

令和4年度

卒業研究発表会 予稿集

情報技術科

日時 令和5年2月17日(金)9:30から
会場 多目的ホール



岩手県立産業技術短期大学校

表 2

目 次

令和4年度卒業研究発表会
開催日：令和5年2月17日(金)
開催場所：多目的ホール

時 刻	No. 研究テーマ	学生氏名	担当教員
9:30	開 会 挨 拶		
9:35 ~ 9:53	1・Mediapipeを用いたスポーツデータ分析	高橋 優也	石館 勝好
9:54 ~ 10:20	2・電子出席簿の作成	浅沼 麟 佐藤 琉希	石館 勝好
	(休憩)		
10:30 ~ 10:56	3・心拍センサーを用いたアート作品の改良と機能拡張	古屋 勇翔 昆野 捷汎	菅野 研一
10:57 ~ 11:15	4・GPUを用いたDeepLearningの処理性能比較、検証	石川 巧弥	菅野 研一
	(休憩)		
11:25 ~ 11:43	5・M5StickCによる自動水やり機能付きプランターの作成	高橋 佳輝	佐々木 建
	(休憩)		
13:00 ~ 13:27	6・A-Frameを用いた学校見学サイトの作成	佐々木 蓮斗 佐々木 悠真 菅原 匠真	安倍 春菜
13:28 ~ 13:55	7・画像認識によるアクションバトルゲームの開発	天沼 優作 宇部 駿 加藤 晴太	安倍 春菜
	(休憩)		
14:05 ~ 14:23	8・深層学習を用いたゴミ分別支援ツールの作成	遠藤 綾李	ソソラバラム バドウジヤルガル
14:24 ~ 14:42	9・Webカメラによる視線トラッキングに関する研究	小松 篤志	ソソラバラム バドウジヤルガル
	(休憩)		
14:52 ~ 15:19	10・Unityを使ったゲームの作成	菊池 愛斗 菅原 遥都 照井 太一 元良 優作	小野 陽子
15:20 ~ 15:38	11・Vue.jsを用いたWebアプリケーションの作成	相馬 加奈	小野 陽子
15:50	講評 閉会		

01 MediaPipe を用いたスポーツデータ分析

17. 高橋 優也

指導教員 石館 勝好

1 はじめに

スポーツをする際に、プレーを向上させるためには客観的視点からプレーを振り返ることが必要不可欠である。そこでプレイヤーの動きを数値化することで客観的にとらえやすくなると考えた。また、データを集める際の機器は高価なものが多く、一般の人は手が出せない状況となっており、一番使うべき中高生のスポーツ指導の場面で使うことのできない状況になっている。

そこで、データ収集の際のコストを抑えることで中高生の指導に生かせるようにし、MediaPipeによる身体の動きの数値化、可視化を今回の研究の目標とすることにした。

2 研究テーマの概要

2. 1 研究の進め方

研究は以下の項目に分けて進める

- ① バレーボールのサーブの動きを動画撮影
- ② MediaPipe を用いて各関節の座標と landmark を取得し、Excel ファイルに書き出す
- ③ データのグラフ化

2. 2 開発環境と使用機材

開発環境と機材は以下のものを使用する。

表 1. 環境と機材

区分	名称	用途
OS	Windows10	-
言語	Python3. 9	-
ライブラリ	OpenCV	画像、動画処理
	NumPy	数値計算処理
	MediaPipe	骨格推定
	Pandas	データ解析
	Matplotlib	グラフ表示
カメラ	ZV-E10L(B)	動画撮影

3 分析の手段

3. 1 MediaPipe による動画の座標取得

バレーボールのサーブの動きをあらかじめ撮影する。動画を読み込むことで身体の関節上に、MediaPipe に設定されている landmark の位置を検出し、画面上の landmark から x, y 座標を取得する。

今回は、肩、ひじ、手首、腰、ひざ、足首の座標を取得した(図 1, 図 2)。

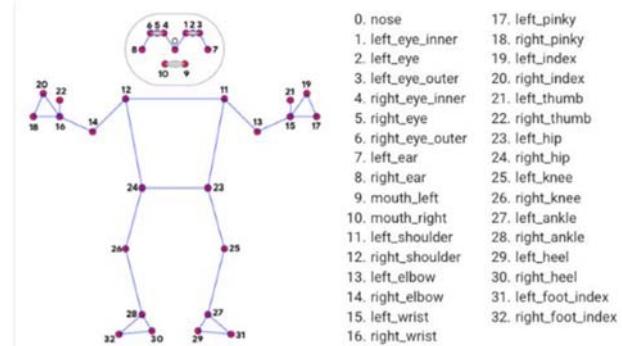


図 1 MediaPipe の landmark



図 2 landmark の検出

3. 2 課題

取得した座標を用いて分析を行おうとしたところ、次のような課題があることが分かった。

① 身長の正確な推定が困難であること

カメラからの人の距離によって画面上に映す人のサイズが変わってしまい、正確な身長や移動距離の推定を行うことができないという課題がある。

② 数値のベクトル

歩く、走るが伴う動作の場合に x 座標は一定の

方向に進むため、数値は増加または減少してしまう。また、右利き、左利きの撮影方向を変えると、x座標が逆転してしまうため修正を行う必要がある。

③ フレームごとの変化は小さくわかりにくい

はじめはフレームごとの変化を棒グラフとして表示していたが、1つ1つの変化は小さく、また1つの動画につき150~200フレームほどになるため、状況の変化が分かりにくくなってしまっていた。

今回は③の課題を解決するために、肘の位置を基準とした手首の移動量のおおよその傾向を分析することにした。

3. 3 座標の計算

今回は同じフレーム数の時の肘の位置を基準とした手首の相対的な位置を取得する（図3）。

これをフレームごとに行い、手首の移動量を算出する。

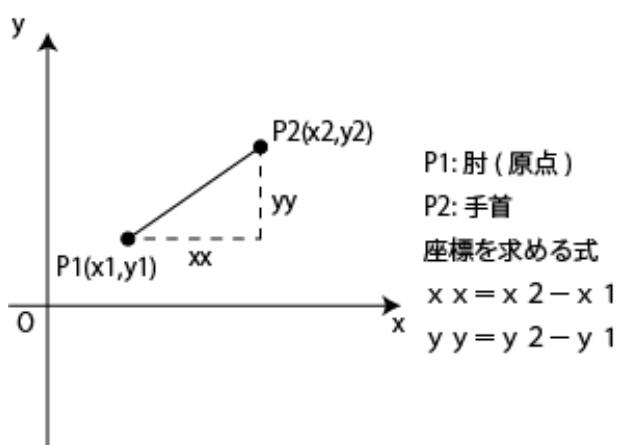


図3 2点間の距離の算出

3. 4 グラフ作成、比較

Matplotlibを用いて取得した値から散布図を作成する。今回は成功時と失敗時の二つで比較する（図4,5）。

サーブ成功時は手の動きにまとまりがあり、細かい分布が少ない。サーブ失敗時はデータが上下に分布していることが分かる。このことから、失敗した場合はサーブトスに合わせようと必要以上に手がぶれたことが想定される。

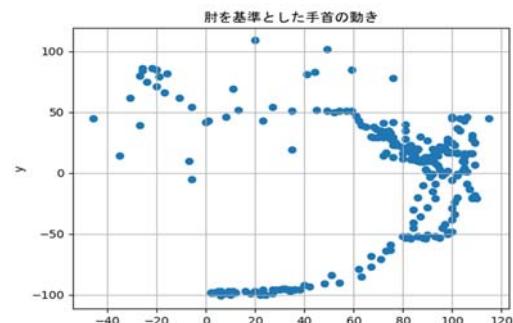


図4 サーブ成功時の手の動き

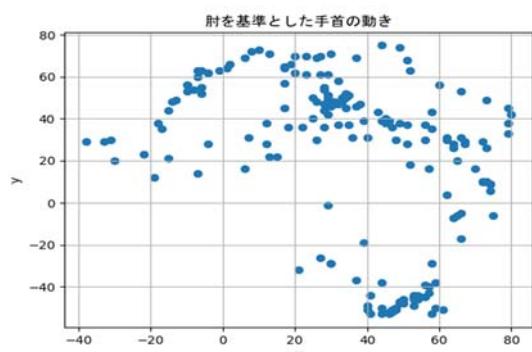


図5 サーブ失敗時の手の動き

4. 終わりに

本研究を通してOpenCVやMediapipeなどの使い方を理解し、landmarkの取得および座標の取得、可視化などに利用することができた。

Mediapipeを用いて、身体の動きに合わせてlandmarkを取得するモジュールについての理解と、開発環境が必要になった。

スポーツと情報分野を組み合わせた研究は情報科では行われていなかったため、先駆けての研究だった。そのため想定外の課題も多く見つかり、プログラムの機能が少なくなってしまった。しかし、骨格推定や身体の動きの可視化などの成果もあったので、今後の研究の基盤となれたなら良いと思う。

5. 参考文献

- MediaPipe・OpenCV・Pythonで体験するAIパーソナルトレーナーアプリ開発 Kindle版
<https://amzn.asia/d/92pernL>
- Home | mediapipe - Google
<https://google.github.io/mediapipe/>

02 電子出席簿の作成

1番 浅沼 麟, 13番 佐藤 琉希

指導教員 石館 勝好

1. はじめに

本校の出欠席の確認作業は教員が紙の様式を用いて手作業で行っている。しかし、紙の様式による確認では手間がかかり、集計時にもミスを起こしやすくなってしまう。そこで教員側の出欠確認の流れをデジタル化することで教員の無駄な時間と負担を軽減することが可能と考えた。そこで我々は既存の欠席連絡システムと時間割システムを連携した電子出席簿を作成することにした。

2. 研究概要

2.1 開発環境について

サーバについて開発では VirtualBox のサーバを使用する。実際に稼働させる際には情報技術科の校内サーバを使用する。

開発環境について表 1 に示す。

表 1 開発環境

OS	Linux
使用言語	HTML, CSS, SQL JavaScript, PHP
データベース	MySQL
ソフトウェア	Visual Studio Code

2.2 システムの設計について

全体の設計として既存の欠席連絡システム、時間割システムのほかに新規のシステムとして電子出席簿を作成する(図 1)。電子出席簿では欠席連絡システムの内部データベースから学生情報を欠席情報を、時間割システムのデータベースから時間割、教員、シラバスを参照する。そして、各データベースから取得した出欠席情報を、まとめて管理するようデータベースに保存する。

電子出席簿で得たデータをもとに出欠情報をエクスポートできるようにする。

また、基本的な使用端末としてスマートフォンやタブレットを想定している。

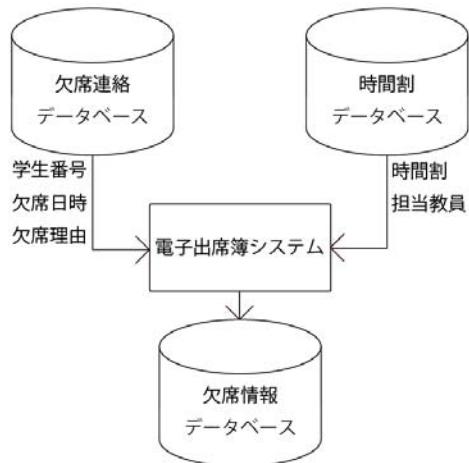


図 1 設計のイメージ図

3. 電子出席簿の仕様

出席簿機能のページでは、1つのページで2つの操作を行える。出欠情報がデータベースに存在しないコマでは出欠情報の登録が行われる。また、出欠情報が存在するコマでは、登録された出欠情報を元に情報の編集が行える。

3.1 セキュリティについて

電子出席簿へのログイン方法として、教員共通のユーザ ID とパスワードでログインする仕様にした。

3.2 TOP ページについて

ログイン後の出席簿 TOP ページを図 2 に示す。このページでは、時間割システムから時間割のデータを取得し、ログイン時の日付の時間割を表示する。



図2 出席簿TOP

3.3 出欠確認について

出欠確認のページを図3に示す。このページでは、欠席者がいる場合には欠席連絡システムから該当学生の欠席情報を取り出し表示する。

AllCheck のチェックボックスで出席学生全員を一括で出席状態にすることができる。

完了ボタンを押下した時点で、チェックされておらず、欠席理由の欄が空白の学生がいた場合にはその学生を「無断欠席」として扱う。その生徒に対して出欠席確認メールを送信し、欠席連絡を促す。



