

オープンデザイン実践 最終レポート「Reaction Flower」

環境情報学部 1年 改野 由尚
学籍番号: 72041901 / CNS: t20190yk

Reaction Flower

教員向け・授業支援ツール



目次

目次

はじめに

Reaction Flowerとは

先生側

生徒側

技術仕様について

なぜObniz?

ハードウェアの仕組み

開発の流れ

ソースコード

Core

Viewer

公開用ページ

Viewer

発表スライド

ライセンスについて

コントリビュートについて

課題

所感

はじめに

もしあなたが大学の先生として、多くの大学生に対して授業をしてください、と言われたらどうでしょうか。それも、対面ではなくオンラインで、ZOOMなどのツールを使って、です。授業の内容さえあれば、しゃべれたら大丈夫だろう。そう思った方もいるのではないかでしょうか。

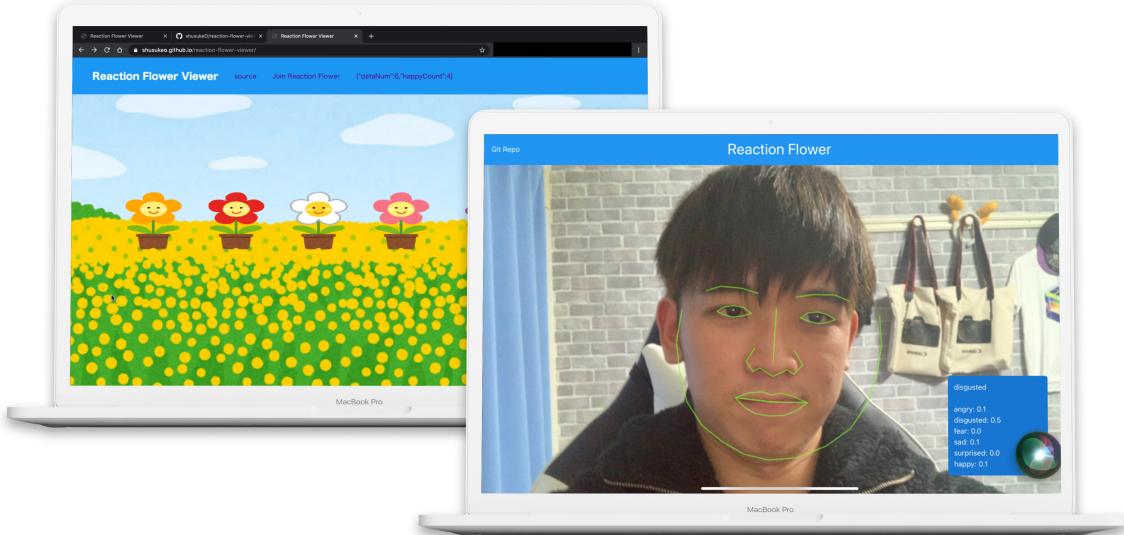
しかし現実はそう甘くありません。生徒からの授業の評価はとても大切です。面白い授業を提供できなければつまらないとSNSで叩かれることはもちろん、授業中に寝てしまう学生が出たりすることや、最終的に自分の評価を下げてしまうことに繋がりかねないでしょう。そのため、授業内での学生の状況を先生はある程度把握する必要があるわけです。

ただ、授業中に全員の表情を見るためにいちいちZOOMで顔を確認していくは、授業の進行の妨げになりますよね。それを解消するために、学生の表情などをひと目見たらわかる、据え置き型のデバイス **Reaction Flower (リアクション・フラワー)** を制作しました。

