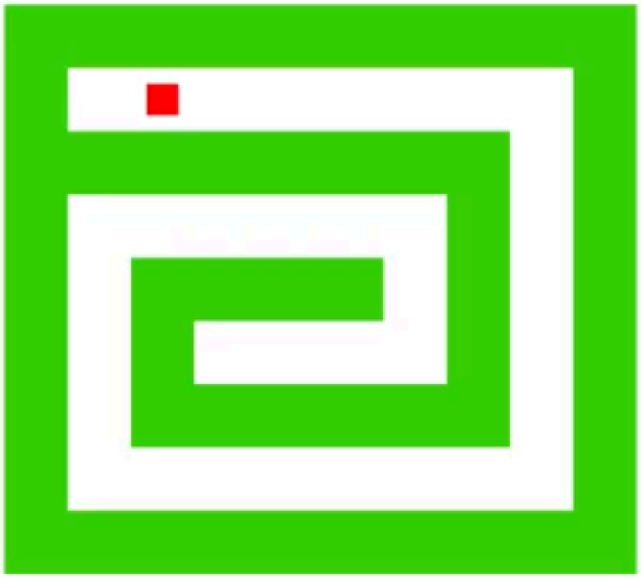
Guide To

EEG based Maze Game

Concept Overview

Real-time EEG-based BCIs (brain-computer interfaces) offer a glimpse into this futuristic reality, holding the potential to revolutionize healthcare, education, entertainment, and countless fields.



This project is the perfect demonstration of application of EEG in machines and its abilities to control objects in real time.

In simple terms, our goal is to control the character in maze using a person's brain waves

OpenBCI hardware

OpenBCI bridges the gap between three previous categories of brain-computer interfaces: Consumer devices, such as the Emotiv, NeuroSky, and Muse headsets, homebrew designs, such as the OpenEEG project, for more confident engineers, and systems for research or clinical use, which cost thousands of dollars.

The OpenBCI arrives pre-assembled and tested, and is relatively straightforward for someone without previous BCI experience to get started with — no soldering, hair-tearing, or midnight coding sessions need apply. I should note, though, that it's not quite as gentle of a learning curve as I was led to expect: it's not plug 'n play and it's fairly complex for a beginner, so there's a lot to learn.

The OpenBCI's software and hardware are open-source, which gives tinkerers huge flexibility in how they use it. The hardware can be modified and improved, just like homebuilt systems. You can use the OpenBCI's app, connect the board to a wide variety of existing processing and display software suites, or, if you're feeling ambitions, write custom software for your needs.

Finally, the OpenBCI's output is far closer to clinical or research grade than most consumer or amateur systems: It has an excellent sample-rate (250 Hz), works with well-established processing and display software, and can be used with any electrodes with industry-standard "touchproof" connectors, so you aren't stuck with the type it comes with.

Requirements

BCI system hardware

EEG

OpenBCI 8-channel wet-electrode cap

OpenBCI electrode cap gel w/ applicator syringe

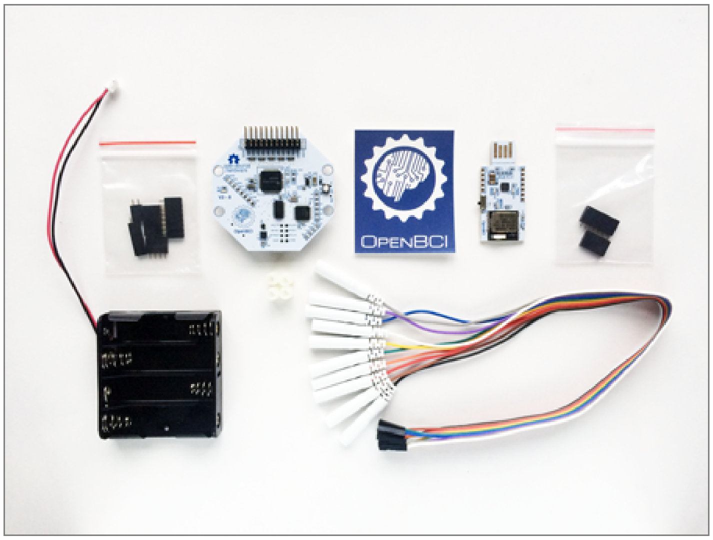
Sampling Hardware

OpenBCI Cyton biosensing board

Power Source (4 x AA batteries)

