

VRChatと脳波測定器を連携させるまで

- 名前：彩瑞スピス
- 日付：2023/8/17

目次

1. 脳波について
2. 制作物の紹介
3. 完成に必要な知識の体系
4. ロードマップ

脳波とは何か

脳波とは

脳の活動を電気信号として捉えたもの

もう少し詳しく言えば「主に大脳皮質の大錐体細胞で生じるシナプス後電位の集合電位の変動を、頭皮上等で電極で記録したもの」です。

つまり、脳の活動を指すものではなく測定データを指します。

今回はこの測定データをVRChatで利用するというのが主旨になります。

※英語表記では**EEG**（Electroencephalogram：脳電図）と記述します。

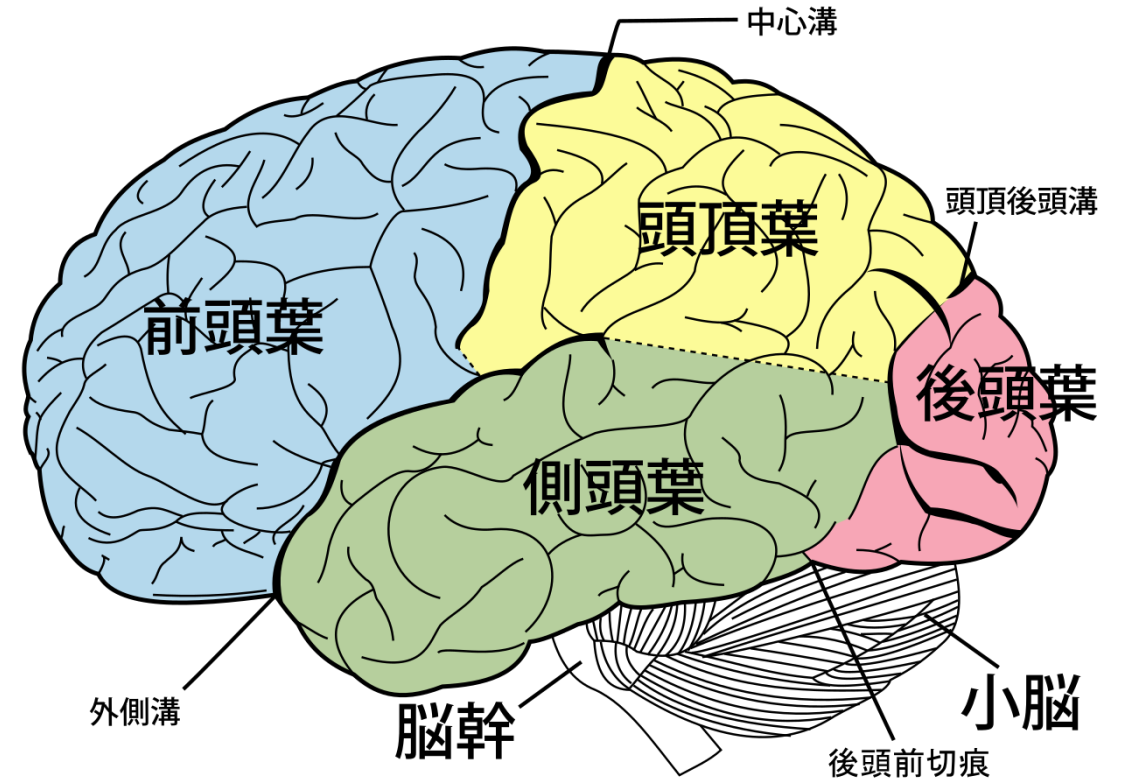
測定方法

大脳を領域別に区分されることが多いですが、現在までに知られているものとしてブロードマンの脳地図

(右図) があります。

脳波の測定ではこのような脳地図を参考に電極を配置します。

今回のように頭皮上に電極を配置する場合、脳波は微小な電圧（数 μV ～数十 μV 程度）で観測されます。



測定で考慮すべき要素

脳波は以下のような複数の要素を考慮して測定する必要があります。

1. **電極の位置**：一般的に「国際10-20法」等を用いる
2. **参照電位の有無**：基準電位として耳たぶ等が用いられる
3. **侵襲/非侵襲**：非侵襲では頭皮上等で測定し、侵襲では開頭手術が伴うことがある
4. **入力抵抗**：電極同士が影響しないように位置を離す
5. **接触抵抗**：頭皮環境や電極等の要素で導電性を高める
6. **誘発対象の選定**：測定対象によって測定方法や測定時間等が異なる
7. **測定環境**：電源線や筋電図等の外部要因の影響を受ける

誘発方法

脳波の電気活動として様々な反応を取得できます。

- 振幅での反応
 - 事象関連電位
 - 誘発電位
- 周期的な反応
 - 基礎律動
 - 事象関連同期・脱同期 など

測定機器

脳波を測定する装置は **脳波測定器（脳波計）** などと呼びます。

近年では **BCI（Brain-Computer Interface）** と呼ばれる、測定した脳波を特定の用途に向けて、脳波測定器（脳波計）と情報通信機器（コンピュータ）を接続する技術が登場しています。

