported.

Events

InputData

Statistics

### Events

This datafile provides you with data about the different events and how well the subject performed.

**Rect\_no :** The number of Event counting both baseline, pauses and rectangles/targets

**EventType :** Baseline (B), Pause (P), Rectangle/Target (R)

**Trigger :** The trigger number send to the NIDAQ board for further distribution

**Height :** The height of the event

**Position :** The absolute position on the screen in pixels

**total\_time\_visible :** The time the event is present on the screen in ms

**Visible\_From :** The absolute time from the beginning of the start of the Trackit run the event is present e.g 1000 means that 1000 ms into the run this event appeared.

**Entry\_time\_ms :** in absolute time when did the subject enter this event. E.g the subject entered the event at 1200 ms into the run

**Calculated\_Entry\_time :** Relatively from Visible\_from time when did the subject enter this event.

**Time\_on\_target\_ms :** How long time did the subject spend inside the given event

**SD\_outside\_Target\_px :** The standard deviation of the distance between the target and the subject when the subject is outside the target.

**Mean\_Inaccuracy\_px\_outside\_target :** The mean of the distances between the target and the subject when the subject is outside of the given target

**Exit\_time\_ms:** When in absolute time did the subject exit the target

**Percent\_Time\_On\_Target :** of all the time the target is visible how big a % is used inside the target

**Reaction\_Time\_ms :** From when the target is visible to the first acceleration = The time it took the subject to react on the target.

**time\_Off\_Target\_ms :** The time spend outside the target in ms, when the target was visible

**overshoot? :** Did the subject manage to apply too much force and thereby miss the target, when the target appeared?

**overshoot\_time :** When did this force application appear in absolute time related to the start of the run

**undershoot? :** Did the subject manage to apply too little force and thereby not reaching the target, when the target appeared?

**undershoot\_time:** When did this force application appear in absolute time related to the start of the run

### InputData

The datafile provides you with the positional data on the screen, the raw input data and the time.

To be more exact this is the three items:

**Digital\_Input :** This is the raw input from the input device whether it is the mouse, NIDAQ or a microcontroller

**Screen\_Position\_Y\_axis(px) :** This is the screen position of the cursor. This can be used to compare with the events data file in case you want to check up different time and space related statistics

**time(ms) :** This is the in game time, which can be directly compared to any time measures given in the events data file.

There will be another Time point, which the time for when the input arrives in the system. This will be work as a parameter to look at flickering in the input.

### Statistics

This file provides the overall statistics calculated based on the current run.

**Mean\_Accuracy :** This is the average percent of time the user was on the target out of the total time the targets were visible.

**Std\_Accuracy :** This is the standard deviation of the Percentage of time the user was on the target put of the total time the targets were visible

**Mean\_Time\_On\_Target :** The average time the subject was on the targets.

**Std\_Time\_On\_Target :** The standard deviation of the time the subject was on the targets

**Total\_Time\_On\_Target :** The total amount of time the user was on targets.

**CompletionTime :** The total amount of time it took the subject to complete the run

**Errors :** The amount of times the subject either over- or undershoots.

# How to Run SVIPT

## Write your own sequence mode

|  |  |  |
| --- | --- | --- |
| Steps: | Screen shots | |
| 1. Open Trackit application |  | |
| 1. Press Configuration 2. Choose Base Configuration |  | |
| 1. Insert Subject ID |  | |
| 1. Insert researcher acronym and project acronym |  | |
| 1. Indicate block No |  | |
| 1. Choose input device, and configure NIDAQ input channel if needed |  | |
| 1. Write the sequence you want or import it from .txt file   Here it is important to follow some basic rules.   1. It has to start with a Baseline 2. All gates including the baseline have to have unique colors 3. Only the start Baseline is required and it has to have some arbitrary four digit number | | |
|  | | |
| 1. Adjust the height of the targets | |  |
| 1. Adjust the lengths of the feedback screen if you want to have any | |  |
| 1. Press start calibration and follow the instructions on the screen | |  |
| 1. Choose Absolute or Relative Maximum voltage | |  |
|  | |  |
| 1. Adjust the maximum percentage of max voltage the subject need to use | |  |
| 1. If absolute is chosen in 13 adjust the maximum absolute voltage | |  |
| 1. Open Game Configuration and mark the area besides SVIPT – Show all targets | |  |
| 1. Open the corresponding Configure menu 2. Adjust the number of trials. Which is complete runs through all the gates 3. Adjust the number of gates to the number of events you wrote in step 8 excluding the baseline 4. Choose whether to have the Original SVIPT colors. Which are those written in step 8, or let the colors be adjusted, so that the current gate is green and the other gates are red. 5. Lastly you can costumize each of the gates heights if you wish to adjust the difficulty level | |  |
| 1. Press Save Configuration from the Home Menu. Trackit will look for this file when Trackit creates the associated datafiles | |  |
| 1. Now you are ready to run the SVIPT game. Press Start Trackit | |  |