図3 出欠確認ページ

3.4 出欠情報のエクスポートについて

ファイル出力ページを図4に、出力したファイルの内容を図5に示す。エクスポートの方法としては CSV ファイルに欠席情報をエクスポートする仕様にした。また、エクスポート後に PC などのマシンにダウンロードできるようにした。

CSVファイル出力

何年度の出欠席情報をCSVファイル出力しますか？

2021 ファイル出力 TOPに戻る

図4 ファイル出力ページ

A1	B	C	D	E	F	G	H
学籍番号	科目	出欠席日時	出欠席時間	出欠席情報	欠席開始時刻	欠席終了時刻	欠席理由
y21501	5016	2023/1/17	1	0	9:30:00	10:20:00	通院・予防接種等
y21502	5016	2023/1/17	1	0	9:30:00	10:20:00	通院・予防接種等
y21503	5016	2023/1/17	1	0	8:50:00	10:20:00	その他
y21504	5016	2023/1/17	1	0	8:50:00	10:20:00	その他
y21505	5016	2023/1/17	1	0	8:50:00	10:20:00	電車遅延・渋滞等
y21506	5016	2023/1/17	1	1	8:50:00	9:10:00	電車遅延・渋滞等
y21507	5016	2023/1/17	1	2	0:00:00	0:00:00	
y21508	5016	2023/1/17	1	2	0:00:00	0:00:00	
y21509	5016	2023/1/17	1	2	0:00:00	0:00:00	
y21510	5016	2023/1/17	1	2	0:00:00	0:00:00	
y21511	5016	2023/1/17	1	2	0:00:00	0:00:00	
y21512	5016	2023/1/17	1	2	0:00:00	0:00:00	
y21513	5016	2023/1/17	1	2	0:00:00	0:00:00	
y21514	5016	2023/1/17	1	2	0:00:00	0:00:00	

図5 CSV ファイル内容

3.5 無断欠席者への確認メールについて

無断欠席者の確認メールは、既存の欠席連絡システム内部のデータベースに保存されてある各生徒のメールアドレスを参照して、無断欠席者に出席確認メールを送信する。

4. おわりに

電子出席簿のシステム設計及び実装は、そのほとんどが完成した。だが、試用をまだ行っておらず実際に運用することでどのような不具合があるのか検証する必要がある。産技短展開始までには試用を行い、実際に使用した先生方の意見を取り入れてデバック作業していく。

03 心拍センサーを用いたアート作品の改良と機能拡張

9番 古屋勇翔 10番 昆野捷汎

指導教員 菅野 研一

1. 研究の背景と目的

以前に産技短の展示で「水と向き合う」^[3]というアート作品を見たことがあり、私も展示了際に実際に触れて体験できるアート作品を作成したいと考えていた。昨年の卒業研究で心拍センサーを用いたアート作品^[5]を作成していたが展示をするには改善が必要だと聞いた。そこで今回、私たちは改善し、展示できるように作成に取り組むことにした。

2. 作品概要

LEDテープ(Neo Pixel)を張り付けたアクリル製の円柱を用意する。心拍センサーを用いて、二人の心拍を取得する。その結果を心拍の類似度に応じて、円柱のLEDを光らせ、音を鳴る仕様になっている。

今年度は、電源の供給のみで安定して動くように改良すること、機能拡張として一致した際に光だけではなく音を出力する。

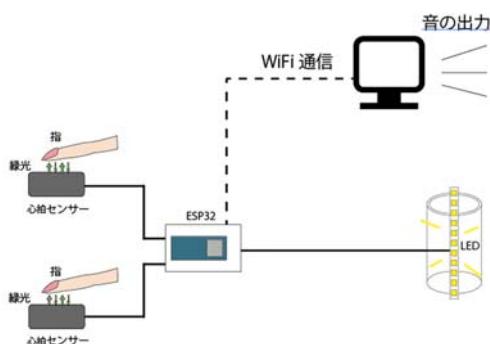


図1 システム構成要素図

3. 開発環境

開発環境は以下の通りである。

3.1.1 ハードウェア

ESPr® Developer 32

無線通信を使用するので、Espressif Systemsの無線通信モジュール(Wi-Fi +Bluetooth)

ESP-WROOM-32 を搭載している ESPr® Developer32 を使用した。

3.2 ソフトウェア

3.2.1 Arduino IDE

ボード上で動作するソフトウェアを開発するためには作られた統合開発環境。ソースコードの編集や、コンパイル、リンク、作成されたオブジェクトのボードへの書き込み機能などが提供される。

今回の研究では、学校の WiFi を用いて ESP32 と接続し、UDP プロトコルを用いて Pure Data との接続を行った。

用いたライブラリは、OSC, WiFi である。

3.2.2 Pure Data

音声、ビデオ、映像処理のためのリアルタイムなグラフィカルプログラミング環境。ネットワークを絡めたプロジェクトや、Arduino と組み合わせた自作楽器の製作も可能である。

4. 使用部品

4.1 心拍センサー

指先などの末梢血管のある部分を光センサ一部に置いて脈波をアナログ出力するセンサーデバイスである。



図2 心拍センサー

4.2 Neo Pixel

光の演出は Neo Pixel を使用する。

Neo Pixel はシリアル伝送により個々の LED の色・明るさを制御できる。

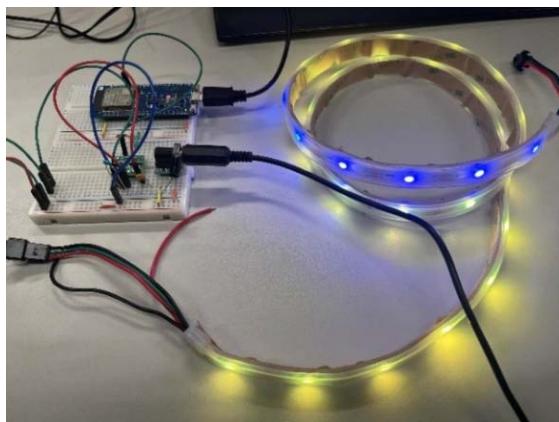


図3 Neo Pixel

5. 改良・修復点

5.1 Neo Pixel

昨年度の研究では安定して光らせることができなかった。原因の一つとして、ESP32 と Neo Pixel を直接つないでしまったことがあると考えた。その理由として、ESP32 は 3.3V で稼働するのに対して、Neo Pixel は 5.0V で稼働するため直接つないでしまうと稼働することもあるが、安定した制御はできない。そこで今回の研究では、3.3V から 5.0V へ変換するロジックレベルコンバーターを使用し、サンプル回路を組み、光らせた。

同じ電源から供給されることによって安定して光らせることができた。

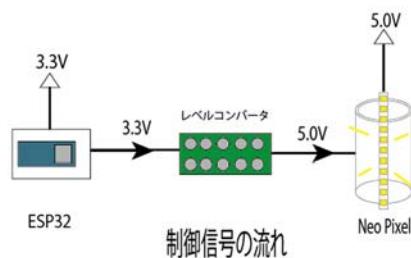


図4 制御の流れについて

5.2 ケーブル類

心拍センサーに電源供給するためのプラグケーブルが一部断線して値を取得しない問題が発生していたため作り直した。

5.3 心拍について

心拍の取得をするためのコードはデータが残っていなかったため報告書[5]を参考に作成した。

5.3.1 移動平均

心拍センサーから取得した値(RAW データ)のままだとばらつきがあるため移動平均を用いて値を平滑化した。

今回の研究では 50 個ごとのデータで平滑化する。



図5 移動平均で平滑化したもの

青:平滑化前 赤:平滑化後

5.3.2 心拍の識別

心拍の波とノイズによる波の識別をするため、平滑化したデータから閾値を設定し、閾値を超えたら心拍の波と判定する。

閾値も 50 個のデータごとに設定をしているため随時更新していく。

課題として、閾値の設定をしてもノイズの波も心拍の波と判定をしてしまうことがあったため、閾値の設定に加えて波の間隔にも閾値を設定した。人の心拍の波の間隔は、0.6~1.2 秒なので閾値も 0.6~1.2 秒に設定した。

5.3.3 類似度判定

二つのセンサーから閾値を超えた値をそれぞれ10個取得すると、類似度判定を行う。類似度の算出は、閾値を超えた10個の移動平均の値に対してユークリッド距離を用いることにした。

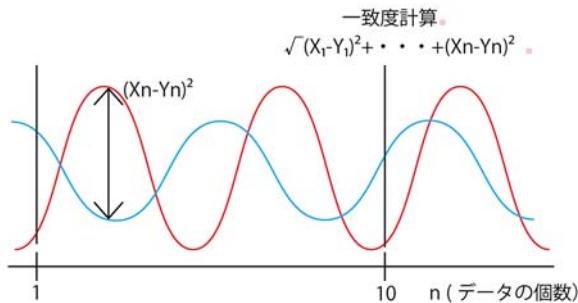


図6 ユークリッド距離

類似度に応じた光り方は一致度が高い順に以下のようないくつかある。

- I. 虹
- II. 赤
- III. 緑
- IV. 青
- V. 紫

6. 機能拡張

6.1 Pure Data

類似度判定によって出た値を、UDPプロトコルを用いたOSC通信を行い、受信する。そして、受信した値によって異なる音が出力されるようにプログラムを作成した。

音は、上記の類似度に応じた光り方の5種類に対応している。一致度が高くなるにつれて音の高さも高くなるように設定した。

下図が、実際のプログラムである。

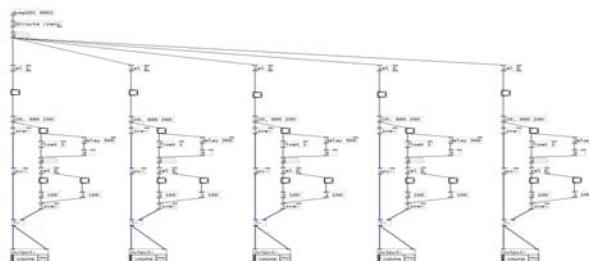


図8 実際のプログラム

7. 通信

7.1 WiFi

ESP32とPC間で通信をするために使用する。今回は、学校のWiFiを使用する。

7.2 UDP

OSC通信を使用するために使用する。

UDPは、インターネットで標準的に使われているプロトコルの一種で、「User Datagram Protocol」の略である。インターネット・プロトコル(IP)を使ったネットワークにおいて、アプリケーション同士が最小限の仕組みでデータを送受信できるように考案された、シンプルなOSI参照モデルにおけるトランスポート層のプロトコルである。

UDPには、送受信されるデータの誤りや順序の違いなどを検出する機能がない。TCPに比べてプロトコルが簡素なため、信頼性には劣るが、通信の処理にかかるコストが少ないというUDPの性質から、応答性の良さを優先したい通信や、その他に音声や動画のストリーミングなどといった、リアルタイム性が重要視される通信で使われている。

7.3 OSC

類似度判定で出た値をPure Data用の変数に代入し、Pure Dataに値を送信するときに使用する。

Open Sound Control(OSC)とは、電子楽器（特にシンセサイザー）やコンピュータなどの機器において音楽演奏データをネットワーク経由でリアルタイムに共有するための通信プロトコルである。

OSCには大きく2つの種類のメッセージがあり、そのメッセージをアプリ間でやりとりしながら通信を行う。

/temperature Message OSCのメッセージ種別を示すもの /temperature/areaA /temperature/areaBなどのように 階層改造にすることが可能	29.0 Arguments OSCのメッセージ内容 数値や文字列などを格納
-----------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------

1つ目は, OSC Message である. 「/」で区切って階層構造を作ることが可能なので, 複雑な通信でも見やすい形で通信フォーマットを作成することが可能. 2つ目は, OSC Arguments である. 数値(int / float)や文字列(string)形式などのデータを受け渡すことができる. 今回は, 過去の卒業研究で使用されていた OSC Message を使用した.

