Ciências ULisboa

Secret Hitler – Agents Simulating Behaviours

Multi-Agent Systems

Group 04 – Bobs Os Construtores

Tiago Varela - 51017

Brian Marques - 51020

José Domingues – 51084

Index

Introduction

The project consists of creating agents capable of playing a version of the deduction game *Secret Hitler* (https://www.secrethitler.com/assets/Secret_Hitler_Rules.pdf). In this game, the players collaborate for a common purpose but some of them are traitors with a single objective. The main task in the project is the development of agents that will be capable of playing the game whilst simulating different personalities that people have.

Objectives

The objective of this project focuses on developing agents that can collaborate between them, having behavioural capabilities similar to human beings.

In many cases nowadays, the sensibilities and opinions of others are based on trust, on likeness, on closeness, and many other attributes that are trivial for people to perceive, but in terms of agents, these are not so easy to define.

Because of that, and using as a base the Secret Hitler game, we intend to create agents that can seemingly replicate those social behaviours much like a person to successfully play the game. Winning is not the primary purpose, as each agent may have flaws and strengths. Depending on luck or how the rest of the players behave, certain strategies they specialise in may be better or worse.

This will be accomplished with attributes that increase or decrease based on the actions observed by the agents. In this environment, the players communicate with each other, not through a proper conversation, but through a rigid protocol of interaction where they have at most one answer. Continuous conversations are not possible.

The observation of the agents while they play the game also allows identifying different types of personalities and therefore allows choosing the best agents that will have a crucial impact on everyday problems.

Game Explanation

The game is played in turns allowing each player to be the president. The president is identified by the token that goes around the table passing through all players. One of the fascist players assigned in the beginning is also Hitler.

The current president chooses the chancellor and together they decide if the government goes forth with a liberal or fascist policy. The decision is made from 3 cards that the president receives from the deck (each card can be liberal or fascist). From there the president chooses 2 to give to the chancellor which chooses one to be the next policy. This way, the players that don't belong to the government can't be sure if it was the cards, president or chancellor that influenced the next policy.

Afterwards, the members of the government can make a claim about each other and try to convince other players about it. The president may claim he gave the Chancellor liberal cards and the Chancellor may claim he only had liberal ones. At this point, all the players should make their judgement on who is lying or if neither can be trusted.

When a president chooses a chancellor, the government goes to elections and needs to be approved by the majority of players before deciding the next policy. Therefore, it's possible that players don't vote in favour of a government constituted by players they suspect.

The game ends if 5 liberal policies have been chosen (victory for liberals), if 6 fascist policies have been chosen (victory for fascists), or if 3 fascist policies have been chosen and Hitler is the current chancellor with an approved government.

The game starts with a deck of 17 cards: 6 liberal and 11 fascists.

To simply run the game the following command can be used:

java -classpath "C:\Users\ABC\Desktop\test.jar";jade/lib/jade.jar jade.Boot -gui gameManager:agent.communication.GameManager;player1:agent.communication.PlayerAgent(shy);player2:agent.communication.PlayerAgent(shy);player3:agent.communication.PlayerAgent(aggressive);player4:agent.communication.PlayerAgent(aggressive);player5:agent.communication.PlayerAgent(default);player6:agent.communication.PlayerAgent(default)

Simplifications

The simplifications are the most important chapter of the study due to the high number of cases of the game. The most difficult part to represent in this game is the discussions between players to justify their choices. Therefore, to simplify the process and do what is proposed the following simplifications were done:

1. The *Special Election* power is replaced with *Peek* power. This case introduces a lot of factors that influence decisions and complicate the communication protocol for the game.
2. When the president has to decide who is the chancellor, the following happens:

- Each agent has to indicate to the president whom they want as chancellor.
- The president chooses the chancellor on his own (the player's votes are not binding, but may be used as reference).
- Observing others' opinions and the president's choice, each player updates their opinion of other agents.

3. When the government passes a new policy, the following happens:
    - The president has to indicate all cards/policies that received (from the deck). (Can lie)
    - The chancellor has to indicate all cards/policies that received (from the president). (Can lie)
    - There were a couple of possible ways this could be done:
        - If one player stated his cards before the other, then the second player could lie accordingly.
        - In the current implementation, the players must make their claims simultaneously, making lies much easier to detect.
    - Observing chancellor and president's cards, the agents update their opinion about them.
4. When the president has the power to *Investigate* or *Execute*, the following happens:
    - Each agent has to indicate to the president whom they want to apply the power to.
    - The president chooses the target of the power (the player's votes are not binding, but may be used as reference).
    - Observing others' opinions and the president's choice, each player updates their opinion of other agents.
    - These are unique moments in the game, as they are the only moments in the simplified where each agent expressly states whom they do not trust

5. When the president has the power to *Peek*, the following happens:
    - The president peeks the next three policies in the deck and then shares what they saw with everyone else. (Can lie)
    - Everyone hears what the president has to say and may choose the believe them and remember the policies or not.

Agents

In terms of the agents that were developed to play Secret Hitler, 3 different personalities were developed. Namely, the Default Agent, the Shy Agent, and the Aggressive Agent.

