

ML Shukai

VRChat Recorder

5/3 Geson Anko

目次

1. 自己紹介
2. 発表目的
3. VRChat Recorderとは。
4. なんで作ったか
5. VRChat Recorderの仕組み
6. 開発秘話
7. VRChat Recorderの使い方
8. 問題と課題
9. 今後の展望

自己紹介

- げそん (GesonAnko)

Twitter@GesonAnkoVR

- ML集会 主催

自律機械知能の研究開発

PythonでML関係ツール作成

全ては自律機械知能を作るために。



目的

- VRChat Recorderをお披露目！
- 技術と作る過程の共有
- 課題と今後の展望を話し、改善に繋がりたい
- データセット作成の協力者募集

VRChat Recorderとは？

VRChatにおける入出力を
全て記録するためのツール！

なぜ、作ったか。

- 「記憶結晶」のため。

記憶結晶とは

VRChatをプレイした情報、すなわちを思い出をニューラルネットワークという流動的で絶大な許容量を持つデータ構造の中に保存し、再生可能にしたもの

→ 記憶という機能を実装するため。

記憶の再生から、今までに経験してない想像上の世界を生み出せれば、まさに思考のようだ。

自律機械知能のコアになる技術

VRChatのI/O

入力

- HMDの姿勢
- 両コントローラの姿勢、スティック、ボタン
- マイク
- ゲームパッド
- キーボード
- マウス
- OSC Inputs

出力

- 映像
- スピーカー
- OSC Feedbacks
- コントローラハップティック

VRChat Recorderで記録できるもの

入力

- HMDの姿勢
- 両コントローラの姿勢、スティック、ボタン
- マイク
- ゲームパッド
- キーボード
- マウス
- OSC Inputs

出力

- 映像
- スピーカー
- OSC Feedbacks
- コントローラハップティック

VR機器の記録方法

- OpenVR (python版)を使う。
- BackgroundアプリとしてOpenVR Systemを起動
- HMD, Controllerの姿勢を取得。(3D回転行列と位置の3x4行列)

```
vrsystem = openvr.init(openvr.VRApplication_Background)
```

```
poses = openvr.TrackedDevicePose_t * openvr.k_unMaxTrackedDeviceCount  
device_poses = poses()  
self.vrsystem.getDeviceToAbsoluteTrackingPose(openvr.TrackingUniverseStanding, 0, device_poses)
```

- Controllerのトリガー、スティックの押し込み情報取得

```
state = self.vrsystem.getControllerState(device_index)[1]  
holder.controller.left.thumb_stick = Axis(state.rAxis[0].x, state.rAxis[0].y)  
holder.controller.left.first_trigger = Axis(state.rAxis[1].x, state.rAxis[1].y)  
holder.controller.left.second_trigger = Axis(state.rAxis[2].x, state.rAxis[2].y)  
holder.controller.left.third_trigger = Axis(state.rAxis[3].x, state.rAxis[3].y)  
holder.controller.left.fourth_trigger = Axis(state.rAxis[4].x, state.rAxis[4].y)
```

← このやり方は Legacy (後述)

VR機器の記録方法

- VRイベントの取得 (これでControllerボタンイベントも取れる)

```
event = openvr.VREvent_t()
vrssystem.pollNextEvent(event)
```

イベントの一覧。(openvr.hより抜粋)

```
enum EVREventType
{
    VREvent_None = 0,

    VREvent_TrackedDeviceActivated          = 100,
    VREvent_TrackedDeviceDeactivated       = 101,
    VREvent_TrackedDeviceUpdated           = 102,
    VREvent_TrackedDeviceUserInteractionStarted = 103,
    VREvent_TrackedDeviceUserInteractionEnded = 104,
    VREvent_IpdChanged                      = 105,
    VREvent_EnterStandbyMode                = 106,
    VREvent_LeaveStandbyMode                 = 107,
    VREvent_TrackedDeviceRoleChanged        = 108,
    VREvent_WatchdogWakeUpRequested         = 109,
    VREvent_LensDistortionChanged           = 110,
    VREvent_PropertyChanged                 = 111,
    VREvent_WirelessDisconnect              = 112,
    VREvent_WirelessReconnect               = 113,

    VREvent_ButtonPress                     = 200, // data is controller
    VREvent_ButtonUnpress                   = 201, // data is controller
    VREvent_ButtonTouch                      = 202, // data is controller
    VREvent_ButtonUntouch                   = 203, // data is controller

    // VREvent_DualAnalog_Press               = 250, // No longer sent
}
```

ここがコントローライベント

映像・音声

- 映像: OBSをPythonから操作する。(obs-websocket)

```
self.client = obs.ReqClient(host=host, port=port, password=password)
self.client.start_record()

video_path = self.client.stop_record().output_path
```

- 音声: SoundCard

loop back指定でspeakerも録れる。

```
sc.get_microphone(speaker_device_name, include_loopback=True)
```

録音はシンプル

```
with self.mic.recorder(
    samplerate=self.sample_rate, blocksize=self.block_size, channels=self.num_channels
) as mic:
    data = mic.record(numframes=self.block_size)
```