8. 終わりに

今回は昨年の研究の引継ぎをし, 展示をできる状態を目指し改善と機能拡張を行った. 昨年の研究で見つかった接続の問題点を改善することができた.

また, 機能拡張として音の機能を追加し, 体験したひとがより楽しめるアート作品を作成することができた.

「水と向き合う」のようにこれからも展示されて実際に触れ, 見て楽しんでもらいたい.

9. 参考文献

[1] 入力デバイスに水を用いて音を生成するアート作品(産業技術短期大学校矢巾校卒業研究報告書, 2016, 上山明江, 東山真実)

[2] 入力デバイスに水を用いて音を出力・制御する(産業技術短期大学校矢巾校卒業研究報告書, 2017, 廣藤美緒)

[3] 「水と向き合う」

<http://www-cg.cis.iwate-u.ac.jp/AT2018/award.html>

[4] 入力デバイスに水を用いて音と光を出力するシステムの制作(産業技術短期大学校矢巾校卒業研究報告書, 2018, 田鎖鴻容)

[5] 心拍信号を利用したアート作品の制作(産業技術短期大学校矢巾校卒業研究報告書, 2022, 橋本佳

太, 福田陸斗)

[6] ESP32 と Neo Pixel フルカラーLED テープで Wi-Fi 卓上イルミネーションオブジェを作ってみた

<https://www.mgo-tec.com/blog-entry-led-tape-neopixel-esp32-artnet.html/2>

[7] 第30回 OSC 通信で Arduino と他のアプリを連携させてみる. (前編)

<https://deviceplus.jp/raspberrypi/entry0030/>

[8] Pure Data Japan 日本の Pure Data ユーザのためのフォーラム

https://puredatajapan.info/?page_id=2

[9] garretlab いろいろ試したこと書き留めるページです

<https://garretlab.web.fc2.com/>

[10] Pure Data チュートリアル&リファレンス(美山千香氏, 株式会社ワークスコーポレーション, 2013)

[11] Pd Recipe Book — Pure Data ではじめるサウンドプログラミング(松村誠一郎, 株式会社ビー・エヌ・エヌ新社, 2012)

[12] UDP とは

<https://www.nic.ad.jp/ja/basics/terms/udp.html>

04 GPU を用いた Deep Learning の処理性能比較、検証

石川 巧弥

指導教員 菅野研一

1. 研究の背景

今年、学校に Deep Learning 専用 PC が導入された。そのため、この PC を用いた研究の例はまだない。そこで、この PC の環境を整え、性能を検証することが、産技短での AI 開発の足掛かりになると想え、この研究に取り組むことにした。

2. 研究概要

2.1 研究内容

- ・目的は GPU の処理性能を比較し、検証する。
- ・方法は Deep Learning を用いたサンプルを GPU と CPU に処理させて、処理速度を比較する。
- ・サンプルの処理量を変化させたり、別の GPU、CPU を用いたりして処理性能を検証していく。

2.2 開発環境

表 1. 開発環境

OS	Ubuntu 18.04.5 LTS
使用言語	Python3.5
使用ライブラリ	TensorFlow
コンテナエンジン	Docker

2.3 Deep Learning 専用 PC の情報

表 2. PC 情報

GPU	GeForce RTX3090
CPU	Intel Xeon W-2223@3.6GHz
RAM	32GB
SSD	2×1TB

2.4 GPU とは

- ・Graphics Processing Unit の略で画像処理装置のこと。並列処理に優れており高速な画像処理ができるため、大量のデータを処理する機械学習での需要が高まっている。

2.5 コンテナ技術とは

- ・アプリケーションと実行環境を 1 つにまとめあげる技術である。ハードウェアと OS をコンテナエンジンを介して繋げることで、仮想的な環境を作り出している。[1]

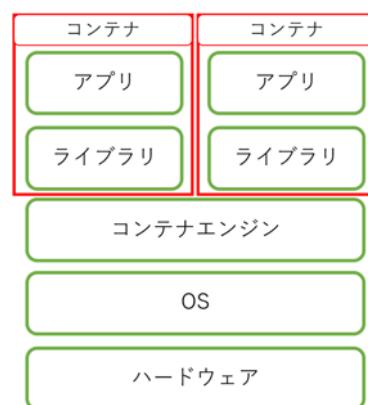


図 1 仮想コンテナ環境

- ・従来の仮想マシンと違い、コンテナ内と OS に依存関係がないため、コンテナを他の環境に移動させても実行できる。また OS を起動する手間もないため、起動・処理も速くなる。

2.6 計測に用いるサンプルデータ

2.6.1 手書き数字の画像分類

- ・MNIST が提供しているデータセット。[2]
- ・学習用データとして 6 万枚、テスト用データとして 1 万枚の画像がある。
- ・それぞれのデータにラベルがついているので、教師あり学習に適している。

2.6.2 キャラクターの物体検知用画像

- ・他の卒業研究[3]で作成した、一つのキャラクターを学習させるデータセット。
- ・学習用データとして、29 枚画像がある。

3. 検証

3.1 環境構築

- TensorFlow のコンテナ環境を作成し、仮想環境内で GPU を利用できるようにした。
- x-window システムを利用し、プログラム実行時に画像やグラフをホストのウィンドウとして表示できるようにした。

3.2 処理時間計測

- 専用 PC の GPU と CPU, Google Colaboratory の GPU を用いて処理速度を計測した。
- 計測方法は、サンプルデータの学習を行う部分の前後に perf_counter 関数を用いて計測する。
- CUDA を使用した GPU で処理速度計測する際は CPU と GPU を同期させるため、perf_counter 関数の前に torch.cuda.synchronize() を利用して計測する。

```

root@7acca523bed:~/workspace/AI_basic_ichinoseki_2021
ファイル(F) フォルダ(E) 表示(V) 検索(S) 終了(T) ヘルプ(H)

torch.cuda.synchronize()
#時間計測スタート
start = time.perf_counter()
#モデルの学習
history = model.fit(
    trainX, trainY,
    batch_size=1000,
    epochs=50,
    validation_data=(validX, validY),
    shuffle=True,
    callbacks = [EarlyStopping(monitor='val_loss', patience=3)],
    verbose=1
)
torch.cuda.synchronize()
#時間計測ストップ
end = time.perf_counter()

156.1      67%

```

図 2. 処理計測位置

4. 結果

4.1 手書き数字の学習時間計測

- それぞれ 5 回ずつ計測した結果を平均し、下の表に表す。比較すると、専用 PC の GPU が CPU に比べ、約 3.4 倍速く処理をしている。

表 3. 計測結果

GPU	3.88 秒
CPU	13.16 秒
colaboGPU	4.32 秒

4.2 キャラクターの学習時間計測

- 10 回学習させる処理を 3 つの処理装置で行い、それぞれの計測結果を下の図と表に表す。

- 専用 PC の GPU が CPU と比べると 7.7 倍以上速く処理を行っている。

表 4. 計測結果

GPU	157.9 秒
CPU	1222 秒
colaboGPU	330.3 秒

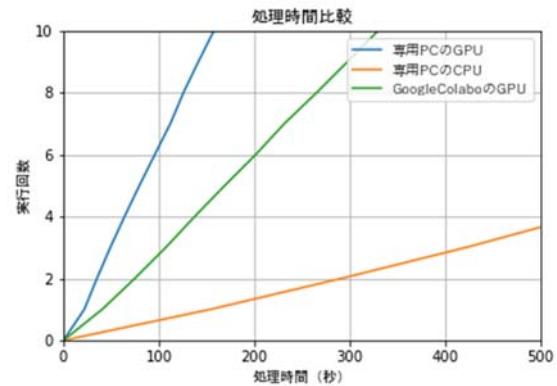


図 3. 各処理装置の処理時間比較

5. 終わりに

- 本研究では GPU の処理性能を検証し、他の処理装置に比べても高速に処理できることを確認できた。複雑で大量のデータを処理する際に、より顕著に性能が現れたので、今後大容量の処理を行う研究や開発に利用されることを期待したい。
- また、環境構築やコンテナの利用方法等はマニュアルを作り、今後の PC の利用に役立てるため残していくたい。

6. 参考文献

- 【話題の IT トレンド】コンテナ技術と仮想マシンの違いとは? | トピックス | PSC Smart Work
<https://psc-smartwork.com/topics/2021/08/162.html>
- [2]MNIST：手書き数字の画像データセット：AI・機械学習のデータセット辞典 - @IT
<https://atmarkit.itmedia.co.jp/ait/articles/2001/22/news012.html>
- [3]遠藤綾李:深層学習を用いたごみ分別支援ツールの作成, 令和4年度卒業研究発表会, 番号 07, 2023

5 M5StickC による自動水やり機能付きプランターの作成

発表者 高橋佳輝

指導教員 佐々木建

1. はじめに

私は、1年次に学んだ Arduino の授業やインターネットの掲載記事をみて IoT について興味を持った。自分でも IoT を用いて何かを自動化できないかと思い模索していたところ、M5Stack シリーズで取り組んでいる事例を目についた。そこで私は、今まで使ったことのないワンボードマイコンである M5Stack シリーズの M5StickCPlus を使い何か形にしたいと考え、卒業研究のテーマとして取り組んだ。

テーマについては、色々と模索をした結果、測定結果をスマートフォンと M5StickCPlus との通信機能の中でやり取りできる研究にしたいと考え、M5Stick 用の水分推定センサをもちいて土壤の水分量のデータを題材に、この研究に取り組みたいと考えた。

2. 開発環境

開発環境は下記の通りである。

開発言語	microPython
使用ハード	M5Stick C plus 水分推定センサ付き給水ポンプ Raspberry Pi 4-b UCAM-C0220FBNBK(webcam)
使用ソフト	UIFlow,
使用サービス	LINENotify,IFTT,LINE, ambient,スプレッドシート

3. 主な使用機材について

・ M5Stick C Plus

→Wi-Fi や Bluetooth が利用
小型で軽量かつ大きいモニターや
ボタンが付いているワンボード
マイコンである。



図.1 実際の Cplus

- ・ M5Stick 用水分推定センサ付き給水ポンプユニット

→水分センサ機能と給水ポンプ機能を備えているユニットである。



図.2 実際のユニット

- ・栽培用のプランター

→貯水機能がついているもの
(2層式)



図.3 プランター

- ・携帯

→LINE の通知確認用。今は自分のスマホにより研究を進める確認をしているが、最終的にはタブレット等に移行していきたい。



図.4 LINE アイコン

4 研究概要

4.1 機能概要

- ・水分推定センサで水分量を計測して、一定値を上回ると水を出し、LINE にて通知を行う。
- ・給水用ポンプにはプランターの土壤にまんべんなく散水するために、散水用のホースに分岐点を設け、複数箇所で散水できるよう工夫をした。
- ・散水していないときは現在の状態をウェブページで確認できるようにした。

4.2 実装機能について

4.2.1 基本機能 (水分測定・散水)

- ①水分推定センサで土壤の水分量を測定し携帯上で確認できる。
- ②測定値が湿っているか乾いているかの判断を条件付けした。(イメージとしては閾値 1900 以上の値で判定。)

③乾いている値（閾値 1900 以上）であれば
散水する。

<乾いているという判定について>

土が完全に乾いているときの測定値 : 2000
全体に程よく水を与えたときの測定値 : 1800
と設定したためその平均値である 1900 を乾いて
いる状態の数値とする判定条件に設定した。



乾いている状態

程よく湿っている状態

図.5 土の状態イメージ

4.2.2 通知機能について

今回は LINENotify と IFTTT を利用して自分のスマホの LINE で水が流れていることを確認す
ることで研究を進めている。

IFTTT とは様々な Web サービスやアプリを連携させて操作の自動化ができるサービスをいう。
LINENotify であらかじめメッセージを追加し,
IFTTT と連携する。