インターフェースとしての役割を担う機器やシステム等もBCIと呼称されます。

したがって、今回の制作物もBCIの一種と言えます。

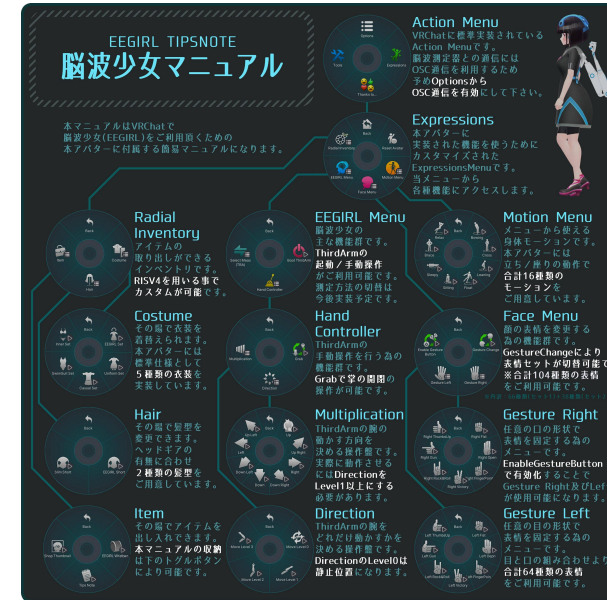
※BMI（Brain Machine Interface）と呼ぶこともあります但明確な区別はないようです
本スライドではVRChatとの連携を前提とするため表記を**BCI**に統一します。

何を作ったのか

VRChatと脳波測定器を 連携するシステム

BCIのシステムと言うと大げさですが
主に以下のものを制作しています。

- 脳波測定器：DIY Kitを利用
- 連携用のアプリ: Pythonで実装
- 対応アバター: 右上のアバター
- ワールドギミック: Shaderなど
- OSCテストツール: PurrData用
etc...



脳波測定器の作成

まずは、脳波測定器の作成から紹介します。

OpenBCI社の自作用のキット
(右図) を輸入して制作しました。

小売価格: 約16万円 (\$1,499.99USD)

