

はじめに

# こんな話をします

---

- 作例の紹介
- 遠隔MTGに映像をつなぐ方法
- TouchDesignerでFaceTrackingの仕方
- TouchDesigner外のPython環境構築
- 作例でFaceTrackingのトラッキングデータから何をしているか

# 今回のサンプルデータ

---

<https://drive.google.com/drive/folders/10VwSJtl2Spesmb0H4RflxwWkbIE9TKpl?fbclid=IwAR16PErIfBRaPnkliZnZO2HuXJERU2uDm76oD7KFJfZgP8VYAfvC7JWPR4>

- **SimpleFaceTracker.zip**

FaceTrackingして顔の上に点を置くだけのシンプルなサンプル

- **TriangleFace.zip**

作例のプロジェクト

- **UseFaceOSC\_win.zip**    •    **UseFaceOSC\_mac.zip**

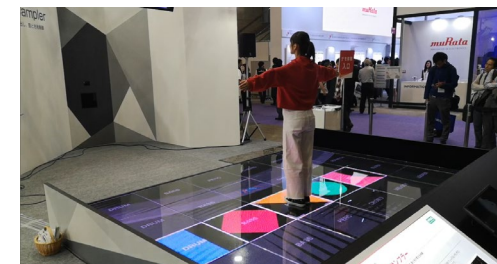
FaceOSCという外部アプリを使ってFaceTrackingするシンプルなサンプル

# 自己紹介

藤岡宏和

Twitter : [@foka22ok](https://twitter.com/foka22ok)

