Table des matières

[1. Projet 2](#_Toc485063106)

[2. Architecture 2](#_Toc485063107)

[2.1. Architecture générale 3](#_Toc485063108)

[2.2. Utilisation de l’agent DF 3](#_Toc485063109)

[2.3. Gestion de la partie 4](#_Toc485063110)

[2.4. Gestion des rôles 6](#_Toc485063111)

[2.5. IA & Gestion du vote 6](#_Toc485063112)

# Projet

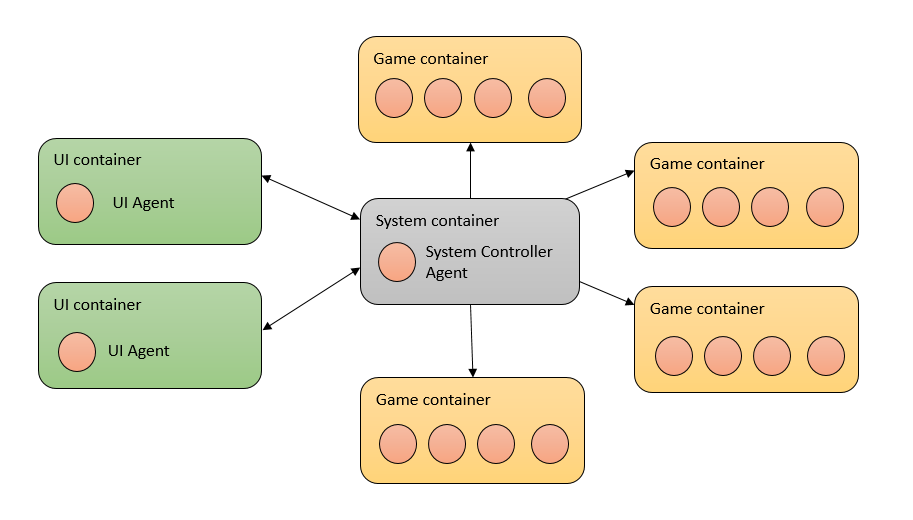
## DESCRIPTION DU JEU

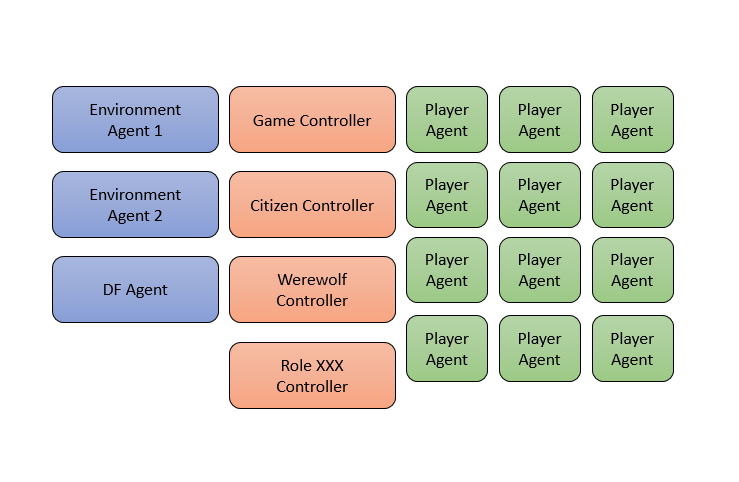
## OBJECTIFS

## ORGANISATION

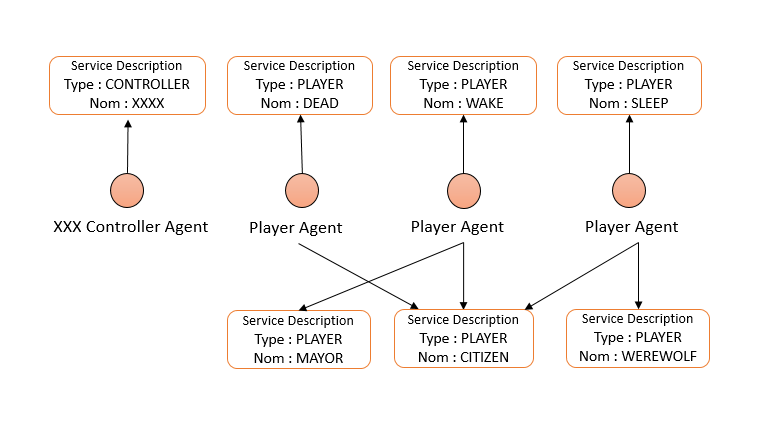
# Architecture

## Architecture générale

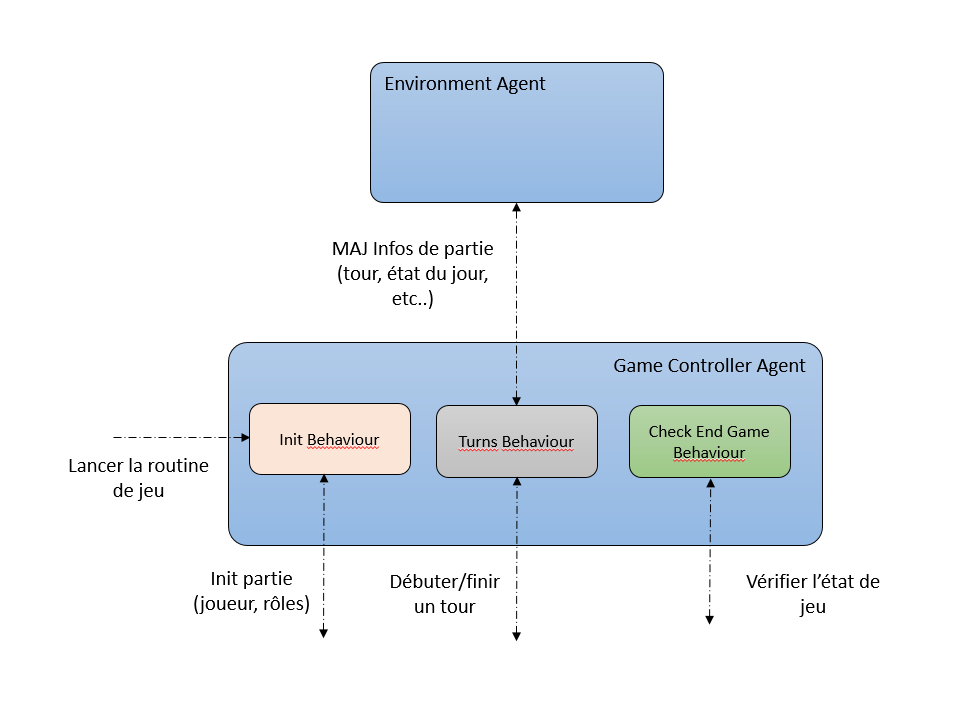




