

## AIWolf ゲームサーバ起動方法

2014 年 6 月 14 日

鳥海不二夫

### 1. AIWolf ゲームサーバに必要なライブラリ

- aiwolf-server-0.1.x.jar
- aiwolf-common-0.1.x.jar
- aiwolf-client-0.1.x.jar
- jsonic-1.3.2.jar

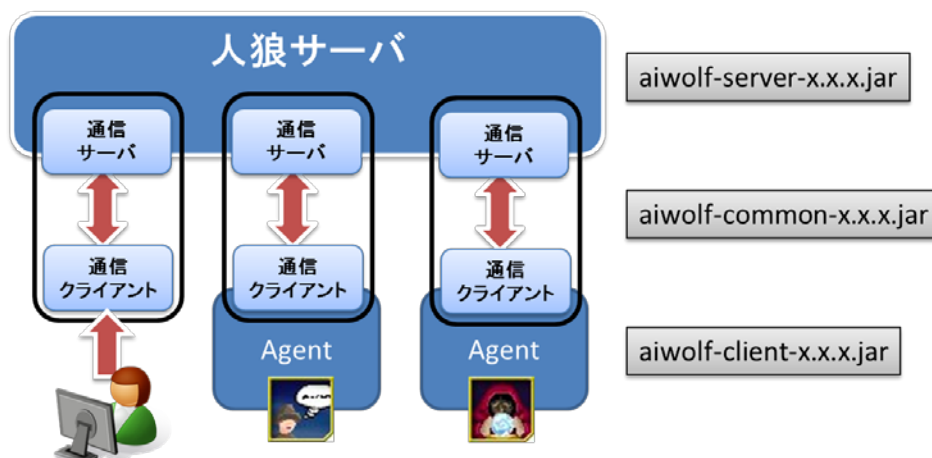
### 2. AIWolf ゲームサーバの基本構成

人狼サーバの基本構成は以下のようになっている。

まず、ゲームマスターとして人狼サーバが存在する。ゲームサーバに参加者が作成したエージェントを接続し、ゲームを行う。人狼サーバを起動するためのプログラム群が aiwolf-server-x.x.x.jar である。なお、サーバには通信サーバも含まれる。

一方、参加者がエージェントを作成するためのプログラム群が aiwolf-client-x.x.x.jar である。この中には通信クライアントとエージェント作成のためのライブラリが含まれている。

また、両方で共通して利用するクラス群が aiwolf-common-x.x.x.jar に含まれる。



参加者は aiwolf-common-x.x.x.jar, aiwolf-client-x.x.x.jar を利用してエージェントプログラムを作成する。エージェントの作成については、エージェントの作成.pdf を参照のこと。

### 3. AIWolf ゲームサーバの起動(直接接続編)

AIWolf ゲームサーバは、基本的にサーバクライアントシステムではあるが、通常のテスト時にはサーバクライアント型ではなく、サーバとクライアントを同時に立ち上げ通信を行わない設定で起動可能

である。

```
> java -cp aiwolf-client-0.1.x.jar;aiwolf-common-0.1.x.jar;aiwolf-server-0.1.x.jar
DirectStarter -c org.aiwolf.client.base.smpl.SamplePlayer 12 -l log
```

ここでは、SamplePlayer を 12 体使ってゲームを開始している。

基本的には、client, common, server の各 jar ファイルを classpath に追加した上で、DirectStarter クラスを起動すれば良い。

Option の設定は以下の通り。

-c Player クラス名 当該クラスを使った Player 数 : プレイヤクラスとプレイヤ数を決める。

-l ログディレクトリ : ゲームログを保存するディレクトリを決定する。デフォルトは./log/

-c の指定はいくつでも可能である。例えば、MyPlayer を作成して、一体だけ参加させたい場合は、

```
> java -cp aiwolf-client-0.1.x.jar;aiwolf-common-0.1.x.jar;aiwolf-server-0.1.x.jar
DirectStarter -c org.aiwolf.client.base.smpl.SamplePlayer 11 -c MyPlayer 1 -l log
```

