

人狼知能Ver.0.4.x 仕様書

2016/11/07 Ver1.00

1. 人狼のルール

1.1. 各役職の人数

15人または5人のプレイヤーで村を構成します。

15人

- ・村人：8人
- ・占い師：1人
- ・霊能者：1人
- ・狩人：1人
- ・人狼：3人
- ・裏切り者：1人

5人

- ・村人：2人
- ・人狼：1人
- ・裏切り者：1人
- ・占い師：1人

2. メソッド

2.1 昼の処理

トーク仕様

- ・0日目のTalkは無し。(GameSettingで0日目Talkありかどうかは設定可能)
- ・同時発話について
 - ・全エージェントのtalkを呼び出す。
 - ・各エージェントはTalkを返すか、skipを返す。
 - ・Talkが返されていたら、順番をランダムにして、Talkリストを作成する。
 - ・Talkにturnフィールドを追加する
 - ・skipは発話に含まない
 - ・全員がOverを返したら、昼の会話フェーズ終了
 - ・最大で10回の発話が可能(GameSettingで設定可能)
 - ・10回までしゃべったら自動でSkipになる
 - ・全員Skipが10回続いたら終了(GameSettingで設定可能)

TalkObjectの仕様について

int idx;

int day;

Agent agent;

String content;

int turn; //これが等しいものは同時に発話された物になる。

2.2 夜の処理

vote()

その日に投票する対象プレイヤーを返します。

最大得票プレイヤーが二人以上いた場合はもう一度投票。それで決まらなかった場合はランダム。

GameSettingで、最大再投票回数と、決まらなかった場合の挙動(ランダム or 処刑なし)を決定することが出来る。

Voteの結果はすぐに算出して、その後のAttack,Guard,Divine時には情報が提示される。

再投票の場合は最大得票数を獲得したエージェントのみが投票対象となる。

それ以外は無効票となる。最大得票を獲得したエージェントは投票権利を持たない。

whisper()

Voteの結果算出後に呼ばれる。Whisperの会話の仕様はTalkに準ずる。

Attack先の相談及び、ComingOutの宣言を行うのが一般的だろう。

0日目にWhisperはある。

attack()

人狼のプレイヤーのみ呼ばれるメソッド。その日に襲撃投票する対象プレイヤーを返します。

Voteの結果を利用することができる。

複数回AttackVoteの仕様はGameSetting設定可能。

再度Whisperも可能 (GameSettingで設定可能)

Attackは襲撃対象となったエージェントになる。(誰を襲撃したのかが分かる。ランダム襲撃で失敗した時に誰を襲撃したのかが分かる)

ただし、実際に襲撃されたかどうかは、deadAgentListに記載されることになる。

Attackedの仕様が変更されることに注意。

guard()

狩人のプレイヤーのみ呼ばれるメソッドです。その日に護衛する対象プレイヤーを返します。

Voteの結果を利用できる

divine()

占い師のプレイヤーのみ呼ばれるメソッドです。その日に占う対象プレイヤーを返します

.

Voteの結果を利用できる

3. 役職について

- ・ FREEMASONは削除？
- ・ 妖狐を導入？
 - ・ 妖狐導入の場合は、GameInfoのAttackedをdeadAgentListに変更する必要がある
 - ・ Teamに妖狐を導入
- ・ GameSettingに基づいて存在するRoleのみを返すメソッド
Role.getExistingRoles()

4. その他

- ・ Agent番号が0から始まるようにする
- ・ Talk内容のチェックをサーバ側で行う
 - ・ 違反しているような発話が行われた場合はnullを返す
 - ・ GameSettingで設定可能
- ・ GameSettingに制限時間を設定できるようにする
- ・ TalkFactory, Parserの仕様変更
 - ・ TemplateTalkFactoryをdeprecatedに
 - ・ 新しいContentBuilderクラス
 - ・ BuilderクラスによるコンストラクタをContentクラスに作る
 - ・ Talkのcontentをtextに変更
 - ・ UtteranceをContentクラスに変更
- ・ ログの仕様変更
 - ・ Talkの仕様
 - ・ ログの吐き出し方->StringでAIWolfGameで直接ログを書くのはやめる
 - ・ ログをどこで出しているのか分からないのは気持ち悪い
- ・ executed=>Banished

5. 検討事項

- ・ ライセンスについて

5.3. 発話可能な内容

本大会では, TemplateTalkFactory, TemplateWhisperFactoryで生成可能な発話のみで会話をを行います. 生成可能な発話は以下の11つです.

- ・ estimate : プレイヤーAの役職は○○だと思う.
- ・ comingout : 私の役職は○○だ.
- ・ divined : プレイヤーAを占った結果, ○○ (人間or人狼) だった.
- ・ inquested : プレイヤーAは霊能の結果, ○○ (人間or人狼) だった.
- ・ guarded : プレイヤーAを護衛した.
- ・ vote : プレイヤーAに投票する.
- ・ attack : プレイヤーAに襲撃投票する. (TemplateWhisperFactoryのみ)
- ・ agree : 発話Tに同意する.
- ・ disagree : 発話Tに反対する.
- ・ request :
- ・ over : もう話すことは無い. (全プレイヤーがOVERなら会話フェーズ終了)
- ・ skip : 様子見 (他のプレイヤーが全員OVERでも会話フェーズが終了しない)

5.4. Playerクラスのパッケージについて

Playerクラスは独自の物を作成してください. デフォルトである
org.aiwolf.player.RoleAssignPlayer
などを直接書き換えることは避けてください.

Playerクラスは、独自パッケージに配置するようにしてください。通常パッケージはドメイン名またはメールアドレスを逆から利用して作成します。

たとえば、gm@aiwolf.orgというメールアドレスをお持ちの方がMyPlayerというPlayerクラスを作成する場合、パッケージとしてorg.aiwolf.gmを指定し、
package org.aiwolf.gm;

```
public class MyPlayer extends AbstractRoleAssignPlayer {  
  
}
```

という書き方になります。

また、ソースコード提出時の「class path:」の欄には、
org.aiwolf.gm.MyPlayer
と指定してください。

6. 他の言語で参加する場合

Java以外の言語はPython, .NETのみ想定しています。これらの言語で参加する場合は、それぞれのライブラリを参照して下さい。独自のライブラリでもTCP-IP接続が出来、エージェントが正しく動けば参加することが出来ます。

ただし、ゲームはLinuxマシンで動きますので、その点を注意して特殊な環境に特化したプログラムをご遠慮下さい。ゲームサーバ上で動かなかったエージェントは自動的に失格となります。

参加されたい方は事前に運営側にご相談(gm@googlegroups.com)ください。

7. 更新

レギュレーションは予告なく変更される可能性があります。変更された場合はプロジェクトページ(<http://aiwolf.org>), 開発者メーリングリスト(aiwolfdev@googlegroups.com)あるいはTwitterアカウント(@aiwolf_org)で告知いたしますので、登録あるいはフォローをしておいてください。

8. 更新履歴

2016/11/07 ver1.00作成