関税・送料・消費税込み: 約20万円

3Dプリントはサイズや価格の面から
品川産業支援交流施設 (SHIP) で行
いました



連動アプリの作成

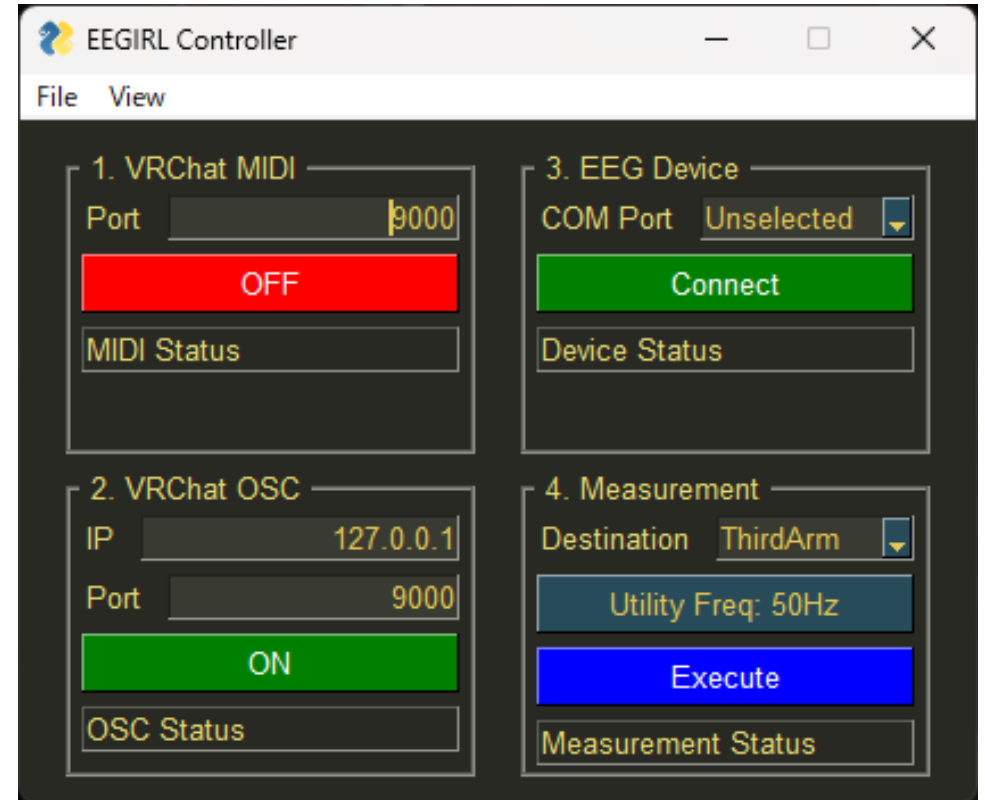
脳波測定器とVRChatを連動させるためのアプリを作成しました。

脳波測定器から取得した生データを加工してOSCとMIDI経由で制御値を送信します。

開発言語：Python

開発環境：VisualStudioCode

ビルド環境：DockerDesktop

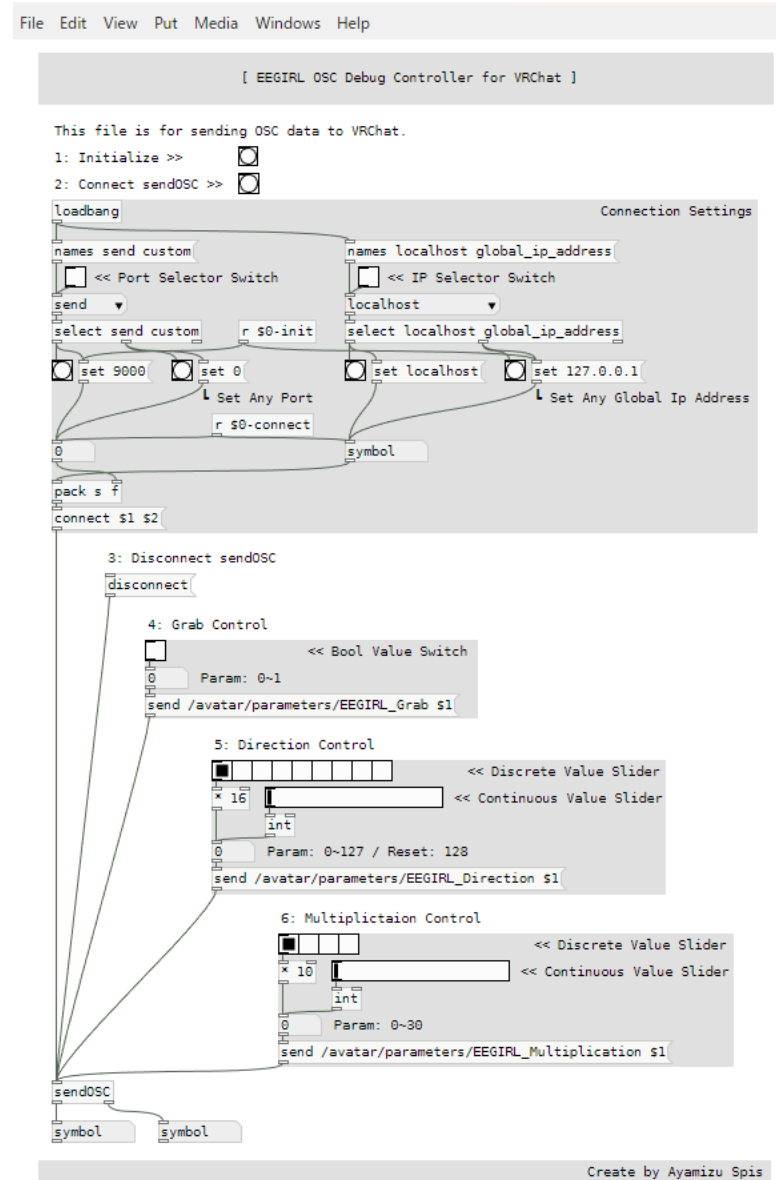


補助ツールの作成