その他

- OSC Feedbacks:

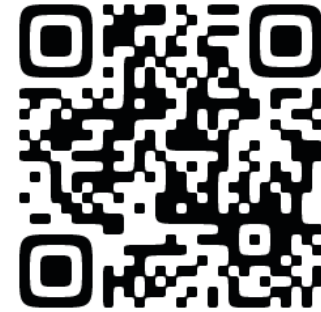
Python OSCのチュートリアルに従う。

```
dispatcher = Dispatcher()
dispatcher.map(self.address, self._osc_callback)
server = osc_server.ThreadingOSCUDPServer((self.host, self.port), dispatcher)
server.timeout = self.timeout
try:
    while self._shutdown is False:
        server.handle_request()
except KeyboardInterrupt:
    pass
```

- Gamepad:

inputsライブラリを使う。

```
events = inputs.get_gamepad()
```



- 開発初期



「VR機器の記録方法なんてわかんないよ～！
Chat GPT～！なんとかしてよ～！」



「ほい」

```
python Copy code  
  
import openvr  
import time  
  
# Initialize OpenVR  
openvr.init(openvr.VRApplication_Background)  
  
# Get the system and the compositor  
system = openvr.VRSystem()  
compositor = openvr.VRCompositor()  
  
# Main loop  
running = True  
while running:  
    try:  
        # Get the device pose
```


- 開発初期



「あーテストコード書くのめんどいな。Copilot！」



「ほい」

```
86 # format: assert unpacked_data[1] == approx(expected)
87
88 assert unpacked_data[0] == approx(123.456)
89 assert unpacked_data[1] == approx(1.0)
90 assert unpacked_data[2] == approx(2.0)
91 assert unpacked_data[3] == approx(3.0)
92 assert unpacked_data[4] == approx(0.1)
93 assert unpacked_data[5] == approx(0.2)
94 assert unpacked_data[6] == approx(0.3)
95 assert unpacked_data[7] == approx(0.4)
96 assert unpacked_data[8] == approx(-1.0)
97 assert unpacked_data[9] == approx(-2.0)
98 assert unpacked_data[10] == approx(-3.0)
99 assert unpacked_data[11] == approx(-0.1)
00 assert unpacked_data[12] == approx(-0.2)
01 assert unpacked_data[13] == approx(-0.3)
02 assert unpacked_data[14] == approx(-0.4)
03 assert unpacked_data[15] == approx(0.5)
04 assert unpacked_data[16] == approx(0.6)
05 assert unpacked_data[17] == approx(0.7)
06 assert unpacked_data[18] == approx(0.8)
07 assert unpacked_data[19] == approx(0.9)
08 assert unpacked_data[20] == approx(1.0)
09 assert unpacked_data[21] == approx(1.1)
10 assert unpacked_data[22] == approx(1.2)
11 assert unpacked_data[23] == approx(1.3)
```


VRChat Recorderの使い方 (わかる人向け)

1. 事前準備

python3.10以降 (仮想環境がおすすめ)
OBSとOBS WebSocketの設定
VRChat のOSCを有効化



2. リポジトリをクローンしてインストール!



```
git clone "<URL>"  
cd vrchat-recorder  
pip install -e .
```

VRChat Recorderの使い方 (わかる人向け)

3. Powershellから、実行！

```
1 & python -m vrchat_recorder `
2     -d "$env:USERPROFILE\Desktop\VRChat Recordings" `
3     --mic "Headset Microphone (Oculus Virtual Audio Device)" `
4     --speaker "ヘッドホン (Oculus Virtual Audio Device)" `
5     --no_gamepad ` # If you don't want to record gamepad info.
6     # --no_obs ` # If you don't want to record by OBS.
7     # --no_osc_feedback ` # If you don't want to record OSC feedback.
8     # --no_mic ` # If you don't want to record mic.
9     # --no_speaker ` # If you don't want to record speaker.
10    # --no_vr ` # If you don't want to record VR info.
```

VRChat Recorderの使い方 (わかる人向け)

3. Powershellから、実行！

```
1 & python -m vrchat_recorder `
2     -d "$env:USERPROFILE\Desktop\VRChat Recordings" `
3     --mic "Headset Microphone (Oculus Virtual Audio Device)" `
4     --speaker "ヘッドホン (Oculus Virtual Audio Device)" `
5     --no_gamepad ` # If you don't want to record gamepad info.
6     # --no_obs ` # If you don't want to record by OBS.
7     # --no_osc_feedback ` # If you don't want to record OSC feedback.
8     # --no_mic ` # If you don't want to record mic.
9     # --no_speaker ` # If you don't want to record speaker.
10    # --no_vr ` # If you don't want to record VR info.
```

