

08 OpenCV を活用したホッケーゲームの制作

佐藤寛大 南舘玲来

指導教員 菅野 研一

1. はじめに

昨年の産技短展で VR ゲームが人気だったが、小学生以下の児童が斜視などの考えうる悪影響により、体験を制限された。また、近年子供の運動習慣の低下が問題視されている。そのため、児童でも安全に楽しめ Unity と Python を使用して、室内でできる健康的でインタラクティブなゲームを開発したいと考え、このテーマを選定した。

2. 研究概要

2.1 開発環境

OS	Windows10
開発環境	Unity, PyCharm
ライブラリ	OpenCV, cvzone
使用言語	C#(Unity)/Python(PyCharm)

2.2 ゲームの仕様について

2.2.1 ゲームのルール

- ・床にプロジェクターで投影した画面でホッケーゲームを行う（図1）。
- ・三脚に固定した Web カメラからフローリングワイパーの赤いポイントを読み取る。
- ・1対1の対戦システムである（図1）。
- ・床での操作と PC 側の操作で分かれる。
- ・時間制の対決にする。
- ・ゲームの流れは図2に記載する。

2.2.2 ゲームの構成

- ・Unity の Scene はスタート画面、遊び方説明画面、プレイ画面、ゲーム終了画面の4つを制作する。
- ・カラートラッキングする HSV 値は、ゲームを行う環境に応じて変化させるものとする。

2.2.3 キャラクター

キャラクターは生成 AI の「ChatGPT」を使用し、フローリングワイパーをモチーフにした「レッドくん」

ん」、キーボードをモチーフにした「キーボちゃん」という2体のキャラクターの画像を生成した。プレイヤーのキャラクターは PC 側がレッドくん、キーボード側がキーボちゃんである。



図1 プレイイメージ

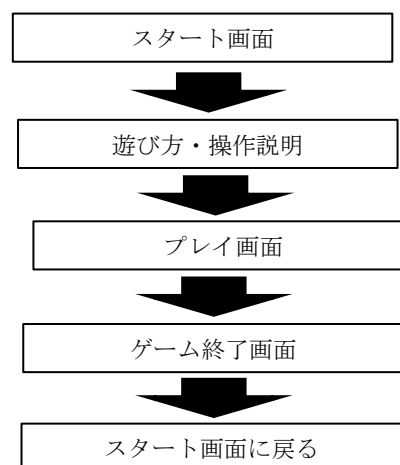


図2 ゲームの流れ

2.3 入力と出力の流れ

- (1) 三脚に固定した Web カメラからフローリングワイパーの色（赤色）をトラッキングする。
- (2) トラッキングした座標を Unity に送る
- (3) 受け取った座標にオブジェクト（マレット）を追従させる。
- (4) プロジェクターから Unity 内の画面の床に出力させる。

2.4 カラートラッキング

2.4.1 カラートラッキングについて

カラートラッキングは Python によって動作して

いる。OpenCV のライブラリを使用しカメラから読み取った映像から HSV 値[1]で色を読み取る。HSV 値とは色相 (Hue) ,彩度 (Saturation) ,明度 (Value) の3つの成分で色を表現する方法。その読み取った値から座標を取得し Unity に送り、リアルタイムでオブジェクトを追従させる。

2.4.2 HSV 値設定

カラートラッキングは cvzone のライブラリを使用し取得したい HSV 値を設定する。

下のコードは赤色の HSV 値

main.py

```
hsvVals = {'hmin':0, 'smin':100, 'vmin':0, 'hmax':11,
'smax':255, 'vmax': 255}
```