1-10在籍のテクニカルディレクター/エンジニア。  
インスタレーションからVR・AR、Web、機械学習など、色々やってます。



遠隔MTGに映像をつなぐ方法

## バーチャルカメラでつなぐ

---

バーチャルカメラに任意の映像を映してWebカメラとして扱うことで、好きな映像表現で遠隔MTGに参加することができます。

機材があるなら、別PCの映像出力をキャプチャーデバイスでWebカメラ扱いするのも手っ取り早い

# Windowsのバーチャルカメラ

---

- NDI Virtual Input

<https://ndi.tv/tools/>

NDI Toolsのアプリの中の一つ。対応アプリが多い。NDI outで繋がられる。おススメ。

- Spout Cam

<https://spout.zeal.co/>

対応している場合はこちらの方が良い。Zoomに限ればこれでOK。Syphon Spout outでつながられる。が、対応アプリが少ない。

- OBS Studio + Virtual Cam Plugin

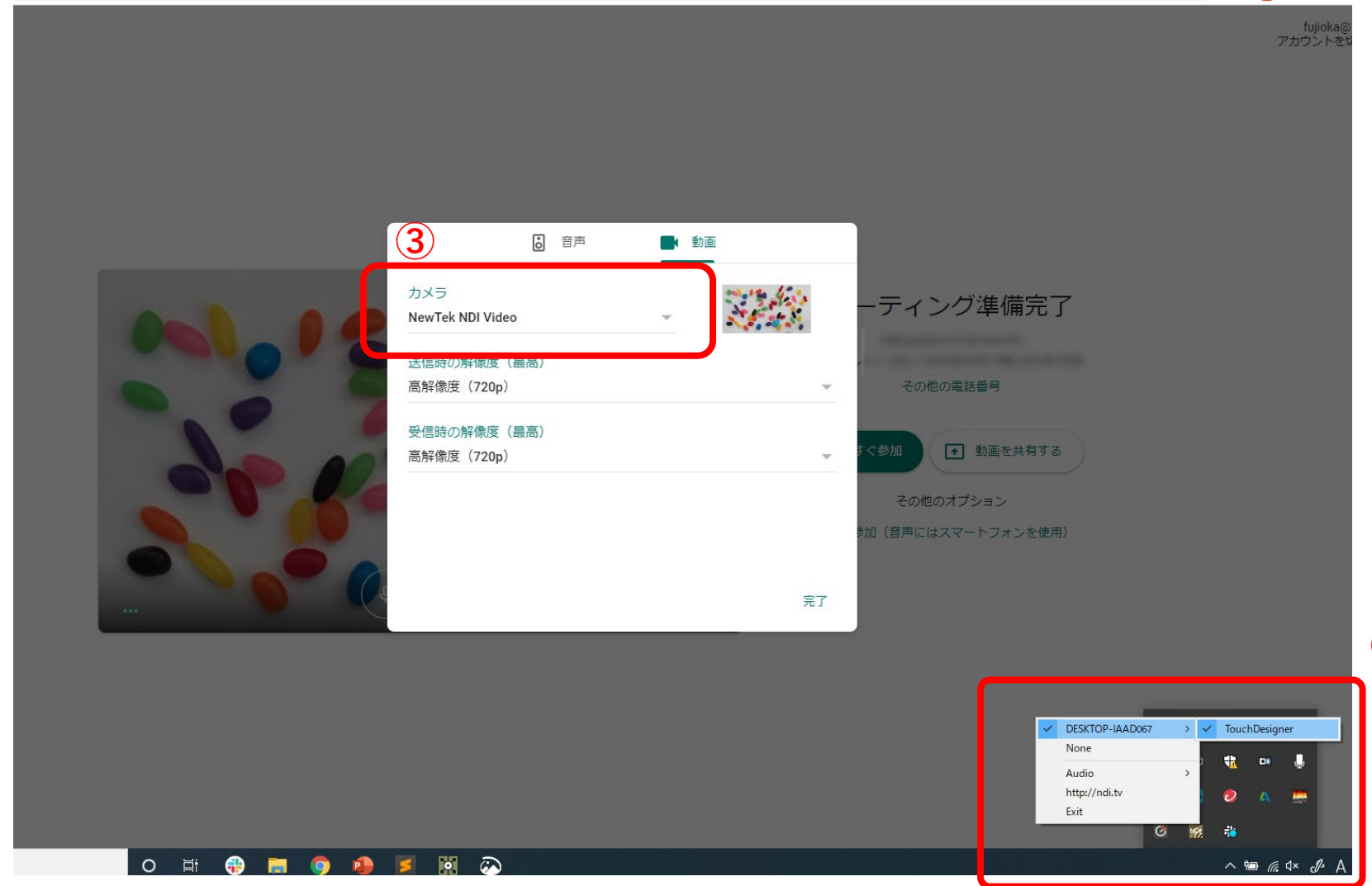
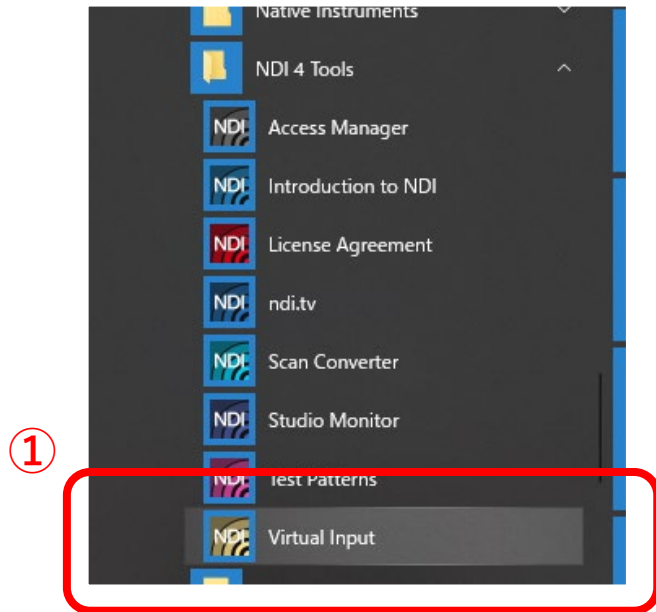
<https://obsproject.com/ja/>

