2022 年度

メディア処理実験

WebGL 最終作品レポート

コース　　：　システムアーキテクト　コース
学籍番号　：　26002005467
氏　名　　：　藤田　フェリペ

提出日：　5 月　26 日

1. 作成した作品について

1-1 作品のタイトルとスクリーンショット

「　ブロック衝突ゲーム　」

作品のスクリーンショット（特徴的な物・何枚でも可）

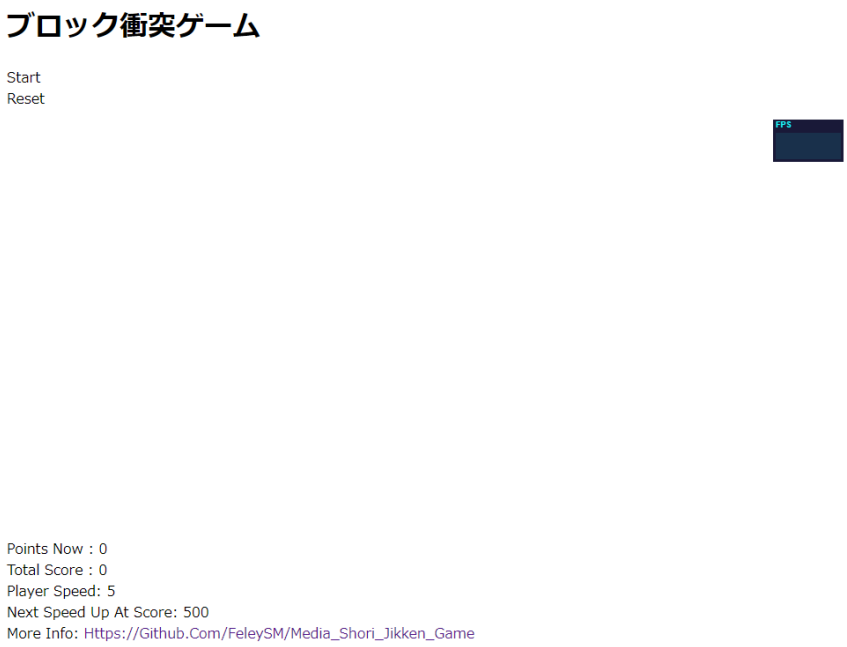

図 1 ：index.html を開くときに表示される初期画面

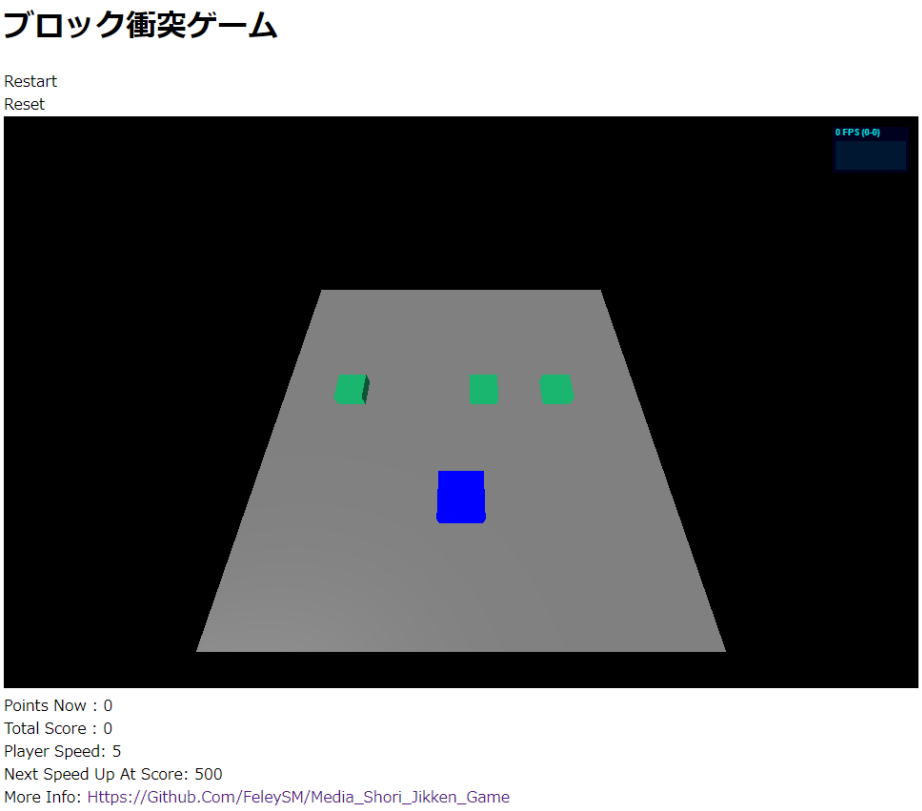

図 2 ：Start ボタンを押されたとき、ゲームが動作する

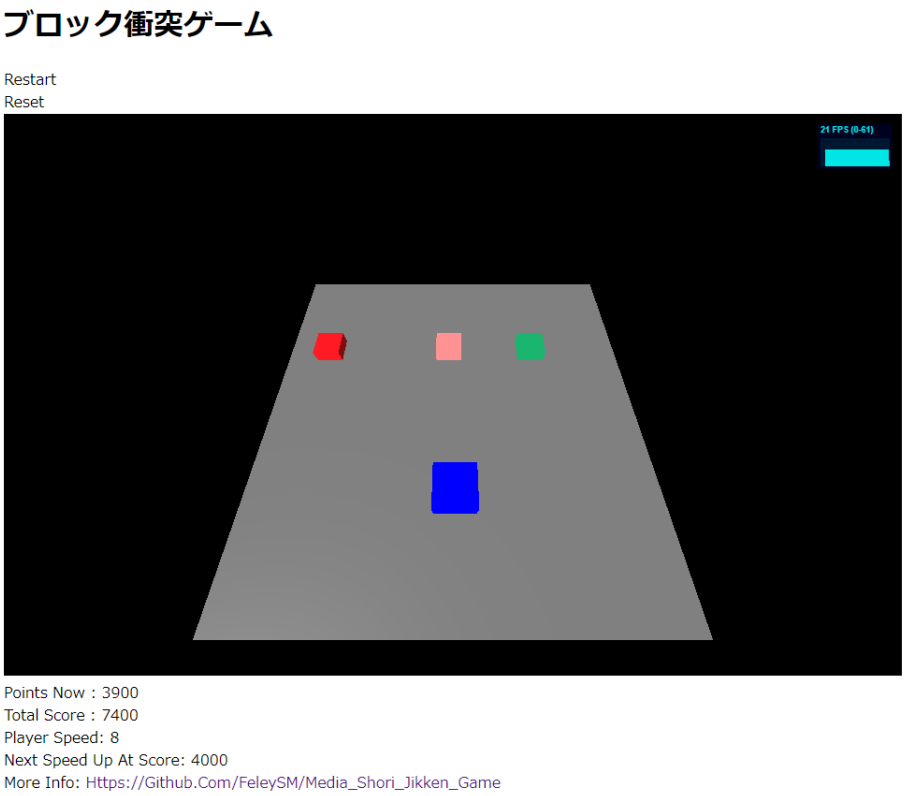

図 3：ゲームがある程度進み、ゲームの下には、今持っているポイント、合計ポイント、プレイヤーの速度、次のスピードアップに必要なポイントを表している。この時点では敵の速度が初期状態より約 2 倍上がっています

1-2 どのような作品なのか，説明しなさい

ゲーム要素について

スマホゲームによくある無限ランナー(Endless Runner)ゲームを論理的に再現しました。このゲームでは、プレイヤーは青ブロックで、キーボードの A と D キーで x 軸方向で動けます。

敵(様々な色ブロック)は画面上（床上）から現れてきて、z 軸の負の方向に一直線に降りてきます。プレイヤーが敵と衝突すると、ポイントを獲得します。敵の色に応じて、獲得するポイントが違います。赤は-100 ポイント、ピンク：+100 ポイント、緑：+200 ポイントになります。