## Random event Generation mode

|  |  |  |
| --- | --- | --- |
| Steps: | Screen shots | |
| 1. Open Trackit application |  | |
| 1. Press Configuration 2. Choose Base Configuration |  | |
| 1. Insert Subject ID |  | |
| 1. Insert researcher acronym and project acronym |  | |
| 1. Indicate block No |  | |
| 1. Choose input device, and configure NIDAQ input channel if needed |  | |
| 1. Adjust the height of the targets | |  |
| 1. Adjust the lengths of the feedback screen if you want to have any | |  |
| 1. Press start calibration and follow the instructions on the screen | |  |
| Choose Absolute or Relative Maximum voltage | | |
|  | | |
| 1. Adjust the maximum percentage of max voltage the subject need to use | |  |
| 1. If absolute is chosen in 13 adjust the maximum absolute voltage | |  |
| 1. Open Game Configuration and mark the area besides SVIPT – Show all targets 2. And mark Random Target Position | |  |
| 1. Adjust how close and far apart the targets are to each other. The amount of targets shown here is irrelevant for SVIPT | |  |
| 1. Open the Configure menu associated with SVIPT 2. Adjust the number of trials. Which is complete runs through all the gates 3. Adjust the number of gates to the number of your desire (max 10) 4. Choose whether to have the Original SVIPT colors, which are colors closely associated with the once written in [1] or colors where the target is green and the rest is red. 5. Lastly you can customize each of the gates heights if you wish to adjust the difficulty level | |  |
| 1. Press Save Configuration from the Home Menu. Trackit will look for this file when Trackit creates the associated datafiles | |  |
| 1. Now you are ready to run the SVIPT game. Press Start Trackit | |  |
| If you are met by the following message, ignore or close it and then repeat step 22 | | |
|  | | |

## SVIPT Datafiles

After a successful SVIPT run three datafiles are exported.

SVIPTEvents

InputData

SVIPTTrials

### SVIPTEvents

Contrary to the eventdata file found when running a Trackit session this event file is nearly not usable. It can however be used to control whether the observed over- undershoots were actually over- undershoots. In the data file the following elements can still be used.

**Height :** The height of the gates

**Position :**  the position of the gates on the screen

**Entry\_time\_ms :**  The time when the program recognized that the subject has entered the gate

**overshoot? :** Did the subject manage to apply too much force and thereby miss the gate?

**overshoot\_time :** When did this force application appear in absolute time related to the start of the run

**undershoot? :** Did the subject manage to apply too little force and thereby not reaching the gate?

**undershoot\_time:** When did this force application appear in absolute time related to the start of the run

The rest of the elements in the datafile is not useful in this scenario. The above mentioned elements can be used in combination with the inputData file to look for data inconsistencies or false over- undershoots.

### InputData

The datafile provides you with the positional data on the screen, the raw input data and the time.

To be more exact this is the three items:

**Digital\_Input :** This is the raw input from the input device whether it is the mouse, NIDAQ or a microcontroller

**Screen\_Position\_Y\_axis(px) :** This is the screen position of the cursor. This can be used to compare with the events data file in case you want to check up different time and space related statistics

**time(ms) :** This is the in game time, which can be directly compared to any time measures given in the events data file.

There will be another Time point, which the time for when the input arrives in the system. This will be work as a parameter to look at flickering in the input.

### SVIPTTrials

This data file contains a short brief dataset showing the subjects SVIPT Performance

**Trial\_no :** Is the trial number that has been completed. It will always start by 0

**CompletionTime :** How long it took the subject to complete the trial

**Error :** The amount of over- undershoots the subject performed during that trial

# References

[1] J. Reis *et al.*, “Noninvasive cortical stimulation enhances motor skill acquisition over multiple days through an effect on consolidation,” *Proc. Natl. Acad. Sci.*, vol. 106, no. 5, pp. 1590–1595, Feb. 2009, doi: 10.1073/pnas.0805413106.

# 