連携時に発行される URL をソースコードに埋め込めば実装可能になる。

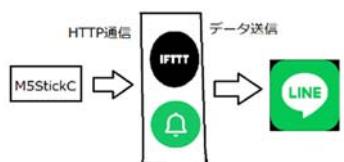


図.6 通信の様子

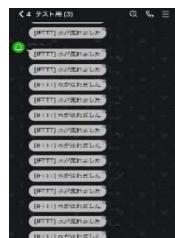


図.7 通知の様子

4.2.3 データ管理ページについて

今回ウェブページにて管理できるようにした。
ウェブページで操作できる内容は以下の通りである。

- ・現在の水分値の確認
- ・遠隔操作
- ・グラフの確認
- ・ストリームの確認



図.8 実際のウェブページ

<水分値の確認>

M5StickCplus から http 通信でスプレッドシートへデータ送信する。

スプレッドシートのデータは FetchAPI を利
用して js でデータを取得、HTML に反映する。

<遠隔操作の動作>

スマホ等で遠隔操作ができるようにした。

Uiflow で実現可能な機能を装着した。

数値の表示、ボタンに操作を割当てた。

元のウェブページが大きすぎるため HTML の
iframe で表示サイズを調整した。

<グラフの確認>

今回は ambient というサービスを利用して水
分量のデータの可視化を図った。ambient とは
細かな初期設定をしなくても送ったデータをア
ルタイムでグラフ化するサービスの一つである。
下画像は実際の動作の様子を示す。

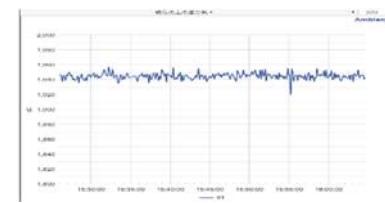


図.9 グラフ作成例

<ストリームの確認>

Raspberry Pi を使用して監視カメラを実装さ
せた。Motion ライブラリを利用して Raspberry
Pi に接続されているウェブカメラの映像をブラ
ウザ上で確認できるようにした。

5. 終わりに

当初の予定であった通信機能を用いた作品がで
きた。データを集約する際に悩む部分もあったが
別にウェブページをつくるなどして解決するこ
とができた。しかし、内容に無駄があつたり通信遅
延が多かったりするのが課題として浮き彫りにさ
れた。更なる調整をしていきたい。

今回の研究でハードウェアやソフトウェアなど
様々な分野に触れることができたので、これから
の仕事などにこのスキルを生かしていきたい。

参考サイト

「トマト栽培をスマート化しよう」

<https://tkrel.com/iot-planter>

参考書籍

「M5Stack & M5Stick C ではじめる IoT 入門」

株式会社アイエンター 高馬 宏典 著

出版社：シーアンドアール研究所

06 A-Frame を用いた学校見学サイトの作成

発表者氏名 佐々木蓮斗, 佐々木悠真, 菅原匠眞

指導教員 安倍春菜

1. はじめに

本校のホームページ上で学校紹介用の 3D 写真が公開されている。しかし、映像を使われている写真的雰囲気が暗いことと、学校の全体図がわかりにくいという問題点があった。また、本校に入学する学生から授業の内容を知りたいという意見が多くあった。

そこで 3D 写真を新しく撮影することで全体の雰囲気を良くし、さらに、各科の授業内容をゲーム形式で紹介することで本校に興味を持ってくれる高校生が増えると考え、本研究テーマを選定した。

2. 研究概要

2.1 使用技術

サイトの作成には A-Frame を用いる。A-Frame とは VR 体験を表現するためのライブラリのことと言う。HTML の簡単なタグを記入するだけで始めることができる。また、JavaScript を用いることで、HTML で記述したオブジェクトの処理や制御が可能である。

表 1 使用技術

開発環境	VScode,
使用言語・使用ライブラリ	HTML,CSS,SCSS,JavaScript,A-Frame
使用ソフト	Blender
対応ブラウザ	Chrome,Safari
使用機材	RICOH THETA

2.2 対応ブラウザ

日本で圧倒的なシェア率を持っている Chrome と PC 利用者のために Mac 標準の Safari を対応ブラウザに選んだ。スマホの対応も実装。ブレイクポイントは JavaScript を用いて閲覧している端末を判定し、その端末にあったサイズに調整している。

2.3 サイトの高速化

スマートフォンで見たときにサイトの表示がかなり遅かったため、Web サイトの表示の高速化を実施した。まず JavaScript と CSS ファイルの圧縮を行った。次に、JavaScript と CSS を別ファイルではなく、インラインで記述できるところはインラインで書き、読み込みの高速化を行った。さらに、JavaScript・CSS の簡略化できる部分は、簡略化しコードの量を減らすことで、読み込み速度を 15 秒だったものが、6 秒程度に縮めることができた。(端末や通信環境により、多少の誤差はあります)

2.4 Web サイトの構成

2.4.1 ページ一覧

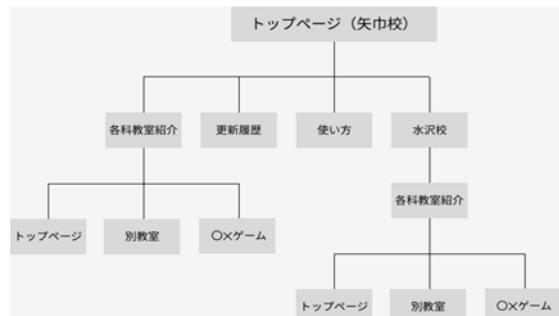


図 1 ページ一覧

2.4.2 3D 写真

矢巾キャンパスと水沢キャンパスの各科や各施設の写真を新しく取り、アップグレードする。

2.5 サイトのデザインについて

サイト内で使用している色に関しては当校で指定されているデザインガイドラインに沿ったものを使用している。

ファーストビューに、モーダルウィンドウでタイトルと簡単な操作方法を表示することで誰でも使いやすいサイトになるよう心掛けた。また、アイソメトリックのデザインは背景を空にし、空中に校舎が浮いているようなデザインに仕上げた。後者の装飾については、校舎だけでは寂しいため、木や車などを設置し、3DCGを動かしたときに楽しめるように工夫した。



図2 ファーストビュー

各科・各教室の紹介ページの部分は、いつでも別の科の教室に移動できるようになっていたが、校舎のアイソメトリックデザインを表示する機会が一番最初しかないのはもったいないという意見をいただき、今いる科の教室にのみ移動できるように変更し、利用者に学校全体の構造をより理解してもらえるようにした。左下に表示されている教室名の隣に自由に表示、非表示にできる教室の説明文を実装した。



図3 説明文

ヘッダーにある使い方と更新履歴にもモーダルウィンドウを使用した。使い方のほうは説明したい箇所が多いので2ページにしたり、文字の大きさや実際に操作している動画を入れることで長い説明でもわかりやすくした。

2.6 サイトのコーディングについて

2.6.1 トップページ

サイトのトップページは、A-Frameを用いて、Blenderで作成した矢巾校・水沢校の校舎のオブ

ジェクトを表示している。校舎の上に配置されている数字は位置を固定してしまうと、校舎を回転して時に見えなくなってしまうため、回転するアニメーションをつけて見えるよう解決した。

また、スマートフォンのジャイロ機能制限について、iPhoneのSafariでサイトを見た場合はジャイロ機能がオフになるのだが、AndroidのChromeで見たときに、ジャイロ機能がオンになりサイトが見づらかったため、ジャイロ機能をオフにするコードを加えた。

2.6.2 各科・各教室の紹介ページ

各科・各教室の紹介ページについては、360度カメラで撮影した写真をA-Frameを用いてVR画面として表示。教室を360度見渡せるようになっている。また、画面右側に、その科の各教室へ移動できるリストメニューを表示する。教室の名前の横に、その教室についての説明文を表示非表示できるように設置。



図4 VR画面

2.6.3 校舎の3DCG

校舎の3DCGの作成について説明する。最初に取り組んだ矢巾校の作成では初めて経験するトラブルが多く苦戦したが、それらの経験が水沢校の作成に活かされた。矢巾校を作成した際に経験したトラブルというのは、

- ① ファイルの容量が大きすぎて、サイト自体の読み込みに時間が掛かってしまう(完成初期は約32MB、読み込みに約7秒)。
- ② Blenderからサイトへの反映を行った際、設定していた色が想定以上に濃くなってしまう。
- ③ Blenderからサイトへの反映を行った際、設定していた色が反映されない場合がある。

といったものだった。①は、Blender 内のマテリアルの個数を減らす、使用する画像の枚数を減らすことで解決した(現在約 4MB、読み込みに約 3 秒)。②は設定する色を初めから薄くすることで解決した。③は色の設定に RGB を用いていたことが原因だったので、色塗り専用の画像を用意することで解決した。また、モデルの中の不要な頂点を減らすことでも 1MB 程の容量削減に成功した。装飾についてだが、矢巾キャンパスの駐車場や両キャンパスの校庭は平面が広がっているため、見栄えが良くない。そこで、空いたスペースに装飾を施すことで、見栄えが悪くなることを回避した。

矢巾校



図 5-1 3D モデル

水沢校

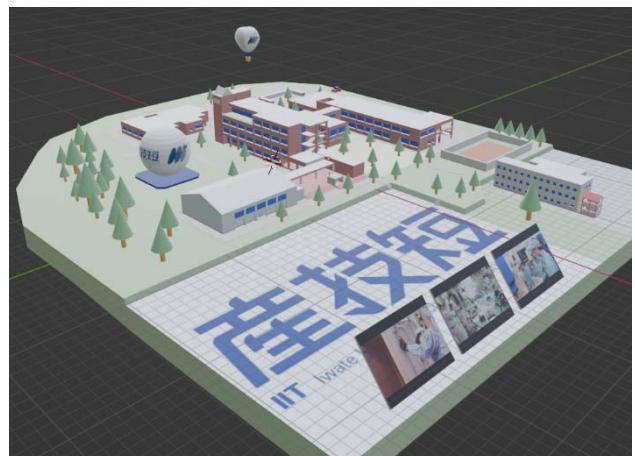


図 5-2 3D モデル

3. 作品の概要

3.1 トップ画面

トップ画面には、水沢キャンパスと矢巾キャンパスの学校の敷地全体を等角投影法(アイソメトリック)で表現し、視覚的にわかりやすく表示する。

3.2 アイソメトリックによる外観

校舎の 3D モデルか、各教室の 3D モデルの上に配置されている、数字の書かれたピンをクリックすることで、指定した場所(各科の教室)の 3D 写真に移動する。各ピンの説明は画面右上のピン一覧メニューにカーソルを合わせることで、見ることができる。



図 6 トップ画面

3.3 校舎見学とゲーム

移動先では、3D 写真で教室内を見てもらう。その選んだ教室についてもっと知りたい場合は、もっと詳しく知りたいボタンをクリックする。その教室にちなんだゲームをプレイし、その場所や科についての理解を深めてもらう。

3.4 ゲーム

各教室の○×ゲームページについては,教室紹介ページに配置されている「もっと詳しく知りたい」ボタンを押すと,○×ゲームページに遷移する.画面のスタートボタンを押すと,○×ゲームがスタートする.問題は1教室5問で,問題を解きながら,各教室で行っている授業内容や,その授業で使用している道具などについて学ぶことができる.

○×ゲームの問題については,各科について詳しく知らなければ作成できないため,各科の職員に作成をしてもらう必要がある.なので,今回の期間内には作成することができなかった.

学：<https://www.iwate-it.ac.jp/vr2020/>



図7 ○×ゲーム

4. おわりに

3Dで分かりやすく学校を紹介することで,受験生が良質な情報を得ることができ,アイソメトリックなデザイン性を持ったサイトにすることで使用感や見栄えが良いサイトになった.

前回までの課題点だった,土台などの装飾やジャイロ機能のいらない部分の削除など,細かいところに力を入れることができ,サイトの読み込みも初期のころに比べて速くなったりと使いやすさにも力を入れることができた.

今後は情報技術科以外の○×ゲームなどを実装することによってより使いやすい作品にし,ターゲットである高校生に産技短の良さを伝えられるようにしていきたい.

