

WebSocket を用いた リアルタイムウェブ環境の構築

釧路工業高等専門学校 情報工学分野

坂上 光紀

指導教員 柳川和徳

目次

1	はじめに	2
2	WebSocket とは	3
3	研究環境	4
4	研究方法	5
5	ファイル構成について	6
6	詳細な処理の流れ	7
7	研究成果	8
8	まとめ	9

図目次

1	イメージ図	2
2	HTTP 通信	3
3	WebSocket 通信	3
4	Node.js	4
5	Socket.IO	4
6	チャットアプリ実行画面	5
7	ファイル構成	6
8	三目並べの PC とスマホの画面	8
9	早押しゲームの PC とスマホの画面	8

1 はじめに

図 1 のように, 一台のパソコンをサーバ兼モニタとして, 複数台のスマートフォンまたはタブレットをクライアント兼コントローラとして扱い, それらが強調動作するためのリアルタイムウェブ環境の構築について検討した. そして, アプリを手軽に開発するためのライブラリを作成した.

なお, クライアント/サーバ間の通信には WebSocket を利用する.



図 1 イメージ図

2 WebSocket とは

従来のウェブ通信では HTTP が多く利用されている。しかし HTTP では、通信を開いてから閉じるまでに一度のやり取りしかできない為、複数のデータを扱う際に手間 (図 2 の青矢印の本数にあたる) が増える。

一方 WebSocket は、一度通信を開くと閉じるまで自由に複数のデータのやり取りが可能で、クライアント／サーバ間で双方向通信も可能である (図 3)。そのため、WebSocket はよりリアルタイムな通信が可能となる。

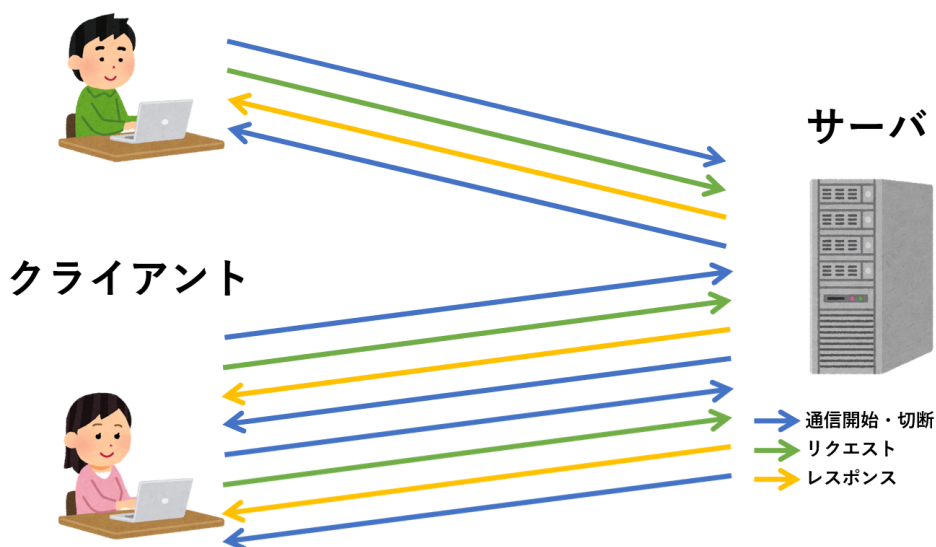


図 2 HTTP 通信

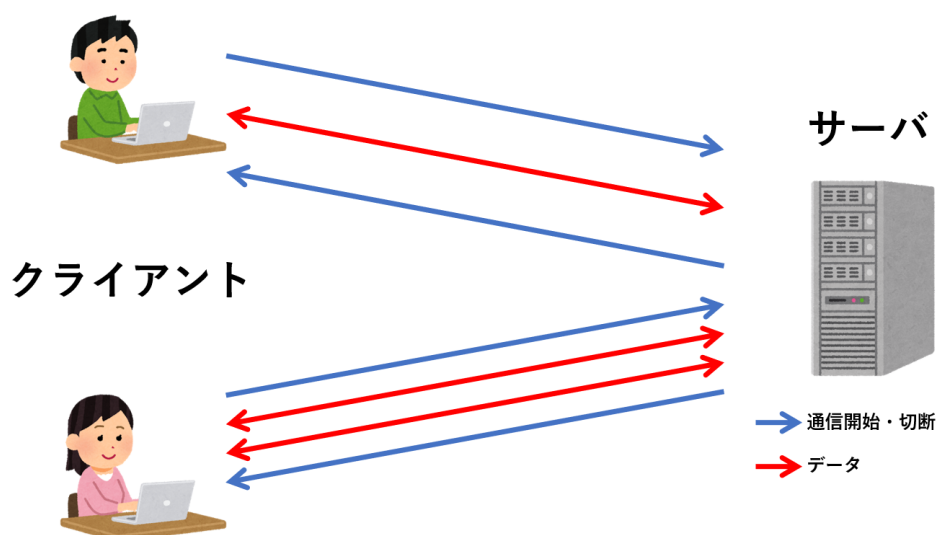


図 3 WebSocket 通信

3 研究環境

本研究には WindowsOS のパソコンを使用している。また, サーバサイドには JavaScript 環境である Node.js [1] を使用し, Node.js のライブラリである Socket.IO [2] で WebSocket 環境を作成した。

