

ANAC 2017

Competition Manual and Rules

Dave de Jonge

April 18, 2017

Abstract

This document describes the setup of the Diplomacy league of the ANAC 2017 competition and explains how to implement an agent to participate in this competition.

1 Introduction

Diplomacy is a strategy game for 7 players involving multilateral negotiations. The main motivation to use Diplomacy as a domain for ANAC is that in Diplomacy, unlike most other existing negotiation domains, the utility functions are not directly given in the form of a closed-form expression. Instead, the agents need to apply heuristics to estimate the value of any deal and one cannot expect an agent to ever have perfect knowledge of its utility function. We therefore think that this domain is more challenging and more realistic than most other negotiation domains.

The game of Diplomacy involves more than just negotiations. However, since the ANAC competition is only about negotiations, participants should *not* implement a full Diplomacy player from scratch. Instead, **you should only implement a negotiation algorithm, which will be used on top of an existing Diplomacy playing agent**. In fact, you are not even allowed to alter anything about the agent other than its negotiation algorithm.

Participants in the competition are expected to be familiar with the Java programming language.

If anything remains unclear after reading this manual, please do not hesitate to send your questions to: davedejonge@iia.csic.es.

2 Diplomacy

In this section we give a short overview of the game of Diplomacy. This is only meant to give you a global idea of the game, but does not serve as a complete introduction to the game. If you are not familiar with the rules of Diplomacy, then please take a look at the complete rulebook:

<https://www.wizards.com/avalonhill/rules/diplomacy.pdf>.

Additional rules specific to the ANAC competition are discussed in the next section.

Diplomacy is a widely played game for 7 players, with no chance moves and no hidden information.¹ Players make their moves simultaneously. It is designed in such a way that each player needs to negotiate with the other players in order to have a chance of winning. It can be played as a classical board game, or it can be played online.²

The game takes place on a map of Europe in the year 1901, which is divided into 75 *Provinces*. Each player plays one of the 7 great *Powers* of that time: *Austria*, *England*, *France*, *Germany*, *Italy*, *Russia* and *Turkey* and each player starts with 3 units (also called armies or fleets) that are placed on the map (except Russia, which starts with 4 units). Some of the Provinces are called *Supply Centers*. There are 34 Supply Centers and the goal of the game is to conquer 18 or more of these Supply Centers.

The game iterates through five types of rounds (or ‘phases’), in the following order:

- Spring phase
- Summer phase
- Fall phase
- Autumn phase
- Winter phase

The first round of the game is referred to as Spring 1901, followed by Summer 1901, Fall 1901, Autumn 1901, Winter 1901, Spring 1902, Summer 1902, etcetera. In each round all players simultaneously ‘submit orders’ for all of their units.

¹One might say that Diplomacy does have hidden information, because players make secret agreements. However, these agreements have no formal meaning, and form part of the players’ strategies rather than of the rules of the game. Therefore, *formally* speaking there is no hidden information.

²<http://www.playdiplomacy.com/>

2.1 Spring and Fall phases

During Spring and Fall phases each player must submit for each of his or her units one of the following three types of orders:

- A *move-to* order, meaning that the unit will try to move from the province where it is currently located to an adjacent province.
- A *hold* order, meaning the the unit intends to stay in the same province where it already is.
- A *support* order, also meaning that the unit will not move, but in this case it will also give extra strength to another unit that is moving.

If two or more units are ordered to move to the same province, then only the unit that receives the most successful supports will succeed and is said to *conquer* that province. A support order is unsuccessful however, if there is another unit moving to the location of the supporting unit. In that case we say the support has been *cut*.

2.2 Summer and Autumn Phases

If, during a Spring or Fall phase, a unit tries to stay in a province, but another unit successfully conquers it, we say the first unit is *dislodged*. This means that during the following Summer or Autumn phase the unit must retreat to another province adjacent to its current location. Units that are not dislodged don't do anything during a Summer or Autumn phase.

2.3 Winter Phases

If a player successfully moves into (or stays in) a Supply Center during a Fall phase that player becomes the *owner* of the Supply Center. That player stays the owner of that Supply Center until any other player manages to occupy it during a Fall phase.

During Winter phases each player may be allowed to build more units, or may be obliged to remove some of his or her units from the map. Specifically, if a player owns m Supply Centers and currently has n units, with $m > n$, then he or she can build $m - n$ new units. On the other hand, if $m < n$ then he or she must remove $n - m$ units.

2.4 End of the Game

A player is eliminated when he or she loses all his or her units. A player wins the game when he or she owns 18 supply centers (a *Solo Victory*), but

a game may also end when all surviving players agree to a draw. Any player may propose a draw when he or she likes to.

