

# 視線計測センサーを利用した応用に関する研究

S18548 Yeoh Chun Hoe

## 1. はじめに

本研究は視線計測センサーの応用に関する研究である。視線計測センサーを利用することで、利用者は視線によって対象物を指定することができたり、操作することが可能である。また、視線の動きを解析することで、その人が現在、物事にどの程度集中しているかや、緊張しているかを計測することができる。

本研究では、ドビー・テクノロジー社製の Tobii アイトラッカー5 を利用する。視線計測に関するプログラムを作成し、どの程度、視線計測が可能であるかを明らかにする。

## 2. センサーの設定と開発環境

本研究に使われている Tobii アイトラッカー5 (図 1) はドビー・テクノロジーが 2020 年から生産した視線計測センサーであり、表 1 は Tobii アイトラッカー5 の寸法などの技術仕様である。



図 1. Tobii アイトラッカー5

表 1. Tobii アイトラッカー5 技術仕様

寸法	285 x 15 x 8.2mm
作動距離	45-95cm / 18-37 "
対応画面 サイズエリア	15~27 インチ [16:9] または 30 インチ [21:9]

本研究では、動作環境として、Windows 10 が搭載された PC を利用した。まず、Tobii Eye Tracker 5 専用のドライバソフト (Eye Tracking

Software) をダウンロードしてインストールする。次に、Tobii Eye Tracker 5 をパソコンの USB ポートに接続する。接続すると Microsoft Store が起動して Tobii Experience (常住ソフトウェア) をインストールするよう求められるので、指示通りにダウンロードしてインストールする。

Tobii Experience を起動して指示通りに Tobii Eye Tracker 5 をパソコンの画面の下に設置し (図 2)、その後はキャリブレーション作業の画面が表示されるので、キャリブレーションを行う。



図 2. 設置された Tobii Eye Tracker 5

本研究では開発環境としてゲーム開発環境である Unity を利用した。Tobii Unity SDK をダウンロードする<sup>1)</sup>。Unity で新しいプロジェクトを作成し、SDK の資源をインポートする。API を Unity の C# スクリプトに適用し、プログラムを作成する。

## 3. 視線計測実験

SDK には、視線によって対象物を選んだり、銃を撃ったりするサンプルプログラムが含まれており、最初に、それらのプログラムを実行した。図 3 は、視線によって、対象物を選択するプログラムの出力画面の様子である。

さらに、インターネットにある資料を調べ、様々なサンプルプログラムを実行してみた。図 4 は、シューティングゲームに応用したサンプ

ルであり、視線で弾の弾道を操作することが可能であった<sup>2)</sup>。

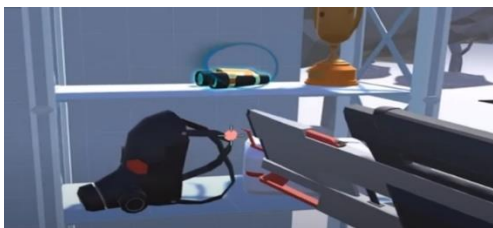


図 3. 視線で物を選択する様子



図 4. 視線で弾の方向を操作できる

実行したシューティングゲームを参考にし、弾の方向ではなく、飛行機(物体)を移動させるプログラムを作成した。まずは視線の座標を Unity に読み取るプログラムを作成した。以下は、そのプログラムの主要な部分である。

```
GazePoint gazePoint = TobiiAPI.GetGazePoint()
if(gazePoint.IsRecent())
{
    print("GazePoint on Screen(X,Y):"
    +gazePoint.Screen.x +", "+gazePoint.Screen.y);
}
```

一行目は、視線の座標を格納した GazePoint オブジェクトを取得するプログラムである。2行目で取得したオブジェクトが有効であるかどうかを判断し、有効である場合には、その座標値を表示させるプログラムである。座標は図 5 に示したように Unity の実行画面の左下に出力される。

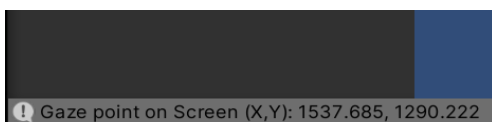


図 5. 右下に表示される x, y 座標

ここで、視線の x と y 軸は実行画面の左下が原点として出力される。実行画面の座標は、中心

が原点となるため、視線座標から画面座標への変換が必要である。視線座標が(0,0)の時に画面座標は(-3,-5)である。また、視線座標の x と y 方向の幅が(770,1350)であり、画面座標の幅が(6,9)であるため、視線座標に対して、その比率を割ることで、画面座標に変換する。以下のその変換プログラムである。

```
position = transform.position;
position.x = (gazePoint.Screen.x/128.0f) -3.0f;
position.y = (gazePoint.Screen.y/150.0f) -5.0f;
transform.position = position;
```

1行目で物体の座標オブジェクトを取得し、2, 3行目で視線座標から画面座標に変換し、4行目でその座標を新たな座標としてセットする。

図 6 は作成したプログラムの出力画面であり、視線の位置にロケットを移動させることができる。

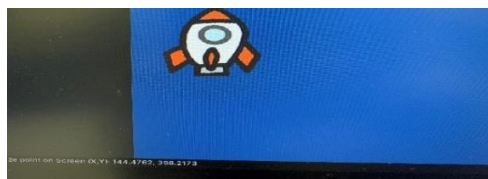


図 6. ロケットを視線で移動する画像

#### 4. おわりに

本研究では、ドビー・テクノロジー社製の Tobii アイトラッカー5 の設定を行い、Unity による開発環境を構築した。SDK をインポートし、サンプルプログラムやインターネットにあるプログラムを実行し、さらに参考として自分なりのプログラムを作成してみた。

今後は視線の動きを計測するプログラムの作成などに取り組んで行き、実用的なシステムの開発を行なっていく。

#### 参考文献

- 1) Scripting API – Tobii Developer Zone  
<https://developer.tobii.com/pc-gaming/unity-sdk/scripting-api/#tobiiapigetgaze-point>
- 2) TobiiEyeTracker シューティング mLAB  
<https://mukai-lab.info/pages/tech/unity/tobii2/>