本来は画面の収録・配信のためのツール。

単体では対応していないが、プラグインを追加することでバーチャルカメラとして使える。

OBS Studioに使い慣れている人は選択肢に入れても良い。

# NDI Virtual Inputの使い方





# Macのバーチャルカメラ

---

- CamTwist

<http://camtwiststudio.com/>

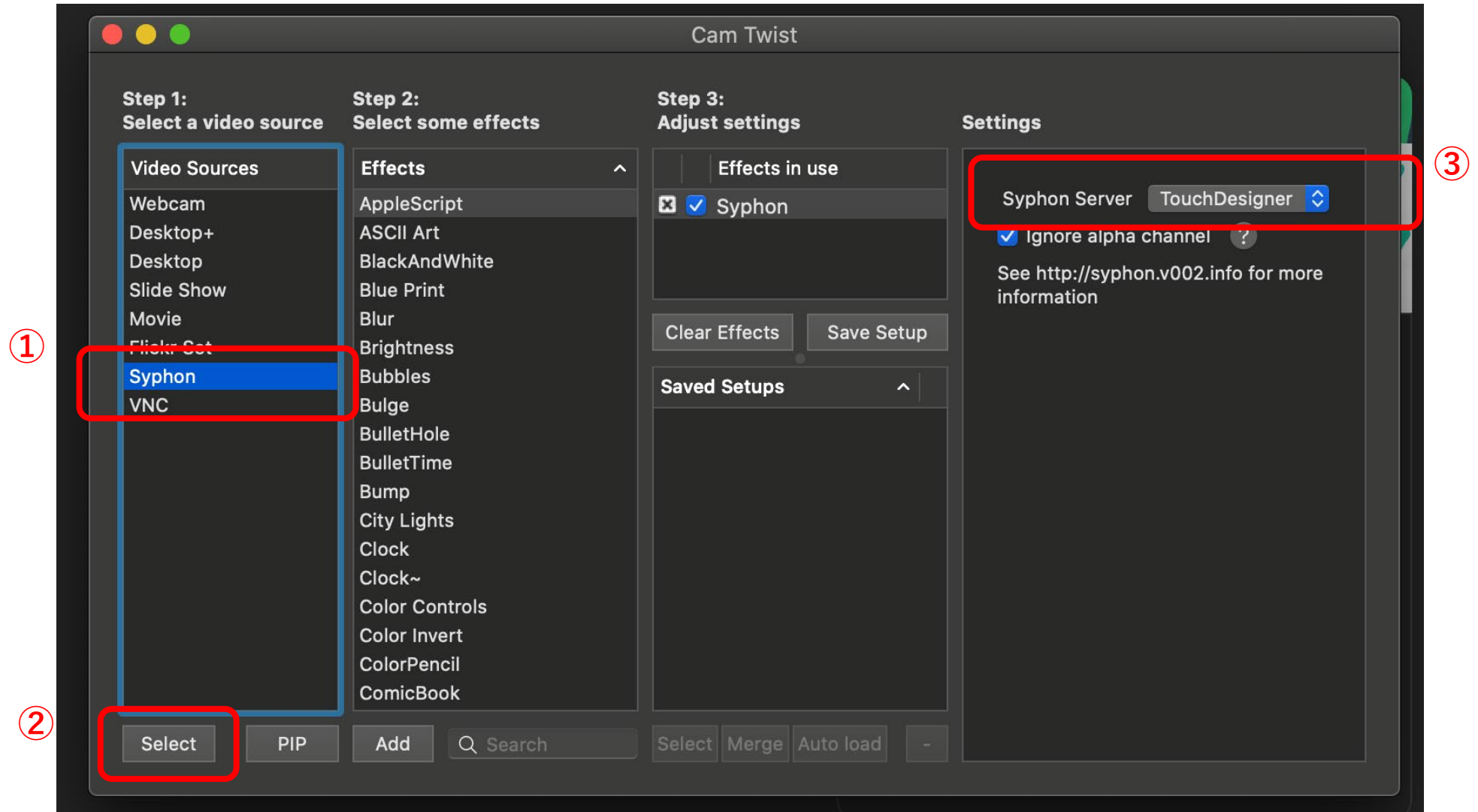
Syphonで映像を送れる。おススメ。

- ManyCam

<https://manycam.com/>

画面キャプチャをバーチャルカメラにできる

# CamTwistの使い方



# バーチャルカメラの応用

---

既存のMTGツールを利用して、リアル ⇄ VRのMTG

HangoutsにVR内から繋いでいる例。Unity使用



<https://www.youtube.com/watch?v=xVCP7Y8407g>

バーチャルカメラを使えば、なんでもアリ  
好きな映像作って参加しよう！

※ただし、MTGツールが対応している場合に限る

# Macユーザーへの残念なお知らせ

---

- ・ MacのZoom最新版はバーチャルカメラが使えなくなっている

<https://note.com/decinormal/n/n5d551b907620>

SnapCameraもCamTwistもどれも使えないので、バーチャルカメラ全部使えない気配。  
一時的なバグの可能性もあるので様子見...

Zoomのダウングレードという手段もあるそうです。

[https://qiita.com/pei\\_1227/items/71234fff6f1b4bb3cf0a](https://qiita.com/pei_1227/items/71234fff6f1b4bb3cf0a)

でも、自分のMacだとCamTwistもSnap Cameraも最新版Zoomで動いた…。

- ・ Slackアプリはバーチャルカメラが使えない

Webブラウザの方でSlack使うとMacのSlackでもバーチャルカメラが使える。

TouchDesignerでFaceTrackingしてみる

# FaceTrackingの仕方

---

TouchDesigner単体でFaceTrackingはありません  
今回はdlibを利用して、PythonでFaceTrackingを行います



<http://dlib.net/>

機械学習アルゴリズムとツールを含むモダンなC++ツールキット。 Boost Software License。  
PythonのWrapperが公式から配布されている。

dlib自体は商用非商用に関わらずどんなアプリケーションでも利用可能。

※ただし、今回利用する学習済みFaceTrackingモデルは、学習素材側に商用利用制限があるので要注意！

ちなみに、Unityのdlibアセットは別の学習素材で学習しなおしているので商用利用可能

# dlibを読み込んで使う

サンプルデータ内には、TouchDesignerのPythonのバージョン 3.7 に合わせたdlibのモジュールが含まれてます。

サンプルデータの「SimpleFaceTracker」が動く場合はそのまま使ってください。

Datでdlib 読み込んでいる箇所

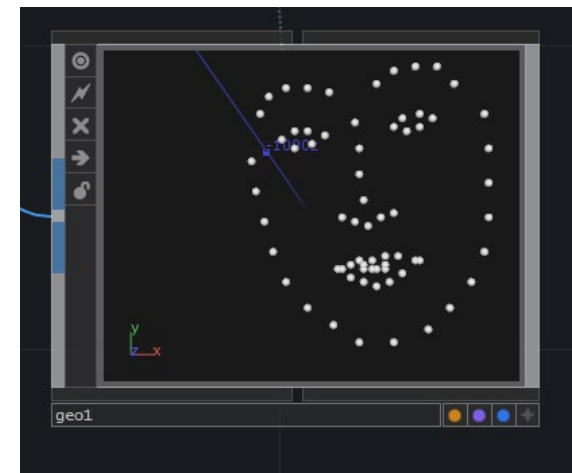
```
import sys
import os
import platform

DLIB_PATH = os.path.abspath('Lib')

if platform.system() == "Darwin":
    DLIB_PATH = os.path.abspath('python_osx')
else:
    DLIB_PATH = os.path.abspath('python_win')

if not DLIB_PATH in sys.path:
    sys.path.insert(0, DLIB_PATH)

import dlib
```