## Utilisation de l’agent DF



## Gestion de la partie



L’agent de gestion de jeu est le GameControllerAgent, il est composé de plusieurs behaviours :

* Init Behaviour, qui permet le démarrage du jeu (configuration, initialisation des joueurs)
* TurnsBehaviour qui assure la routine de jeu (déroulement des tours)
* CheckEndGameBehaviour permettant de tester la fin de jeu.

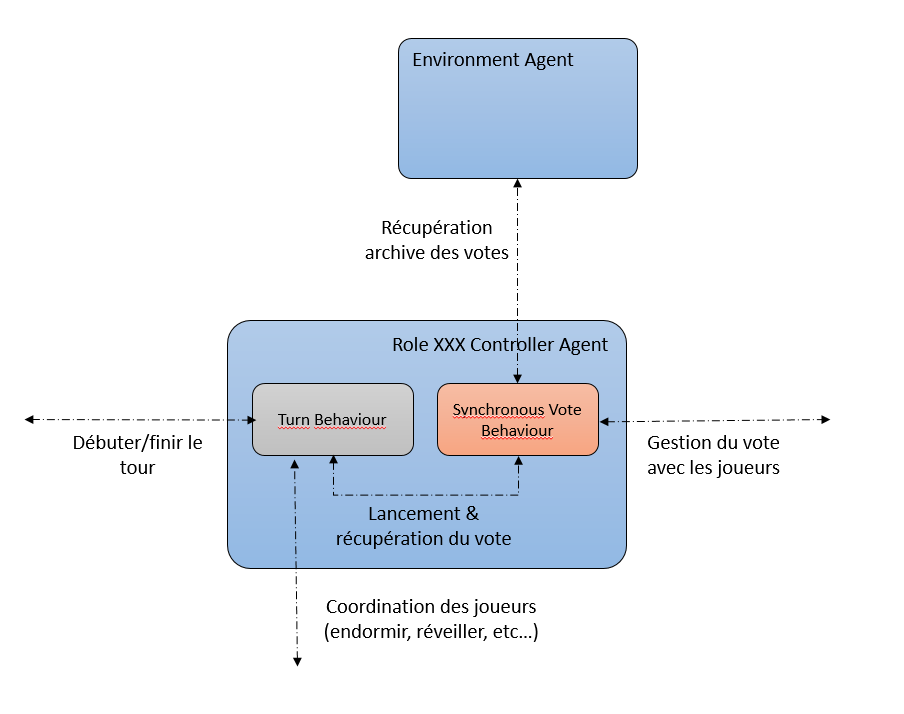
C’est un agent qui communique beaucoup avec l’agent d’environnement afin de lui fournir des infos relatives à la partie (nom du tour, nombre de tours, état du jeu).