OpenBCI Bluetooth dongle

Auxiliary hardware



Windows PC (w/ Python 3.10)

Test setup:

I7 10th gen hexacore CPU

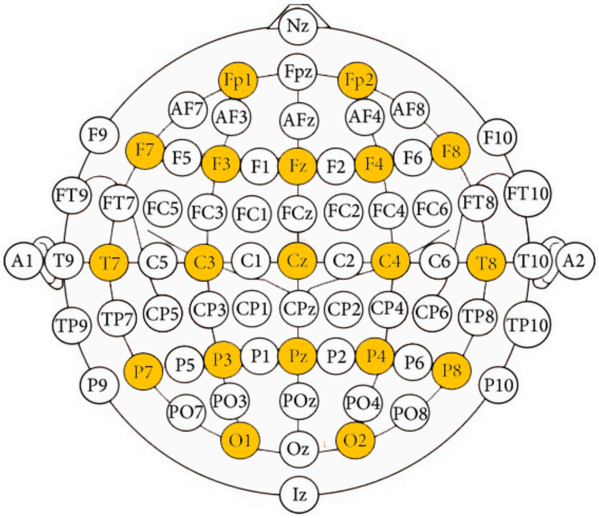
Nvidia Gtx 1660ti GPU

32 GB DDR5 ram

****Note:**The Auxiliary hardware used in this project is the bare minimum we recommend to see some sort of progress and running. WIth this setup we were able to run the game for 15-20 seconds where the character moved in every 2 seconds after reading brain signals till it crashes after 15-20 seconds due to the system not being able to handle the load of real time processing.

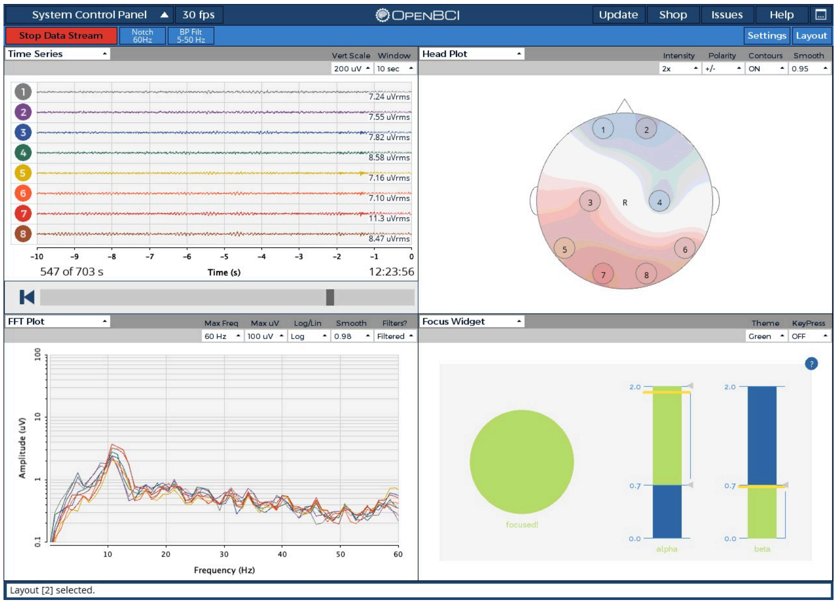
Step-by-step to reproduce the project

- Connect OpenBCI EEG cap to Cyton board
    - **Channels**:
        - White - REF
        - Black - GND
        - Gray (1) - O1
        - Purple (2) - O2
        - Blue (3) - P3
        - Green (4) - P4
        - Yellow (5) - PZ
        - Orange (6) - C3