開発中にアプリから送信する信号情報を読み取ったり、逆にVRChatに手動で信号を送信したいと思いました。

そこで、OSC通信の送受信テストを行うツールを作成しました。

使用ソフト：PurrData



専用ワールド

視覚からの刺激を用いた測定方法であれば、VR空間内に測定環境を構築できるのではないかと。

そう思ったので前々から専用ワールドを作成したいと思っていました。折角ならそのワールドで技術交流もできたらいいなと思いました。

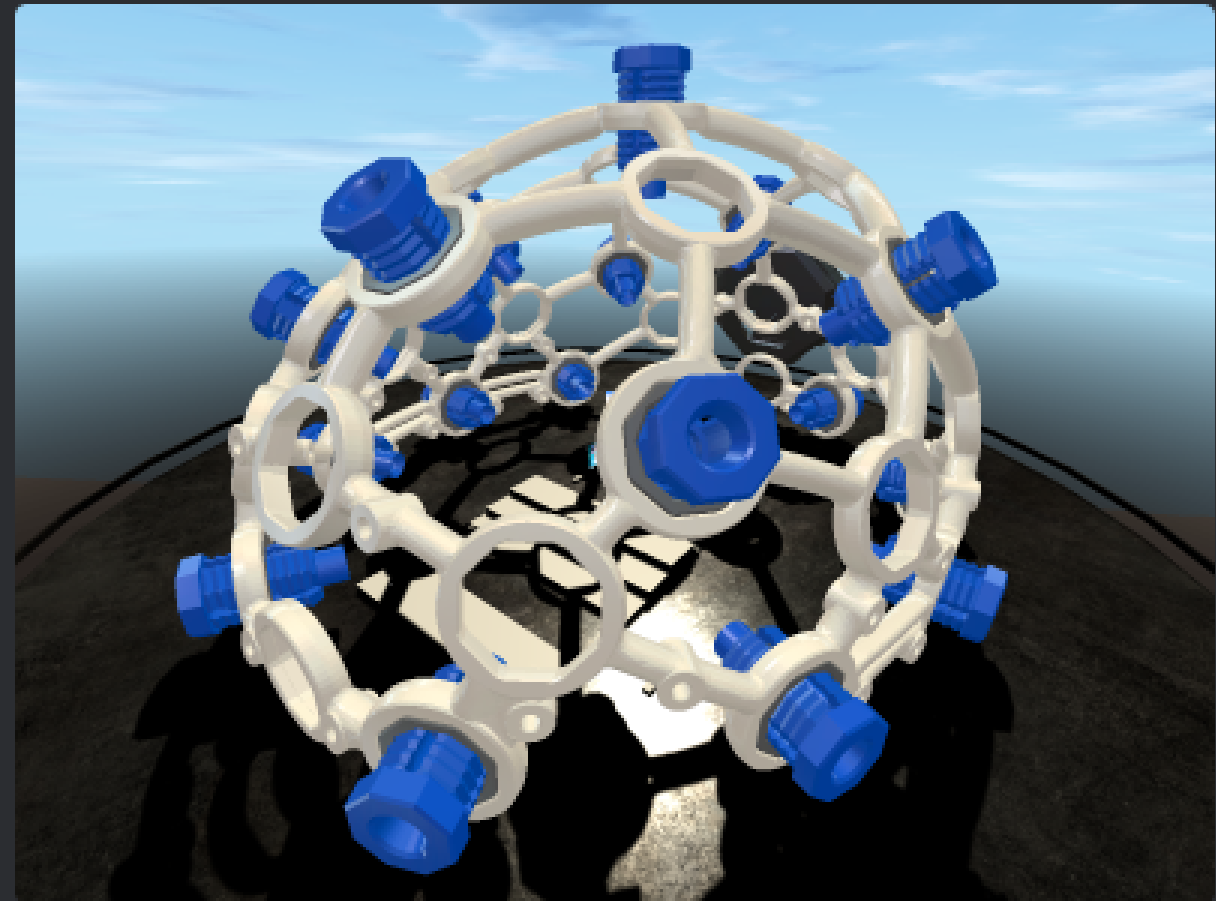
そのような経緯で

- ・ **実験用の専用ワールドの制作**
- ・ **VRC脳波技術集会の開催**

に至りました。

VRC脳波技術集會会場 by Spis (スピス)