なお作成したアプリは, パソコンと AndroidOS のスマートフォン, タブレットで Firefox と GoogleChrome, iPhone で Safari での動作が確認できている。



図 4 Node.js



図 5 Socket.IO

4 研究方法

はじめに,WebSocket を用いてリアルタイムウェブ環境で動くチャットアプリの作成を行った [3]. 図 6 は実際のチャットアプリの画面で,ID を見ると別端末との通信が行われていることがわかる.

このチャットアプリを基に,スマホ画面のボタンを押すと対応する文字列が PC に送られ,その文字列に応じて PC の画面を書き換えるような動作に変更した. また,これらの処理をライブラリ化し, ボタンを用いた三目並べと早押しゲームを作成した.

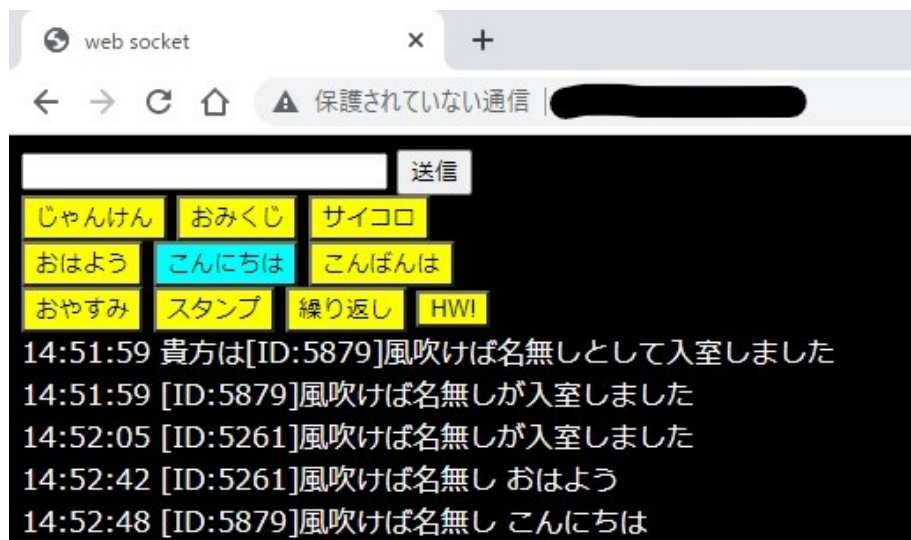


図 6 チャットアプリ実行画面

5 ファイル構成について

図 7 のように 3 つのファイルで構成されており, サーバファイルでサーバを立て, HTML ファイルで画面を構成し, スマホ画面のボタンが押されるなどの動作によりライブラリが呼び出され, 対応する文字列を WebSocket を用いてサーバへ送り, サーバからアプリへ送られる. その後, 送られた文字列に応じてアプリ内で処理を行い, 画面を書き換えるという流れになっている.