参考

岩手県立産業技術短期大学校バーチャル学校見

7. 画像認識を用いたロールプレイングゲームの開発

天沼 優作, 宇部 駿, 加藤 晴太

指導教員 安倍 春菜

1. はじめに

前期の授業で在学生に入学時の当校の認知度に関するアンケートを実施した際、受験した学科については知っているが、他科の授業内容について知られていないことがわかった。そこで、ゲームを用いて簡単に他科の雰囲気を知ってもらいたいと考えた。

ゲーム作成にあたり、近年注目されている画像認識に着目した。昨年の卒業研究にも利用されていた Mediapipe によるハンドトラッキングを応用したいと考えこのテーマを選定した。

2. 研究概要

2.1 開発環境

OS	Windows11
ソフトウェア	Unity2019.4
	Blender
ライブラリ	Mediapipe
使用言語	C#(Unity)/Python(Mediapipe)

2.2 ゲームの仕様について

ゲームの内容は、産業技術短期大学校(以下「産技短」と称する)を舞台としたロール扮演游戏とした。キャラクター選択画面から各科のイメージキャラクターを選択し、産技短を再現したマップを探索できる。

※ゲームの流れは図1に記載。

各科のキャラクターと戦闘し、最上階にいるボスを目指す。最上階にいるボスを無事に倒すことが出来ればクリアとなる。攻略のカギは各科のキャラクターの特徴を生かして進むこと。

※戦闘シーンの流れを2ページ目に記載。

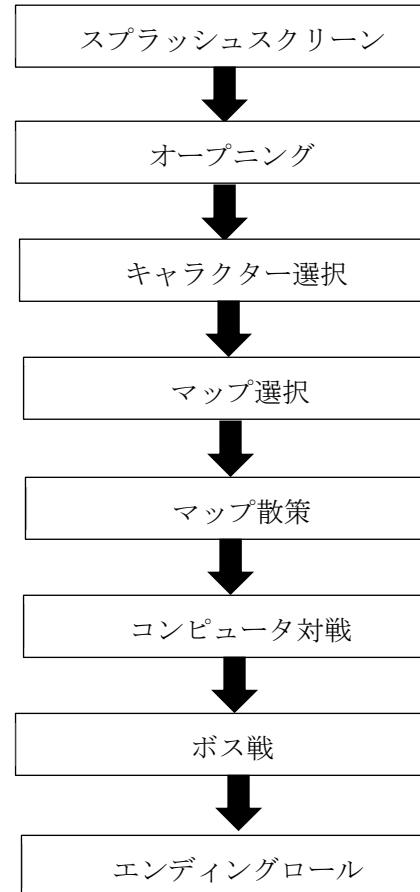


図1 ゲームの流れ

2.4 ジェスチャータイム

ジェスチャータイムは成功したポーズの数によって技の威力が変わるものとする。ジェスチャータイムは、12個のポーズの中から5個ランダムで指定される。指定されたポーズを手で連続してとる仕様にする。

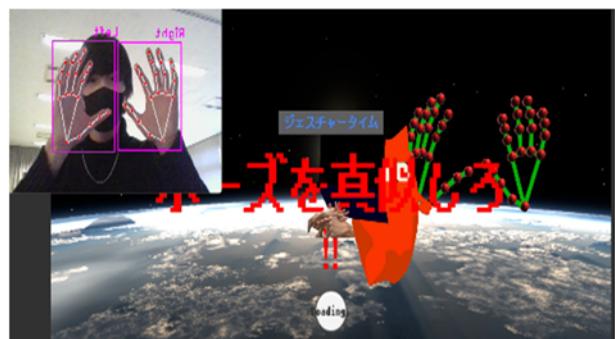
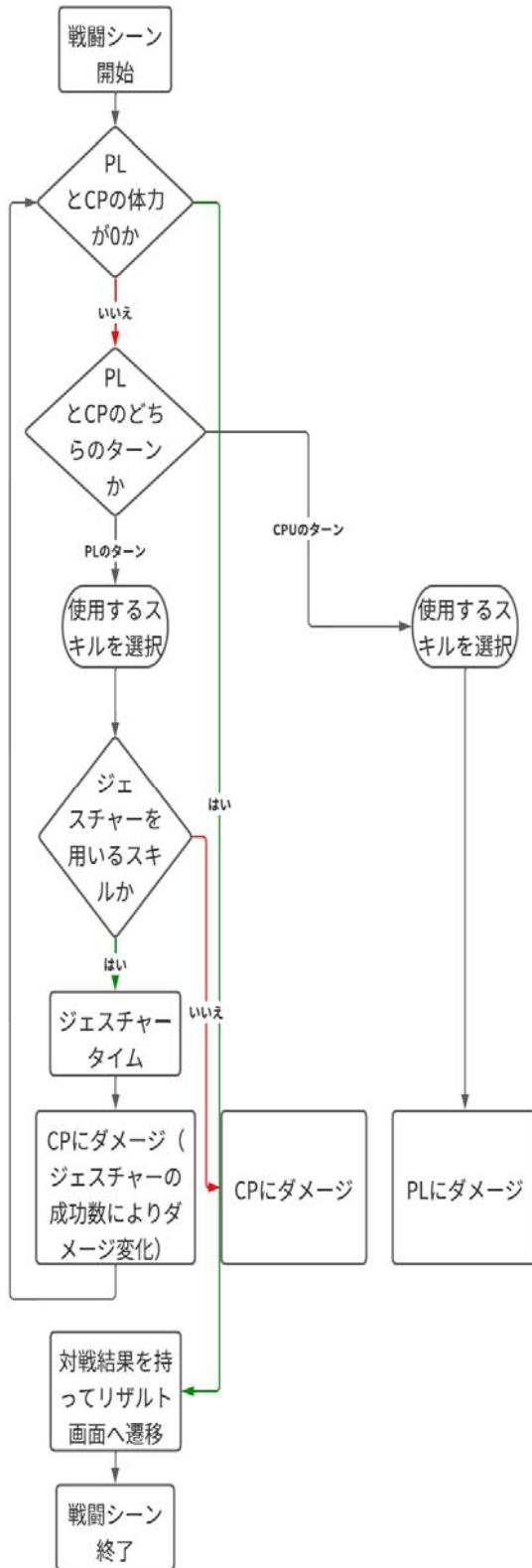


図2 ジェスチャータイム

以下、プレイヤーを PL と称し、コンピュータを CP とする。



2.5 ハンドトラッキング

ハンドトラッキングは Python によって動作している。手の関節一つ一つの座標を取得する Python プログラムを C#で呼び出す。Python で取得した手の座標情報を Python から C#へソケット通信で文字列として送る。受け取った文字列を C#が配列に変換し、Unity のゲームオブジェクトにアタッチする仕組みになっている。この動作を毎秒行い、自分の手がゲーム内で動いているかのように見せている。Unity のゲームオブジェクトは、関節を表す赤い球体と骨を表す緑の棒でできている。球体同士を棒でつなぐことでできており、手の座標情報を球体に与えることで手の形を表現する仕組みになっている。

ジェスチャータイムではこの関節の座標同士の距離を判定式に用いることでどんなポーズをとっているかがわかるようになっている。
(取得する手の関節の座標情報は図 5 に記載)

Mediapipe の仕様上、両手を重ねるようなポーズをとった際に片手しか読み込まれないとといった課題があった。その解決策として、両手が重ならないようなポーズにしたり、重なるポーズであっても判定を甘くすることによって、円滑にゲームを進められるようにしている。

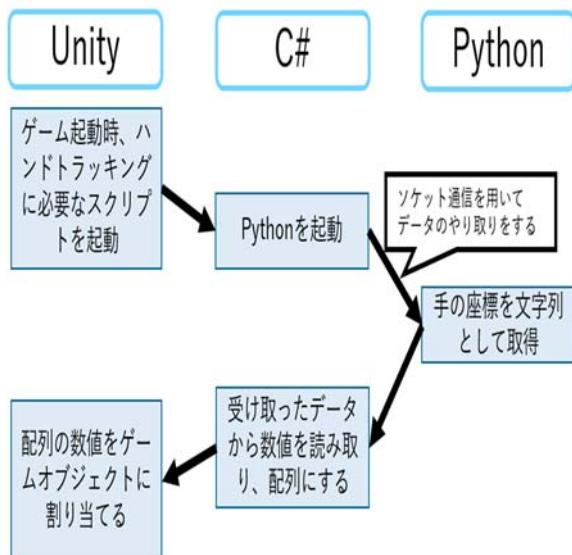


図 3 戦闘シーンのフローチャート

図 4 Mediapipe と Unity の連動の仕組み

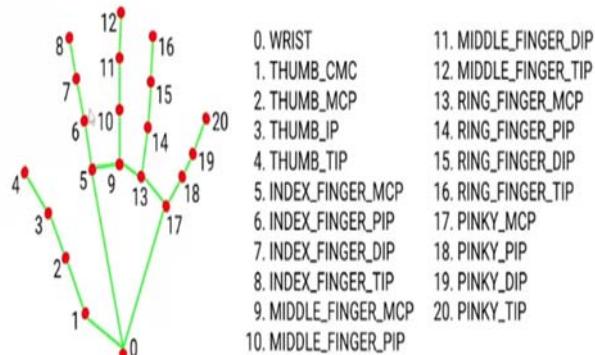


図 5 取得する手の座標図解

3. キャラクター・エフェクト

3.1 使用技術について

オープンソースの統合型3DCGソフトウェアの一つであるBlenderを使用する。3Dモデリング、アーマチュア(骨組み)、アニメーション、テクスチャ(模様)をBlenderで作成する。Blenderで作成したメッシュ(形状データ)やアーマチュアはUnityにそのままインポートできるが、マテリアル(色、影など)はインポートすることができない。そのためUnity上で設定する必要がある。Unityでは主にマテリアルの設定とエフェクトのパーティクルの設定を行っていく。

3.2 キャラクターについて

使用できるキャラクターは5体になっている。

それぞれ産技短(矢巾キャンパス)の学科をイメージしたものになっており、キャラクターデザインは産業デザイン科より許可を得て使用している。スキルの内容も各学科の特徴がわかるようなものにしている。マップ内にいる生徒も各学科で使用されている道具や関する用語をモチーフにして作成している。

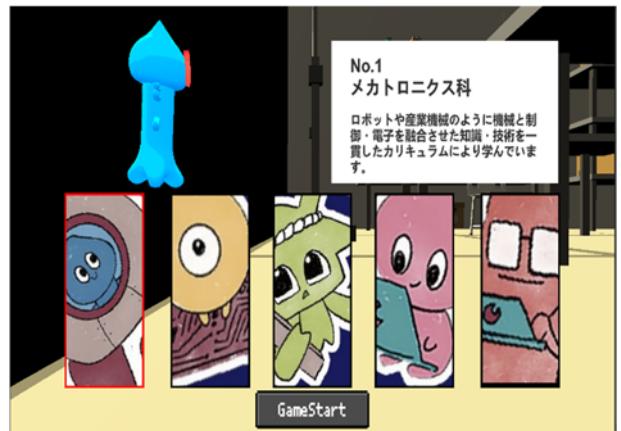


図 6 キャラクター選択画面

3.3 キャラクター作成

それぞれのモデリングにアーマチュアを設定、アニメーションの作成を行った。アニメーションは攻撃1、攻撃2、受け身、走り、通常の立ち状態の5つを用意。キャラクターのマテリアルはUnityで行う。キャラクターのイメージに合うようにアニメ調の色合いに設定。

3.4 エフェクト作成

それぞれの攻撃技に合わせてエフェクトの作成を行った。Blenderで作成したメッシュ、透過PNGをUnity上にインポートし、作成。Unityの機能であるパーティクルシステムを活用し、放出量や物理演算などを設定することでエフェクトを作ることができる。