TouchDesigner外に  
Python環境構築してをdlib使う

# Anaconda

---

今回はあえてAnacondaを使った環境構築を紹介します。



<https://www.anaconda.com/>

データサイエンティストや機械学習利用者のための環境構築ソリューション

Python環境を作ったり壊したり切り替えたりが簡単。  
今回の内容では過剰な環境構築ですが…

# Python環境構築できていると何が嬉しいのか

---

TouchDesignerと機械学習の連携ができるようになります！

こんなのとか



TouchDesigner + Pose Estimation + Style Transfer

<https://www.youtube.com/watch?v=LPFGLQo1ucg>

# 機械学習と組み合わせてできそうなこと

---

Semantic Segmentationで背景演出で遊ぶとか、人物シルエットを使って遊ぶとか。

Zoomのバーチャル背景等もこれ。

ただ、MTGは固定カメラになるので、MTGに限れば背景差分の方が楽。

画像認識で、特定のものが映った時に演出変えるとか

物体検知で何かをトラッキングして演出するとか

Style Transferでエフェクト効果かけるとか

# Python環境でdlibを動かそうにしてみる

---

最新のPython3.8を使わない。3.7にすること。

- TouchDesignerのPythonが3.7
- Tensorflowが公式にはPython3.7までの対応
- Spout for Pythonのビルド済みライブラリは3.5～3.7まで  
(自力ビルドは面倒なのでビルド済をありがたく使わせてもらいましょう)

Spout for Python (派生。3.5～3.7までビルド済)  
<https://github.com/spiraltechnica/Spout-for-Python>

Spout for Python (オリジナル。3.5用)  
<https://github.com/WaltzBinaire/Spout-for-Python>  
ビルド方法を書いているので、ビルドすれば3.6～でも使える