2.5 Negotiations

The main difference between Diplomacy and other deterministic games like chess and checkers, is that in Diplomacy players are allowed to negotiate with each other and form coalitions. Each round, before the players submit their orders, the players get some time to negotiate with each other and make agreements about the orders they will submit. Negotiations take place in private, and agreements that are made are only known to the players who are involved in the agreement. Players that are not involved in it are not aware of the agreement.

Typically, players may agree not to invade certain provinces, or they may agree that one player will use some of his or her units to support a unit of the other player. In this way, players essentially form coalitions against other players. These coalitions are not given beforehand. During the course of the game players may form coalitions and break coalitions as they like.

In a real Diplomacy game there are no real rules for the negotiations. Players are allowed to negotiate anything and there is no guarantee that they will indeed obey the agreements they make. **For the ANAC competition, however, the players may only negotiate specific deals**, as we will discuss in the next section, and **the players will automatically obey their agreements**.

3 Diplomacy for ANAC

3.1 Framework

The competition will be run on the BANDANA framework:

<http://www.iiia.csic.es/~davedejonge/bandana>

which is an extension of the DipGame framework. The BANDANA framework provides all the necessary Java classes to run a Diplomacy tournament (except for the Parlance game server) and analyze the results. Furthermore, it provides the Java class that the participants need to use as the base class of their agents.

3.2 The Negotiation Protocol

The agents do not negotiate according to the Alternating Offers protocol or any other turn-taking protocol. Instead, each agent may propose any deal whenever it wants. A deal may involve at least 2 and at most 7 agents. Once all players involved in the deal have accepted it, a special *Notary* agent checks whether it is consistent with earlier made agreements. If this is indeed the case then the Notary will send a confirmation message to all agents involved in the deal. The deal is only considered officially binding once the Notary has sent this confirmation message. During any Spring or Fall phase players may make as many proposals as they wish.

Note that a proposal is only sent to the players that are involved in the deal, so the other players will not be aware of the fact that this deal has been proposed. Also, the Notary sends its confirmation message only to the players involved in the deal, so the agreement remains secret.

3.3 Allowed Proposals

As explained, in real Diplomacy there are no formal rules for the negotiations. However, for the ANAC competition there is a well-defined restricted set of proposals the players can make.

The kinds of deals the agents may propose to one another are those deals that are represented by a BasicDeal object in the BANDANA framework. That is: a deal may consist of any number of Order Commitments (a promise to submit a certain order during a specified phase) and any number of Demilitarized Zones (an agreement between a set of players that neither of them will move into a given set of provinces during a specified phase).

In the ANAC competition negotiations will only take place during each Spring or Fall phase, and the Order Commitments and Demilitarized Zones may only refer to orders during Spring and Fall phases.

Definition 1. An *Order Commitment* oc is a tuple: $oc = (y, \phi, o)$, where y is a 'year' (an integer number higher than 1900), $\phi \in \{Spring, Fall\}$ and o is any legal Move-to order, Hold order, or Support order.

An order commitment is a promise that a power will submit a certain order during a certain phase and year. For example: "*The army in Holland will move to Belgium in the Spring of 1902*". However, if the order is a HLDOrder, then the corresponding power is still allowed to submit a SUPOrder or SUPMTOOrder for that unit instead of the HLDOrder.

Definition 2. A *Demilitarized Zone* dmz is a tuple: $dmz = (y, \phi, A, B)$ with y and ϕ as above, A a nonempty set of Powers:

$$A \subset \{Austria, England, France, Germany, Italy, Russia, Turkey\}$$

and B a nonempty set of Provinces.

A Demilitarized Zone consists of a phase, a year, a set of Powers and a set of Provinces, with the interpretation that none of these Powers is allowed to enter (or stay inside) any of these Provinces during that phase and year. For example the Demilitarized Zone (1903, *Fall*, $\{FRA, GER, ENG\}$, $\{NTH, ECH\}$) has the interpretation "In the Fall of 1903 *FRA*, *GER*, and *ENG* will not enter the North Sea and will not enter the English Channel".

Definition 3. A *Deal* d is a set:

$$d = \{oc_1, \dots, oc_n, dmz_1, \dots, dmz_m\}$$

where each oc_i is an Order Commitment, each dmz_i is a Demilitarized Zone, and with $n \geq 0$, $m \geq 0$, and $d \neq \emptyset$.

A proposed deal can only be accepted or rejected in its entirety. It is not possible to only accept part of the deal. In the case you wish to accept only a part of the deal, you simply need to propose a new deal which only consists of those Order Commitments and Demilitarized Zones you wish to include in it. For example, if Austria proposes a deal $d = \{oc_1, oc_2\}$ to Germany, then Germany can choose to either accept d or to reject d , but cannot choose to only accept $\{oc_1\}$. Instead however, Germany can decide to make a new proposal $d' = \{oc_1\}$ to Austria, so then it is up to Austria to determine whether to accept or reject d' .