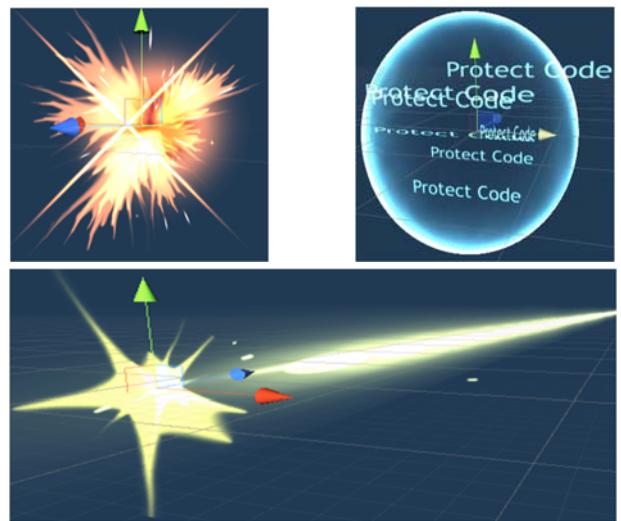


図 7, 8, 9 エフェクト各種

4. マップ

4.1 ゲームエンジンについて

C#を用いたプログラミングでコンテンツの開発が可能であるUnityを使用する。Unityは、PythonやBlenderとの連携も可能であり、Asset(ゲーム素材)が豊富である。

4.2 マップ作成

マップは産技短(矢巾キャンパス)の各科フロアをモチーフにして作成した。ボスステージでは、学校の崩壊をテーマに雰囲気のある空間を目的とした。マップ作りでは、オブジェクトを作成し、全体の外枠を組むことや、Assetを使用し、備品などの細かいところの再現。また、マテリアルを変更し、キャラクターと合うように設定を行う。



図 10 情報技術科フロア



図 11 ボスステージ

4.3 マップ概要

キャラクターが敵を探すマップについて説明する。Unityの仕様上オブジェクトを置くだけでは、キャラクターは歩くことができないので、マップの床や壁に当たり判定を付けることで、キャラクターが壁などを貫通するのを防いでいる。

また、ドアに近づくと自動で開くように設定したり、生徒に近づくと話しかけられることや、マップ内に散りばめられたヒントとともにクリアできるクイズやクエストなど、ユーザーが楽しめるギミックを盛り込んでいる。

5. おわりに

今回の研究では、ハンドトラッキングによるプロジェクトの実装が予定より早く進むことができ、ゲーム開発に費やす時間を増やすことができた。

結果として、ただの学科紹介ではなく、誰もが楽しめる、ユニークでゲーム性の高いホールプレイングゲームが完成した。

6. 参考文献

[UnityでMediaPipeを実行してVTuberになる](#)

[3d Hand Tracking in Virtual Environment / Computer Vision](#)

[Unityゲームエフェクト:初めてのシェーダグラフ\[ネクストンCG\]](#)

[\[Blender\] ずぼらな人の為のキャラクターの作り方](#)

[作って学べるUnity超入門](#)

08 深層学習を用いたゴミ分別支援ツールの作成

5 遠藤 紗李

指導教員 ソソラ

1. はじめに

資源回収において、ペットボトル回収箱に他のゴミが捨てられている等の異物混入^{*1}が問題になっている。一方、手をかざすと自動的に開くセンサー付きのゴミ箱が普及しつつあることからカメラ付きの回収箱があれば便利だと考えた。つまり、図1に示したペットボトルの回収箱はカメラにペットボトルを写すと、蓋が開き、ペットボトル以外のものであれば蓋が開かないものとする。

そこで、授業で学習した深層学習アルゴリズムを用いてカメラの画像から回収できる、または回収できないというようにゴミの分別ができるモデルを制作してみたいと考えた。



図1.イメージ

2. 研究概要

2.1 概要

本研究では、リアルタイムでカメラの画像からゴミの検出する技術に着目する。物体検出アルゴリズムの一つである YOLO^{*2}を用いて「ペットボトル」、「カン」の2種類の半別モデルを作成し、実験検証を行った。

2.2 開発環境

開発環境は以下の通りである。

表 1. 開発環境

OS	Windows10
主ツール	YOLOv5, labelImg Google Colaboratory
言語	Python = 3.8.3
パッケージ	OpenCV=4.5.4.60, Keras=2.10.0 Tensorflow=2.10.0, cvzone=1.5.6

2.3 開発フロー

開発フローを以下に示す。

1. データセット(画像)の用意
2. LabelImg でタグ(正答)付け、YOLO 形式のファイルに変換
3. Google Colab(GPU)で学習
4. 学習済みのモデルを local PC に移植
5. Local PC で YOLO を実行
6. リアルタイムで物体検出

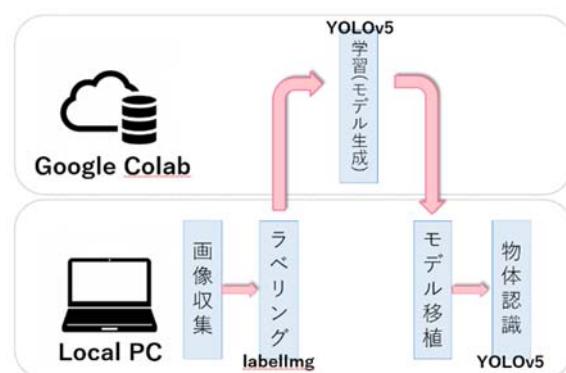


図2. 開発フロー

2.4 YOLO とは

YOLO とは、コンピューターが外部の物体を検出するときに使用される代表的なアルゴリズムのことである。YOLO という名前の由来は、「You Only Look Once」という英文の頭文字をつなげて作られた造語である。物体の検出した時のイメージを以

下の図に示す。

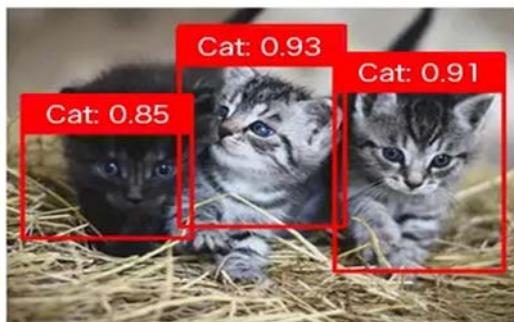


図 3.YOLO で物体検出している様子

3. 研究内容

3.1 データセット準備と学習

自作データセットの詳細は以下の通りである。

画像枚数 100 枚(ペットボトル)/100 枚(缶)

画像サイズ 640*480

学習回数 1000 回

所要時間 6 時間

学習ツールは、ブラウザから直接 Python を記入でき、GPU が使用可能なサービスである Google Colaboratory を採用した。

3.2 検証と結果

学習後の「ペットボトル」分類器において回収できるまたは回収できない物体と分けて、実験を行った。以下に、その認識率をまとめて表示する。

「ペットボトル」モデル		
回収できる	回収できない	
99.2%	2%	20%

図 4. 「ペットボトル」モデルにおける認識率

ここで、認識率およそ 80%を閾値とし、閾値より大きい場合は回収できる、そうではない場合は回収できないと判断するように考えている。同様に「カン」モデルに対して実験を行った。

「カン」モデル		
回収できる	回収できない	
96.7%	1%	17%

図 5. 「カン」モデルにおける認識率

3.3 学習における損失

図 6 には「ペットボトル」モデルの学習における損失数のグラフを表示した。この損失関数のグラフを見ると、初期段階で大きかった loss が徐々に小さくなっているため、正しく学習されていることが分かる。

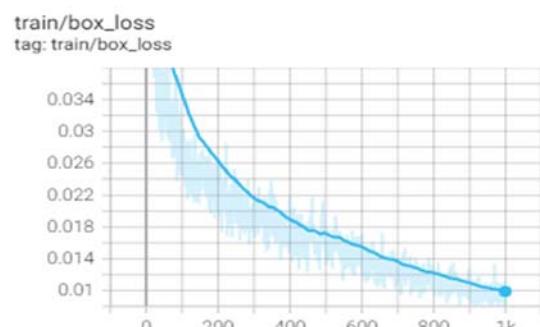


図 6. 損失関数グラフ

4. おわりに

本研究を通じ、Python の開発環境構築から記述方法、そして画像認識や AI への応用方法を学ぶことが出来た。

AI の分野はブラックボックスな面が多く、エラーの原因が推測しにくい場面も多々あった。苦労した点ではあったが、エラーの検索力の向上に大いに役立てることが出来、有意義な時間となった。

引用文献

[1]公明党:資源の 3 割 異物混入(2022)

<https://www.komei.or.jp/komeinews/p224430/>

[2]YOLO アルゴリズム解説

https://www.renom.jp/ja/notebooks/tutorial/image_processing/yolo/notebook.html

[3]yolo v5 自作データの学習方法

<https://qiita.com/enya314/items/1bd053d6a81a156ff81>

09 Web カメラによる視線トラッキングに関する研究

小松篤志

指導教員 ソソラ

1. はじめに

近年、世界的に非接触の操作デバイスの需要が拡大している。音声認識や体を使ったジェスチャーを画像認識するなど多くの手法がある。

そこで今回私は、眼の動きをトラッキングして何かを操作できるツールを製作したいと考えた。

2. 開発環境

OS	Windows 10
統合開発環境	Visual Studio Code
開発言語	Python 3.8.10
使用ライブラリ	OpenCV MediaPipe faceMesh Pandas Numpy Kivy Concurrent.futures winsound
カメラ	Integrated Camera(標準装備)

3. 研究概要

本研究では、特殊なハードウェア [1] を必要とせずに、WEB カメラを使用してリアルタイムで虹彩、瞳孔、および眼の輪郭を含むキーポイントにより、眼の動きを追跡するアルゴリズムを開発する。

3.1 眼の左右の判定について

提案したアルゴリズムは以下の通りである。

ステップ 1 : MediaPipe の FaceMesh を使用し、眼のキーポイントのうち図 3.1 に示す①～⑤（①—左目頭、②—右目頭、③—左虹彩の中心、④—右虹彩の中心、⑤—目頭と目頭の中心）のキーポイントの座標を取得する。

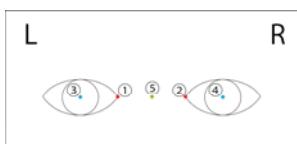


図 3.1 本手法に用いられる眼のキーポイントの位置情報

ステップ 2 : ①～⑤のキーポイントの座標から異なるキーポイント同士の距離求める。ここで、i 番目のキーポイントの座標 $P_i(x_i, y_i)$ 、j 番目のキーポイントの座標を $P_j(x_j, y_j)$ とすると i, j の距離 D_{i-j} を式 3.1 により求める。そして、図 3.2 (ア) には前へ向いている初期状態、(イ) には眼球を右に動かす状態の距離をそれぞれ色別で示した。

$$D_{i-j} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \quad (3.1)$$

ステップ 3: 頭を動かさない状態で、眼を右に動かす場合は D_{3-1} と D_{2-4} の値が変化される。つまり、眼球を右に動かすと D_{3-1} (III) の値がある閾値より小さくなり、 D_{2-4} (IV) の値もある閾値より大きくなる。

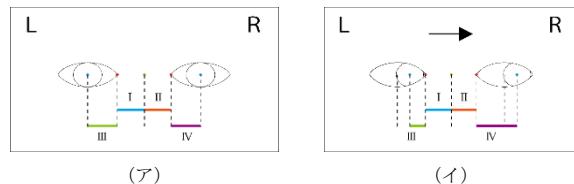


図 3.2 眼を右へ動かしたときの変化の様子

3.2 眼の上下判定について

上を向いた時のアルゴリズムと下を向いた時のアルゴリズムを分けて提案した。

3.2.1 上を向いた時のアルゴリズム

ステップ 1: 新たに図に示す眼のキーポイントの座標を取得する。

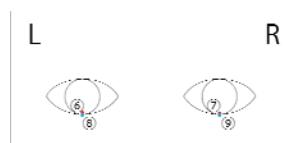


図 3.3 上向き判定のキーポイント

ステップ 2: 3.1 ステップ 2 同様にキーポイント同士の距離を取得する。

ステップ3: 眼を上に動かすと D₆₋₈ と D₇₋₉ の距離が変化する。これを左右どちらの目でも判定する。

実験中の試行錯誤

左右の判定に比べ、変化距離が非常に小さく、誤差値と正常値で区別に難航した。

そこで指数平滑移動平均法 [2] を利用し、データの値を均することで誤差値と正常値を区別しやすくなれた。

3.2.2 下を向いた時のアルゴリズム

ステップ1: 図 3.1 に加えて新たなキーポイントの座標を取得する。



図 3.4 下向き判定のキーポイント

ステップ2: 線 6-8 と点 4 の距離を計算(式)