## Gestion des rôles

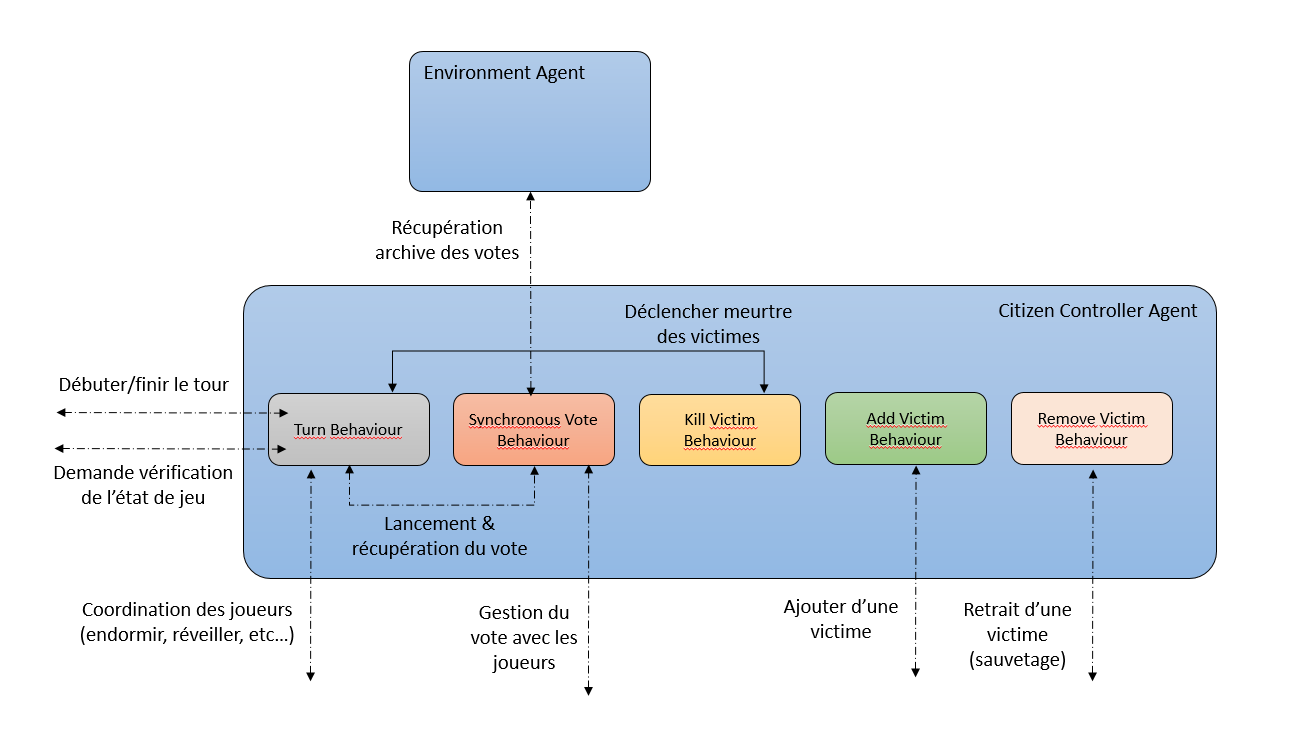
La gestion de rôles et le déroulement concret des tours sont sous-traités par le GameControllerAgent, il s’agit de la responsabilité des contrôleurs de rôles.

Ceux-ci ont une structure similaire :

* TurnBehaviour, qui permet le déroulement concret du tour
* SynchronousBehaviour, qui coordonne le jeu pendant les phases de vote.



Le plus important des contrôleurs de rôles est le CitizenControllerAgent.