VRC脳波技術集會の会場になります！ #VRC脳波技術集會 @ Spis (スピス) World Version: 1.0



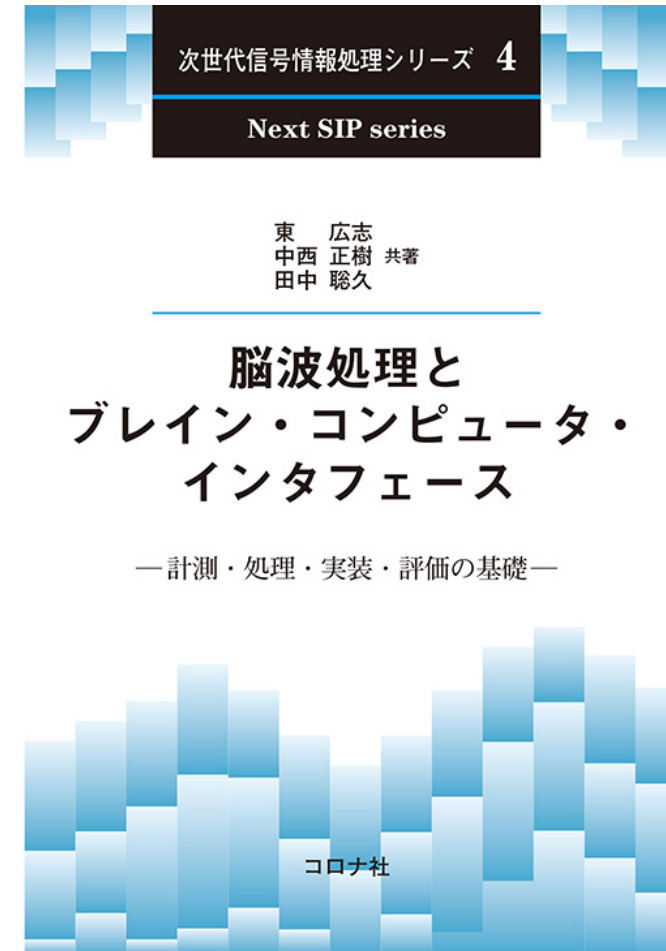
どうやったのか

技術調査

まずは技術調査を行いました。

偶然、脳波技術に触れている知り合いがいたので、その方に相談しながら論文や書籍を参考に技術調査していきました。

右図の書籍は特に良かったオススメ本として紹介します。

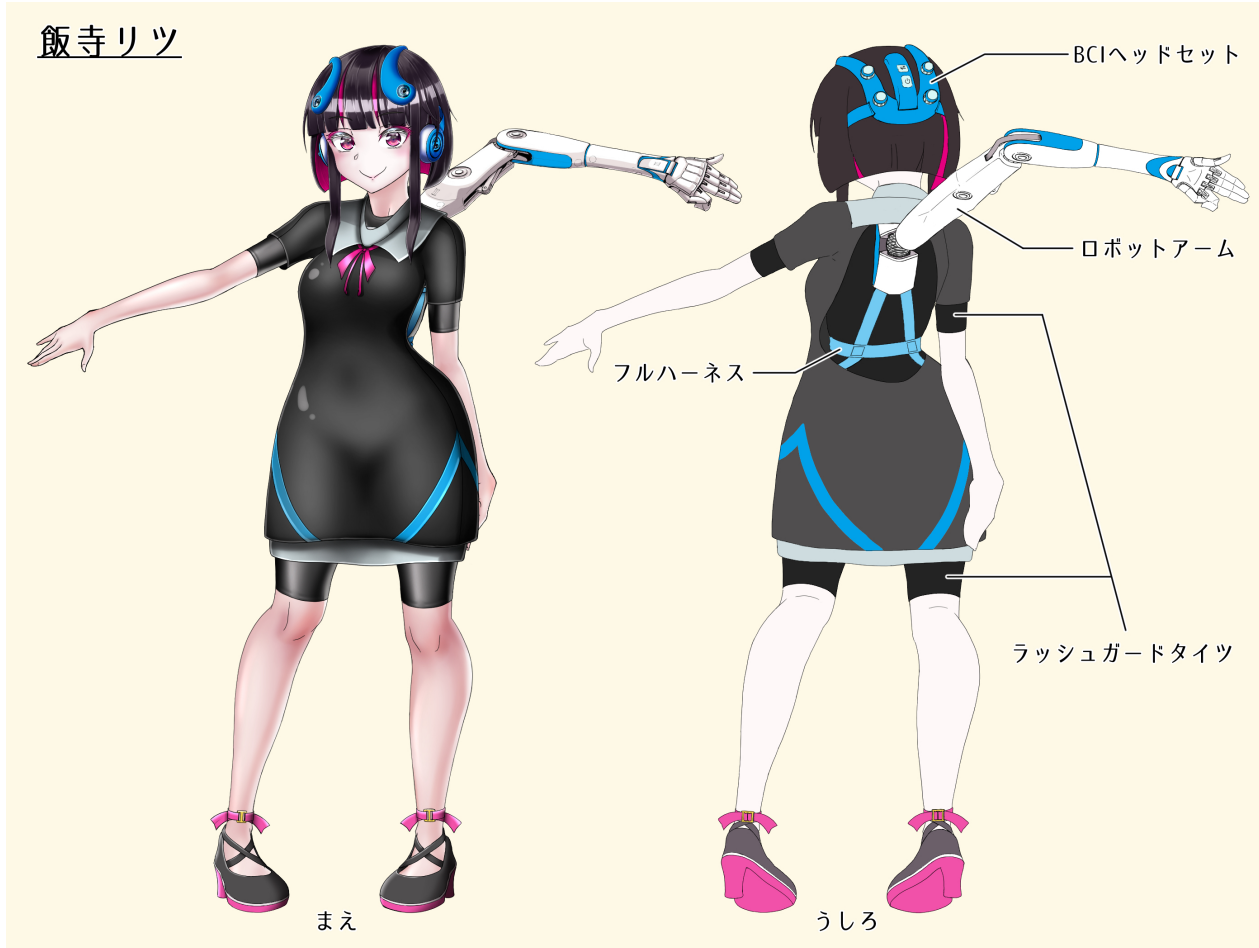


アバターのコンセプト

次に技術調査と並行で制作する
3Dアバターのイメージを固めました

衣装デザインを考えるのが素人には
難しかったので、髪型・ヘッドギ
ア・配色からあたりをつけて描きま
した。

使用ソフト：ClipStudioPaint

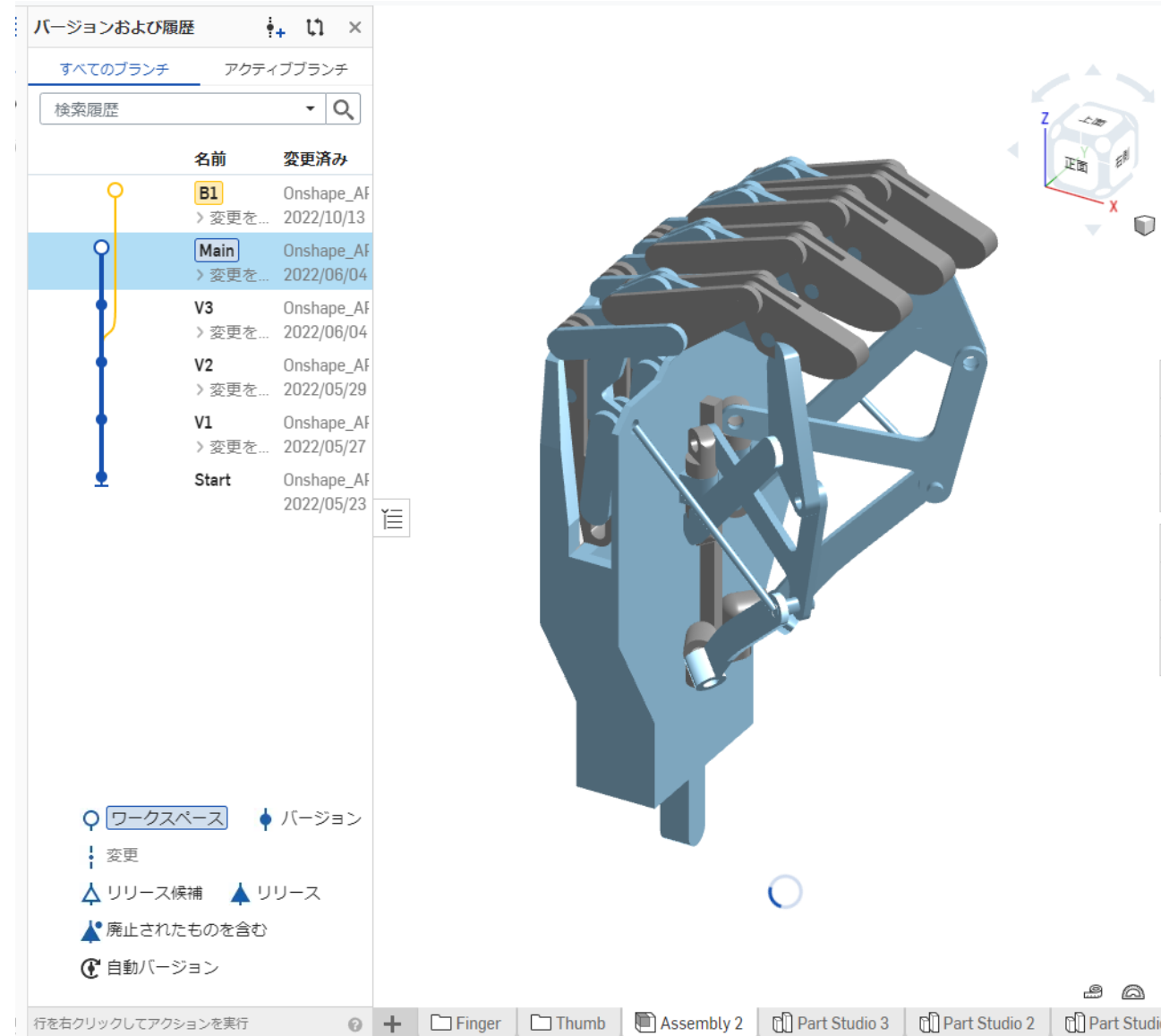


ロボットアーム設計

アバターのデザイン決めの際に、ロボットアームがちゃんと動くのかを確認したくなりました。

そこで、CAD（3D設計支援ソフトウェア）で構造設計しました。

使用CAD：Onshape

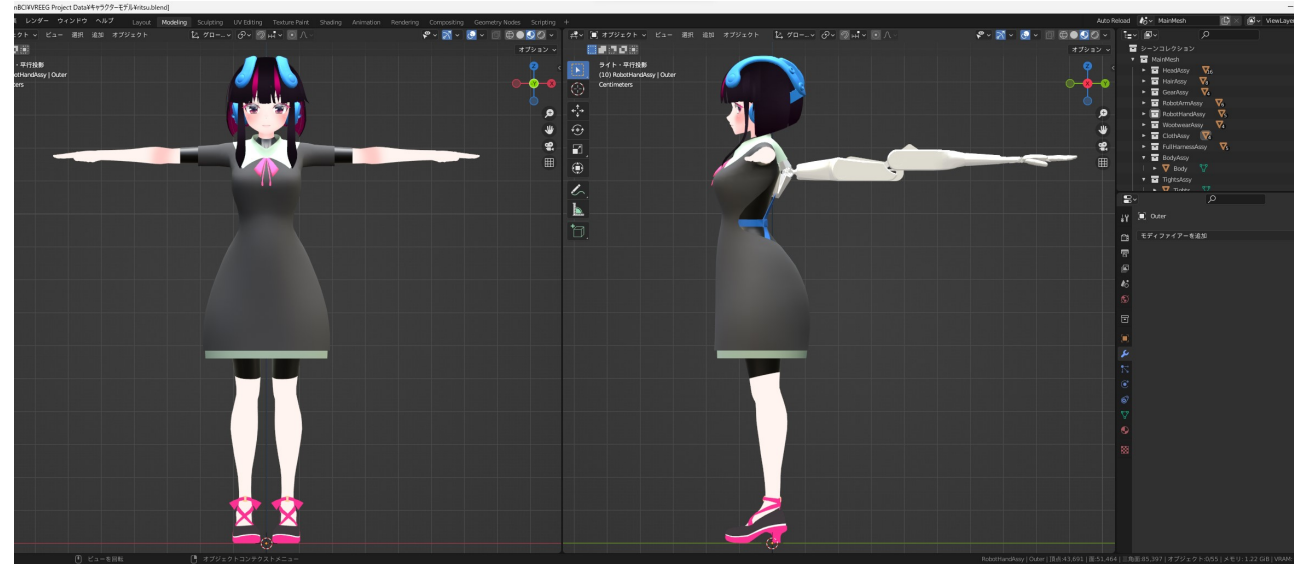


モデリング

3DアバターのモデリングはBlenderを使用して行いました。

なお、BlenderやUnityで駆動情報が引き継げなかったため、代替としてAnimationClipで拳動を再現することで対応しました。

使用ソフト: Blender, ClipStudioPaint



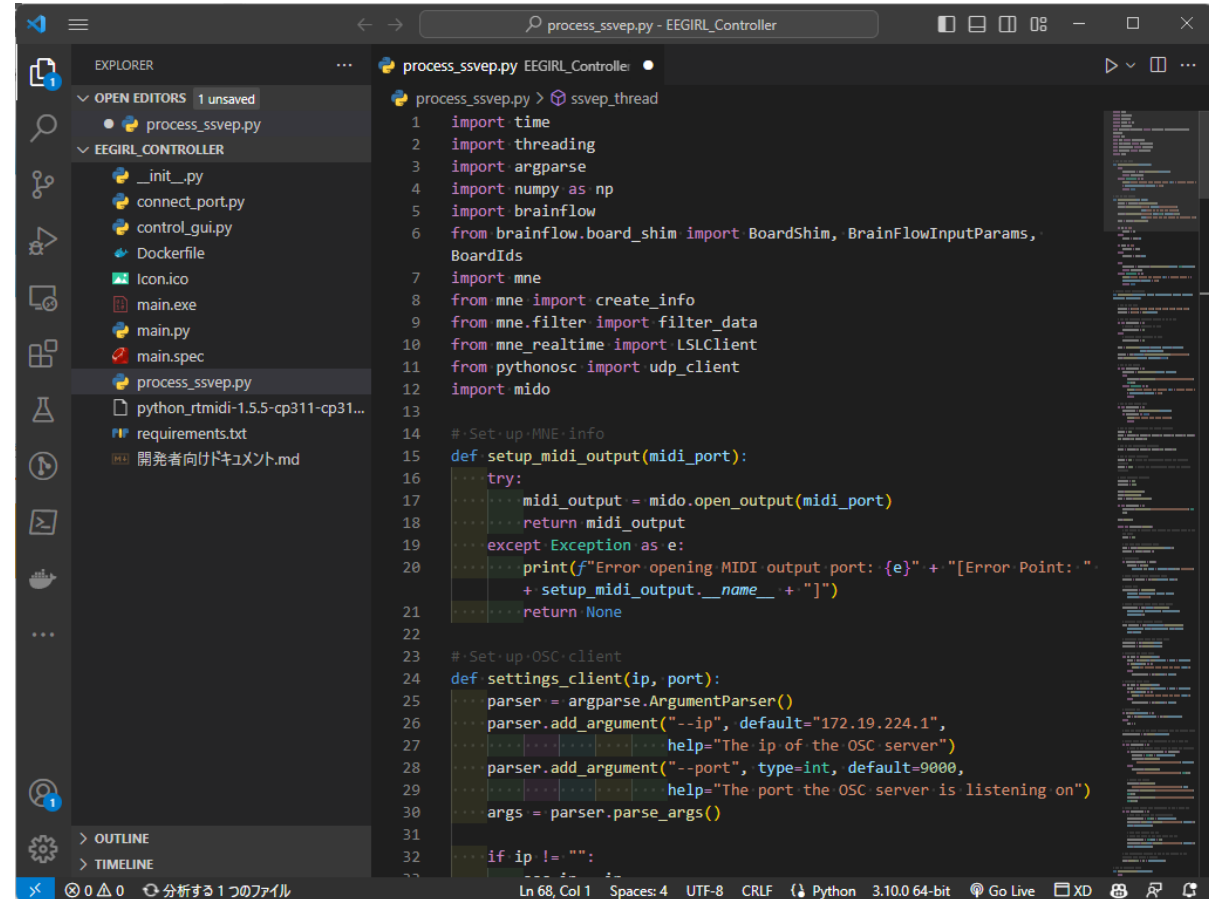
アプリ開発の構成

脳波測定器とPCの通信

