

## 1 動作環境

エージェント本体は Python3 で記述しています。パッケージは numpy, scipy, pandas などを使用しています。通信などについては公開されている Harada さんの AIwolfPy のものを使用しています。

## 2 起動方法

1. AIwolf を人狼知能 HP より DL
2. 適当な場所に展開
3. StartServer.sh を実行、人数を決め、Connect をクリック
4. python3 agent\_nea.py -h localhost -p 10000 を実行

人数合わせのために他の java エージェントなどをつなぐ場合は StartGUIClient.sh を実行するとすぐにできる

## 3 メソッド

### 3.1 talk

turn 制で、生存プレイヤーが全員一斉に発話する。return で話す内容を指定する。プロトコル部門では contentbuilder のメソッドを返し、自然言語部門では文字列を返す。

### 3.2 whisper

人狼が 2 人以上生存している場合にのみ実行可能。人狼にしか送信されない talk。return で話す内容を指定する。プロトコル部門では contentbuilder のメソッドを返し、自然言語部門では文字列を返す。

### 3.3 vote

処刑対象の指定。return で投票対象のエージェント番号を指定する。

### 3.4 attack

人狼のみ可能なメソッド。return で攻撃先のエージェント番号を指定する。

### 3.5 divine

占い師のみ可能なメソッド。return で占い先のエージェント番号を指定する。

### 3.6 guard

狩人のみ可能なメソッド。return で護衛先のエージェント番号を指定する。

### 3.7 initialize

ゲームの初期化時に呼ばれるメソッド。

### 3.8 update

更新情報がある時に呼ばれるメソッド。Java 版と挙動が異なるため注意。daily\_finish や request を内包する。

### 3.9 dayStart

前日のいろいろな結果が来るメソッド。

### 3.10 finish

ゲーム終了時に呼ばれるメソッド。何故か 2 回呼ばれるらしい。

## 4 データ構造

### 4.1 base\_info

辞書型で基礎的な情報すべてを保持している。initialize や update 時に渡される。

agentIdx 自身のエージェント番号

myRole 自身のエージェントの役職

roleMap 役職の dict。他のエージェントの役職が確定的にわかる場合 (人狼の場合など) はここに入る

statusMap エージェントの静止を表す dict

remainTalkMap 生存エージェントのその日の残り発話可能回数を表す dict

remainWhisperMap Agent のその日の残りの可能な囁き回数を表す dict

## 4.2 diff\_data

pandas DataFrame で、ゲームに関わる情報を保持している。6 列存在し、agent, day, idx, text, turn, type がある。type によって内容の意味が異なる。

### 4.2.1 type = initialize, finish

agent = idx = agentIdx (自分自身)  
initialize の場合は day = 0  
turn = 0  
text = comingout 文

### 4.2.2 type = talk, whisper

agent = 発話者  
day = 日付  
idx = talk/whisper の id  
turn = talk/whisper の turn  
text = 発話文そのまま

### 4.2.3 type = attack\_vote, vote

agent = 投票対象  
idx = 投票者  
turn = 0 再投票時は-1  
text = vote/attack 文

### 4.2.4 type = execute, dead

agent = 死者  
idx = turn = 0  
text = Over

### 4.2.5 type = divine, identify, guard

agent = 能力の対象  
idx = 能力使用者  
turn = 0  
text = devined/identified/guarded 文

### 4.2.6 type = attack

agent = 襲撃対象  
idx = turn = 0  
text = attack 文