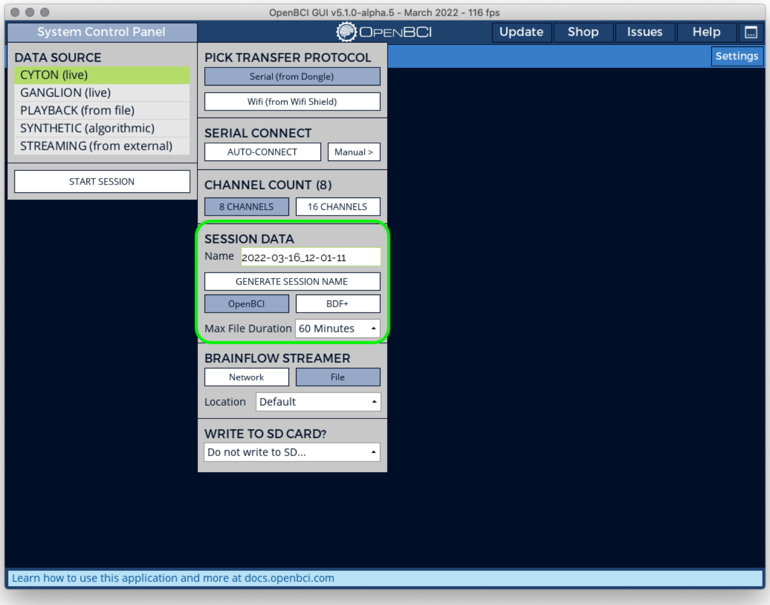


        - Red (7) - C4
        - Brown (8) - CZ
- **Wear EEG cap**
    - Using applicator syringe, apply gel to REF, GND, O1, O2, P3, P4, PZ, C3, C4, and CZ.
        - If unfamiliar with gel application, download OpenBCI GUI and stream data. Change vertical scale to automatic and apply gel slowly until each channel's scale is minimized.
        - Remember, the markings are always present on the cap so you don't have to worry about finding it.

- OpenBCI GUI
        - The OpenBCI GUI is OpenBCI's powerful software tool for visualizing, recording, and streaming data from the OpenBCI Boards. Data can be displayed in live-time, played back, saved to your computer in .txt format, as well as streamed in live-time to third-party software such as MATLAB. It can be launched as a standalone application or as a sketch from Processing (a Java-based programming language). In this guide, we will cover both scenarios.
        - If it's your first time working with OpenBCI and you own the Cyton or Cyton+Daisy, make sure to install the latest FTDI driver for your operating system.
        - info
        - **The FTDI driver is only necessary for Windows 8, Windows 10, and Mac OS X 10.9 through 10.15. If you are running a Mac that is mid 2015 or newer, you do not need to install the FTDI driver!**
        - Install the OpenBCI GUI as a standalone software. VIsit the OpenBCI GUI install page or documentation page. You shall find the link in references if needed.



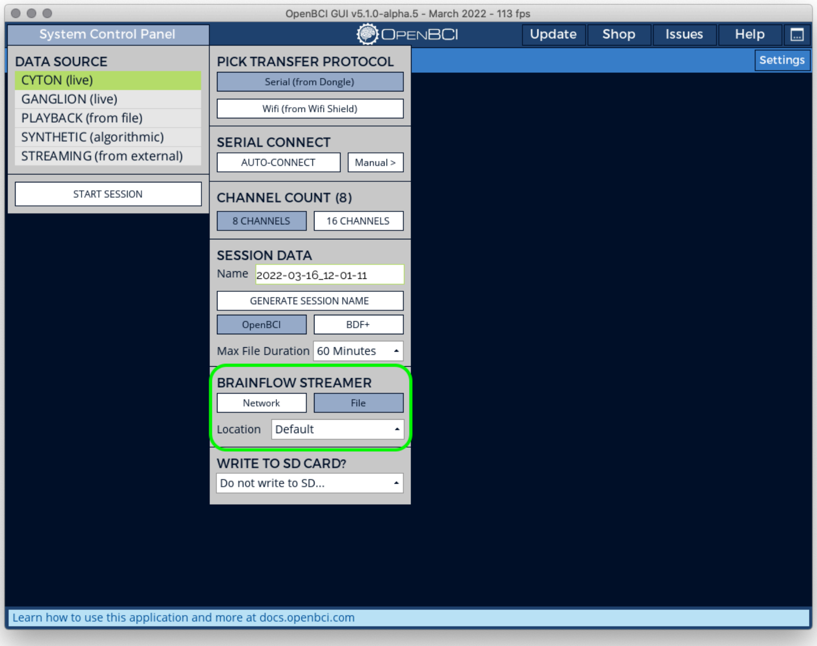
OpenBCI users always have free access to live and recorded data! Recorded data is saved to your local computer to respect privacy and freedom of use. Also, you have the option to save data as a CSV text file or BDF+ format.



