2013年度 ソフトウェア創造実習

実施計画・報告書

「機体の操縦の疑似体験を可能にしたレーシングゲームの開発」

情報通信工学コース 3年　７番

大島浩太

1.　実習の目的

　モバイルをレーシングゲームの機体のハンドルとして、モバイルを左右に動かしたり画面をタップしたりすることにより、機体を動かせ、ユーザーは機体を実際に動かしているかのような体験ができること。

2. 既存プログラムの概要

　プログラム名称：Mobile Driver

　動作説明

1. サーバー側

　サーバーはnode.jsを使用して、WebSocketやAJAX long pollingなどのブラウザ上でリアルタイムに通信できる機能を持つsocket.ioをモジュールとして組み込んだ。

　サーバーに認証機能をもたせ、PCに対してはPINを配布し、モバイルに入力されたPINが、配布したPCのPINと一致していれば認証成功とした。

　socket.ioのroom機能を使用して、モバイルから送られてきたセンサーの情報を、一致しているPINのPCに転送するようにした。

1. クライアント（PC）側

　WebGLの描画ライブラリであるThree.jsを用いてレーシングゲームの３D描画を行った。これはHexGLを用いた。

1. クライアント（モバイル）側

　サーバーによる認証後、傾きセンサーや画面のタップ状態をサーバーに送り続け、この情報をPC側に転送することにより、機体の操縦を行った。

PC

・WebGL

・センサーの値を受信

サーバー

・認証

・センサーの値を転送

モバイル

・センサーの値を取得して送信

add\_room\_pc

added\_room\_pc

add\_room\_mobile

added\_room\_mobile

added\_room\_mobile

report\_motion

report\_motion

Figure 1 サーバー、PC、モバイルの大まかな通信手順（青字は転送）

　使用したソフトウェアまたはライブラリは、別紙「Mobile Driver Quotes」に記した。

3. 開発項目の目標＜工夫点＞

　この形式に近いものは任天堂株式会社製造のWiiおよびWiiリモコンが挙げられる。しかし、WiiおよびWiiリモコンには次の欠点がある。

* WiiリモコンはWiiの近くで操作しなければならない
* WiiのソフトウェアはWiiでしか動かない

　今回は、これらの欠点を以下のように解消するように工夫した。

* サーバーに接続できればよいので、どこからでも操作できる
* WebSocketとWebGL（HTML5 Canvas）を対応したブラウザであれば環境は問わない

　PC側でPINを表示させ、モバイル側でPINを入力させることにより、PCとモバイルを関連付けることができた。また、接続時に通信時間を表示させ、どのくらいの通信ラグが発生するかを表示するようにした。



Figure 2 PCのPIN画面



Figure 3 モバイルのPIN入力画面

　モバイルからサーバーにモバイルの情報を送信するタイミングをondeviceorientation、touchstart、touchendイベント発生時にしていたが、これらのイベントが捕獲できずに常にタップしたままの状態があったため、setTimeout関数を用いたイベントを用いて、一定時間ごとにタップの状態などを判定してサーバーに送信することにした結果、ほぼ正常に動作することができた。（Figure 4 イベントの取り扱い方法参照）

モバイル

PC

touchstart/touchend event

setTimeout event

setTimeout event

setTimeout event

Figure 4 イベントの取り扱い方法

タップする画面を、HTMLのMeta要素変更及びiPhoneのホーム画面に追加をすることにより、従来のブラウザのサイズより広く使えるようにし、操作性を向上させた。

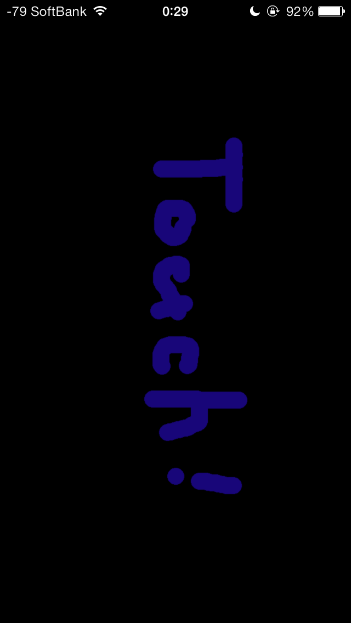
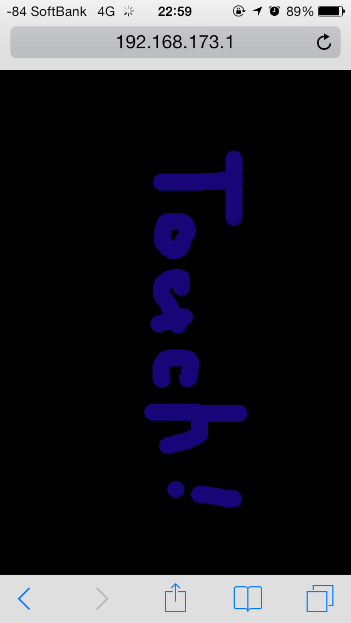


Figure 5 拡張されたiPhone Safari Meta要素修正の相違（左が修正前、右が修正後）

4. 開発結果

　規定通り実装できた。

ゲーム中のPCの参考画面については、Figure 6 に示した。ゲーム前のPCの画面は別紙「Mobile Driver」に示した。

　ゲーム中のモバイルの画面はFigure 5 に示した。ゲーム前のモバイルの画面はFigure 4 に示した。

5. 開発に対する考察

　多くのJavaScriptファイルを読み込まなければならなくなったため、ブラウザの読み込み時間が長くなった。JavaScriptファイルを一つのファイルにし、最適化を行うことでJavaScriptファイルの読み込みにかかる時間を短縮できるが、そうした場合、プログラムが正常に動かなくなってしまったため、今回は行わなかった。次回はこのエラーの原因を探り、解決したいと思う。

　CSSファイルの最適化は行った。



Figure 6 参考：レーシングゲームのPCの画面

参考：node.jsサーバーの起動方法：

1. node.jsを<http://nodejs.org/> よりダウンロードし、インストールする
2. 端末で以下のコマンドを入力する。

npm install

node app.js

既定で80番ポートをリッスンするようにした。

JavaScriptのソースコードを参考として別紙に示した。

その他のソースコードは<https://github.com/zirrusc/MobileDriver>に示した。