The most important feature of the agents is how they build relations with each other. This is done by using a number value (integer) that represents the agent opinion of the others (and how they think the others see them). These values start at 50 and go from 0 to 100 (the higher the value, the more liked an agent is).

As the game moves along, this value for each agent is changed based on the behaviours the other agents have. When evaluating certain decisions, a function is called to evaluate how good it was in the perspective of each agent and readjusts the value for that agent according to that function.

When building a different personality and behaviour, besides redefining how some functions are handled, the BIAS value can also be changed to assign a bigger or smaller penalty depending on the situation and personality. 5 bias (shy) lowers the value necessary to like someone and lowers the score interval in which you dislike someone. -5 (aggressive) does the opposite.

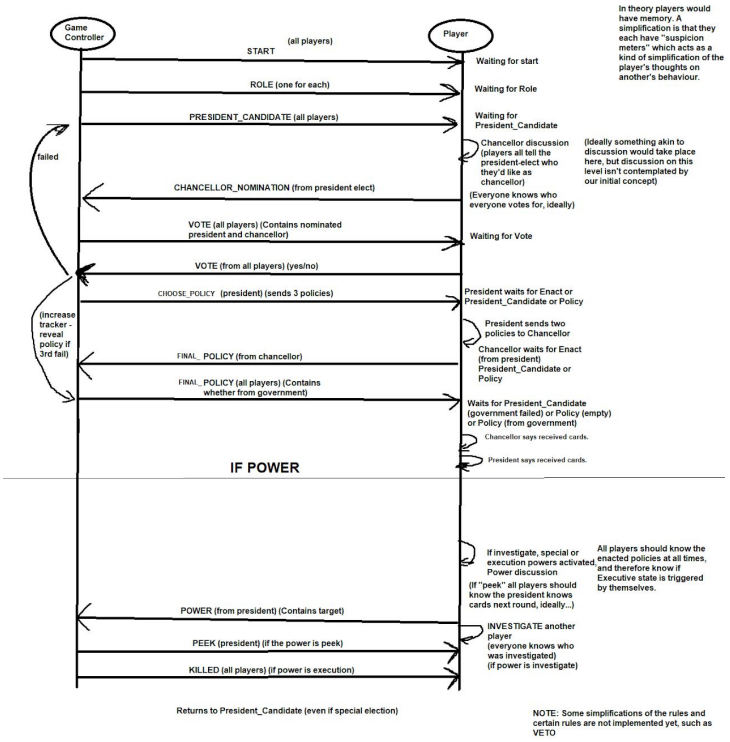
What follows are examples of the threshold values the agents use to interpret how they think about each other.

Thresholds Values	**0**	**40**	**50**	**60**	**100**
Feeling	Hate/Do not trust at all	Dislike	Okay	Great	Loves/Fully trusts

In terms of expressing agents' thoughts, a variable "verbose" was added to print those thoughts. When the player is created, a variable will pass either true or false to indicate if the respective agent will print his thoughts. Notice that due to the fact of not having a GUI the information is printed in the console or redirected to a text file, and the amount of information can make the console log unperceivable for the user, therefore the use of the variable verbose should be used carefully to avoid information density.

