

2018年 後期 HoloLens班 活動報告書

目次

1. 活動概要
2. HoloLensについて
3. 学習内容
4. 活動で得られたもの
5. 問題点
6. 展望

##活動概要

本プロジェクトでは、HoloLensの使用方法、アプリの作成方法を学び、HoloLensを使ったアプリを作る。

HoloLensについて

HoloLensとは、Microsoft社が開発したMixedReality(以下MR)デバイスであり、様々な分野において活用されている。

活動内容

今回のプロジェクトでは、まずHoloLensを使ったUnityでのアプリの開発方法を学習し、その後アプリ開発においてよく使われる有用な関数の紹介などを踏まえて、実際にアプリを開発した。

学習内容

今回のプロジェクトでは、Unityに触れたことのなかった班員も複数いたことから、Unityの導入から開発環境を整えた。

開発はUnity2017.4.16以降を使用し、UnityでのHoloLens開発で使用されるMixedRealityToolkitをMicrosoft社が公開しているOSSの[Gitリポジトリ \(https://github.com/Microsoft/MixedRealityToolkit-Unity\)](https://github.com/Microsoft/MixedRealityToolkit-Unity)からクローンした。

勉強会ではMicrosoft社公式ホームページのMixedRealityAcademy内のtutorialを使用し、基本的なメソッドやクラスを学習した。

HoloLensにおける操作は、基本的にHoloToolkit.UnityInputModuleを用いて行われ、その中には様々な手の動きの検出用メソッドが格納されている。

以下に主なメソッドを挙げる。

- フォーカス

```
public interface IFocusable : IEventSystemHandler
{
    void OnForcusEnter(); // フォーカス（注視）し始めたとき
    void OnForcusExit(); // フォーカス（注視）が外れたとき
}
```

- ジェスチャー

- インプットの検出

```
public interface IInputHandler : IEventSystemHandler
{
void OnInputDown(InputEventData eventData); // オブジェクト (対象) へのInputDown
void OnInputUp(InputEventData eventData); // オブジェクト (対象) へのInputUp
}
```

- クリックの検出

```
public interface IInputClickHandler : IEventSystemHandler
{
void OnInputClicked(InputClickedEventData eventData); //オブジェクト (対象) へのクリックイベント
}
```

- 手の検出

```
public interface ISourceStateHandler : IEventSystemHandler
{
void OnSourceDetected(SourceStateEventData eventData); // 対象を見ている状態で手が認識されている
void OnSourceLost(SourceStateEventData eventData); // 対象を見ている状態で手が認識から外れた
}
```

- ホールドの検出

```
public interface IHoldHandle : IEventSystemHandler
{
void OnHoldStarted(HoldEventData eventData); // 対象をつまみだした
void OnHoldCompleted(HoldEventData eventData); // 対象をつまんでいる
void OnHoldCanceled(HoldEventData eventData); // 対象のつまみを外した
}
```

- ドラッグ操作

```
public interface IManipulation : IEventSystemHandler
{
void OnManipulationStarted(ManipulationEventData eventData); // 対象をつまみだした
void OnManipulationUpdated(ManipulationEventData eventData); // 対象をつまんでいる
void OnManipulationCompleted(ManipulationEventData eventData); // 対象のつまみを外した
void OnManipulationCanceled(ManipulationEventData eventData); // 途中でトラッキングが外れた
}
```

- ナビゲーション操作

```
public interface INavigationHandler : IEventSystemHandler
{
void OnNavigationStarted(NavigationEventData eventData); // 対象をつまみだした
void OnNavigationUpdated(NavigationEventData eventData); //対象をつまんでいる
void OnNavigationCompleted(NavigationEventData eventData); // 対象のつまみを外した
void OnNavigationCanceled(NavigationEventData eventData); // 途中でトラッキングが外れた
}
```

また、今回のアプリを作成するにあたり、UnityでOpenCVをかたんに使用できる、OpenCV for Unityを使用した。

これは、HoloLensについているカメラをUnity上で使用するように設定することで、簡単にOpenCVが扱える。

```
using OpenCVForUnity;
```

をスクリプトの戦闘に加えることで、このアセットを使用することができる。
実装は以下の手順で行う。

1. スクリプトでCamera.mainを参照するためHoloLensCameraStreamをダウンロードし、Hierarchyに追加し、TagをMainCameraに設定する。
2. カメラの画像から顔の位置を検出する。
3. 取得した位置情報に対してモデルを配置する位置を決定し、配置する。

1は省略する。

次に2だが、これをまた細かく3つの段階に分けようと思う。

- 1. 顔検出用のカスケード分類機のロード

```
cascade = new CascadeClassifier (Utils.getFilePath
("lbpcascade_frontalface.xml"));
```

- オブジェクト検出をするための事前準備として、検出器の読み込みが必要になる。検出器とは検出対象とするオブジェクトの特徴を機械学習によって蓄積させた学習データのことである。今回のサンプルでは"lbpcascade_frontalface.xml"という正面顔検出用のカスケードファイルを読み込む。
- 2. WebCamTextureのフレーム画像をMat形式に変換（このとき、顔検出に不要な色情報を排除するためにグレースケールにする）

```
Utils.webCamTextureToMat (webCamTexture, rgbaMat, colors);
Imgproc.cvtColor (rgbaMat, grayMat, Imgproc.COLOR_RGBA2GRAY);
Imgproc.equalizeHist (grayMat, grayMat);
```

- また、ここでヒストグラムの平坦化を行い検出処理に適した画像情報に加工している。
- 3. 画像に対して分類機で顔検出を行う

```
cascade.detectMultiScale (grayMat, faces, 1.1, 2, 2, // TODO:
objdetect.CV_HAAR_SCALE_IMAGE
                        new Size (webCamTexture.height * 0.2,
webCamTexture.height * 0.2), new Size ());
```

- 入力画像、検出結果を格納するMatOfRectオブジェクト、スケールファクタ、オブジェクトを構成する近傍矩形の最小数、処理モード、最小ウィンドウサイズを引数に入れて顔検出処理をする。

3. 取得した位置情報に対してモデルを配置する位置を決定し、配置する。

```
OpenCVForUnity.Rect[] rects = faces.toArray ();
hoge.transform.position = new Vector3((float)rects[Length].x,
(float)rects[0].y, HogePos);
```

これにより、OpenCVをUnity上で使用し、顔検出をした後、モデルを配置することができるようになった。

活動で得られたもの

今回の活動では、Unityにあまり触れたことのない班員もいた中で活動を開始したため、班員の多くはHoloLensの使用方法に加え、Unityの使用方法を学べたと考えられる。

最終的に一つのアプリが制作途中ではあるものの、基本的な技術は使用することができたため、Unityに使用するアプリの作成方法においては学べたのではないかと思う。

問題点

このプロジェクトは、一回生がプロジェクトリーダーを務め、班員の中には実際に使用し、アプリを製作したことがある者がいたこともあり、円滑な活動は行われることは難しかった。

展望

今回はHoloLensの使用方法から学習を始めたため、短期間でのアプリの開発となった。そのため、今回のアプリ開発において学習・使用した技術を応用したアプリの制作にも携わっていきたい。