反対の眼も対応する線と点で計算

ステップ3: 左右それぞれの眼の面積を計算

ステップ4: ステップ2で算出した距離が線 6-8 より下へ移動したことを判定する。同時に、眼の面積が閾値より低くなった時に判定結果を返す。

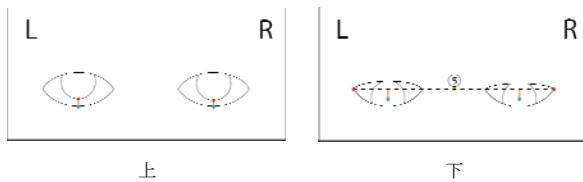


図 3.5 眼を上下へ動かしたときの変化の様子

3.3 視線方向取得を活用したアプリ

作成した視線方向取得ツールを用いて音楽を演奏することのできるデスクトップアプリケーションを作成した。



図 3.6 アプリメイン画面

3.3.1 アプリ仕様

- ・視線方向に合わせて音程が変わる
- ・画面右側が操作オブジェクトである
- ・画面右上の「BPM」欄で BPM(リズム)の変更が可能

各種ボタンの機能

ボタンの名前	機能
mute	演奏の一時停止
play	演奏再開
up, down	音程の切替
set(BPM 欄)	BPM を確定する
?	操作説明
終了	アプリの終了

4. 終わりに

本研究を通して、眼の動きをトラッキングする専用ハードウェアがなくとも眼の動きを検知することが可能であると確認できた。視線を取得する方法として角膜反射法 [3] がある。しかしこの方法では眼に光を当てなければならず目への負担がどうしても避けられない、しかし本研究では、外部から的人体への影響を極力減らして実行することが可能となた。しかし、課題点も残った。再初期段階で考えていた案は視線で PC のマウスのように画面上を操作できるようなツールだった。本研究のような、市販の web カメラから画像認識をして目の動きを取得する方法であると細かい目の動きを検知する精度が出せなかつた。妥協案として方向を取得するだけなら誤差と区別できる明らかなデータが取得できると考えた。

今後の課題として左右の判定の精度は十分に作ることができたが、上下の判定の精度は改善の余地が残されている。

参考文献

- [1] 特殊なハードウェア
<https://www.tobii.com/ja/products/eye-trackers>
- [2] 指数平滑移動平均法
<https://mirukognosis.com/?p=1020>
- [3] 角膜反射法
<https://techblog.nhn-techorus.com/kakumaku>

10 Unity を使ったゲームの作成

菊池 愛斗 菅原 遥都 照井 太一 元良 優作
指導教員 小野 陽子

1. 研究の背景と目的

今まで業務システム中心に作成していたため身近にあるものを作成したいと考えた。その中でも本校で学んだ様々な技術の復習と活用ができ、今後必要となる知識も学べるゲーム作成をしようと考えた。ゲーム作成はシステム、デザイン、アイディアなどの考えを交換しながら作成するためグループとしての開発スキルも高めることができる。私たちが作成したゲームを産技短展や文化祭などに来た人がプレイすることで、パソコンに親しんでもらったり、産技短に興味を持ってもらうきっかけの一つになったりすればいいと考える。以上の点からUnityを使ったゲームを作成することにした。

2. 研究概要

1～2人の開発でルールが簡単で誰しもが楽しめるようなゲームを複数個作成する。作成したゲームに飛ぶためのホーム画面を作成する。

2.1. 開発環境

開発環境を表1に示す。

表1 開発環境

OS	Windows10
ツール	Unity Photon
言語	C#

3. 成果物

8つのゲーム、及びホーム画面を作成した。

3.1. タンク



図1 タンクゲームの完成図

このゲームは戦車を動かし、弾を発射して、敵戦車を破壊するゲームである。敵の攻撃により、HPが0になるか、制限時間を過ぎると、ゲームオーバーとなる。ステージが全部で5つあり、そのすべてのステージのクリア条件を満たせば、ゲームクリアとなる。

ゲームクリアした場合、倒した戦車の数とクリア時間に応じたスコアとこれまでのハイスコアが表示される。

操作方法は、移動がカーソルキーか、「wasd」キー(上移動が「w」、左移動が「a」、下移動が「s」、右移動が「d」)で、弾の発射が左クリックか、「スペース」キーとなっている。

開発に当たり、苦労した点は3つある。

1点目は同じプログラムを実行しても、プレイするパソコンによってゲーム内のオブジェクトの動きが変化する現象に対して、フレーム単位ではなく、時間単位で動かすことで修正した。

2点目は敵オブジェクトが、自分が操作しているオブジェクトを認識するときに予想外の動きをしてしまうバグを修正したことである。例を一つ

上げると、図2のように、本来地面を走るように設計したはずが宙に浮かんでしまう。



図2 宙に浮かんでしまうタンク

3点目はゲームのスコアを、ステージをクリアした場合は値を保持し、ステージをオールクリア、または、ゲームオーバーになった場合は値をリセットする仕様にしたことである。

3.2. カードゲーム

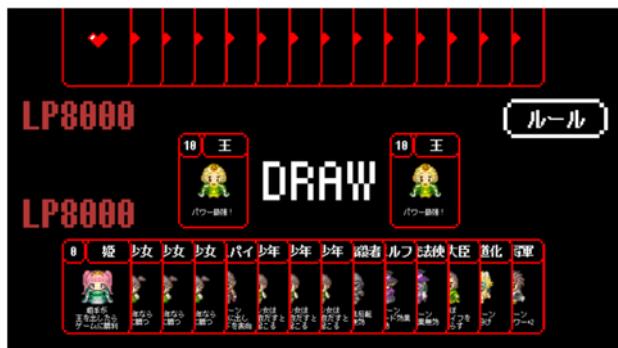


図3 カードゲームのプレイ画面

このゲームは、毎ターン、カードを1枚選択(クリック)し、決定ボタンをクリックする。自分が選んだカードと相手が選んだカードの左上のパワーの数字が大きい方がバトルに勝つ。

バトルに勝つと、相手のライフを1000減らすことができ、先に相手のライフを0にした方が勝ちとなる。お互い、ライフが0にならずに手札がなくなった場合、ライフの残量が多い方が勝ちとなる。

それぞれのカードには、特殊効果があり、パワーを上げる、相手の効果を無効にする、条件を満たせば、ゲームに勝つなどがある。

また、このゲームはNPC対戦と通信対戦の両方でできるようにした。大まかな流れは同じだが、通信対戦のほうは、相手とのマッチング処理や、プレイ中に自分が選択したカードや、相手に影響を及ぼすカード情報などを相手のゲームに伝えなければならない。そのため、Photonというオンライン対戦をするために必要な機能が備わっているツールをUnityにインポートすることでこれらのことができるようになった。

苦労した点は、ゲームバランスが狂わないように、一発逆転の効果のカードをバグが発生しないように実装したことである。

3.2.1. Photon

カードゲームは、CPU戦のほかに通信対戦もしたいと考えた。Photonというツールを使用することで可能になった。Photonとは、サーバーの管理やマルチプレイをする上での面倒な処理をSDKとして提供しているネットワークエンジンのことである。このツールを使うことで、ネットワークに詳しくなくても、比較的簡単にオンラインゲームを作ることができる。



図4 Photonの公式サイト

苦労した点は、Photonを使ったオンライン対戦の時とCPU戦の時とで違う動きをしてしまい、それを修正したことである。

3.2.2. カードゲームのモード選択



図 5 カードゲームのプレイ画面



図 6 カードゲームのマッチング画面

カードゲームを始める時に、初めに図 5 の画面が表示され、CPU 対戦とオンライン通信対戦のどちらか一つを選び、遊ぶことができる。CPU ボタンを選択すると、図 3 の画面に遷移して、すぐに遊べるが、オンラインボタンを選んだ場合、図 6 の画面に遷移し、対戦相手が見つかった後は、CPU 戰と同様に図 3 の画面に遷移する。

3.3. シューティングゲーム

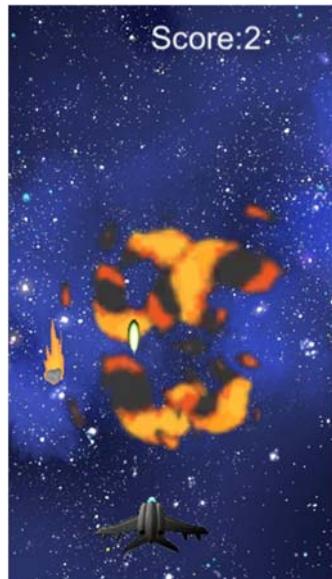


図 7 シューティングゲームの完成図

降ってくる敵や攻撃をよけながら弾を当てていき、敵や攻撃に当たるとゲームオーバーとなる。

「A」で左、「D」で右に動かす。もしくは「カーソル左」で左、「カーソル右」で右に動かす。

スコア機能、ハイスコア機能も実装している。

左クリックか「スペース」キーを押すことで弾が発射される。

敵にはただ降ってくるだけの通常敵、特定の技を放つボスキャラがおり、敵を倒すごとにスコアが上がり、一定値になるごとにボスを出現させる。

ボスはそれぞれ異なる攻撃を繰り出してくれる。

最後のボスを倒したら、ゲームオーバーとなるまで通常敵が降ってくるのでハイスコアの更新を目的としてもゲームを楽しめる。

ゲーム開始時難易度「Normal」、「Hard」のどちらかを選択し、ゲームをプレイする。

「Hard」の変更点としては、「Normal」より敵の HP を増やしたこと、雑魚敵の出現頻度を上げたことの 2 つある。

苦労した点は、2 つある。

1 点目は、楽しく遊ぶための難易度調整をしたことである。具体例を挙げると、敵キャラクターの

出現頻度や HP を初心者から上級者まで楽しく遊べるように微調整を繰り返した。

2点目は、ゲーム内のキャラクターの技をそのキャラクターに合うように設定したことである。具体例を挙げると、ビームをくりだす敵を出現させたいと考えた時に、それに合ったキャラクターを作ったり、登場させたいキャラクターを考えそれが剣を使うキャラクターならば、そのキャラクターの技を斬撃にしたりした。

3.4. 反射神経ゲーム

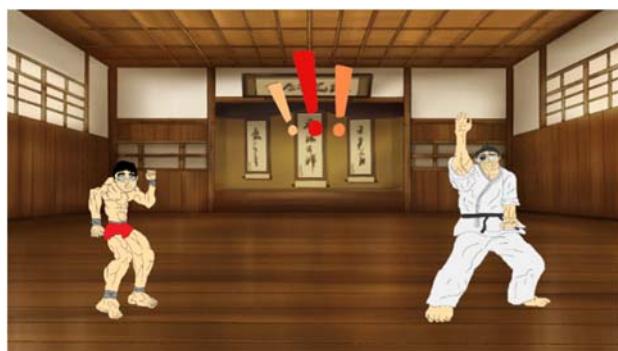


図 8 反射神経ゲームの完成図

「！」が表示されたら CPU(右側)よりも速く左クリックすることで勝ちとなる反射神経ゲームである。

待機していると指定時間内のランダムなタイミングで画面に「！」が表示される。

「！」が表示されたら素早く左クリックを押す。クリックの速さが CPU(右側)よりも速ければ勝利となる。

「！」が表示されるよりも前にクリックするとゲームオーバーとなる。

ステージは全部で5つあり、すべて勝利することでゲームクリアとなる。

途中で負けるとリトライ画面に移動し、リトライボタンを押すとまた最初のステージからゲームが始まる。

次のステージに上がるごとに敵のスピード、「！」が表示されるタイミングが異なる。

苦労した点は、「戦い」というイメージでこのゲ

ームを作成したので、戦いに合った背景と登場キャラクターを設定したことと、誰でも遊べるようにキーボード操作を使わずにクリックのみで遊べるようにしたことである。

3.5. テトリス風ゲーム

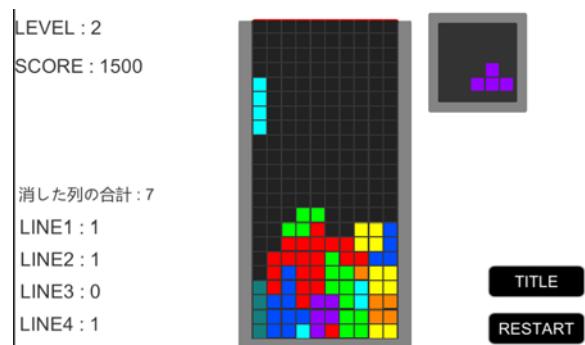


図 9 テトリス風ゲームの完成図

7種類のブロックがフィールド上部からランダムに1つずつ落下し、フィールドに敷き詰めると揃った横一列を消すことができるゲームである。

「A」で左、「D」で右、「S」で下、「W」で右回転することができる。もしくは「カーソル左」で左、「カーソル右」で右、「カーソル下」で下、「カーソル上」で右回転することができる。