Agents Protocol

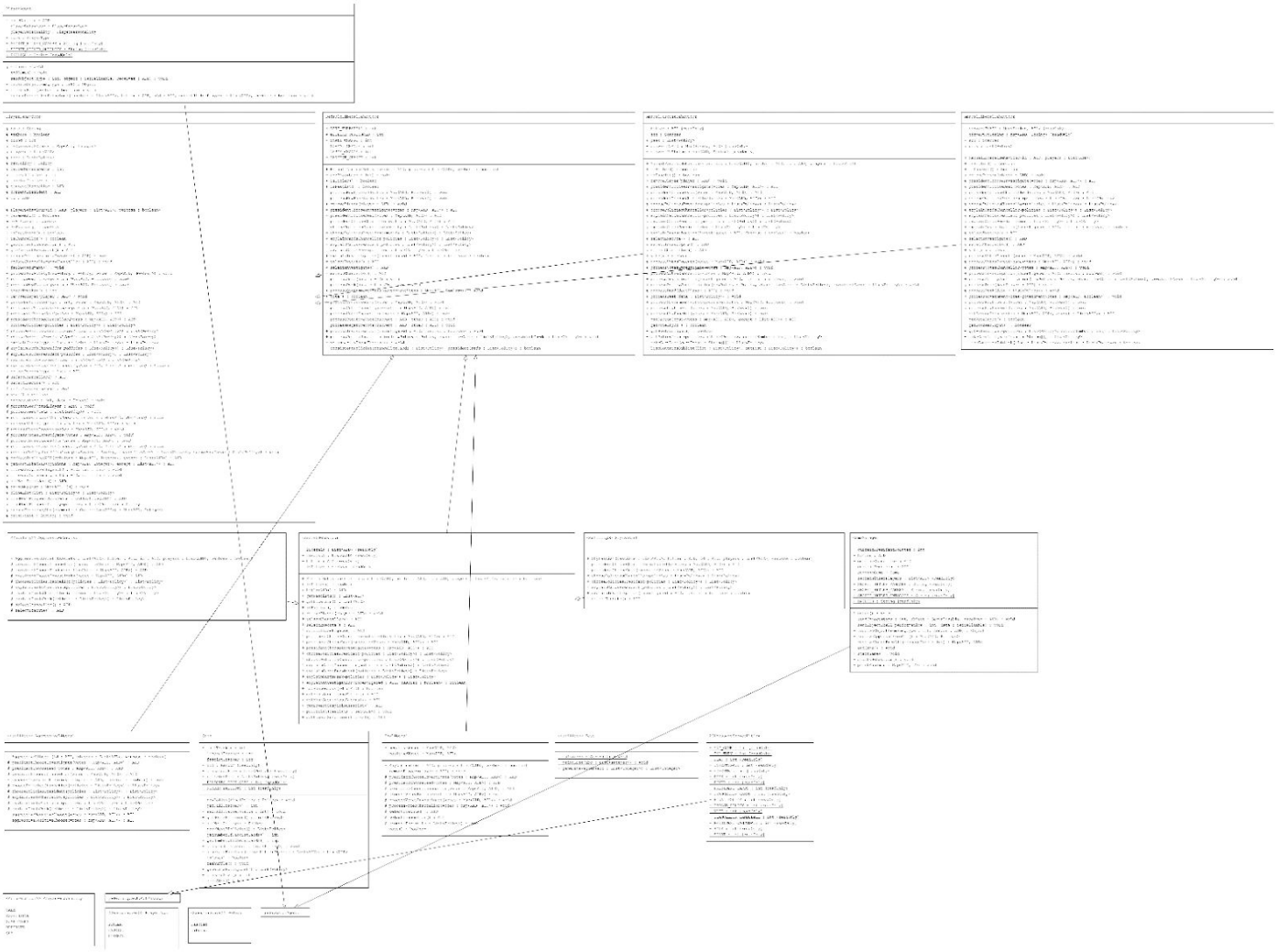
For the agents' protocol, besides the following flow showing how they interact, new performatives were also defined to identify different parts of the protocol. Those performatives were named *ACLMessageSecretHitler*.




```
// ACLMessage performatives go from -1 to 19;
public static final int START = 20;
public static final int ROLE = 21;
public static final int PRESIDENT_CANDIDATE = 22;
public static final int CHANCELLOR_CANDIDATE = 23;
public static final int VOTE = 24;
public static final int CHOOSE_POLICY = 25;
public static final int FINAL_POLICY = 26;
public static final int CHANCELLOR_CARDS = 27;
public static final int PRESIDENT_CARDS = 28;
public static final int POWER = 29;
public static final int PEEK = 30;
public static final int EXECUTE = 31;
public static final int INVESTIGATE = 32;
public static final int DEAD = 33;
public static final int END_TURN = 34;
public static final int END_GAME = 35;
```

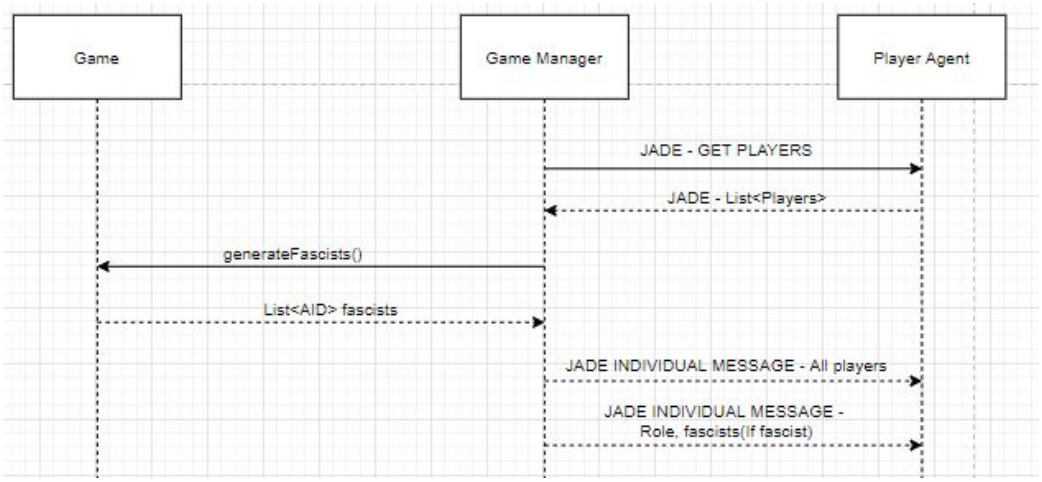
Domain Model & SSD's

In terms of agent modelling and construction, we built a domain model and SSD's to have a way to interpret and understand how the workflow is going to be processed. The image below is too big to fit and properly be analysed, but it's possible to find it in the attachments. The following was developed:

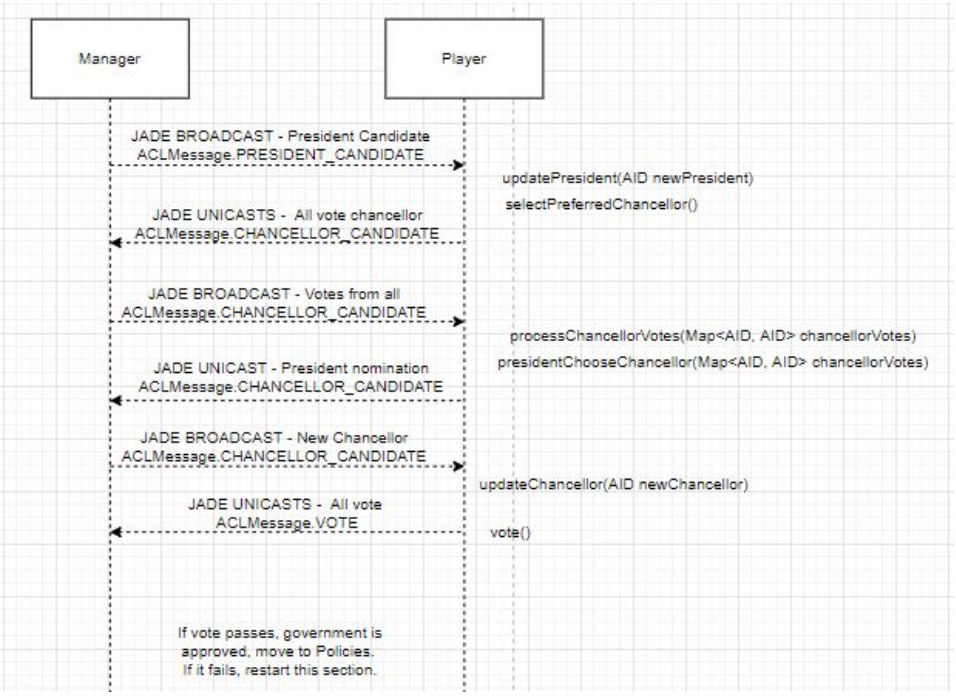


In terms of the SSD's the following was done:

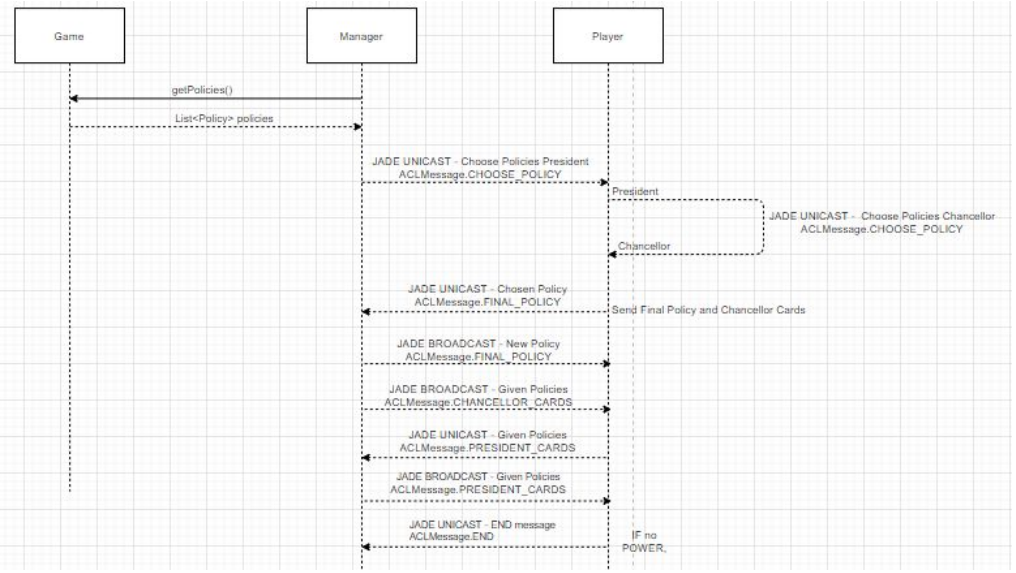
Role Attribution



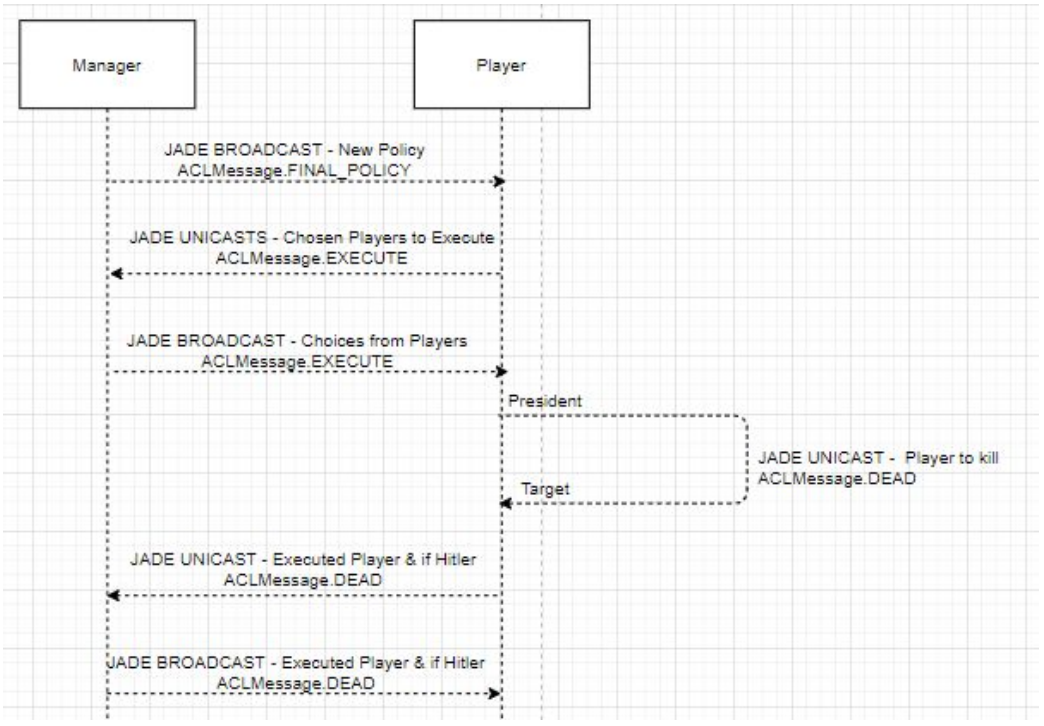
President & Chancellor Attribution



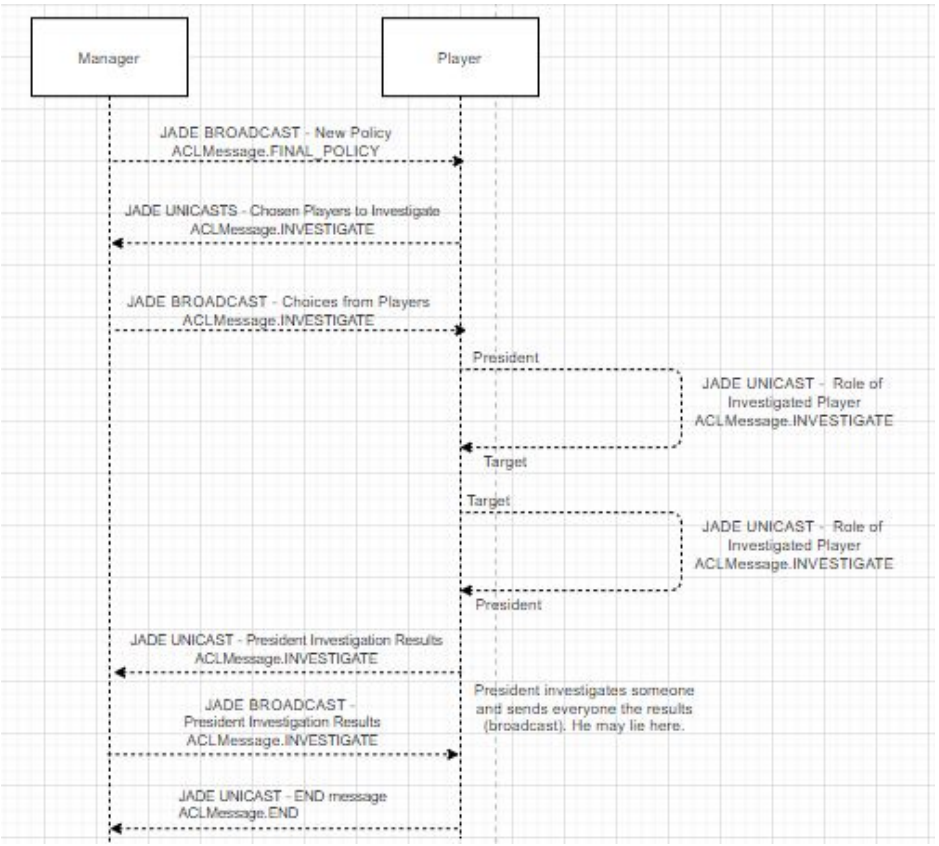
Policies



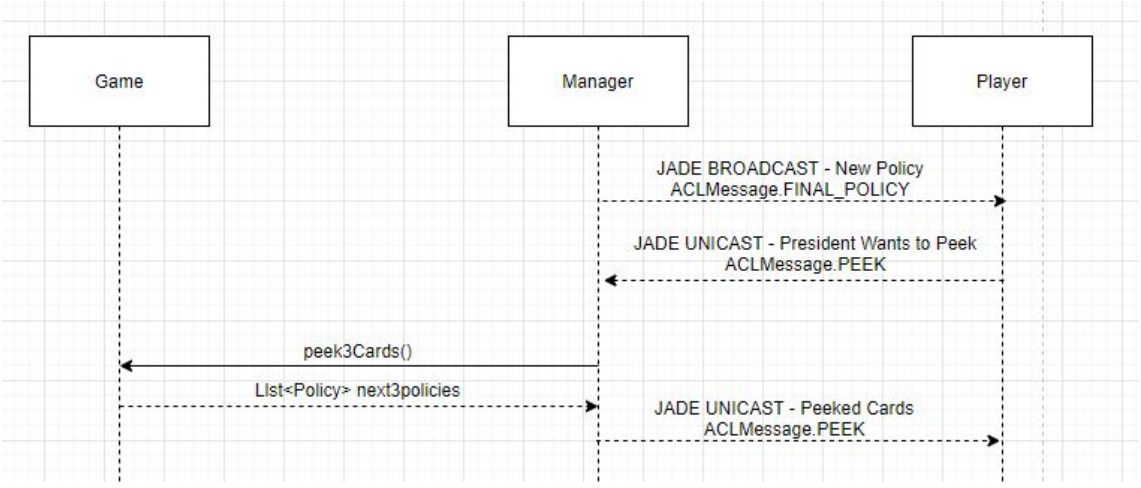
Powers - Execution



Powers - Investigate



Powers - Peek



Comparison Between Personalities

Like mentioned before, different personalities lead to different behaviours, and therefore different ways to deal with each action. In the table below, we show all the agents created and the way they approach the different situations in the protocol.

Function	Default Liberal	Default Fascist	Shy Liberal	Shy Fascist	Aggressive Liberal	Aggressive Fascist
processNewLiberal	Increases the opinion of the current president and chancellor. Increase opinion for each agent that voted in favour of the government and decrease opinion of those who voted against.	DefaultLiberal	DefaultLiberal with bias of 5	DefaultLiberal with bias of 5	DefaultLiberal with bias of -5	DefaultLiberal with bias of -5
processNewFascist	Increase opinion for each agent that voted against the government and decrease opinion of those who voted in favour. If there is no previous peek decreases president and chancellor opinion. Otherwise, only decreases opinion if peek had liberal policies.	DefaultLiberal	DefaultLiberal with bias of 5	DefaultLiberal with bias of 5	DefaultLiberal with bias of -5	DefaultLiberal with bias of -5