敵が床の面積を超えると、最初に現れた z 方向の位置に戻ります。出現するとき、x 軸の位置は、決められた範囲内でランダムに決まります。色もランダムです。

ポイントを獲得するにつれて敵の動く速度が上がります。また、今持っているポイントを消費して、プレイヤーの速度を上げることができます(キーボードのスペースキー)。

また、画面の右上に、フレームレート等のステータスを表示しています。mrdoob/stats.js のプログラムを使っています。さらに、シンプルなスタイルシート(CSS)を加えました。

メニューについて

ゲームの Canvas の上に、二つのボタンを作りました。index.html プログラムを開くと、最初は、Canvas が描写されておらず、Canvas の上に Start ボタンと Reset ボタンがあります。

Start ボタンを押すとゲームが始まり（Canvas 表示)、Start ボタンが Pause ボタンに変わります。Pause ボタンを押すと、ゲームはボタンが押された瞬間に止まり、Pause ボタンが Restart ボタンに変わります。Restart ボタンを押すと、ゲームが再開され、Restart ボタンが Pause ボタンに変わります。

Reset ボタンを押すとゲームが一回止まり、すべてのゲームデータ（ポイント等）が消えます。そして、上のボタンが Start ボタンに戻り、Start ボタンを押すとゲームが 1 からスタートします。

ゲームの Canvas の下に、現在のポイント、合計ポイント、プレイヤーの速度、次の速度パワーアップに必要なポイント、を順に表示しています。

*ゲームは終了条件がありません。永遠に繰り返されます。

1-3 インタラクションについて

Canvas（ゲーム）内のインタラクション

a または A キー	x 軸負の方向に動く
d または D キー	x 軸正の方向に動く
スペースキー	ポイントを消費してプレイヤーの動く速度を+1

ボタン（文章）によるインタラクション

Start ボタン	ゲームを開始（exec 関数呼び出し+変数制御）
Reset ボタン	レンダリングを停止し、ゲームにかかわる変数を初期化し、ゲームをリセットする
Pause ボタン	レンダリングを停止する
Restart ボタン	Pause 中において、レンダリングを再開する

1-4 苦労した点はどこか

最初は、敵クラスを作って、インスタンス化して出現させようと思っていました。いろいろ工夫してみましたが、なかなかうまくできなくて、敵の数や出現等を関数処理で行うことにしました。

また、ゲームっぽさを出すために、メニューを作ってみましたが、思っていたより難しかったです。授業では、レンダリングを update 関数内で行っていたが、ボタンを押してゲームをフリーズするには、レンダリングを別の関数で行い、update 内で呼び起こす必要がありました。そうしないと、Pause ボタンを押したら、すべてが最初の状態に戻りリセットされます。分割することで、Pause ボタンでレンダリングだけを、Reset ボタンですべてをストップできます。ここにたどり着くにはすこし時間がかかりました。

さらに、カメラの位置、プレイヤーと敵の速度、ポイント等の要素を調整しつつコードを書きましたので、後でグラフィックスや最適化を行うときは、より簡単でいろいろ実装できると思います（いずれきちんとしたゲームにしたいと思います)。

2. CG について感想など

（この演習では CG 基礎とそれに関わる演習を混ぜて行いました．興味を持った点やもっと知りたかった点などあれば自由に記載して下さい）

この授業では、初めて JavaScript で CGI を扱いました。Canvas や Three.JS のすばらしさに感動しました。こんな簡単にブラウザ上で図を書いたり、アニメーションを表示したりできるとは知りませんでした。Unity でゲームを作ったことがあって、WebGL にエクスポートする機能がありますが、実際に WebGL を使ったことなく、ドキュメンテーションを少し読んだだけで頭が痛くなりました。Three.JS を作った人には感謝しかないです。私は JavaScript が大好きで、今後も Three.JS を勉強しようと思っています。

先生は基本だけを教える目的でこの授業を作ったと思いますが、個人的には、JSX と最近の JS の書き方にプログラムを書いたほうがいいと思います。現代の JS では var は非推奨ですし、Node 等で環境を設定することが一般化されていると思います。音や glb ファイルを扱いたかったが、CORS のポリシー上ローカルサーバーを立ち上げる必要があるので、簡単なものしか作れないのが少し残念なところだと思います。しかし、授業全体がすごく面白く、パワーポイントの説明がわかりやすく、やる気が出ると思いました。

参考文献

特になし