Task: Create a script to start the fNIRS recording in Aurora once a task starts in Unity.

Official Progress

Add LSL Outlet to Unity

- https://github.com/labstreaminglayer/LSL4Unity
- I went with option 1
- Scene: Simple Physics Event Outlet
    - Script used: SimpleOutletTriggerEvent
    - Modified the script by duplicating the on-collision method and changing the name
        - Pressing the start button it is easier to use onClick than a collision event.
- Where in the code should it go
    - Attach to start buttons
        - Where ever there is a start button in the next scene, the LSL outlet function will be called
            - lslStuff() aka OnTriggerEnter()
    - All start buttons should attach lslStuff from findSM (find system manager) through the onClick() function.

Add LSL Inlet to Aurora

- Automatically built into Aurora.
- Proceed through the application as normal.
- However, before selecting the desired montage, first select edit
    - In the third tab, modify the LSL name
    - In our case, the LSL name is “LSL4Unity.Samples.SimpleCollisionEvent”

Add LSL Input to BrainVision

- Downloads
    - BrainVision LSL Viewer: https://www.brainproducts.com/downloads/more-software/#utilities
    - LSL Connector: https://github.com/brain-products/LSL-LiveAmp/releases
    - LabRecorder: https://github.com/labstreaminglayer/App-LabRecorder
        - It appears to be something different from BrainVision Recorder
        - Provides a recording of both the LiveAmp and Unity triggers in a .XDF file
            - More information on how to read this below
- GitHub Resources
    - LSL for LiveAmp: https://github.com/brain-products/LSL-LiveAmp/tree/master
- Explanations
    - Introduction to BrainVision LSL Viewer: https://pressrelease.brainproducts.com/lsl-viewer/

    - How to setup and use LSL with BrainVision: https://bci.plus/data-processing-with-lsl-bv/
    - Multiple GitHub pages for all BrainVision devices that use LSL: https://pressrelease.brainproducts.com/lsl-github/

Read .XDF Files With MatLab

- GitHub Resources
    - Variety of software options in addition to MatLab: https://github.com/sccn/xdf
    - MatLab specific: https://github.com/xdf-modules/xdf-Matlab/tree/0cdf054391fff7f0ea3416ee632ad1bd73d6623b
- Download
    - MatLab via CSU (License Issues): https://www.engr.colostate.edu/ets/matlab/

Send Triggers to start Recording.

- Initial thoughts:
    - It will require additional scripting
    - Could involve LSL API
    - Manual Instructions: https://nirx.sharefile.com/share/view/s3abb703753d4e2a8
    - Common Issues: https://support.nirx.de/archives/knowledge/common-lsl-triggering-issues
- The above is unnecessary
    - Unity sends triggers automatically with the onClick event
    - Just make sure to start the Aurora recording before users trigger any events

Idea: Use LSL to communicate between Unity and Aurora (Correct: Unity → Aurora)

Add LSL to Unity

- GitHub Instructions

Use Turbo Satori for Real-Time Analysis of fNIR data.

- NIRx

LSL is Compatible with Aurora

- LSL Supported Devices
- Supposedly Unity as well but when I tried the GitHub instructions it did not work for me
    - Original GitHub Repo
    - Alternative GitHub Repo
        - Does not work for me either

        - Is three years old

NIRStar

- James, Emily, and Jeremy would be great resources
- I would like a Zoom on how to set this up
    - Jeremy should be able to help
- RA: Sara
    - Train with James next week
    - We will have to get her familiar with the headset

Idea: Using alternative software (Turbo Satori) (Backwards: Aurora → Unity)

Unity receiving data from Turbo Satori via TCP

- GitHub Instructions

Real-Time Analysis - Use signals to influence Unity

- Real-Time Analysis

Idea: Via Lucus

- Web Sockets
    - Networking
    - Using ports
    - Unity can send and Aurora can listen
    - Overview
        - https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
    - Writing Scripts
        - Client applications: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API/Writing_WebSocket_client_applications
        - C#: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API/Writing_WebSocket_server
- Auto Hot Keys
    -