Reaction Flowerとは

Reaction Flowerは、ハードウェアデバイスとアプリケーションがセットになり動作します。先程のシナリオのように先生側と生徒側で手順をまとめました。

Reaction Flower Viewer



Reaction Flower - Core

先生側

先生側はハードウェアを用意するだけ。かんたんです。

学生の過半数が笑ったり、驚いたりした表情をすると揺れはじめると、学生の顔を見ることなくリアクションを確認することができます。

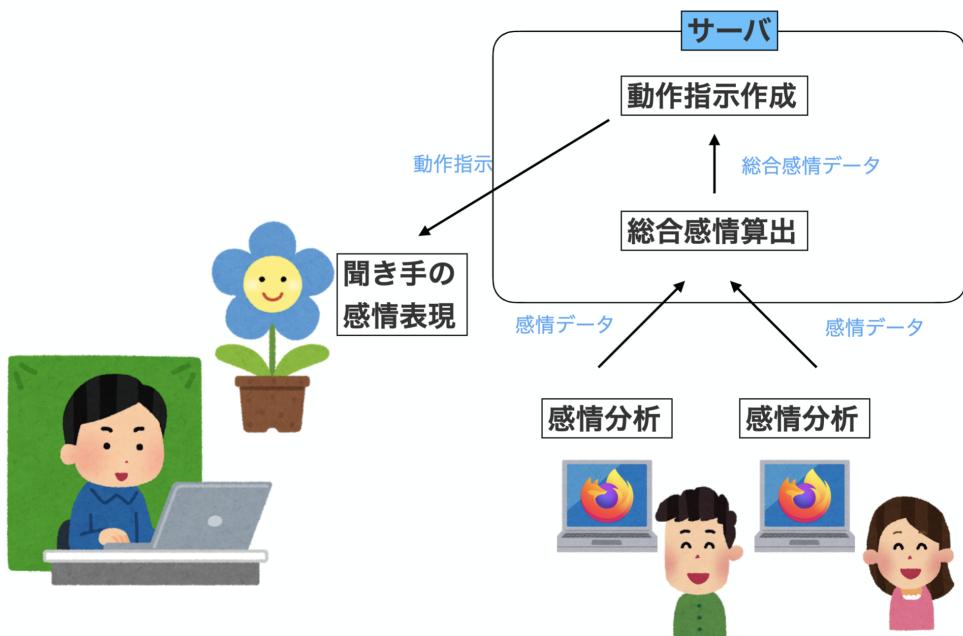
デバイスなしで動作させたいときはWebページから確認できる [Reaction Flower Viewer](#) もあります。

生徒側

ZOOMの他に、ブラウザなどでこのアプリケーションの画面を開いておくだけです。表情を読み取り、サーバーに送信、結果が先生側に反映される。とても単純な仕組みです。

技術仕様について

技術仕様は以下のとおりです。



- node.js (コア)
 - socket.io (情報の送受信)
 - express (ページ表示)
 - clmtrackr.js (表情分析)
 - materialize.css (ページの装飾)
 - obniz.js
- HTML (Viewer)
- obniz (ハードウェア)
 - 電磁石で動作する振り子
 - LED
- heroku (ホスティング)

私はnode.jsのコア部分、Obnizの動作の部分などを包括的に担当しました。

なぜObniz?

Obnizはインターネットを通してプログラムを実行できることと、node.js で動作させることができる点、自身が書き慣れておりコーディングしやすいことが魅力だったため採用しました。

また、モバイルバッテリーなどでも動作するため、今回のコンセプトであるスタンダローンで動くという点でもコードを外に出すことなく動作させることができました。

ハードウェアの仕組み



100円ショップなどで市販されている、日光などの光で動く振り子を使用しました。

それぞれのコードをObnizのIOに接続し動作させています。

開発の流れ

スケジュール

Aa 工程名	担当者	日時
<u>全体：アイデア固め</u>	Hayashi Kaino Okihara	@2020/11/04
<u>フロントエンド：顔認識</u>	Kaino Okihara	@2020/11/06
<u>ハードウェア：構築</u>	Hayashi	@2020/11/10
<u>バックエンド：Socket.ioでの通信</u>	Kaino	@2020/11/10
<u>バックエンド：Obnizの動作</u>	Kaino	@2020/11/11
<u>全体：バックエンドとハードの接続</u>	Hayashi Kaino	@2020/11/11
<u>フロントエンド：WebAPIの実装</u>	Okihara	@2020/11/27
<u>フロントエンド：Viewer作成</u>	Okihara	@2020/11/28

