

# werewolf

---

## directory

```
.
├── README.md
├── classes
│   ├── abst_classes
│   │   ├── action_abst.py
│   │   ├── knowledge_abst.py
│   │   └── role_abst.py
│   ├── roles
│   │   ├── __init__.py
│   │   ├── villager
│   │   │   ├── __init__.py
│   │   │   ├── action.py
│   │   │   ├── knowledge.py
│   │   │   └── role.py
│   │   └── wolf
│   │       └── __init__.py
│   └── util.py
├── game
│   ├── __init__.py
│   ├── master.py
│   ├── player.py
│   └── reception.py
└── main.py
```

---

## main.py

1. masterの起動
2. reception をdemonでうごかす
3. あとは無限ループ

---

## ./game

- ゲーム運営のための人たち

## reception.py

- 受付の人
- 基本的にncのコネクションを待ちつつ、接続があれば`ReceptionThread`につなぐ
- demonで動かしておく

## master.py

- ゲームマスターの人
- ゲームの管理、全体へのアナウンスなど
- みんなが参照するデータ`GlobalObject`を持つてる

1. プレーヤの準備を待つ
2. 役職選択の開始を通知
3. 役職選択の終了待ち
4. 0日目の処理トリガー (未実装)
5. 以降、n日目の処理をトリガー (集計とかもやる)

## player.py

- 各プレーヤの代理人みたいな感じ
  - ユーザとの情報のやりとりを担当
1. ゲーム中ならごめんねして終了
  2. 1人目ならなんでプレーかを聞く
  3. 名前を聞く
  4. プレーヤの準備を待つ
  5. 役職の選択
  6. ゲームマスターからのトリガーを受けて、行動や情報提供を行う
    - 何をやるかは各役職クラスに丸投げ

---

## ./classes/abst\_classes

- 役職、行動、知識用の抽象クラス
- インターフェースを揃えるために一応用意
- 各役職クラスの実装で利用

---

## ./classes/util.py

- いろんなところで使う定数を定義
- `TIME_OF_DAY`: 朝昼晩
- `WIN_CONDITION`: 勝利条件

---

## ./classes/roles

- 役職ごとにフォルダを作って、その中に`role.py`, `action.py`, `knowledge.py`を作る
- `__init__.py`を入れることで勝手に選択画面に出てくるよう細工

## ./classes/roles/.../action.py

- 行動を定義
- `role.py`で利用
- 朝昼晩での行動を定義しておいて、日にちに応じてアクションを変えたらいいなってる
- 行動に応じて、`PlayerThread`のソケットを利用して、ユーザと通信したらいいと思う

## ./classes/roles/.../knowledge.py

- 得られる知識を定義
- `role.py`で利用
- 同じく、朝昼晩での知識を定義しておいて、日にちに応じて返り値を変えれたらいいなってる
- 知識は`PlayerThread`のソケットを利用して、ユーザに送れそう

## ./classes/roles/.../role.py

- 役職のクラス
- 役職名
- `action.py`, `knowledge.py`のインスタンスを持つ
- 以下を持つ
  - 勝利条件
    - どの条件で勝つかを持っておく
  - 0日目の行動
    - 他の人狼が誰とかをmaster経由で調べて通知
  - 朝,昼,晩の行動
    - `action.py`のインスタンスを使って対応 (丸投げ)
  - 朝,昼,晩に得られる知識
    - `knowledge.py`のインスタンスを使って対応 (丸投げ)