# Anacondaにdlibが使える環境を構築する

Windowsの場合、CMakeのためにC++デスクトップ開発環境が必要  
Visual Studio Installerでインストールできます



# Anacondaにdlibが使える環境を構築する

Anacondaに環境を作る

コンソール開いて

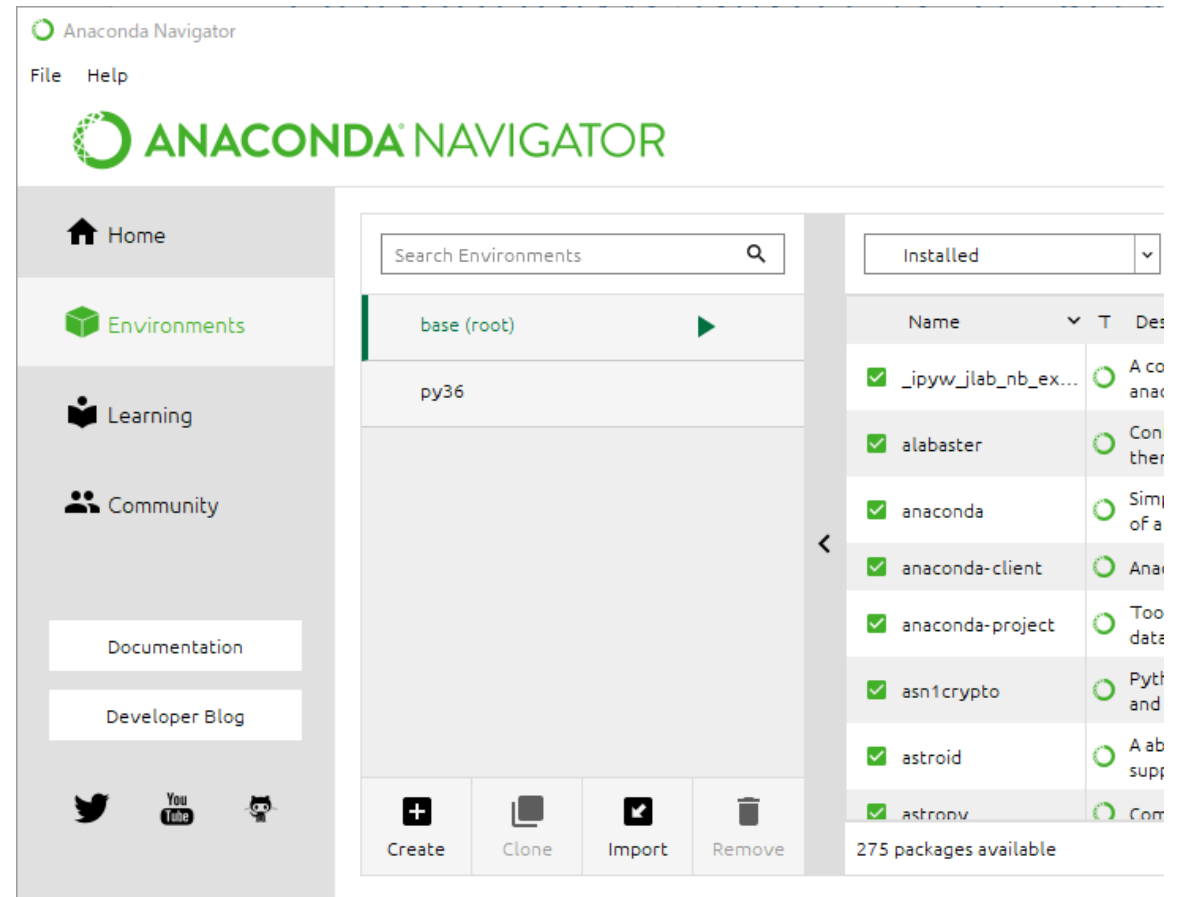
**pip install dlib**

or

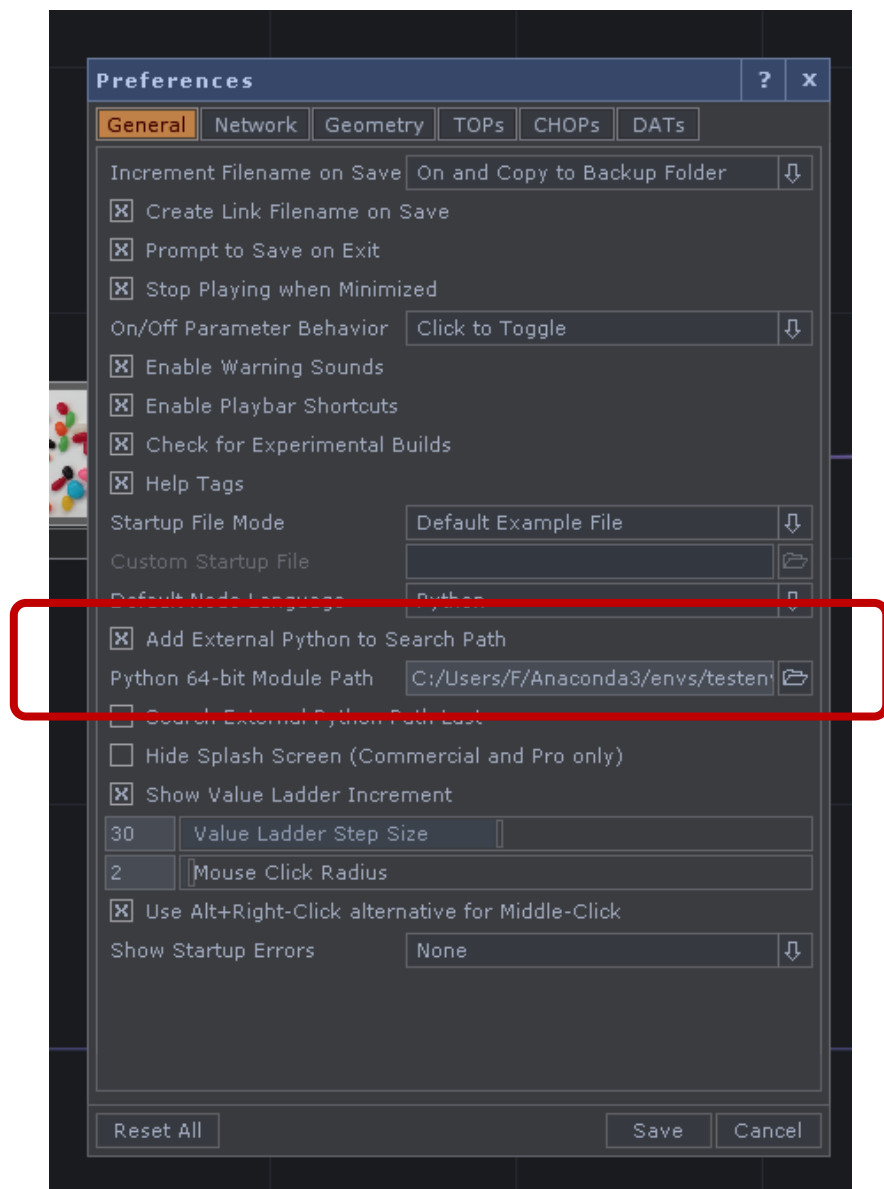
**conda install -c menpo dlib**

Cmakeが無いと怒られる場合

**pip install cmake**



# External Python Pathを指定する



ユーザー名「F」のBase環境を使う場合

**C:/Users/F/Anaconda3/Lib/site-packages**

ユーザー名「F」の「testenv」環境を使う場合

**C:/Users/F/Anaconda3/envs/testenv/Lib/site-packages**



# dlibでFace Tracking

---

dlibでFaceTrackingするためには、顔の特徴点検出モデルが必要

以下からダウンロードできます。

[http://dlib.net/files/shape\\_predictor\\_68\\_face\\_landmarks.dat.bz2](http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2)