Aa 工程名	☰ 担当者	🕒 日時
<u>ハードウェア: LEDなどの追加</u>	Hayashi	@2020/11/28

かなりギリギリのスケジュールながらも、ある程度形にすることができたのではないかと思います。

(まだまだ課題もたくさんあり、心残りはあります)

ソースコード

Core

yosh1/reaction-flower

This application can detect your emotions through your facial expression. There are six emotions that can be detected and the emotions are angry, disgusted, fear, sad, surprised and

⌚ <https://github.com/yosh1/reaction-flower>



Viewer

shusukeO/reaction-flower-viewer

Contribute to shusukeO/reaction-flower-viewer development by creating an account on GitHub.

⌚ <https://github.com/shusukeO/reaction-flower-viewer>



公開用ページ

Viewer

Reaction Flower Viewer

<https://shusukeo.github.io/reaction-flower-viewer/>

発表スライド

<https://github.com/yosh1/reaction-flower/blob/master/assets/発表資料.pdf>



▲ 画面上では文字化けているので、ダウンロードして確認してください。

ライセンスについて

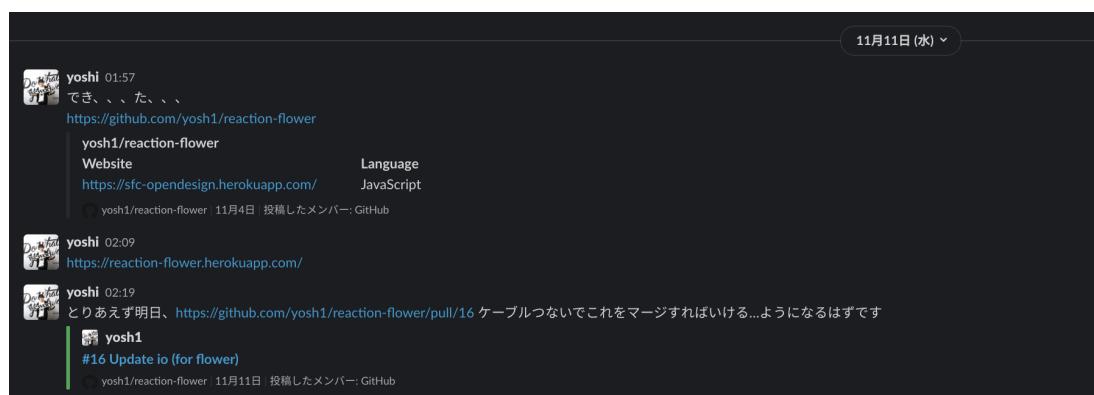
ライセンスには、自由に使っていいが、本許諾表示は書かないといけない。また、元の作成者は関知しないという、数あるライセンスの中で非常に制限のゆるいライセンスと言えるMITライセンスを設定しました。

コントリビュートについて

基本的にIssue / PRは自由に行っていただいて構いません。
通常1日以内に確認して返信するようにしています。

課題

- node.js + socket.io に関してはもっとやりようがあったのではないかと思う
 - Firebase Realtime DBなどを活用するのが良かったのでは
- Obnizの導入についても少し検討の余地がある
 - RasPi Zero + CHIRIMEN.org でも同じようなことができたような.....
- 分担の下手さ
 - 結局核となる部分を一手に引き受けてしまったのでプレゼン前日～当日に地獄を見ました。
当日も3時ぐらいまでコードを書いていました...



- 次回やるとしたら、ちゃんと分担を考えて、負担が均等になるようにしたいです。

所感

アイデアが右往左往し、はじめのほうは先生たちを焦らせた（であろう）このチームも、無事にアウトプットを出すことができたりあえずほっとしています。

一連の作業の中で、共同で行うからこそ生じる難しさがありました。先の課題の部分で述べたとおり、平等な仕事配分で各々が動くことができれば、幾分か楽に課題を進められたと思います。

このチーム開発で色々な経験を得られたと思うので、それを生かしてこれからも色々なものづくりに取り組んでいきたいと思います。

半期間という短い間でしたが、ありがとうございました。