NeuroController Proposal

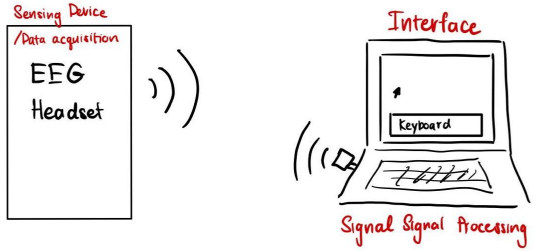
Team A0: Jean, Jonathan, and Wendy

Carnegie Mellon University

Use Case

Increase accessibility of computer interfaces for amputees and individuals with difficulty controlling muscle functions such as those with ALS, through EEG.

Areas covered: Signals, Software Systems



Use Case Requirements (User)

- Aim for above 90% accuracy in converting user intentions into the corresponding output within the interface.
- Desire for a user to on average score 500 ms on reaction time in the human benchmark test using our device.
- For a point-and-click test, aim to hit 5 targets within 30 seconds with 75% accuracy

Use Case Requirements (Signals)

- Obtain 4 distinct and easily transmittable brain signals to send from our EEG device
- Process signals to usable control data
- Send controller signals across a wireless network and process signal within latency of 250 ms

Use Case Requirements (Software)

- Software configuration to override and control the standard desktop mouse interface
- An onscreen accessibility keyboard that provides the ability to send all standard keyboard inputs, modified with text autocomplete and customizable widgets based on user preference

Technical Challenges

- Choosing or generating pattern recognition algorithm to discern the different signals
- Discern and process a common signal that may vary across multiple subjects
- Accessibility in performing a certain task (UI design)
- Developing both a keyboard and mouse driver for standard operating systems

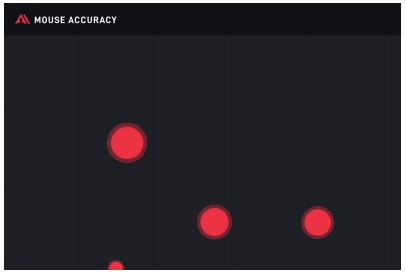
Risk Mitigation

- In the case of inability to obtain a stable usable EEG signals, we plan to use EMG (muscle) signal detection for control data in replacement
- If we desire to stick to non-physical transmission of control signals, we will pivot to using CV detection through eye-tracking

Solution Approach

- Collect and sample data from Emotiv Insight for research
- Implement processing algorithm in C/C++ for low latency
- Incorporate an installable OS driver to interface to our custom accessibility keyboard and mouse application
- Construct an iOS App
  - Built using Swift or Flutter

Testing, Verification, and Metrics



1. Link our EEG device directly to a desktop operating our signal processing algorithm. Check that data transmission is working properly according to defined movement tasks
2. Test the entire live system: clock the time required for a signal from the EEG device to reach our desktop interface
3. General accessibility tests: series of tasks comparing the time a normal desktop user takes to the time it requires our device user to accomplish the same task

Tasks and Division of Labor

- Jean- Signals
  - Signal processing algorithm/research
- Jonathan- Hardware/Software Interfacing
  - Create acquisition and processing backend from EEG device
- Wendy- Software
  - Design interface for users to navigate through

Schedule

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		1/31/22	2/7/22	2/14/22	2/21/22	2/28/22	3/14/22	3/21/22	3/28/22	4/4/22	4/11/22	4/18/22	4/25/22	5/2/22
2	**Deliverables**											SLACK	SLACK	
3	Proposal Presentation Slides	All												
4	Proposal Presentation	JK												
5	Design Presentation Slides			All										
6	Design Presentation			JU, WM (TBD)										
7	Final Presentation Slides											All		
8	Final Presentation											JU, WM (TBD)		
9	Public Demo													All
10	Final Report													All
11														
12	**Logistics**													
13	Ordering all components	All												
14	Ethics Section							All						
15	Weekly Status Reports		All											
16	Individual Status Reports		All											
17														
18	**Signals**													
19	Play with Emotiv headset to get an idea of its capabilities and features	All												
20	Research signal processing techniques for EEG signals	JU												
21	Design movement tasks to obtain different control data sets		All											
22	Design & implement signal processing algorithm					JU								
23	Build our own EEG (expansion) eg. motor cortex detection													
24														
25	**Hardware Drivers**													
26	Link EEG headset to hardware (RPi/computer)			JK, JU										
27	Create a device driver/acquisition system for OS/Pi						JK							
28														
29	**Interface (Software)**													
30	Create wireframes for design of desktop app		WM											
31	Reading documentation for Swift and Flutter		JK, WM											
32	Setting of RPi (expansion)			JK (TBD if needed)										
33	Testing communiciations between device/interface (integration & debugging)									All				
34	Code a user interface design for desktop app						JK, WM (layout/design)	(design)	(implement)	(implement)	(integrate)			
35	Refining UI											JK, WM		

Conclusion

- Aim to increase computer accessibility for those where a traditional mouse and keyboard would be impossible or difficult to use
- Provide an integrated control interface system that is easily accessible and customizable for users