Apart from proposing this type of deals, your agent is also allowed to propose a draw to all the other surviving players, by calling the `propose-Draw()` method of the `ANACNegotiator` class. The game ends in a draw if all agents that have not been eliminated propose a draw in the same round of the game (see Section 'Proposing Draws' of the BANDANA manual).

For more information about the negotiation protocol we refer to the BANDANA manual.

3.4 Commitments

In a real Diplomacy game the agreements made between the players do not have any formal consequences and players may break their promises. In

the ANAC competition, however, your agent will be based on the *D-Brane tactical module* (see the BANDANA manual) which will choose the orders your player will submit, and which will do this in such a way that it always obeys all agreements that your negotiation algorithm has made. Therefore, when your agent makes a deal, you can be sure that all agents involved in the deal (including your own agent) will indeed respect this commitment.

4 Overview of the Competition

The agents participating in the competition are divided into groups. Ideally, each group contains 7 participants. However, If the number of participants is not a multiple of 7 then (some of) the groups will be supplemented with 1 or more agents provided by the organization. If there is more than 1 group, the best players of each group will advance to the final round (the number of participants in each group that advance to the final will depend on the total number of participants). In each group (and in the final) the players will play a large number of games against each other. In each game players are randomly assigned to any of the 7 'Great Powers'.

Every Spring and Fall phase of the game will have a deadline of 30 seconds. It is the participants own responsibility that their players' negotiation algorithm finishes within that deadline. A game may either end in a 'Solo Victory', which happens if one player owns 18 or more Supply Centers, or in a draw between all surviving players. In any round, any player may propose a draw to the other players. The game ends in a draw if each player that has not yet been eliminated proposes a draw during the same phase. If the game advances to the Winter 1920 phase all players will automatically propose a draw.

4.1 Scoring System

Each player that is eliminated before the end of the game receives 0 points. If a player wins the game by a Solo Victory then that player receives 12 points, while all other players receive 0 points. If the game ends in a draw then points are awarded to the surviving players as follows:

- If there are 2 surviving players, each of them receives 6 points.
- If there are 3 surviving players, each of them receives 4 points.
- If there are 4 surviving players, each of them receives 3 points.
- If there are 5 surviving players, each of them receives 2 points.
- If there are 6 surviving players, each of them receives 2 points.

- If there are 7 surviving players, each of them receives 1 point.

If 2 or more players in a group end with an equal number of points, the total number of Supply Centers conquered in all games will be used as a tie-breaker. If players still rank equal, or if the difference between the players is so small that it cannot be considered statistically significant, the organization may decide that more games will be played.

5 How to Implement a Negotiator for the Competition

To participate in this competition you must implement a Java class which extends the `ANACNegotiator` class which is provided with the BANDANA framework. The BANDANA framework and its manual can be downloaded from:

<http://www.iiia.csic.es/~davedejonge/bandana>

The BANDANA framework also includes the source code of an agent called `ANACExampleNegotiator`. Please take a look at this code to see an example of how to implement your agent. For an explanation of this code we refer to the general BANDANA manual.

We should remark that the BANDANA framework also includes classes to implement a full Diplomacy agent completely from scratch, but that is not what you are supposed to do. **Participants of the ANAC competition are not allowed to participate with a Diplomacy agent implemented completely by themselves. Instead, participants must only extend the `ANACNegotiator` class.**

This means that as a participant you cannot directly control which moves your player is going to make. You are only writing a negotiation algorithm. Your player will use an existing algorithm to determine which moves to make.

The `ANACNegotiator` is an abstract class that has a private field of type `ANACPlayer` which extends the `Player` class. Therefore, most of the options that you have to implement a player are hidden for participants of ANAC.

The `ANACNegotiator` class contains an abstract method with the signature `void negotiate(long deadline)`, which you must implement. This method is called at the beginning of each Spring phase and each Fall phase. The given deadline is a Unix timestamp and it is your own responsibility to make sure that the method returns before this deadline. Your player must also implement the following main method and constructor:


```

public static void main(String[] args){
    MyANACNegotiator myNegotiator = new MyANACNegotiator(args);
    myNegotiator.run();
}

public MyANACNegotiator(String[] args) {
    super(args);

    //here you can put whatever you like...
}

```

You may give any name you like to your class, so you can change the name `MyANACNegotiator` to the name you like (but please make sure that it is a clear and recognizable name, because the name of your agent will by default be set equal to the class name).