presidentChooseChancellor	Nominates player he likes he most, this is the biggest value of the opinion.	If I'm Hitler don't want to mess around choose most voted, otherwise choose Hitler if I'm not too suspicious and fascistTracker >= 3, otherwise most voted	Most votes	Choose Hitler If fascistTracker >= 3, otherwise most voted	Chooses most liked unless they are most voted.	Always choose fascists
presidentChooseDead	Nominates player he dislikes the most, this is the lowest value of the opinion.	If I am Hitler most voted, otherwise choose the most suspicious liberal	Most votes	Most voted if is not Hitler, otherwise liberal	Chooses least liked unless they are most voted.	Most suspicious liberal
presidentChooseInvestigate	Nominates player he dislikes the most, this is the lowest value of the opinion.	Choose whom others voted more	Most votes	DefaultFascist	Chooses least liked unless they are most voted.	Most suspicious liberal
choosePoliciesChancellor	Removes a fascist policy. Otherwise, pass the first policy.	If I'm Hitler remove fascist, otherwise remove liberal	DefaultLiberal	Choose fascist	If they love the president, they may choose to activate a Power, given that it isn't too dangerous to do so. Otherwise, they enact Liberal if they can.	Choose fascist
choosePoliciesPresident	Removes a fascist policy if possible. Otherwise, removes the first policy.	Removes liberal	DefaultLiberal	Removes always to make it possible to have one liberal and one fascist to let chancellor make the final choice	If they love the Chancellor, they remove a fascist. If they don't and have two liberal cards, they test the Chancellor. If they have two fascist cards, they give	Removes always liberal