Where do I find my Data?

By default, **the GUI stores all user data and raw EEG recordings in** **`[USER]/Documents/OpenBCI_GUI`** and names each session with an autogenerated timestamp. Saved data does not have filters applied. This gives you the freedom to change and experiment with other filters during playback.

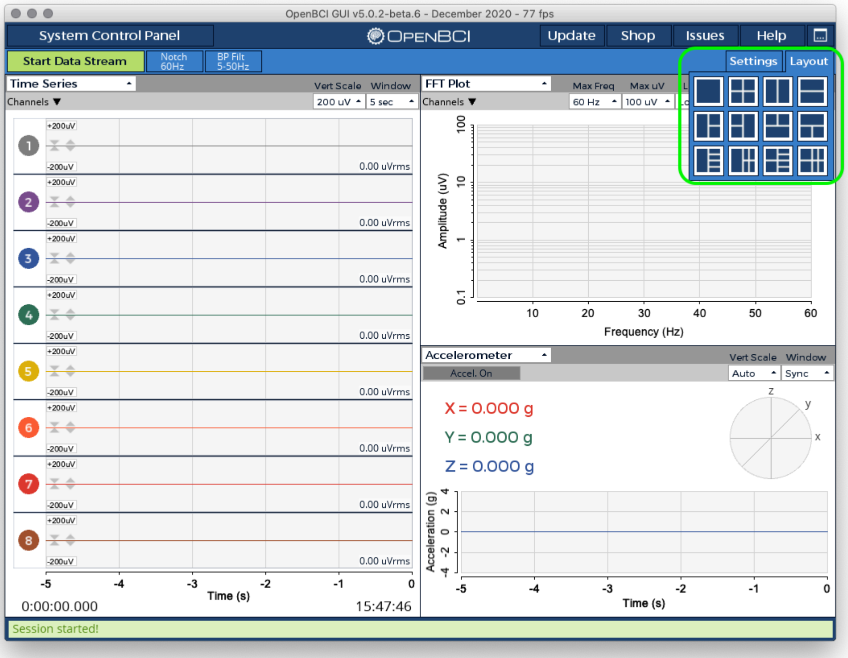
Starting with GUI 5.1.0, you can save data simultaneously to the BrainFlow CSV format, which [can be used directly in BrainFlow](#) for offline processing, training, and experimentation.



Cyton users can save data to an SD card. To playback these files using the GUI, copy the files to your computer from the SD card. This works better than reading files from the SD card. With GUI v5, it is no longer necessary to convert SD files for playback in the GUI. A few users may still want to [convert large SD file recordings](#).

Customize Your Layout

The OpenBCI GUI displays up to six customizable windows in twelve layouts! You can choose what each window displays by clicking the dropdown menu at the upper left of each window. The bottom row of layouts work best on higher resolution monitors.