Once your agent has negotiated its agreements with the other players, the `ANACPlayer` will automatically choose a set of orders that obey those agreements and submit those. These orders come from the D-Brane Tactical Module as explained in Section 7 of the BANDANA 1.2 manual. You can use this same tactical module in your negotiation algorithm. You can access it by calling `getTacticalModule()`. It allows you to see which orders your player will submit if a certain proposal gets accepted. Based on this information you may decide to propose or not to propose a certain deal. Other tools that you may find useful for the development of your negotiator are the `Adjudicator` and the `GameBuilder`, which are also explained in the BANDANA manual.

As explained, you cannot directly control which orders your agent will submit, but you can do this indirectly, by making deals with the other players. For example, suppose you are playing as England and you want to invade Paris. In order to do so you could for example make an agreement with Germany that one of your fleets will move into Paris and that Germany will give you support. If Germany accepts this agreement, then your agent will certainly try to move your fleet into Paris. However, if your negotiation algorithm does not make such an agreement, then you cannot control whether your agent will attack Paris or not. In that case, if you want to find out whether your player will invade Paris or not, you need to call the D-Brane Tactical Module, and provide it with the agreements you have made. It will then return the set of orders it will submit under those agreements.

Just like the `Player` class, the `ANACNegotiator` class has two public fields `me` and `game` which represent the power you play and the current state of the

game. These are explained in the BANDANA manual. More information about the deals you can propose, how you can accept incoming proposals, and how the negotiation protocol works can be found in the BANDANA manual.

Finally, we should stress a couple of important differences between the things explained in the BANDANA manual, and the ANAC competition:

- Instead of extending the `Player` class you should extend the `ANACNegotiator` class.
- It is not possible to send informal messages (Section 6.8 of the manual) in the ANAC competition.
- You do not need to check whether previously confirmed deals are still valid (Section 6.3.3), because this is already done automatically by the `ANACNegotiator` class.
- You do not have to check whether deals are consistent with the confirmed deals (Section 6.7) because the Notary will do this.
- The BANDANA manual warns that you can never be sure the other players obey their deals. This is however not the case in ANAC because each player will automatically obey all confirmed agreements.

5.1 Command Line Parameters

The constructor of your agent will call the constructor of its parent class `ANACNegotiator` with an array of Strings. The Strings in this array can be the same as the arguments of the other example agents provided with BANDANA. That is: you can set the name of the agent (-name), the log path (-log), the final year (-fy), the game server port (-gamePort) and the negotiation server port (-negoPort). All of these are only optional. If they are not specified then the `ANACNegotiator` will use default values. Any other strings passed to the super constructor will simply be ignored.

Please note, however, that during the competition the command line parameters will be set by the organizers of the competition, so you cannot set them yourself. Also, your agent should be able to start without setting any other command line parameters.

5.2 Reading from and Writing to Hard Disk

In order to debug your agent you may find it useful to write log files. You can do so by calling the `getLogger()` method from the `ANACNegotiator` class and then call the `logln()` method of the `Logger` class (or any of its other

logging methods). You don't need to call the `enable()` method because it will be enabled automatically.

It is important to know that your agent is only allowed to write to disk via this Logger class, and that **your agent is not allowed to read anything from hard disk**. Therefore, the log files are only useful while you are developing your agent, but cannot be used during the competition itself (for example to store any knowledge about your opponents).

5.3 Bugs and Hacks

Of course the BANDANA framework may contain some bugs. If you find any such bug keep in mind that **you are not allowed to exploit such bugs to give your agent any kind of advantage**. Any of the classes or methods provided in the BANDANA framework may only be used in the way that we *intended* you to use them, which should be clear from this manual or the BANDANA manual. Whenever you find a bug, or find any way to make use of a method or class that you are not sure of whether it is legal, please contact us.

6 How to Submit my Agent to the Competition

Once your agent is finished you will have to compile it into a runnable jar file and put it in a zip file *together with your java source code*. You should send this zip file by e-mail to `davedejonge@iia.csic.es`. Please make sure you put [ANAC Diplomacy] in the subject of the e-mail, and clearly include your name and the name of your agent in the e-mail.

The deadline for submission is 1 June 2017 23:59 UTC-12.

7 Frequently Asked Questions

A list of Frequently Asked Questions can be found here: https://docs.google.com/document/d/1CubJLcQz02_6yNeKh9t2DHx0cTmdGOCjjHrqWU1SB30/edit?usp=sharing

8 Contact

If you still have any questions about Diplomacy, BANDANA, or the ANAC Diplomacy competition, please do not hesitate to contact us. Also if you have any suggestions on how to improve the BANDANA framework, the

BANDANA manual, or this competition manual, then we are happy to hear from you. You can contact us by sending an e-mail to davedejonge@iia.csic.es