## Gestion du vote & IA

### Vote entre LES JOUEURS

La gestion du vote, phase cruciale du jeu, se déroule de manière synchrone, c’est-à-dire chacun des joueurs vote successivement (ce qui leurs permet de voter en fonction du résultat partiel).

Cependant, ce vote synchrone peut prendre énormément de temps lorsque le nombre de votant est important, nous avons donc fait une optimisation en permettant **le vote par « paquet »,** c’est-à-dire qu’au mieux de faire voter les joueurs un par un, nous les faisons voter petit groupe par petit groupe).

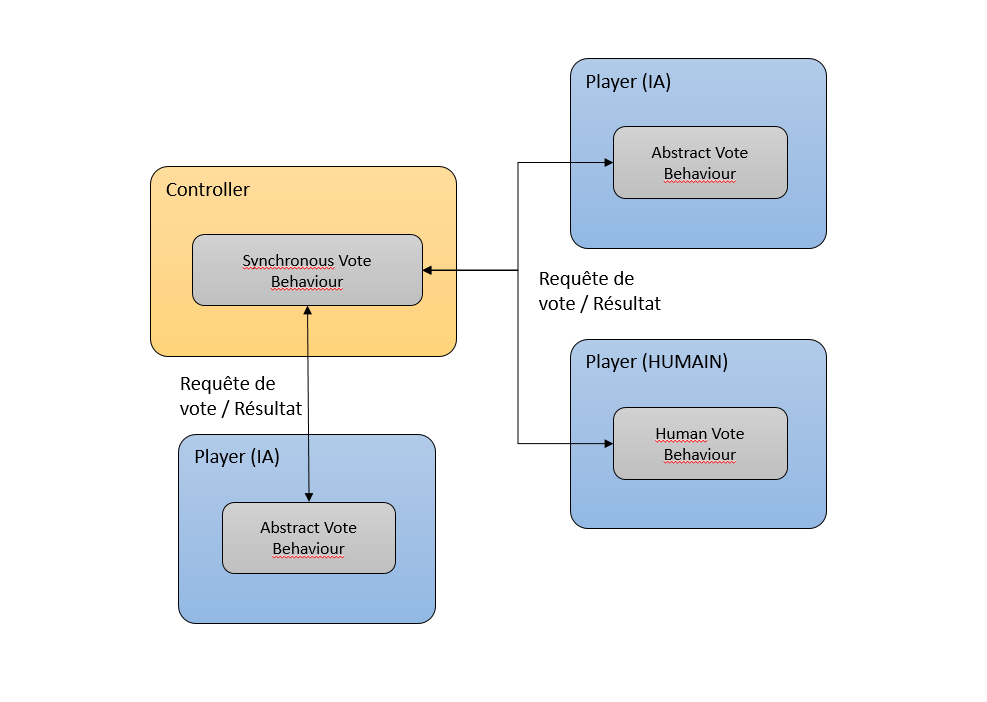
Plusieurs types de votes sont possibles :

* Vote « pour » : le joueur doit favoriser quelqu’un
* Vote « contre » le joueur doit défavoriser un autre joueur (un type de vote qui conduit très souvent à l’élimination du joueur désigné).
* Demander si le joueur souhaite voter : utile lorsqu’il y’a des actions qui se produisent seulement lorsqu’un joueur est sûr de son choix (exemple : rôle WITCH).

La désignation se fait suivant le processus suivant :

* Le joueur ayant recueilli le plus de voix est désigné
* Si il y’a une égalité, un nouveau vote est organisé en réduisant la liste des choix possibles.
* Si il y’a une situation d’interblocage (chacun des joueurs restent sur leurs choix initiaux, provoquant de nouveau une égalité), le joueur désigné est choisi aléatoirement.

Comme le montre le schéma suivant, la gestion de vote est gérée par SynchronousVoteBehaviour : behaviour présent dans chacun des agents contrôleurs de rôles.