Impedance Testing

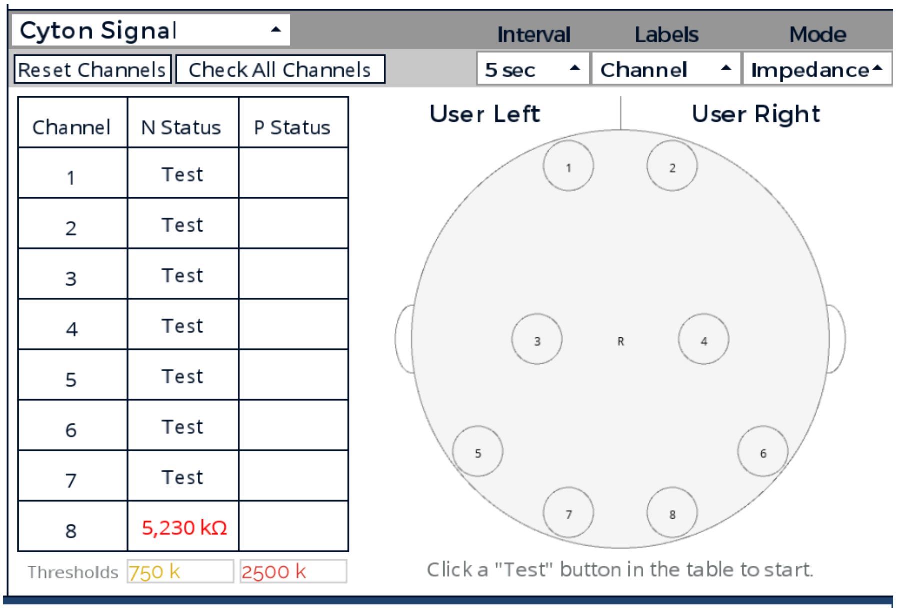
The impedance widget is a valuable tool for evaluating electrode contact before data acquisition. Press `Test` to start impedance test on an individual channel.

The impedance value is in colored font as a visual guide to the pre-set thresholds. A red impedance value means you should adjust your electrodes, part your hair, add gel, use paste, or such measures as appropriate for the electrode you're using. Experienced users can also adjust these thresholds in the bracket-field beneath the table.

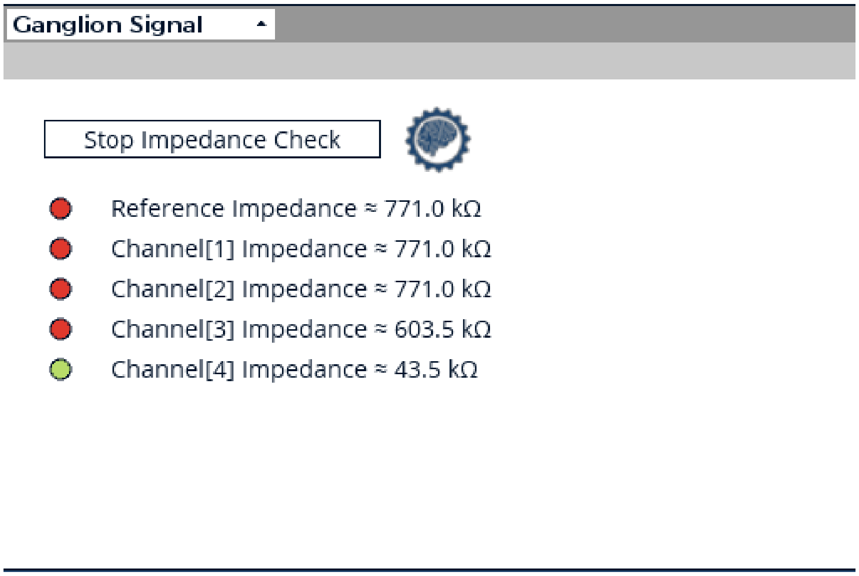
info

During the impedance test, the board sends a small current through the selected channel to obtain the impedance value. **For this reason, you won't be able to stream data on a channel and obtain the impedance value simultaneously from the channel.**

The screenshot below shows the GUI Impedance Widget for Cyton:

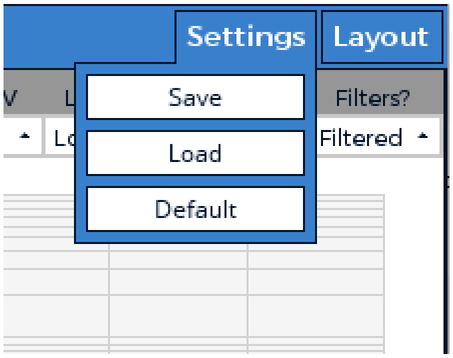


To test impedance for Ganglion, use the Ganglion Signal Widget.



Decrease Setup Time by Saving/Loading Settings

Starting with GUI V3, you can save and load a snapshot of nearly every setting in the GUI (ex. Layout, Time Series Channel Settings, Networking Settings, etc.) by using a dropdown menu (pictured below) or keyboard shortcuts. If you change anything during a session, save your settings or click "Stop System" before exiting. **Save with lowercase 'n'** and **Load with capital 'N'** on your keyboard to and from */Documents/OpenBCI_GUI/Settings/*.



