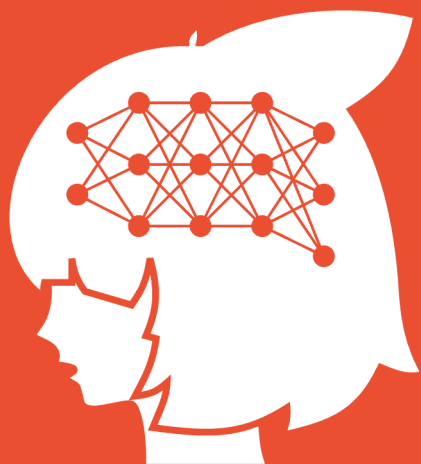




ML Shukai

Weekly P-AMI<Q>

2025/04/09 GesonAnko

自己紹介

- げそん (GesonAnko)

X(Twitter)@GesonAnkoVR

- ML集会 主催

- 自律機械知能の研究開発
- Pythonで機械学習のツール作る（機械学習より得意かもしれない）



自律機械知能に脳みそを焼かれた人。

今週の進捗



- P-AMI<Q> Core プロジェクト
 - ついに動作のコア “Thread” クラスを実装
- ライブラリ “pamiq-io” を実装
 - ぱみきゅーの入出力システムを色々まとめたライブラリ
 - 映像、音声、OSC、キーボード、マウス
- P-AMI<Q>とのコミュニケーション理論
 - 複数体の通信をどのように起こすか
 - 通信 → 情報の伝達 (コピー) → 模倣 → 思考

P-AMI <Q> Core

P-AMI<Q> Coreプロジェクト

- P-AMI<Q> Coreとは
 - 自律機械知能 “P-AMI<Q>” を実装するためのコアライブラリ
- 今週の進捗
 - 基底 Threadクラスの完成
 - **P-AMI<Q>の処理単位の基本**
 - イベントフックを定義
 - サブクラスでオーバーライド
 - 実は UnityのMonoBehaviorを参考にしている)

• アイシアさんのおかげ！



```
class Thread(PersistentStateMixin, ThreadEventMixin):
    def run(self) -> None:
        """The main method that runs the thread.
        self._logger.info(f"Start '{self.THREAD_TYPE.thread_name}' thread.")
        try:
            self.on_start()
            while self.is_running():
                self.on_tick()
                time.sleep(self.LOOP_DELAY)
            self.on_end()
        except Exception:
            self._logger.exception(
                f"An exception has occurred in '{self.THREAD_TYPE.thread_name}' thread."
            )
            self.on_exception()
            raise
        finally:
            self.on_finally()
            self._logger.info(f"End '{self.THREAD_TYPE.thread_name}' thread.")
```

```
def on_start(self) -> None:
    """Called when the thread starts."""
    pass

def on_tick(self) -> None:
    """Called on each loop iteration.

    Main processing logic should be implemented here.
    """
    pass

def on_end(self) -> None:
    """Called when the thread ends."""
    pass
```

pamiq-io

ライブラリ “pamiq-io” を実装

- “pamiq-io” とは
 - VRChatなどとの入手力システムをまとめたライブラリ
 - (以前 “vrchat-io”を作っていたが、整理し、拡張)
 - 映像: **OBS** + **OpenCV**  
 - 音声
 - **PulseAudio** (Linux サウンドサーバ) 
 - **Soundcard** (Pythonライブラリ) ← ループバックも記録できる
 - OSC: **python-osc**
 - キーボード, マウス: **inputtino** ← Python Bindingは僕が作った😊



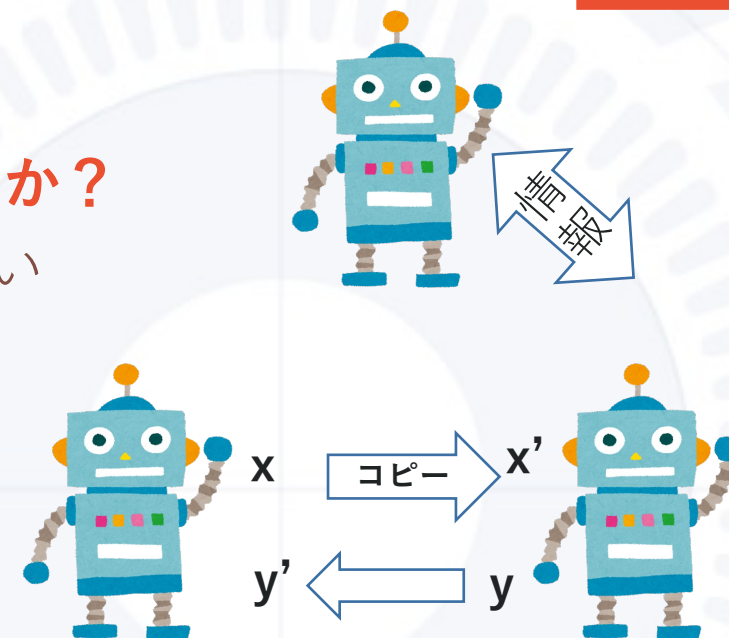
“pamiq-io”の苦しいところ

- Docker上で開発するときデバイス設定が面倒
 - “—privileged” (特権) オプションをつける
 - カメラやキーボードデバイスへのアクセスが可能になる。
- PulseAudioにDocker上からアクセスがめんどい
 - PulseAudioのソケットファイルをコンテナにバインドマウント
 - “-v \${XDG_RUNTIME_DIR}/pulse/native:\${XDG_RUNTIME_DIR}/pulse/native”
- PipeWire (別のサウンドサーバ) だとうまくいかなかったりする。
 - 新しめのLinuxは PulseAudioから移行していたりする。ぐぬぬ…
- 結局 Dockerファイルを作っても環境の再現が難しい …

$P\text{-AMI}\langle Q\rangle$ との コミュニケーション

P-AMI<Q>とのコミュニケーション

- 問
 - **どのようにして機械知能間で情報伝達を可能にするか？**
 - トップダウンにプロトコルを定めるのはセンスが悪い
 - 自発的なコミュニケーションを促したい
- 情報が伝わるとは？
 - **単純には、「コピー」**
 - 相互に情報をコピーし合えたら、それは通信では？
- 「コピー」とは、「模倣」
 - 相手の起こした現象を模倣する。



模倣について

- 模倣をどのように実現する？（どのように定義するか）
 - **✗** 「相手と同じ行動をすること」
 - 相手の行動を自分の身体の動作に変換することは難しい
 - そもそも身体の構造が違う場合は不可能
 - **✓** 「観測した現象を再現すること」
 - 相手が起こした現象を観測し、それを再現してみる
 - 説明できること
 - 犬が飼い主をスイッチを押す
 - カラスが蛇口を捻って水を出す
 - **?** 再現する対象をどのように選択する…？
 - 好奇心？ よくわからない、面白そうな観測の評価…？