サンプルプロジェクトのAssetフォルダにも入っているので、今回はそれを使ってください。

※今回利用する学習済みFaceTrackingモデルは、学習素材側に商用利用制限があるので要注意！

# dlib Face Trackingコード解説

```
def onFrameEnd(frame):
    arr = cameraVisionOp.numpyArray() #TopからPython NumPy配列でデータを取る
    gray = arr[:, :, 0] #白黒画像で良いので、赤色だけ抽出
    gray = gray * 255.0 #numpyArray()は0.0~1.0の少数点データになっているので0.0~255.0にする
    imgData = gray.astype(np.uint8) # 0~255の整数にする
    imgData = np.flipud(imgData) #numpyArray()はdlibと上下の正負が逆なので反転する。
    dets=detector(imgData,0) #顔の領域を検出する 第二引数はupsamplingの回数。FPSと精度がトレードオフ

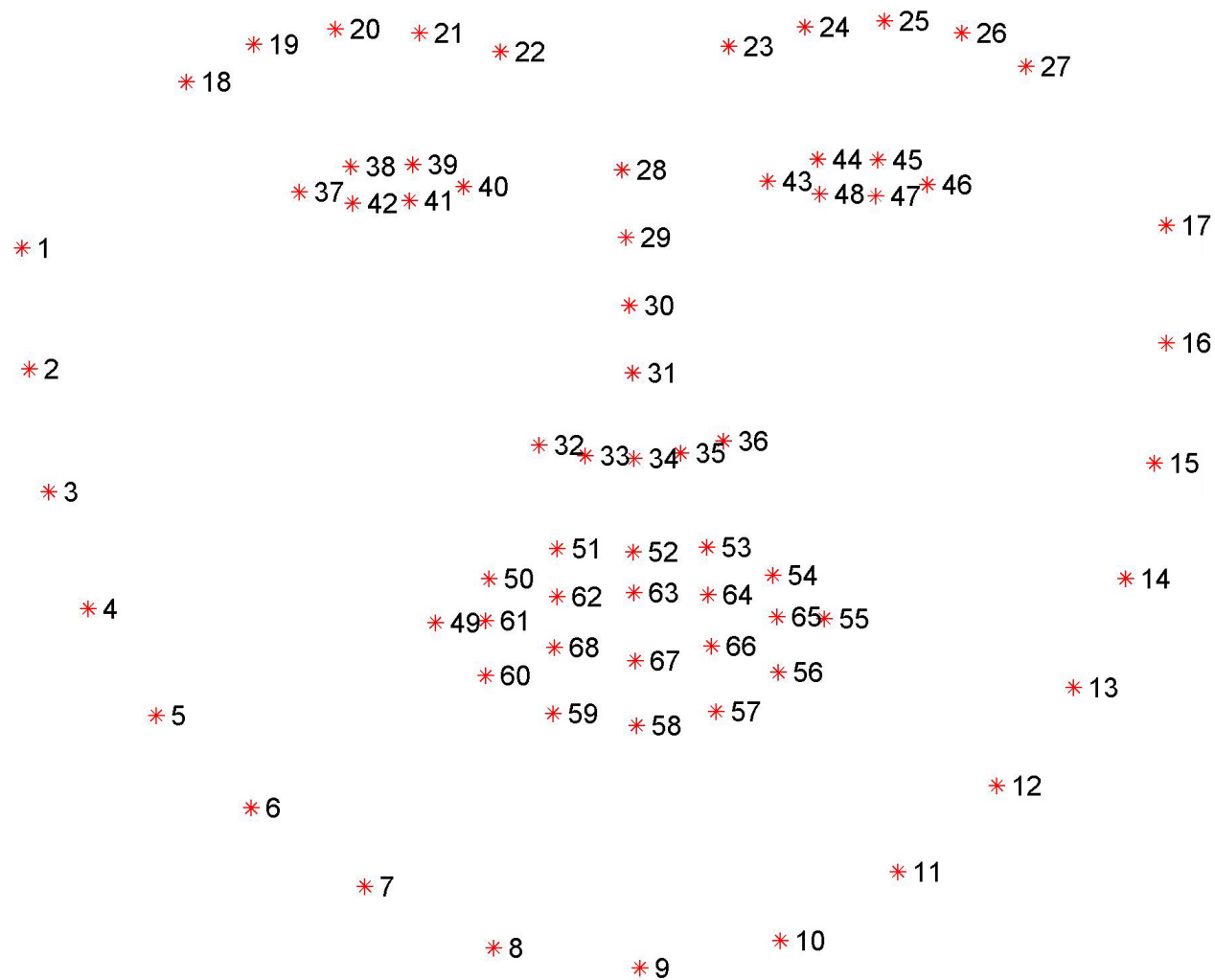
    if len(dets)>0:
        d=dets[0]

        #検出した顔の上下左右の座標
        detectTable[0,0]=d.left()
        detectTable[0,1]=d.top()
        detectTable[0,2]=d.right()
        detectTable[0,3]=d.bottom()

        #検出した顔領域から68の特徴点を取得する
        faceShape = predictor(imgData, d)
        for i in range(68):
            shapePart=faceShape.part(i)
            landmarkTable[i,0]=shapePart.x
            landmarkTable[i,1]=shapePart.y

    return
```