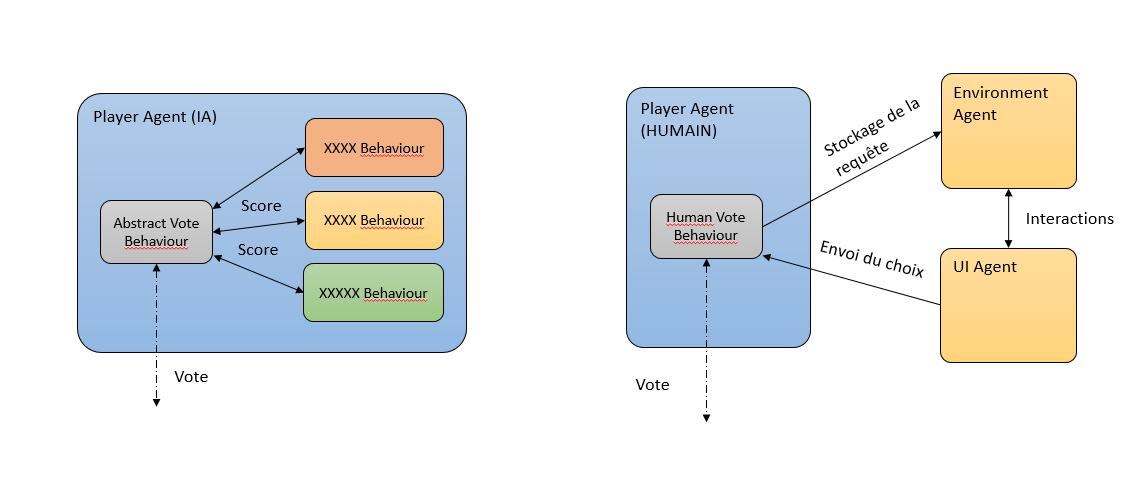
### CHOIX DU JOUEUR : VOTE INTERNE

Lors d’un vote, les choix des joueurs sont complétement différents :

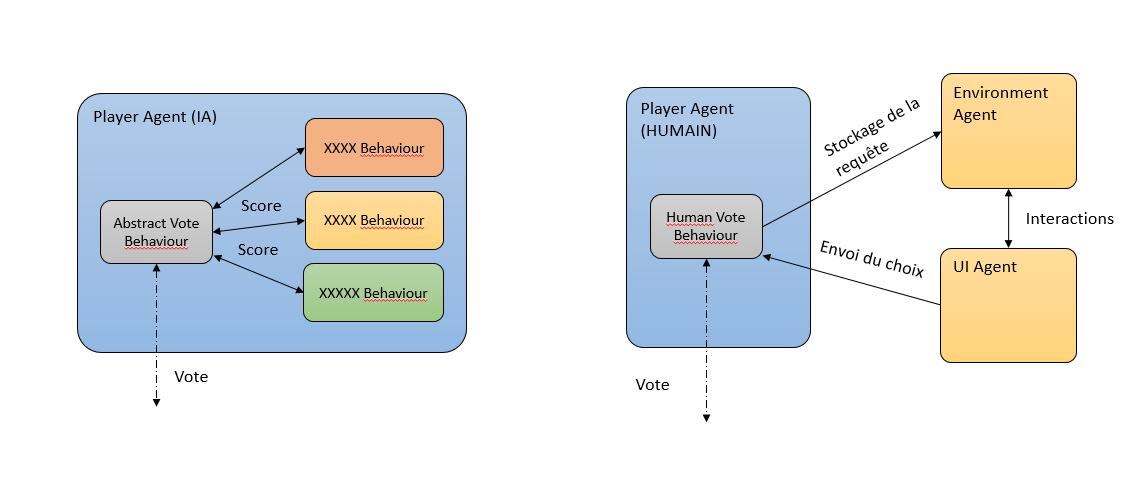
* Quand le joueur est humain, c’est à l’utilisateur d’indiquer pour qui il souhaite voter.
* Quand le joueur est une IA, il doit avoir un choix cohérent avec les rôles qu’il possède.

Ainsi, le behaviour de vote est différent entre les joueurs humains et IA.