					themselves a power, if not too dangerous. Otherwise DefaultLiberal	
explainCardsChancellor	Tells the truth.	If Hitler or I'm too suspicious or current president is Hitler tell truth, otherwise say there are two fascists	DefaultLiberal	DefaultFacist	If they hate the President, they'll try to lie to increase suspicion on them. Otherwise tells the truth.	Lie
explainCardsPresident	Tells the truth.	If the chancellor is fascist and I am not suspicious say there are 3 fascist cards, if I'm Hitler say 2 fascists 1 liberal, otherwise truth	DefaultLiberal	Say 2 fascist 1 liberal if there is 1 fascist, otherwise, tell the truth	If they hate the Chancellor, they'll try to lie to increase suspicion on them. Otherwise tells the truth.	Lie
explainCardsPeek	Tells the truth.	Truth	DefaultLiberal	DefaultFascist	If they hate the next president, they may claim the deck has a liberal card when it does not. Otherwise tell the truth.	Lie
explainInvestigation	Tells the truth.	Say liberals are fascists and fascists are liberals	DefaultLiberal	Say liberals	DefaultLiberal	DefaultFascist
selectChancellor	Chooses the player he likes he most, this is the biggest value of the opinion, except the current president and the previous chancellor.	If I'm Hitler don't want to mess around, similar to liberal, otherwise choose Hitler if I'm not too suspicious and facistTracker >= 3, otherwise similar to liberal	Selects whoever got most votes last time. If not possible, DefaultLiberal	DefaultFascist	DefaultLiberal	Choose Fascist
selectExecute	Nominates player he dislikes the most, this is the lowest value of the opinion, except himself and the president.	If Hitler choose the most suspicious player, otherwise most suspicious liberal	Selects whoever got most votes last time. If not possible, DefaultLiberal	Most suspicious liberal	DefaultLiberal	Most suspicious liberal
selectInvestigate	Nominates player he dislikes the most, this is the lowest value of the opinion, or itself if it is suspicious, except the president.	Choose whom I dislike most	DefaultLiberal	DefaultFascist	DefaultLiberal	DefaultFascist
vote	Votes in favour if the opinion of the president and chancellor is both okay, or if the opinion of one of them is great.	DefaultLiberal	Accepts if currentChancellor got most votes.	DefaultLiberal with a bias of 5	DefaultLiberal with bias of -5	DefaultLiberal with bias of -5
processDead	If I liked the victim decrease the opinion of the president. Increase if I did not like the victim.	DefaultLiberal	DefaultLiberal with bias of 5	DefaultLiberal with bias of 5	DefaultLiberal with bias of -5	DefaultLiberal with bias of -5
processPeek	If I trust the president, I save the peeked cards. Ignore otherwise.	DefaultLiberal	DefaultLiberal with bias of 5	DefaultLiberal with bias of 5	DefaultLiberal with bias of -5	DefaultLiberal with bias of -5
processGovernmentVotes	If voted the same as I, increase opinion about that agent. Decrease if not.	DefaultLiberal	DefaultLiberal with bias of 5	DefaultLiberal with bias of 5	DefaultLiberal with bias of -5	DefaultLiberal with bias of -5
processVotesExecute	If someone voted in me decrease my opinion of him. If I like them both reduce my opinion about the voter and voted. If I do not like the voted	DefaultLiberal	DefaultLiberal with bias of 5	DefaultLiberal with bias of 5	DefaultLiberal with bias of -5	DefaultLiberal with bias of -5

	player increase the opinion of the voter and decrease the voted. If I do not like them both I increase my opinion of the voter and voted. If I don't like the voter, but I like the voted player I decrease my opinion of the voter and increase the voted. If I like the voted I decrease my opinion of the voter. If I do not like the voted I increase my opinion of the voter.					
processVotesInvestigate	Same as processVotesExecute	DefaultLiberal	DefaultLiberal with bias of 5	DefaultLiberal with bias of 5	DefaultLiberal with bias of -5	DefaultLiberal with bias of -5
processVotesChancellor	If they voted in me increase my opinion. If voted on themselves ignore. If I like both increase the opinion of the voter and voted. If I do not like the voted but I like the voter, I decrease the opinion of the voter and increase the voted. If I do not like both I reduce the opinion of both. If I don't like the voter but I like the voted I increase the voter and decrease the voted. If I like voted I increase the voter. If I do not like the voted I decrease the opinion of the voter.	DefaultLiberal	DefaultLiberal with bias of 5	DefaultLiberal with bias of 5	DefaultLiberal with bias of -5	DefaultLiberal with bias of -5
processInvestigation	If I hate the president, I assume the opposite of what he said. If I just do not like him, I ignore. If the investigated player is fascist, I set my opinion of him to 0. If not, I set my opinion of him to 100.	DefaultLiberal	DefaultLiberal with bias of 5	DefaultLiberal with bias of 5	Believes the president if they love him, otherwise ignores. If they hate the president, they assume he said the opposite instead.	DefaultLiberal with bias of -5
processPolicyJustification	If I'm president and the chancellor passes a fascist policy with the opportunity of passing a liberal then adjusts opinion according to the tracker of fascists and liberal cards (*). If chancellor and president lied, then decrease the opinion of both. If a fascist policy was passed, and chancellor admitted it had liberal cards then adjusts opinion according to the tracker of fascists and liberal cards (*). If a fascist policy was passed but chancellor only had fascists cards and both president and chancellor didn't lie, then increase opinion of both.	DefaultLiberal	DefaultLiberal with bias of 5	DefaultLiberal with bias of 5	DefaultLiberal with bias of -5	DefaultLiberal with bias of -5

Human-Agent

As a complement to the project, a way of adding human players was implemented y defining the different personalities of the agents at the start of the game. A human player can then take different actions or force some things to happen based on that to see how the agents react.

For further studies, this might come in handy since the agent can be manually tested, evaluated, and adjusted according to its purpose.

To be able to play the game as a human the command to run the game should have the personality “calm”. Each calm player is human controlled. This was done by default but can be changed.

The commands that are accepted are simply indicating a player, approving/disapproving of a government, and claiming and choosing policies. When approving or disapproving the words “yay” or “nay” are accepted and mean/indicate respectively, yes (in favor), or no (against). When the player is asked to choose another agent, they choose the agent by typing their name on the console. Only the valid players for the operation in question are accepted (case sensitive). For example, “player1” can be a valid player if there’s an agent with that name. When claiming or choosing policies, the player uses “f” for a fascist policy or “l” for a liberal policy and should separate them with commas and no spaces when constructing a list of policies.