# dlib FaceTrackingの特徴点

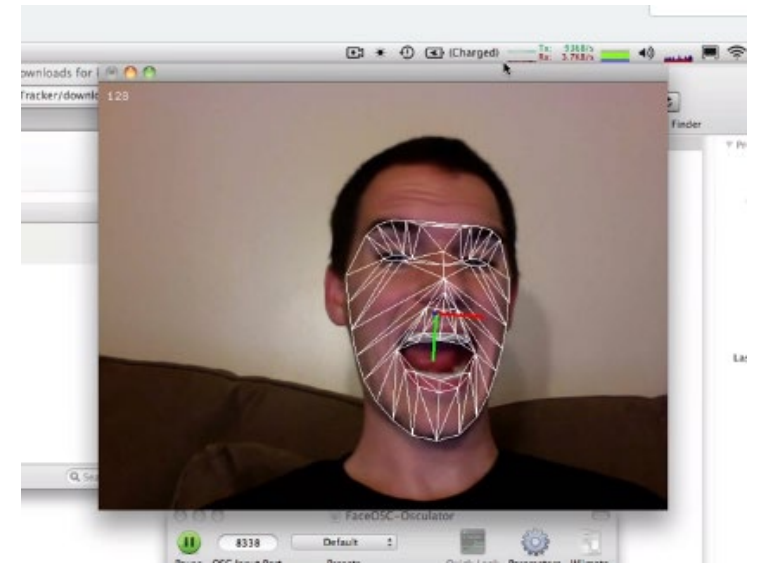


この特徴点はiBUG 300-W datasetの学習データに依存

# 外部アプリ使って環境構築無しでFaceTrackingする

## FaceOSCを使う

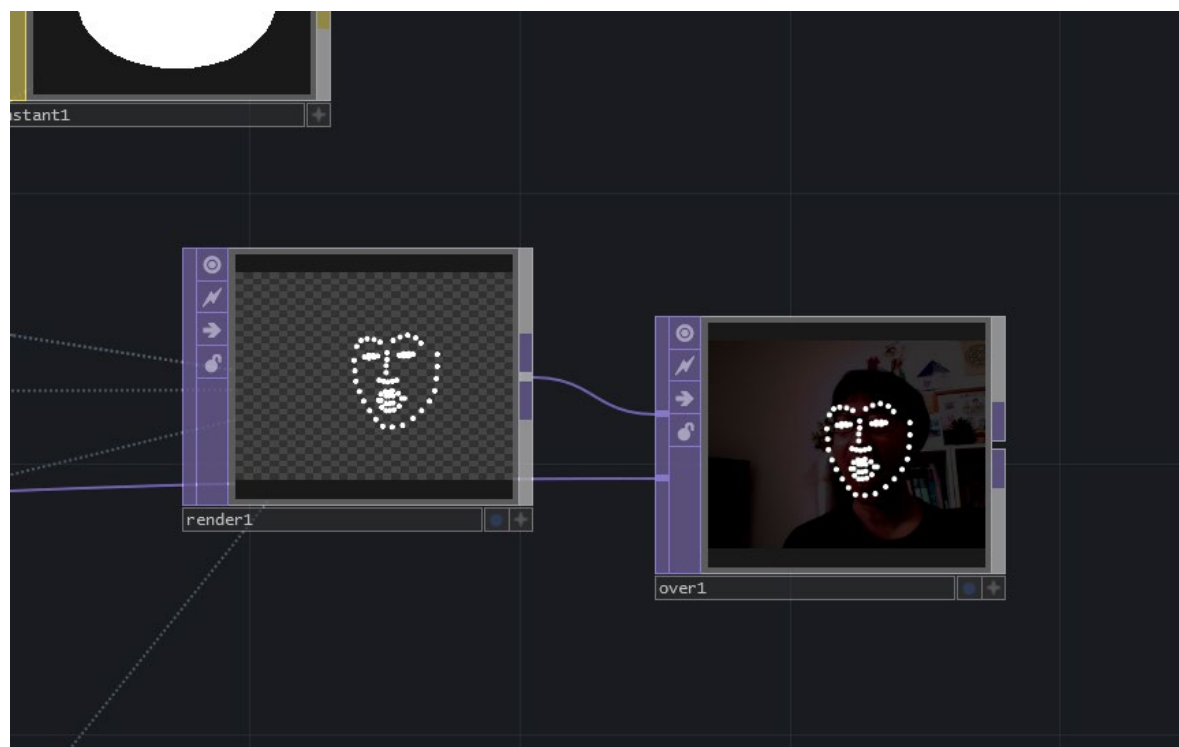
<https://github.com/kylemcdonald/ofxFaceTracker/releases>