Neuromore等のソフトウェアを利用しても良かったのですが、諸課題により通信処理も実装しました。

実装はPython言語で行い、以下のライブラリを用いました。

- PySerial : シリアル通信用
- Brainflow : 脳波測定器の測定用



```
process_ssvep.py EEGIRL_Controller
process_ssvep.py > ssvep_thread
1 import time
2 import threading
3 import argparse
4 import numpy as np
5 import brainflow
6 from brainflow.board_shim import BoardShim, BrainFlowInputParams, BoardIds
7 import mne
8 from mne import create_info
9 from mne.filter import filter_data
10 from mne_realtime import LSLClient
11 from pythonosc import udp_client
12 import mido
13
14 # Set-up MNE info
15 def setup_midi_output(midi_port):
16     try:
17         midi_output = mido.open_output(midi_port)
18         return midi_output
19     except Exception as e:
20         print(f"Error opening MIDI output port: {e}" + "[Error Point: "
21             + setup_midi_output.__name__ + "]")
22         return None
23
24 # Set-up OSC client
25 def settings_client(ip, port):
26     parser = argparse.ArgumentParser()
27     parser.add_argument("--ip", default="172.19.224.1",
28                         help="The ip of the OSC server")
29     parser.add_argument("--port", type=int, default=9000,
30                         help="The port the OSC server is listening on")
31     args = parser.parse_args()
32     if ip != "":
```

アプリ開発の構成

測定値の解析と制御値への変換

Python言語とmneという脳波解析に有用なライブラリを用いて脳波測定器から取得した生データを解析します。

解析の構成はざっくりと以下のようになっています。

- 前処：フィルタリングやノイズ除去等
- 識別：誘発電位の周波数の検出等

最終的に閾値を用いて制御値に変換します。

アプリ開発の構成

VRChatへの転送

連携アプリを用いてOSC通信経由で制御値をVRChat OSCに送信します。
MIDI通信にも対応しているので、
MIDI経由でワールドに制御値を送信することも可能です。