とすればよい。

#### 4. AIWolf ゲームサーバの起動(職業指定型起動編)

エージェントを作る際に、デバッグのため特定のエージェントに特定の職業を割り当てたいことがある。指定したクラスのエージェントを指定した職業に就けるための起動方法が以下の通りである。

ここでは、org.aiwolf.sample.MyPlayer を占い師(seer)として起動し、そのほかの職業はすべて SamplePlayer によって構成させるようにする。なお、エージェント数は 12 とする。

```
> java -cp aiwolf-client-0.1.x.jar;aiwolf-common-0.1.x.jar;aiwolf-server-0.1.x.jar
RoleRequestStarter -n 12 -c org.aiwolf.sample.MyPlayer seer -d
org.aiwolf.client.base.smpl.SamplePlayer -l log
```

起動クラスは、RoleRequestStarter である。「-n 12」オプションは全部で 12 体のエージェントを登録することを指す。「-c クラス名 役割名」オプションで、MyPlayer クラスのインスタンスを seer として登録することを指定する。なお、-c オプションは複数使うことが可能であり、複数のエージェントそれぞれに異なる職業を割り当てることが出来る。

「-d クラス名」オプションは、指定した-c で指定した以外のエージェントのクラスである。今回の場合は、1 対の MyPlayer と 11 体の SamplePlayer によって起動する。なお、-d オプションを省略した場合は、自動的に SamplePlayer が残りのエージェントを担当することになる。なお、職業として

指定可能なものは以下の通りである.

- villager (村人)
- werewolf (人狼)
- seer (占い師)
- medium (霊媒師)
- Bodyguard (狩人)
- possessed (狂人)

## 5. AIWolf ゲームサーバの起動(サーバクライアント編)

AIWolf をサーバクライアントシステムで起動する方法を説明する. 基本的にはエージェント開発時にこのモードを使う必要は無い.

ゲームサーバ起動には,

- aiwolf-server-0.1.x.jar
- aiwolf-common-0.1.x.jar
- aiwolf-client-0.1.x.jar
- jsonic-1.3.2.jar

が必要となる.

### I. ゲームサーバの起動

AIWolf ゲームサーバの起動には, aiwolf-server-0.1.x.jar を使う.

```
> java -cp aiwolf-server-0.1.x.jar:aiwolf-common-0.1.x.jar:jsonic-1.3.2.jar
ServerStarter -p port -n playerNum
```

たとえば, port:10000 を使って 12 人でプレイすることを考えると,

```
> java -cp aiwolf-server-0.1.x.jar:aiwolf-common-0.1.x.jar:jsonic-1.3.2.jar
ServerStarter -p 10000 -n 12
```

となる.

立ち上げると

Waiting for connection...

と表示されるので, プレイヤを立ち上げる.

### II. AIWolf プレイヤー起動方法

プレイヤーの起動には, aiwolf-client-0.1.x.jar を使う.

```
> java -cp aiwolf-client-0.1.x.jar:aiwolf-common-0.1.x.jar:jsonic-1.3.2.jar  
ClientStarter -h host -p port -c className
```

で, className のプレイヤーが起動する.

ただし, className は org.aiwolf.comon.data.Player を実装したクラスである必要がある.

```
> java -cp aiwolf-client-0.1.x.jar:aiwolf-common-0.1.x.jar:jsonic-1.3.2.jar  
ClientStarter -h localhost -p 10000 -c org.aiwolf.client.base.smpl.SamplePlayer
```

Server 起動時に指定した分だけ Client が接続されればゲームが開始される.

Player の作り方はまたエージェントの作成.pdf 参照.

### III. JSONIC について

jsonic に関する詳細は <http://jsonic.sourceforge.jp/> を参照のこと.