図3は HSV 値色相を変化させ緑色を読み取っている。

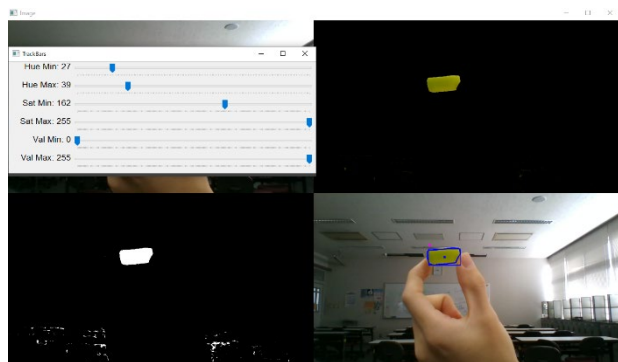


図3 HSV 値設定後

2.5 Python と Unity の連動

Python と C#とのデータのやり取りをするためにソケット通信[2]を行う (図4)。

そのためには,Socket というライブラリを使用し,自分自身を指す IP アドレスとポート番号を指定している。

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
serverAddressPort = ("127.0.0.1", 5052)
```

図4 ソケットの生成

ソケット通信は UDP と TCP があるが,通信速度が UDP のほうが速いため UDP のソケット通信を使用する。

2.6 シーンについて

2.6.1 スタート画面の説明・機能

スタート画面は文字を見やすくすることを意識して作成した。文字を動かすことで,アクティブな印象にした (図5)。

- ・スタートボタンをクリックしたときに遊び方説明画面に遷移する。
- ・BGM は Asset Store からダウンロードした,「Casual Game BGM #5」をループ再生している。



図5 スタート画面

2.6.2 遊び方説明画面の説明・機能

背景にゲームのプレイ画面が表示されている。背景を半透明な黒背景とし,画面が切り替わったことが分かりやすくした (図6)。

- ・START ボタンをクリックすることでプレイ画面に遷移できる。



図6 遊び方説明画面

2.6.3 プレイ画面の説明・機能

キャラクターの画像を貼り付けて、プレイヤーが別々に操作している違和感を軽減している。

プレイ中にわくわくできるような背景やテクスチャを使用した（図7）。

- ・遊び方説明画面のBGMが消える。
- ・カウントダウンタイマーが「3,2,1,GO!」と表示された後にBGMが変化する。
- ・制限時間が画面上部に表示されている。
- ・パックが衝突したときの効果音は「効果音ラボ」から参照した。<https://soundeffect-lab.info/>
- ・パックが壁に埋まってしまう場合があるので、Enter キーで画面中央にパックをリスポーンできるようにした。
- ・制限時間が0になったら、ゲーム終了画面に遷移する。



図7 プレイ画面

2.6.4 ゲーム終了画面の説明・機能

背景を半透明な白背景とし、画面が切り替わったことが分かりやすくした（図8）。

- ・タイマーが0になったとき、「ただ今の結果…」と表示し、両プレイヤーの得点を比較して、得点が高いプレイヤーを **Winner** として表示する（図8）。また、同じ得点だった場合は **Draw** と表示する。
- ・RESTART ボタンをクリックすることでスタート画面に遷移する。
- ・プレイ画面のBGMを止める。



図8 ゲーム終了画面

2.7 プレイ環境の設備

2.7.1 プロジェクター

床に映すために木でできた台にプロジェクターを設置する（図9）。

この台は過去の卒業研究[3]で使われたもの。プロジェクターの設定は通常の画面の向きだと、プレイヤーの背中や、Web カメラの三脚によって、プロジェクターの光が遮られてしまう。そこで、プロジェクターの設定の設置モードをフロント・天吊りに設定し、プレイヤーの位置を画面横にずらすことによって光が遮られずに床に投影できる。また、台形補正機能で投影される画面が長方形になるように設定した。



図9 プロジェクターの設置台

2.7.2 Web カメラ

Web カメラでフローリングワイパーを動かす範囲を真上から撮影しなければならないため,Web カメラを固定したい.そこで,マイクスタンドを使い固定した.マイクスタンドのネジと Web カメラのネジが違うため直接接続することはできない.そのため,アルミ板を切断し,Web カメラのネジとマイクスタンドのネジそれぞれに合う穴をあけ,三脚と Web カメラを留め具で結合させた (図 10) .

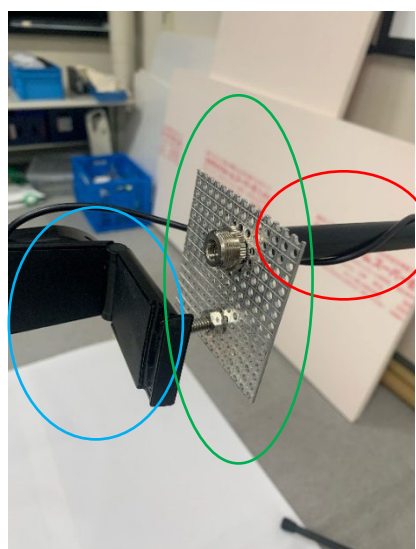


図 10 三脚と Web カメラの結合部分



図 11 Web カメラの全体写真

2.7.3 プロジェクタースクリーン

床に映像を投影するので,床が白いほうが綺麗に描画される.また,フローリングワイパーの摩擦が

少ないほうがストレスなくゲームをプレイできるので,610mm×幅のインクジェット普通紙を養生テープやのりで貼り,画面のサイズに合わせた一枚のスクリーンを作成した.

2.8 ビルドについて

ソースコードを実行ファイルで 1 つのファイルにする.ビルドすることで,アプリ化でき,オブジェクトファイルの生成にかかるオーバーヘッドが減少する[4].

3 おわりに

研究を通して,UnityやPythonの知識が身についた.また,思った通りにプログラムが動かないときの問題解決能力が身についた.本来は,2人でフローリングワイパーを動かして,プレイできるゲームを作りたいかったが,カラートラッキングした座標を 2 つのオブジェクトにアタッチすると,両方ともオブジェクトが動かないという問題が発生し,研究が滞ってしまったため,カラートラッキングで追従させるオブジェクトを 1 つとし,片方のプレイヤーはキーボードで操作するという形式になった.

結果として,子供だけでなく大人も楽しめるホッケーゲームが制作できた.

参考文献

[1] HSV 色空間 :

<https://www.peko-step.com/html/hsv.html>

[2] 【Unity】UDP 通信する :

<https://note.com/hikohiro/n/n17e98f4b2897>

[3] 斎藤 英比古 : Kinect を使った NUI の研究, 平成 23 年度産技短卒研報告書,pp6,2012

[4] Unity (jumbo) ビルドについてのメモ :

<https://lambda00.hatenablog.com/entry/2023/07/02/125251>