「スペース」キーでブロックを即時落下させるドロップすることができ、「左コントロール」キー、「Q」で左回転することができる。

次に降ってくるブロックを表示させるブロックを作成し、ブロックの落下地点が表示させるブロックを作成した。

消した段階に応じてスコアが上がっていき、スコアが一定値に達するごとにスピードも上がっていく。

規定のスコアに達することでゲームクリアとなる。ブロックが詰めなくなり、フィールド外に出ることでゲームオーバーとなる。

苦労した点は、ブロックの落下地点が表示させるブロックを表示させることである

3.6. 横スクロールアクションゲーム 1



図 10 アクションゲーム 1 の完成図

キャラクターを操作してゴールを目指すゲームである。

キャラクターの操作方法は「A」で左移動、「D」で右移動、「スペース」キーでジャンプすることができる。もしくは「カーソル左」で左、「カーソル右」で右に動かすことができる。

キャラクターがブロックに触れるとブロックが移動するものを制作した。

画面上でアイテムに触れるとスコアが増えるものを制作した。

難易度はイージー、ノーマル、ハードを選択することができ、1つの難易度につきステージ数は5つとなっている。ステージをクリアすると次のステージに進むことができ、5つのステージをクリアするとスコアを表示してタイトル画面に戻る。

ステージには制限時間があり、時間が0になるとゲームオーバーになる。キャラクターが敵に当たる、穴に落ちるなどをしてゲームオーバーになる。ハードのステージでは強制的にスクロールするステージを制作した。迫ってくる壁に触れるとゲームオーバーになる。

苦労した点は、ステージ作りにおいて、適切な難易度のステージを複数作ったことである。難易度によってステージの難しさを調整してクリアできるようにした。

3.7. 横スクロールアクションゲーム 2

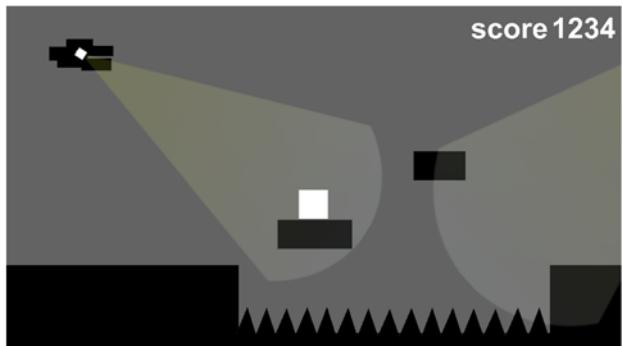


図 11 アクションゲーム 2 の完成図

基本的なキャラクターの動きは 3.6 のものと同様である。

キャラクター操作は、「カーソル左」で左、「カーソル右」、「カーソル上」でジャンプを行う。

3.6 の横スクロールアクションゲームとの違いとして難易度の選択は無く、ステージも一つにまとめてている。

穴に落ちる、敵や仕掛けにあたる、敵の攻撃にあたるなどでゲームオーバーとなり、時間経過によるゲームオーバーは無い。

ゲームは進行していくほど難易度が上がり、どこまで進んだかのハイスコアを目指してプレイする。

仕掛けとして、プレイヤーを認識し、ある程度の範囲まで追跡しながら攻撃してくる敵を配置している。また自動で動き、出たり消えたりする仕掛けもあり、それらにあたるとゲームオーバーになる。

苦労した点は、プレイヤーを認識している時としていない時で、敵の動きが大きく変わる設定にする際の制御プログラムの作成である。

3.8. メニュー画面

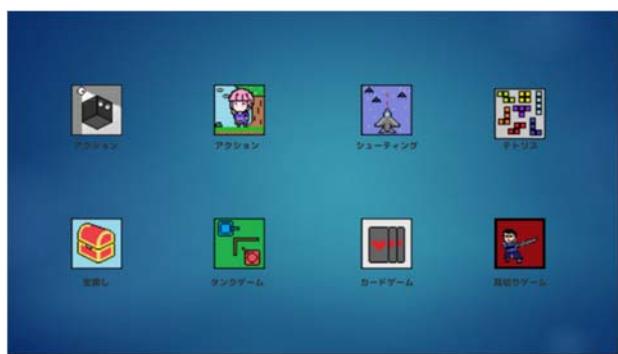


図 12 メニュー画面

ゲームごとに一目見てどんなゲームかがわかる印象的で記憶に残りやすいアイコンを作成した。

メニュー画面を図 12 のようにし、画面内のアイコンのボタンをクリックすることで、各ゲームに遷移することができる。ボタンにマウスカーソルを合わせると、ボタンが拡大表示されるようにした。



図 13 タイトル画面

また、こだわった点としては、ゲームを実行した時にすぐにメニュー画面が表示されるのではなく、市販の人気ゲームのように、初めにロゴを表示させ、次に図 13 のようなタイトル画面を表示させ、その画面をクリックすることで図 12 のようなメニュー画面に遷移できるようにした。

4. 終わりに

本研究では Unity を使ったゲームの作成を行った。研究を通して、ゲームの種類は違いながらも世界観だけはある程度統一することを意識し、作品を作り上げる難しさを痛感した。

Unity を使うこと自体が初めてであったので、ゲームの仕様、難易度、デザインなどを考えながら作成するのに苦労したが、仕様や難易度を調節し、誰しもが楽しめるようなゲームを作成することができたと思うので、産技短展や文化祭などで体験してほしい。

研究を通して計画を立てることの重要性、グループで作りあげることの難しさを学ぶことができた。

5. 参考文献

たのしい 2D ゲームの作り方

Unity ではじめるゲーム開発入門

STUDIO SHIN 著

<https://nnhokuson.hatenablog.com/entry/2016/07/04/13231>

<https://www.youtube.com/@dkrevel6735>

https://www.youtube.com/@studio_shimazu/featured

11 Vue.js を用いた Web アプリケーションの作成

相馬加奈

指導教員 小野 陽子

1. はじめに

以前から JavaScript を使ったフレームワークでのアプリケーション開発に興味があった。そのため、Vue.js を使用した Web アプリケーションを開発したいと考えた。

2. 研究の意義

フレームワークの導入は授業では学習しないため、その学習例を卒業研究の形で残すことで、他の生徒にとって、フレームワークやライブラリを使うハードルが下がるのではないかと考えた。今回は学生にとって手軽で使いやすい学習支援アプリを作成することにした。

2.1 本研究の概要

当初から計画していた学習支援アプリケーションについては、追加機能の実装を考えた時、GoogleAPI 等の連携など、追加で学習する内容が増え、本筋から逸れた内容になることが懸念されたため、より Vue.js を使って作ることに意味のあるアプリケーションも作成したいと考えた。

そのため、「標語コンクールの集計アプリケーション（以下標語アプリと表記）」の開発も同時進行で進めることとした。

2.2 開発環境

以下に開発環境を示す。

表 1 開発環境

IDE	Visual Studio Code
プラグイン	Vue-CLI Vuex Vue-router

2.3 学習支援アプリケーションについて

ToDo アプリ機能と単語帳機能を作成した。単語帳について、以下にリストページの UI と、追加画面の UI を記載する。



図 1 登録済単語のリスト表示画面

図 2 単語の追加画面

2.4 標語アプリについて

標語の登録画面と、投票するシステム、Node.js を使ったログイン機能を作成した。

リスト表示画面について、ジャンルごとの切り替えが出来るよう、タイプ別で切り分けをし、タブ

形式で表示ができるようにしている。

リスト表示画面で投票する標語をジャンルごとに各1つ仮選択し、投票確認画面に遷移する形になっている。

以下に標語追加画面とリストページのUIを記載する。

図3 標語アプリ 標語登録画面

ID	タイトル	タイプ	投票数	状態	投票ボタン
122	安全標語とSDG標語を各自選択して投票して下さい。	safe	3	○	投票ボタン
121	安全標語安全標語安全標語	safe	0	○	投票ボタン
119	安全標語安全標語安全標語	safe	2	○	投票ボタン
116	安全標語安全標語安全標語	safe	2	○	投票ボタン
115	安全標語安全標語安全標語	safe	0	○	投票ボタン
113	安全標語安全標語安全標語	safe	2	○	投票ボタン

図4 標語アプリ リスト表示画面

3. 同様のアプリをHTML等で作成した場合との比較

3.1 全体としての利点

Vue.jsを使うことの利点は、特定のページの内容をコンポーネントとして部品化して、他のページでも読み込むことが出来る点である。今回のアプリを作成する中でも、コンポーネントとビューを分けることで、ページの一部を変更したい時、どのコードを変更したら良いかが分かり易いと感じた。

また、ビューの中では、簡単に複数のコンポーネントを読み込むことが出来るので、表示形式なども工夫できると感じた。

3.2 学習支援アプリを制作した時の利点

まず、Vue.jsを用いて開発する都合上、SPAに近い構造になるのでページ遷移が速くなる。今回はローカルストレージを使用してデータの保存をしているが、Node.jsとの連携も可能なので、一般的なデータベースを利用することもできる。

3.3 標語アプリを制作した時の利点

現在、標語を応募するフォームをコンポーネント化し、タブ上に並べる形式で表示しているが、これをJavaScriptで再現しようとすると、HTMLコード上に全てのコンテンツの内容を記載した後に、JavaScriptで表示する内容を選択し、表示するといった形になる。

Vue.jsを使った場合と比べて、個々のコンテンツ内容を変更したい時にどこを変更するのかが分かり辛くなると感じた。

Vue.jsを用いた場合、個々のコンポーネントが別ページになっているため、どこを変更すべきかが分かり易く、またビューページではコンポーネントの読み込みと切り替え表示のみのコードを記載するだけで良いため、動作が軽く、変更もしやすいと考える。新しい項目を増やしたい場合の拡張性にも優れている。

また標語リスト表示画面では、選択中の単語IDを下部にリアルタイムで表示できるため、どの項目を選択しているかが視覚的に分かり易い。仕組み自体はJavaScriptを用いても作成することは可能だが、Vue.jsを使うとコード数も短く、簡単に同じ機能を実装できる。

4. おわりに

Vue.jsを用いたアプリケーション作成の一連の流れを知ることができたので今後役立つと考えた。フレームワークを使った開発は通常のページ作成より作業自体は少ないが、開発例などを調べても資料が見つかり辛いため、注意が必要だと感じた。

特に今回用いたVue.jsについては、Vue3での開発例が少なかったが、文法は分かり易く、一度理解したらページを作成するのは簡単だったため効率的に開発が出来たのではないかと思う。

令和4年度卒業研究発表会予稿集 情報技術科
岩手県立産業技術短期大学校
令和5(2023)年2月17日発行



<https://www.iwate-it.ac.jp/>