

## 6 メンタル管理を可能にするアプリの開発

鎌田 樹, 笹森 悠誠, 田鎖 礼夢

指導教員 飯坂 ちひろ

### 1. はじめに

1年次に学んだ Arduino の授業やインターネットでの掲載記事等を通じて, IoT に強い興味を抱いた.自分でも IoT 技術を用いて, 人々のメンタル管理を支援するアプリケーションを制作できないかと模索していた.

卒業研究テーマとして,「メンタル管理を可能とするアプリ (メンカノ) の制作」を設定した.本研究では「音楽で人のメンタルをコントロールすること」を目標とする. このプロジェクトでは, Raspberry Pi を活用して, 心拍センサーなどのセンサーを用いてユーザーのメンタル状態をリアルタイムでモニタリングし, 適切な音楽を再生するなどの手段を通じてメンタルの変化を実現する.

### 2. 研究概要

#### 2.1 概要

本研究ではメンタルを変化させるために, イライラ棒のような高ストレスが掛かるゲームをプレイしてもらい, センサーを使いリアルタイムで各種値を測定しリラックスする音楽や逆に更にストレスが掛かるような音楽を流す.

また, センサーを使い各種値を取得する部分を「オブザーバー」, 各種値を使い値を処理する部分を「処理システム」, ゲーム画面等のフロントエンド部分を「怒落絵」といい, これら三つを合わせたものを「VIOS」とする. また, 音楽を流す処理と VIOS を合わせて「メンカノ」とする.



### 2.2 開発環境

OS	Windows10
開発環境	Visual Studio Code
言語	C#,C,HTML,CSS,SCSS,JavaScript
CO2 センサー	Grove-VOC and eCO2 Gas Sensor
パルスオキシメーター/心拍センサー	Qwiic - MAX30101/MAX32664
サーマルカメラ	amg8833

### 3. 研究内容

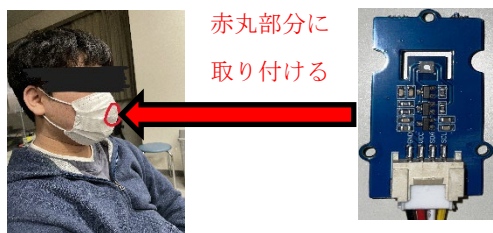
#### 3.1 「オブザーバー」について

##### 3.1.1 開発環境

作業するにあたり, RaspberryPi はヘッドレスで開発することにした. GUI は RealVNCViewer で操作することとし, コーディング及びデバックなどは WindowsPC の VisualStudioCode から SSH 接続して行った. これらの設定によって現在は RaspberryPi を電源につないでだけで, すべての操作を WindowsPC から行えるようになっている.

##### 3.1.2 センサー

センサーとの通信はすべて I2C で行う. 「Grove-VOC and eCO2 Gas Sensor」と「Qwiic-パルスオキシメーター/心拍センサーモジュール」はメス-メスのジャンプワイヤで RaspberryPi に接続することが難しそうだったため, オス-メスのジャンプワイヤを自作した. パルスオキシメーター/心拍センサーモジュールはピンヘッダがはんだ付けされていないため自分でつけた.



3.2「処理システム」について

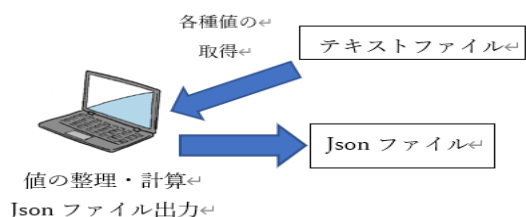
オブザーバーで得た各種値をテキストファイルで受け取り,その情報を基に「リラックス度」を計算し「心拍数」,「体温」,「血中酸素濃度」,「CO2濃度」,「リラックス度」を怒落絵に json ファイルとして送っている。

リラックス度を出すにあたって,先行研究がなかったので心拍数の最大値と最小値(45~130),体温の最大値と最小値(35.5~38.0)を設定し,それを基に独自で計算式を出しそれぞれ 0~50 で評価し合算してリラックス度を算出している。計算式は以下に示す。

式: $50 - \text{Math.Max}(0, \text{Math.Min}(50, ((\text{センサーの値}) - \text{最小値}) / (\text{最大値}) - (\text{最小値}) * 50))$

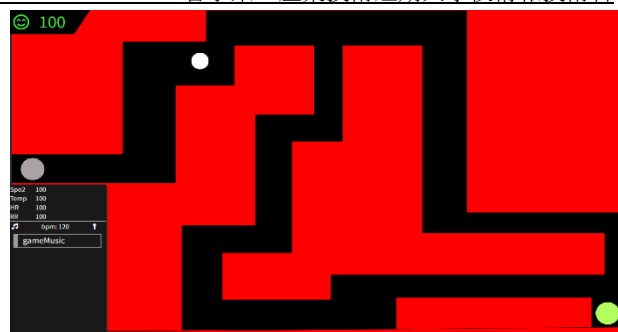
※Math.Max(値 1, 値 2) = 値 1 と値 2 を比較して大きい方の値を返す。

また,データの整理やオブザーバーのデータで容量を圧迫しないために各種値を読み取ったら読み取った値を消し,自動的に「年/月/日/時間\_分\_秒」の json ファイルに値を保存させている。



### 3.3「怒落絵」について

メイン画面, 楽曲画面, 設定画面, メニュー画面を実装し, メニュー画面からそれぞれの画面へ移動できるようにした。メイン画面ではゲームを, 設定画面ではゲームに関する設定や音量の調整や表示ウィンドウの大きさ等の設定を出来るようにし, 楽曲画面では制作した楽曲をシャッフルや単体で試聴出来るようにした。以下にメイン画面を示す。



また, 難易度機能を実装し「Easy」「Normal」「Hard」の順徐々に道が狭くなり, 曲がり角が増えゲームが難しくなっていく曲も激しくなり, 許容される「リラックス度」の値も厳しくなりクリア難易度が上がる。ゲームについては, Unity を用いて自作した。

また, ゲーム画面では処理システムが作った json ファイルを読み取り, 「リラックス度」「心拍数」「流している曲名」「リラックス度のグラフ」を常時表示している。

### 3.4「楽曲制作」について

本アプリの音声再生機能は, あらかじめ用意している音声ファイルをメンタル状態に応じて選択する仕組みになっているため, 曲にラベル付けがされているべきだと考えた。そのため作成した曲のファイル名には, 曲の BPM とジャンル, 受ける印象などを含めるようにした。さらに, メンタル状態に与える効果として, 曲は「興奮作用のあるもの」と「鎮静作用のあるもの」の2つに分けた。以下に現在制作した曲のファイル名を列挙する。

○興奮作用のあるもの

- ・ bpm173\_bluesrockhuu
- ・ bpm72\_rockhuu …etc

○鎮静作用のあるもの

- ・ bpm40\_bukimi
- ・ bpm111\_melow …etc

## 4. 終わりに

本研究では音楽でメンタル管理を可能にすることを念頭にアプリケーションを作成してきた。アプリケーションの作成は概ね順調に進んでいったが, センサーから値を取得するのに想定より時間を掛けてしまい苦労した点はあったが, 問題解決能力向上に役立ち, 有意義な時間となった。