Notice that the roles of players are printed in the beginning of the game for informative purposes. As a result, a human player playing as liberal will know who is who and can use that for experiments and behaviour analysis. Besides that, they can benefit from that information to win easily.

In the example below, the player1 will be the human player, and player3 will display its thoughts.

java -classpath "C:\Users\ABC\Desktop\test.jar";jade/lib/jade.jar jade.Boot -gui gameManager:agent.communication.GameManager;player1:agent.communication.PlayerAgent(calm);player2:agent.communication.PlayerAgent(shy);player3:agent.communication.PlayerAgent(aggressive,true);player4:agent.communication.PlayerAgent(aggressive);player5:agent.communication.PlayerAgent(default);player6:agent.communication.PlayerAgent(default)

Extra – As included in the initial proposal report.

In terms of trying to extend the project and aiming for further development, some extras could be done.

Event Memory

Each agent can have a memory of every other agent in terms of decisions they have made and evaluate the opinion based on the whole flow of the game. Doing this, memories can be positive or negative.

- Positive Memories
- ENACTED_LIBERAL – Was part of a government that passed a liberal policy.
- GAVE_LIBERAL – When I was chancellor, the player gave me a liberal policy.
- GAVE_FULL_LIBERAL – When I was chancellor, the player only gave me liberal policies.
- INVESTIGATED_LIBERAL – When I investigated the player I found out he was liberal.
- SHOT_FASCIST – Executed someone I thought it was fascist.
- VETOED_FASCIST – Rejected a fascista policy with me.
    - The game would've ended with a fascist victory if that didn't happen.

- Negative Memories
- ENACTED_FASCIST – Was part of a government that passed a fascist policy.
- GAVE_FULL_FASCIST – When I was chancellor, the player only gave me fascist policies.
- INVESTIGATED_FASCIST – When I investigated the player I found out he was fascist.
- SHOT_LIBERAL – Executed someone I thought it was liberal.
- INCONSISTENT_WITH_PEEK – I saw the cards on the top of the deck and his government lied.
    - This event requires saving the result of the peek to compare with the cards the current government says. We have to save the previous state for this one.

Use Memories to Justify Votes and Actions

In the game context, there is a limited number of explanations and justifications, which allows being modelled as follows:

- Each vote in A) can include a justification:
- NO_ARGUMENT- The agent doesn't want to argue.
- RANDOM – I don't have any opinion yet (Choosing randomly).
- Any positive memories about him.

- If the player passed a fascist policy on purpose in B), he can include a justification:
- NO_ARGUMENT - The agent doesn't want to argue.
- WANT_PEEK – The agent wants the peek power.
- WANT_INVESTIGATE – The agent wants the investigate power.
- WANT_SHOOT – The agent wants to try to kill Hitler with the execution power.
- WANT_VETO – The agent wants veto power.

- If someone is accusing someone in C) he can justify:

1. NO_ARGUMENT - The agent doesn't want to argue.
2. RANDOM – I don't have any opinion yet (Choosing randomly).
3. Any negative memories about him.

React to Justifications

The agents observe each other's votes and their justifications, plus if they believe in the justification.

- If a player belies in an argument in favour of player A, that event is added to its memory list about the player A
  - If the player already has that argument on the list then it won't be more effective. For this reason, seeing that player Z ENACTED_LIBERAL 3 times doesn't matter even if those 3 times were done in different governments.
  - The agentes don't remember who argued on the saved memories, so even if he finds out that was a fascists that argued someting the things he said won't be invalidated.

- If the player doesn't believe in the argument nothing happens except the normal opinion updates of who's arguing.

Even if the argument doesn't change opinions, the original base remains. If someone votes in favour of someone we think it's liberal our opinion about him grows even if the presented argument for that opinion is not good or is repeated.

Limitations and Difficulties

In terms of limitations and difficulties, it was felt problems in terms of performance related to receiving messages, in terms of the game itself and also how the agents communicate. With this some simplifications and assumptions were made.

In terms of processing time and performance, a difficulty was found related to the game flow, where an action has to wait for another action before it happens. This means that each agent should only do something after another agent act. With this, the way found to solve the problem was simply waiting for the respective message until the next step. This however was done without any *Cyclic Behaviour with Block*, which impacts in terms of performance since each agent will wait infinitely for a certain message. Afterwards, this could be improved recurring to the behaviours but in terms of perceiving what happens the communication was done this way.

As for the limitations related to the game itself and agents communication, the game itself has a lot of conditions and special rules which could lead to several problems and a high level of complexity when the main purpose isn't creating agents masters in the game, but instead agents that can simulate people's behaviours and personalities. For this reason, some special cases were made simpler or removed at all, and everything was assumed to go well when playing the game, this is, avoiding cases when an agent dies or a message is lost, since in this kind of game that kind of situation doesn't happen.