使いやすさとか、そういうの、ない。

- 使いにくい。
- OpenVRはOpenXRへ…

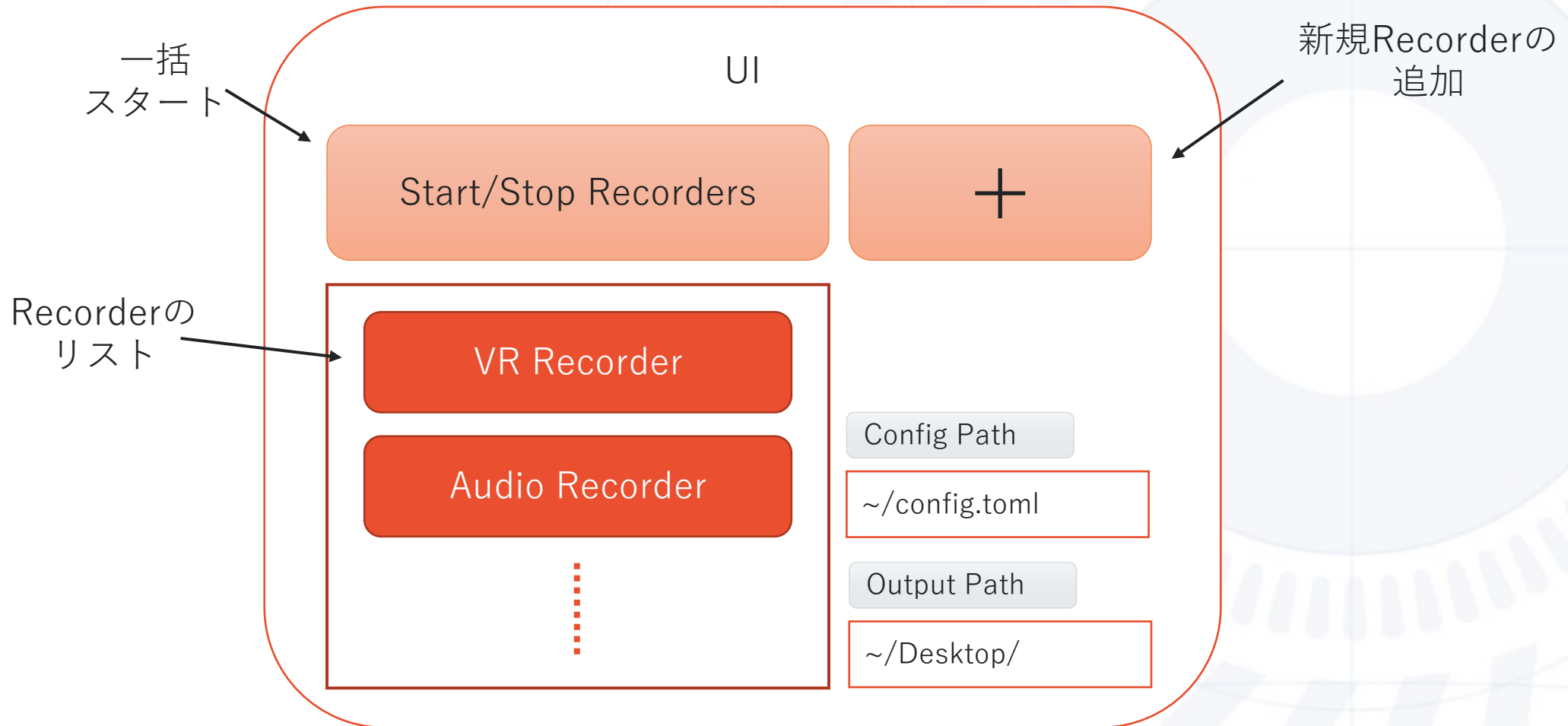
OpenXRのドキュメントは最近良くなった。(前は国語辞典)

Controllerごとにbindingの設定ファイル用意するのめんどい。

- SteamVRを落とすとPythonごと落ちる。 ←Backup処理 未実装
- 完全なVR映像をキャプチャしたい。
- VRChat起動中になぜかMouseの位置が固定化。
- ↑のおかげでKeyboardの記録処理の実装がほったらかし

今後の展望

- 使いやすく、したいね。(GUIとか)



意外とこういう類のツールは需要がありそう。

エンジニア作業飲み集会とか
OSS集会の人たちと組んでも良さげ。

- OpenXRへの移行とか（めんどい）
- デスクトップモードに対応
- Backupを取りながら記録する機能
- VR映像のキャプチャ機能

誰かやってくんないかな…

お願い

- データセット作成の協力者 募集!
 - とりあえず記録し続けていて欲しい。



參考資料

- OpenVR
<https://github.com/ValveSoftware/openvr>
<https://github.com/ValveSoftware/openvr/wiki>
<https://github.com/ValveSoftware/openvr/blob/master/headers/openvr.h>
- obsws-python
<https://github.com/aatikturk/obsws-python>
- python-osc
<https://pypi.org/project/python-osc/>
- VRChat OSC Overview
<https://docs.vrchat.com/docs/osc-overview>
- Python Inputs
<https://pypi.org/project/inputs/>
- SoundCard
<https://pypi.org/project/SoundCard/>



ML Shukai

EOF