UseFaceOSC\_win・UseFaceOSC\_macを用意しているので、  
環境構築しない場合はこちらを使ってください。

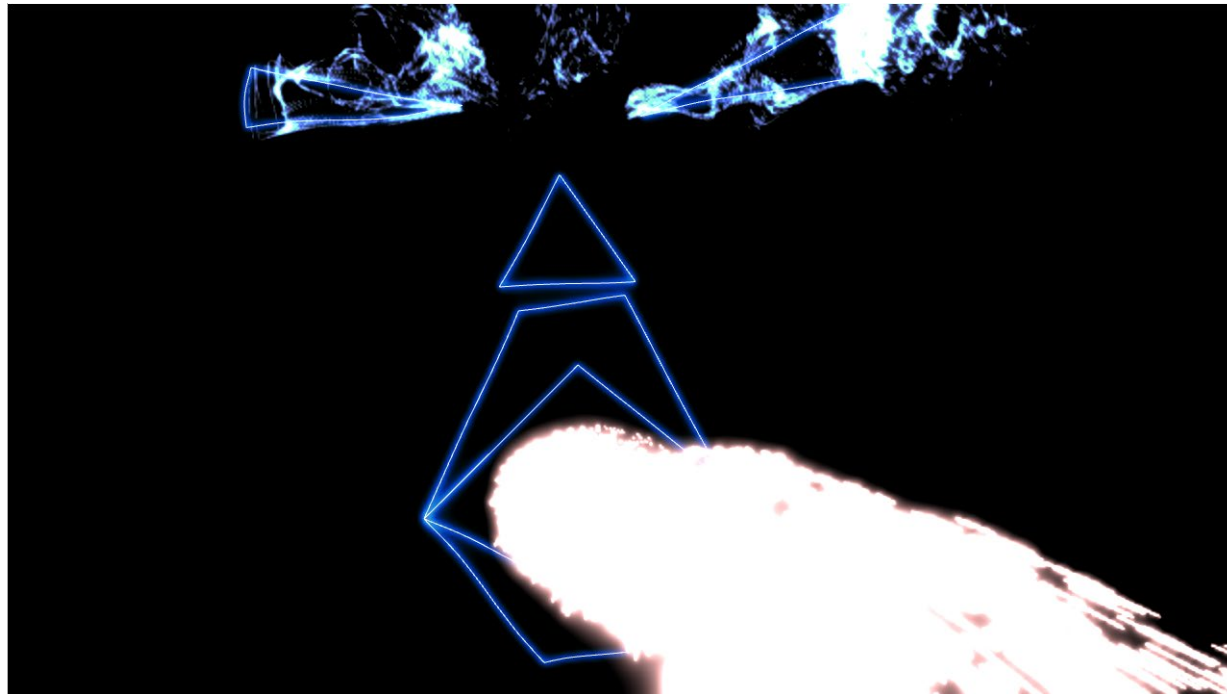
# とりあえずFaceTrackingを動かしてみる

## SimpleFaceTracker



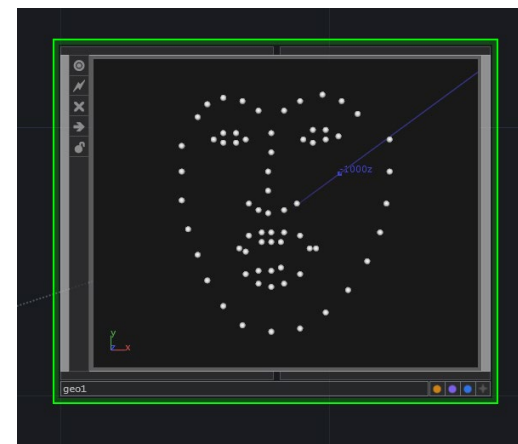
# 作例を作ってみる

## TriangleFace

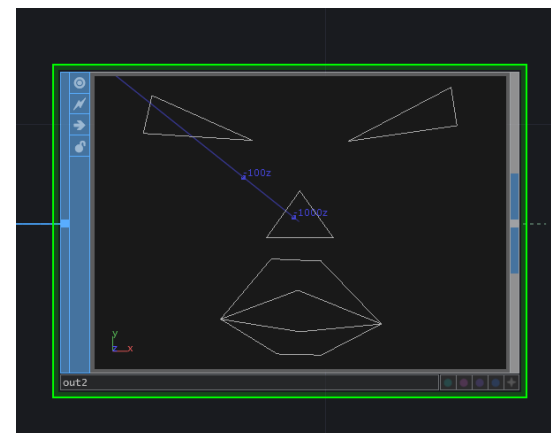


# 顔の形状を整える

- パーツに分解する



- パーツごとに形と大きさを調整する



# エフェクトをつけてみる

---

- 目を燃やしてみる
- 口を開けると火を吹く
- オーディオリアクティブにしてみる