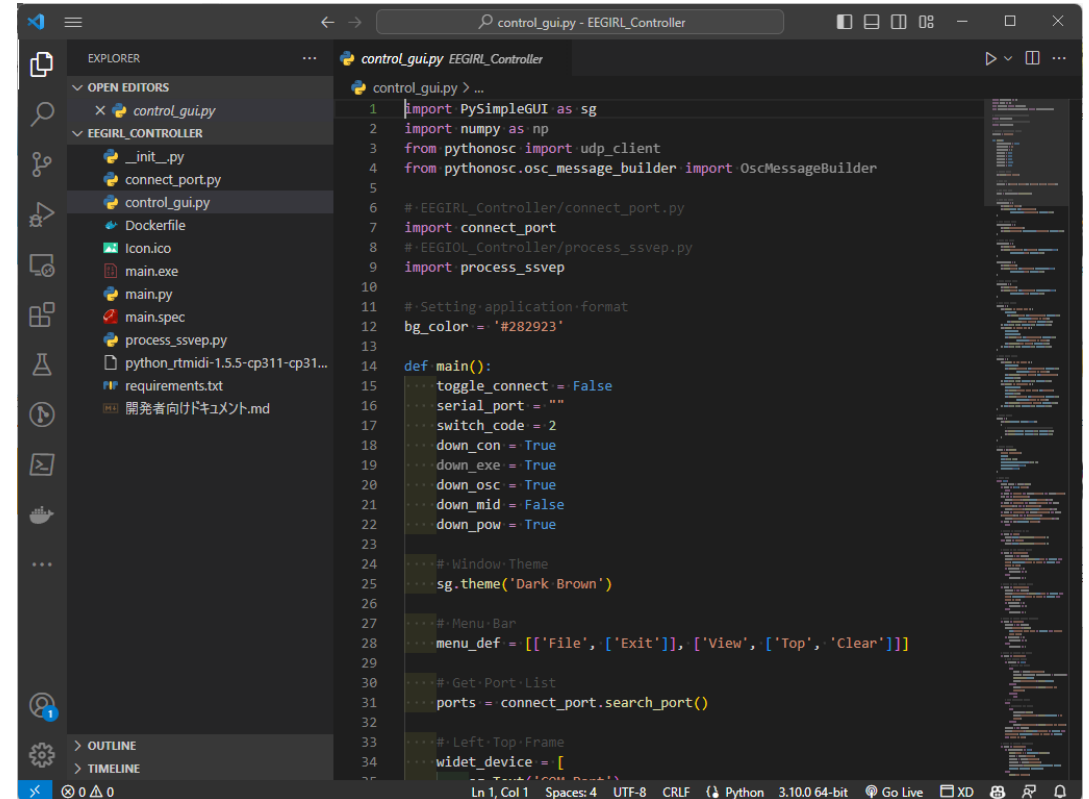
- python-osc: OSC通信用
- mido: MIDI通信用

アプリ開発の構成

GUI操作

一連の過程をGUI操作で実行したかったのでPySimpleGUIを用いてGUIを実装しました。

- PySimpleGUI: GUI用



```
control_gui.py - EEGIRL_Controller
control_gui.py > ...
1 import PySimpleGUI as sg
2 import numpy as np
3 from pythonosc import udp_client
4 from pythonosc.osc_message_builder import OscMessageBuilder
5
6 # EEGIRL_Controller/connect_port.py
7 import connect_port
8 # EEGIRL_Controller/process_ssvep.py
9 import process_ssvep
10
11 # Setting application format
12 bg_color = '#282923'
13
14 def main():
15     toggle_connect = False
16     serial_port = ""
17     switch_code = 2
18     down_con = True
19     down_exe = True
20     down_osc = True
21     down_mid = False
22     down_pow = True
23
24     # Window Theme
25     sg.theme('Dark Brown')
26
27     # Menu Bar
28     menu_def = [['File', ['Exit']], ['View', ['Top', 'Clear']]]
29
30     # Get Port List
31     ports = connect_port.search_port()
32
33     # Left Top Frame
34     widget_device = [
35         sg.Text('COM Port')
36     ]
```

アバター側の実装

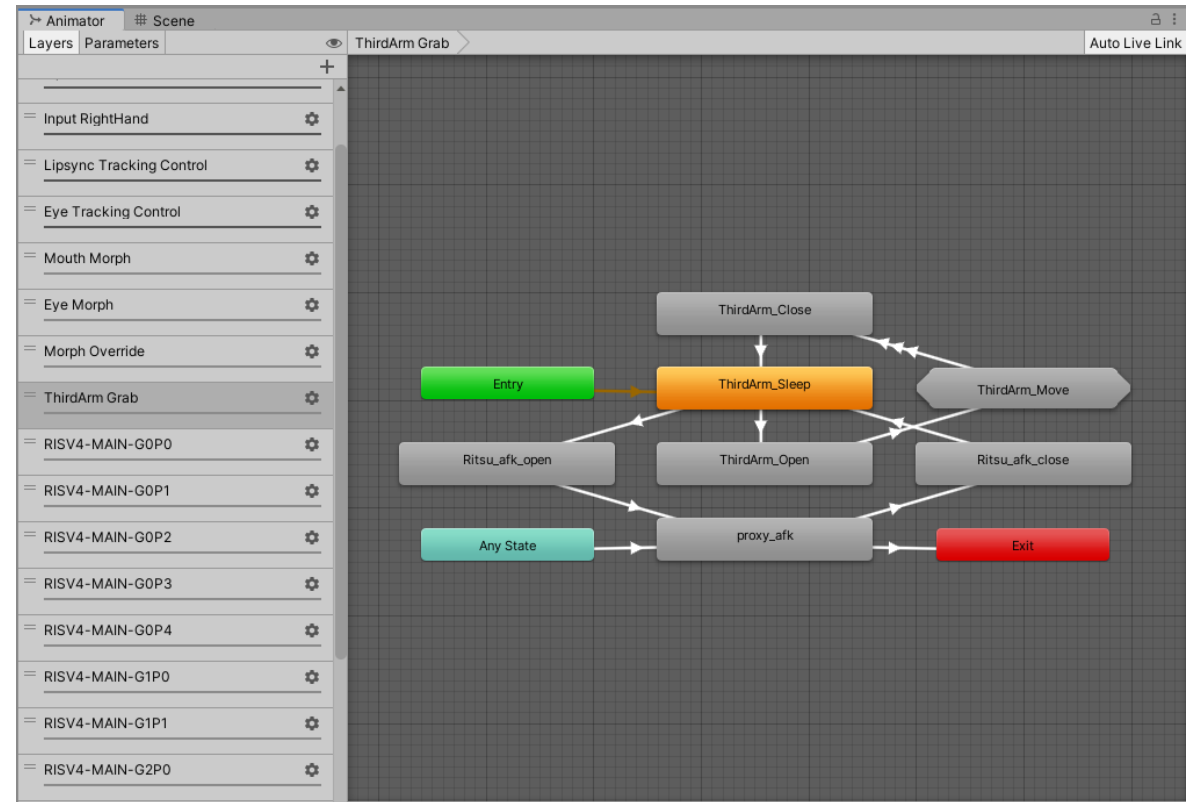
OSC通信を用いてVRChatのアバターのアニメーションを動かします。

VRChatSDKや特有の制約が多く、アバターの制御には苦労しました。

今後の派生コンテンツの制作で一番大変になりそうな部分です。

開発環境：Unity

使用言語：C#, U#, Shaderlab



課題と目標

課題 1

正直完成できると思わなかったが、測定環境も相まって精度の低いのが悩み。
15分試行を10回繰り返しても、1回も正解できないこともある。

課題 2

現在使っているBCIが高すぎる上に、医療機器対象であるためにコンシューマ向けに販売する国内メーカーが存在せず敷居が高い。安い端末を輸入してテストしたり、あわよくば自作したいとも思ってる。

課題 3

コンテンツを作って盛り上げることが目的だけでも、自分が出しているコンテンツの量も少なすぎるので今後も作り続けたいと思ってる

課題 4

機器不足・機能不足・人手不足。イベントを通じて増やしたい