Start the system first. Then, arrange the GUI to suit your needs and **click "Settings"-->"Save"**. A dialog box will open to confirm file name and path *(e.g. GanglionUserSettings.json)*. After, you will see a message saying "Settings Saved!" When you click "Stop System", all settings will be auto-saved.

The GUI automatically loads settings from */Documents/OpenBCI_GUI/Settings/* when the system starts!

To load settings, just **click "Settings"-->"Load"**. Select a settings file from the dialog box that opens. If the settings are incompatible (ex. loading 16 channels while using 8), the GUI will display an error at the bottom or "Default Settings Loaded". If all is well, the GUI will display "Settings Loaded!" as the GUI snaps directly to your desired settings.

Click "Settings"-->"Default" to revert the GUI to default settings.

For all data modes, the GUI will load all GUI-related settings automatically. After starting a session, hardware settings (channel on/off, gain, etc.) will need to be set by the user manually

For information, kindly refer to the official documentation at :
https://docs.openbci.com/Software/OpenBCISoftware/GUIDocs/

Setting and Running the Code

Step 1:

Once you have downloaded the GUI and understood how different waves work, the most important part for us being the **Focus Widget.** The Focus Widget uses the concept of using alpha and Beta waves frequencies to determine your mental state

Alpha- focused

Beta - relaxed

They are other waves too with different frequencies, but for the sake of this project we will consider these only.

Step 2:

Once you have cloned the repo, and have the requirements installed, it is time to set the COM port as per your device. You need to open open_bci_v3.py file and make sure to enter the correct port address. For windows it is COM and for linux/unix (Mac OS) it is dev/tty/. Please make sure you update the port address correctly in the following two files:

Open_bci_v3.py

brain.py

Then we run the codes one by one in the following manner:

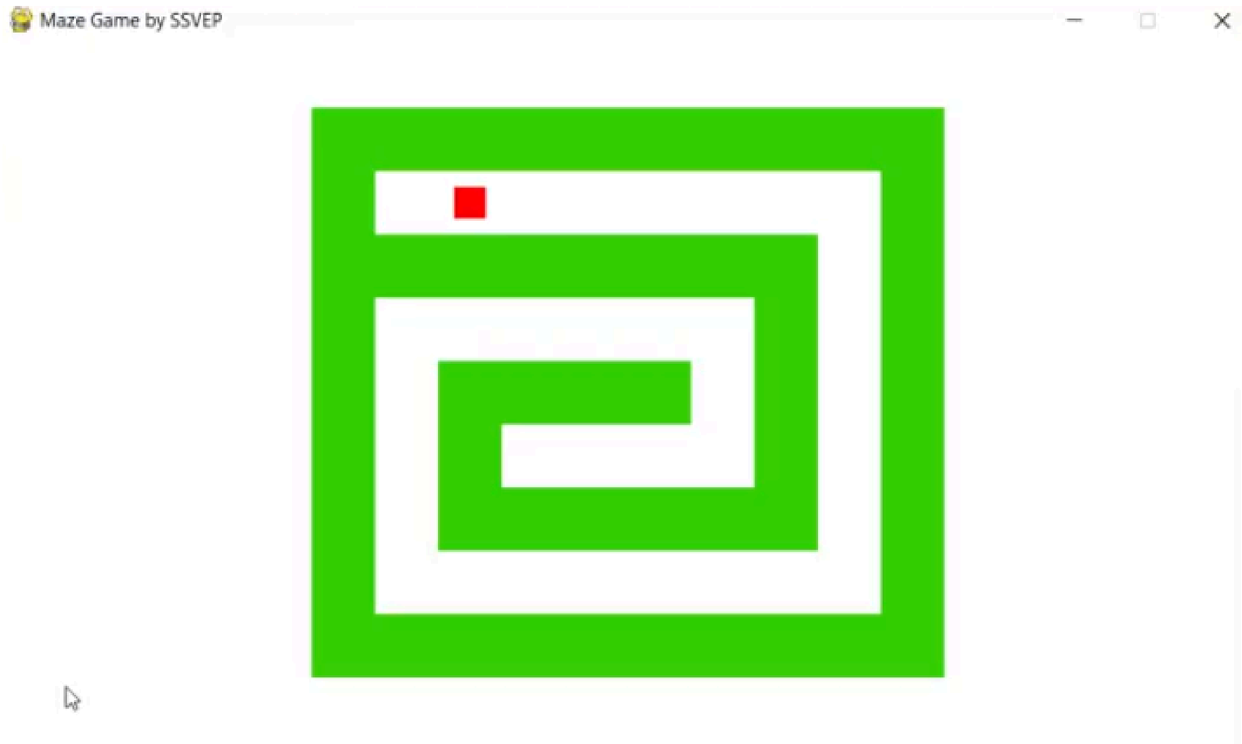
Open_bci_v3.py - this will not return any output unless there is an error brain.py - should show some output of the following manner:

```
(base) PS C:\Users\Vansh Santdasani> cd SSVEP
(base) PS C:\Users\Vansh Santdasani\SSVEP> python open_bci_v3.py
(base) PS C:\Users\Vansh Santdasani\SSVEP> python maze_improved.py
pygame 2.5.2 (SDL 2.28.3, Python 3.11.5)
Hello from the pygame community. https://www.pygame.org/contribute.html
Connecting to V3 at port COM5
Serial established...
OpenBCI V3 8-16 channel
On Board ADS1299 Device ID: 0x3E
On Daisy ADS1299 Device ID: 0x3E
LIS3DH Device ID: 0x33
Firmware: v3.1.2
$$$
Board Instantiated
Captured data from channel 6: -3367832
Captured data from channel 6: -1166878
Captured data from channel 6: -6509434
Captured data from channel 6: -2300650
Captured data from channel 6: -6509089
Captured data from channel 6: -2304576
Captured data from channel 6: -6509205
Captured data from channel 6: -2322499
Captured data from channel 6: -6509170
Captured data from channel 6: -2292415
Captured data from channel 6: -6509151
Captured data from channel 6: -2324764
Captured data from channel 6: -6509397
Captured data from channel 6: -2300941
Captured data from channel 6: -6509214
Captured data from channel 6: -2304498
Captured data from channel 6: -6509373
Captured data from channel 6: -2322522
Captured data from channel 6: -6509250
Captured data from channel 6: -2292443
Captured data from channel 6: -6509279
Captured data from channel 6: -2324712
Captured data from channel 6: -6509458
Captured data from channel 6: -2301116
Captured data from channel 6: -6509353
Captured data from channel 6: -2304603
Captured data from channel 6: -6509554
Captured data from channel 6: -2323027
Captured data from channel 6: -6509275
```

This ensures that everything is working as expected and the data is being processed.

Step 3:

Now, finally you can run maze.py and see the game in action for yourself.




```
(base) PS C:\Users\Vansh Santdasani> cd SSVEP
(base) PS C:\Users\Vansh Santdasani\SSVEP> python open_bci_v3.py
(base) PS C:\Users\Vansh Santdasani\SSVEP> python maze_improved.py
pygame 2.5.2 (SDL 2.28.3, Python 3.11.5)
Hello from the pygame community. https://www.pygame.org/contribute.html
Connecting to V3 at port COM5
Serial established...
OpenBCI V3 8-16 channel
On Board ADS1299 Device ID: 0x3E
On Daisy ADS1299 Device ID: 0x3E
LIS3DH Device ID: 0x33
Firmware: v3.1.2
$$$
Board Instantiated
Captured data from channel 6: -3367832
Captured data from channel 6: -1166878
Captured data from channel 6: -6509434
Captured data from channel 6: -2300650
Captured data from channel 6: -6509089
Captured data from channel 6: -2304576
Captured data from channel 6: -6509205
Captured data from channel 6: -2322499
Captured data from channel 6: -6509170
Captured data from channel 6: -2292415
Captured data from channel 6: -6509151
Captured data from channel 6: -2324764
Captured data from channel 6: -6509397
Captured data from channel 6: -2300941
Captured data from channel 6: -6509214
Captured data from channel 6: -2304498
Captured data from channel 6: -6509373
Captured data from channel 6: -2322522
Captured data from channel 6: -6509250
Captured data from channel 6: -2292443
Captured data from channel 6: -6509279
Captured data from channel 6: -2324712
Captured data from channel 6: -6509458
Captured data from channel 6: -2301116
Captured data from channel 6: -6509353
Captured data from channel 6: -2304603
Captured data from channel 6: -6509554
Captured data from channel 6: -2323027
Captured data from channel 6: -6509275
```