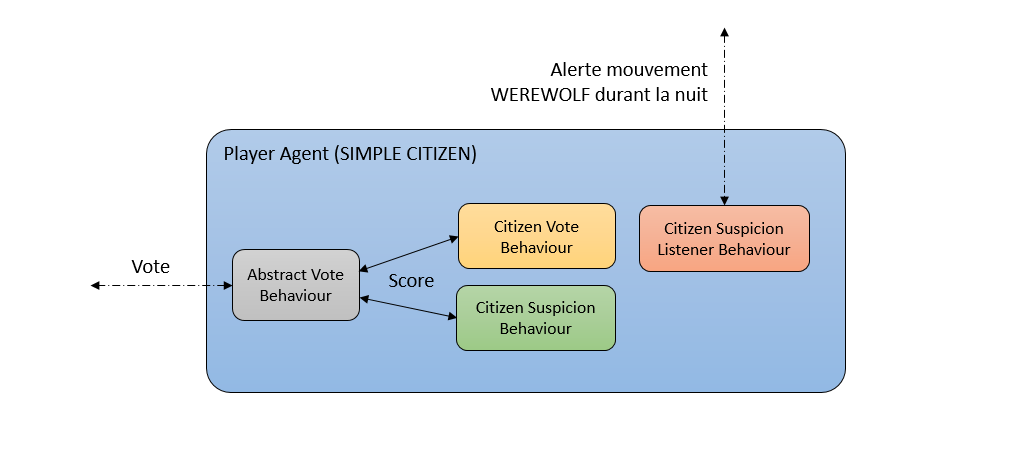
Les joueurs humains stockent la requête de vote dans l’agent d’environnement, lorsqu’ils en reçoivent une ; afin que celle-ci s’affiche dans l’IHM et que l’utilisateur puisse voter.



Pour proposer un choix pertinent en fonction des rôles, la méthode scoring a été retenue. L’intelligence de chacun des rôles va être encapsulée dans une ou plusieurs unités de réflexion.



Ci-dessous un exemple représentation interne en termes d’unités de réflexions.



### Exemple de scoring

Les 2 exemples de scoring que nous allons étudier sont issues d’un déroulement du jeu. Ils représentent les scores associés au vote du joueur E, lorsqu’il souhaite éliminer une personne.

Ce joueur possède plusieurs rôles :

* FLUTE\_PLAYER : son but est de charmer l’ensemble des joueurs vivants. Il remporte le jeu s’il réussit cet objectif. A ce stade de la partie, les joueurs A et D ont déjà été charmés.
* LOVER : au cours de la partie, le joueur CUPIDON la désigné pour être amoureux avec le joueur A. Si l’un meurt, l’autre aussi.

Afin de mettre en valeur le raisonnement de l’IA, nous allons montrer les scores générés lorsque le joueur possède davantage de point de capacités pour sa réflexion stratégique, puis lorsqu’il y’a davantage de points sur la réflexion suspicieuse.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type stratégique | 3 | / 5 |  |  |  |
| Type suspicieux | 2 | / 5 |  |  |  |
|  |  |  |  |  |  |
|  | Joueur A | Joueur B | Joueur C | Joueur D | Joueur E (Votant) |
| IA Role CITIZEN | 1 | 2 | 0 | 1 | -1000 |
| IA Role LOVER | -500 | 0 | 0 | 0 | 0 |
| IA Role FLUTE\_PLAYER | -10 | 4 | 1 | -2 | 0 |
| IA Suspicion | 0 | 0 | 5 | 1 | 0 |
|  | -1527 | 18 | 13 | -1 | -3000 |

Dans ce premier exemple, il s’agit du joueur B qui a été désigné (score maximum).

* Afin que le joueur E ne vote pas pour lui-même (s’agissant d’un vote d’élimination, ça serait du « suicide »), l’unité de réflexion du rôle CITIZEN lui a affecté un grand score négatif.
* Pour éviter de voter pour son amoureux, l’unité de réflexion associé au rôle LOVER a également effectué un score négatif au joueur A.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type stratégique | 2 | / 5 |  |  |  |
| Type suspicieux | 3 | / 5 |  |  |  |
|  |  |  |  |  |  |
|  | Joueur A | Joueur B | Joueur C | Joueur D | Joueur E (Votant) |
| IA Role CITIZEN | 1 | 2 | 0 | 1 | -1000 |
| IA Role LOVER | -500 | 0 | 0 | 0 | 0 |
| IA Role FLUTE\_PLAYER | -10 | 4 | 1 | -2 | 0 |
| IA Suspicion | 0 | 0 | 5 | 1 | 0 |
|  | -1018 | 12 | 17 | 1 | -2000 |

Ici la réflexion stratégique possède un poids plus important et cela fait la différence par rapport à la configuration précédente.

Le joueur vote ici contre le joueur C, contre qui il a le plus de soupçon.

Nous constatons que la répartition des points de capacités influence fortement le vote du joueur, cela permet ici d’avoir des joueurs aux raisonnements différents même si ils possèdent exactement les mêmes rôles.

# CONCLUSION

## AmeliorationS POSSIBLES

## BILAN