

## プレイヤークラス作成マニュアル

2014年6月14日

鳥海不二夫

### 1. プレイヤークラス

プレイヤークラスは、ゲームに参加するプレイヤー部分である。人狼において最も重要なクラスとなる。情報参照、対話、投票、占い対象の決定など全てをこのクラス内で行う。

Ver0.1.4からは各職業ごとに Player を作成することが可能となる。詳細は、5.RoleBasePlayer 参照のこと。

### 2. プレイヤークラスの作成条件

基本的には、プレイヤークラスは、org.aiwolf.common.data.Player インターフェースを実装することで作成される。

ただし、引数無しのデフォルトコンストラクタが存在しなければならない。

### 3. プレイヤークラス内で設定すべきメソッド

getName: Player の名前。Player 名は自由に設定可能。なお、ゲームには直接利用されず、単に表示に利用するだけである。

update: 現在のゲームの状況が変更されたときに呼ばれる。

initialize: ゲーム開始時に呼ばれる

dayStart: 一日の始まりに呼ばれる

talk: 発言を行うためのクラス。発言を返すメソッド。返された発言はそのまま他のエージェントに送られる。Talk の内容は org.aiwolf.client.lib.TemplatetalkFactory を利用して作成する。

whisper: プレイヤの役割が人狼だった場合に、発言するささやきを定めるメソッド。

vote: 一日の終わりに追放する Agent に投票するメソッド

attack: 人狼だった場合に、襲う Agent を指定するメソッド

divine: 占う対象を指定するメソッド

guard: 守る対象を指定するメソッド

finish: ゲームが終了した際に呼ばれるメソッド。このとき、全 Agent の職業が明かされる

### 4. GameInfo

ゲームの情報をまとめたクラス。

ゲームの状況を取得するために利用する。

詳細は JavaDoc 参照のこと。

### 5. RoleBasePlayer の作成

Ver0.1.4 から与えられた役割ごとに Player を作成することが可能になった。

各 Role ごとに実装すべきクラスはそれぞれ、

org.aiwolf.client.base.player.AbstractVillagerPlayer : 村人

org.aiwolf.client.base.player.AbstractSeerPlayer : 占い師

org.aiwolf.client.base.player.AbstractMediumPlayer : 霊媒師

org.aiwolf.client.base.player.AbstractBodyGuardPlayer : ボディーガード

org.aiwolf.client.base.player.AbstractPossesedPlayer : 狂人

org.aiwolf.client.base.player.AbstractWereWolfPlayer : 人狼

である。これらの Player を実装することで、各役職を与えられたときにのみ実行される Player が作成できる。

これらの Player を使って人狼を実行する場合は、RoleBasePlayer を継承したクラスを作成する。

下記に My~Player を作成し、それらを利用して実行するためのクラス作成例を示す。

```
import org.aiwolf.client.base.player.RoleBasePlayer;
public class MyPlayer extends RoleBasePlayer {
    public MyPlayer () {
        setVillagerPlayer (new MyVillagerPlayer ());
        setSeerPlayer (new MySeerPlayer ());
        setMediumPlayer (new MyMediumPlayer ());
        setBodyGuardPlayer (new MyBodyGuardPlayer ());
        setPossesedPlayer (new MyPossesedPlayer ());
        setWerewolfPlayer (new MyWereWolfPlayer ());
    }
    @Override
    public String getName () {
        return MyPlayer.class.getSimpleName ();
    }
}
```

なお、作成していない職業があれば、何も set しないことで SamplePlayer が自動的に利用される。そのため、一部職業だけ作成してテストすることも可能である。