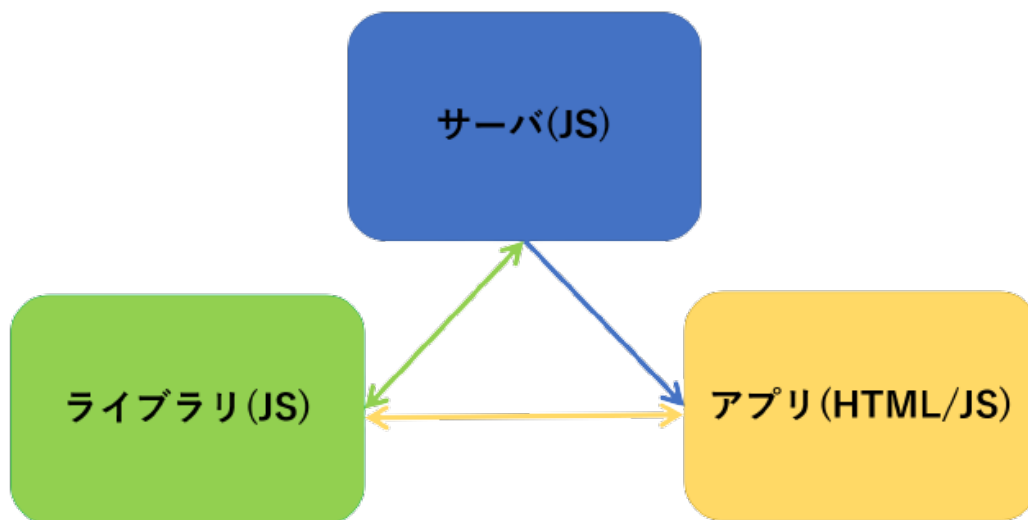


図 7 ファイル構成

6 詳細な処理の流れ

例として、三目並べアプリのボタンが押された際に、ボタンに対応する数字を引数として、ライブラリ内の関数 1 が呼ばれ、5 行目で受け取った数字をサーバへ送っている。

Listing 1 ライブラリ

```
1  function change_color(no){
2      var vname = myName;
3      socket.emit("button", {value:no});
4      socket.emit("Bingo", {value:vname});
5  }
```

次に、サーバ内の関数 2 で数字を受け取り、2 行目でアプリへそのまま送っている。

Listing 2 サーバ

```
1  socket.on("button", function(no){
2      io.emit("button", {value:no.value});
3  });
```

最後にアプリ内の関数 3 で数字を受け取り、その数字を引数として同じアプリ内の関数 4 を呼び出し、6 行目の switch 文で受け取った数字、即ち押されたボタンを判断し、8 行目の if 文ですでに押されているボタンかを確認し、押されていないならば 9 行目の PC の画面に表示されている対応するマスの色を変えるという動作を行う。

Listing 3 アプリ 1

```
1  socket.on("button", function(no){ change(no.value);});
```

Listing 4 アプリ 2

```
1  function change(no){
2      ct++;
3      if (ct % 2 === 1){
4          switch (no) {
5              case "0":
6                  if (bt1 === 0){
7                      button11.style="WIDTH:80px;HEIGHT:80px;background-color:
                        blue";
8                      bt1 = 1;
9                      break;
10                     } else ct++; break;
```

7 研究成果

図 8 と図 9 はそれぞれ、作成した三目並べと早押しゲームで、左から PC の画面、スマホの画面となっているが、ゲームの操作はほぼスマホのみで行うことができ、PC 画面が操作に応じてリアルタイムで変化するようにになっている。なお、スマホは複数台接続が可能である。

作成したゲームはどちらも HTML ファイルのみを書き換えて作られているため、ライブラリやサーバ側のファイルは書き換える必要がほぼなく、HTML と JS を修得していれば簡単に別の作品に応用できる。

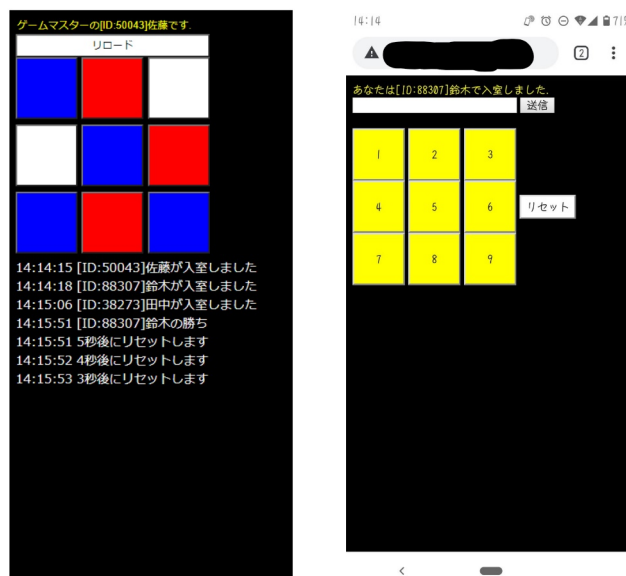


図 8 三目並べの PC とスマホの画面



図 9 早押しゲームの PC とスマホの画面

8 まとめ

リアルタイムウェブとライブラリを作成し、それを利用してチャットアプリと2つのゲームを作成した。また、作成したゲームはHTMLファイルのみ書き換えることで、簡単にほかの作品に応用が可能である。

参考文献

- [1] “Node.js”, <https://nodejs.org/en/>, 参照 Jul.2020.
- [2] “Socket.IO”, <https://socket.io/>, 参照 Jul.2020.
- [3] 齋藤 毅, “Node.js と Socket.IO でリアルタイムのチャットアプリを作ってみる”, [https://saitodev.co/article/Node.js と Socket.IO でリアルタイムのチャットアプリを作ってみる/](https://saitodev.co/article/Node.js%20と%20Socket.IO%20でリアルタイムのチャットアプリを作ってみる/), 参照 Jul.2020.