References

[1] https://docs.openbci.com/Software/OpenBCISoftware/GUIDocs/

[2] https://openbci.com/community/getting-started-with-openbci/

[3] https://github.com/DarrenVawter/P300_BCI/tree/main?tab=readme-ov-file - To understand how the cap is connected

If you are interested in reading more about EEG based BCI and what all can be done with it, you are free to refer to the below references:

1. D. Zhang, L. Yao, K. Chen, S. Wang, X. Chang and Y. Liu, "Making Sense of Spatio-Temporal Preserving Representations for EEG-Based Human Intention Recognition," in IEEE Transactions on Cybernetics, vol. 50, no. 7, pp. 3033-3044, July 2020, doi: 10.1109/TCYB.2019.2905157.

2. Shahini, N.; Bahrami, Z.; Sheykhivand, S.; Marandi, S.; Danishvar, M.; Danishvar, S.; Roosta, Y. Automatically Identified EEG Signals of Movement Intention Based on CNN Network (End-To-End). Electronics 2022, 11, 3297. https://doi.org/10.3390/electronics11203297

3. Ou Bai, Peter Lin, Sherry Vorbach, Jiang Li, Steve Furlani, Mark Hallett, Exploration of computational methods for classification of movement intention during human voluntary movement from single trial EEG, Clinical Neurophysiology, Volume 118, Issue 12, 2007, Pages 2637-2655, ISSN 1388-2457, https://doi.org/10.1016/j.clinph.2007.08.025.

4. Shafiul Hasan, S.M., Siddiquee, M.R., Atri, R. et al. Prediction of gait intention from pre-movement EEG signals: a feasibility study. J NeuroEngineering Rehabil 17, 50 (2020). https://doi.org/10.1186/s12984-020-00675-5

5. Maryam Mahmoodi, Bahador Makkiabadi, Mehran Mahmoudi, Saeid Sanei, A new method for accurate detection of movement intention from single channel EEG for online BCI, Computer Methods and Programs in Biomedicine Update, Volume 1, 2021, 100027, ISSN 2666-9900, https://doi.org/10.1016/j.cmpbup.2021.100027.

6. Lin Yue, Hao Shen, Sen Wang, Robert Boots, Guodong Long, Weitong Chen, and Xiaowei Zhao. 2021. Exploring BCI Control in Smart Environments: Intention Recognition Via EEG Representation Enhancement Learning. ACM Trans. Knowl. Discov. Data 15, 5, Article 90 (October 2021), 20 pages. https://doi.org/10.1145/3450449

7. John Atkinson, Daniel Campos, Improving BCI-based emotion recognition by combining EEG feature selection and kernel classifiers, Expert Systems with Applications, Volume 47, 2016, Pages 35-41, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2015.10.049.

8. Torres, E.P.; Torres, E.A.; Hernández-Álvarez, M.; Yoo, S.G. EEG-Based BCI Emotion Recognition: A Survey. Sensors 2020, 20, 5083. https://doi.org/10.3390/s20185083

9. Schwartz AB, Cui XT, Weber DJ, Moran DW. Brain-controlled interfaces: movement restoration with neural prosthetics. Neuron. 2006 Oct 5;52(1):205-20. doi: 10.1016/j.neuron.2006.09.019. PMID: 17015237.

10. Pires, Gabriel & Nunes, Urbano & Castelo-Branco, Miguel. (2007). Single-Trial EEG Classification of Movement Related Potential. 569 - 574. 10.1109/ICORR.2007.4428482.

11. Tucker, M.R., Olivier, J., Pagel, A. et al. Control strategies for active lower extremity prosthetics and orthotics: a review. J NeuroEngineering Rehabil 12, 1 (2015). https://doi.org/10.1186/1743-0003-12-1