DarkBlade: a program that plays Diplomacy

João Ribeiro[1], Pedro Mariano[2], and Luís Seabra Lopes[2]

[1] XLM - Serviços de Informática, Lda
jbsr@ua.pt
[2] Transverse Activity on Intelligent Robotics – IEETA – DETI
Universidade de Aveiro
Portugal
plsm@ua.pt lsl@ua.pt

Abstract. Diplomacy is a 7-player game that requires coordination between players in order to achieve victory. Its huge search space makes existing search algorithms useless. In this paper we present Darkblade, a player designed as a Multi-Agent System that uses potential fields to calculate moves and evaluate board positions. We tested our player against other recent players. Although there are some limitations, the results are promising.

1 Introduction

Two-player games have been the subject of intense research in order to come up with an agent capable of beating the human champion. Checkers and Chess have already seen the appearance of such agents. Moreover, in the game of Checkers a partial proof of being a draw has been established [1]. Research has been focusing on building databases of end-positions, which can be used to classify the position as a win for one of the players or as a draw. Search algorithms and specialised hardware have been the tools to achieve this goal. However, research in games with more than two players where alliances between players are crucial in order to reach the winning position has been neglected. The relation between any two players will affect the strategy of a player [2]. If relations may shift during the course of a game, agents must take into account backstabbing behaviour if there is no entity to enforce agreements [3].

Work by Krauss [4], Loeb [5] and others on multi-player games used Diplomacy and show its potential for serving as a test-bed to new theories and practical approaches for solving such highly complex games. The game's rules are simple but the sheer number of possible plays for each turn is staggering. The number of possible openings is determined to be over 4×10^{15} which is far superior to the 20 different opening moves in chess [5]. As for the number of possible movements per turn it is roughly estimated to be around 34^{16} (see [4]). This presents a new problem that requires a different approach, even if Diplomacy and Chess have similar mechanics, the size of the search space makes a purely search space based solution unfeasible due to its computational requirements.

2 Diplomacy

2.1 Rules

The game of Diplomacy is played on a board that depicts nations and regions in Europe in the beginning of the 20th century [6]. The map is divided in provinces which can be inland, water and coastal. Some inland and coastal provinces have supply centres which may be owned by a player. There are a total of 34 supply centres and victory is achieved when a player owns 18 supply centres.

Each player controls units. These may be either armies or fleets. Only one unit may be located in a province. Common sense dictates what province type may hold some unit type. An exception occurs with fleets in that they do not occupy a coastal province, but rather its coast or one of its coasts. In the case of coastal provinces with two provinces, fleets can only travel to some coasts in adjacent coastal provinces.

At the start of a game each player is randomly assigned a nation, which corresponds to 3 supply centres and 3 units, with the exception of Russia that has 4 of each. The initial supply centres are termed *home supply centres*. They are the only places where a player can put new units. Figure 1 shows the map in the beginning of a game.

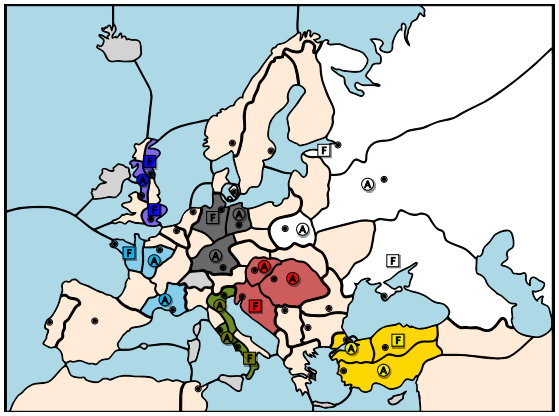

Fig. 1. Map game and initial state at *Spring* of 1900.

Normal play is divided in two types of turns called *Spring* and *Fall*. Both have the following phases: 1) diplomatic; 2) order writing; 3) order resolution; 4) retreat and disband. Fall turns have a fifth phase: gain or lose units. A game starts at turn Spring of year 1900.

Each player assigns to its unit one of the following orders: *hold* its position, *move* to a reachable province, *convoy* a land unit, or *support* another unit. Only fleets can convoy. A unit may give support to a unit to: 1) hold its position if the supported unit is in a reachable province; 2) move to a reachable province.

This province must be reachable by the supporting unit. Support orders are an important aspect of this game, since if two units try to move to the same province without any support, they will fail. If they are supported, the unit with highest support wins.

The game focuses on negotiation, alliances, defection and treason (agreements are not enforceable).

2.2 DAIDE Application

DAIDE (Diplomacy AI Development Environment) is a client/server application which was developed in order to foster the development of artificial agents to play Diplomacy. Clients are players and they send orders and diplomatic messages to the server. The server is responsible for managing a game, processing orders, and dispatch diplomatic messages to players. Diplomatic messages are organised in levels, ranging from alliance proposals (level 10) to explanations (level 130). Currently few agents use level 10 messages. DAIDE can also be used to play no-press games, i.e., no diplomatic messages are allowed. From this point we will use the word bot to refer to an artificial player designed to play Diplomacy through DAIDE.

3 Previous Work

3.1 Israeli Diplomat

The Israeli Diplomat [4], by Kraus and Lehmann, was a continued effort in developing a Diplomacy agent first published in 1988 [7] with impressive results and providing an obvious inspiration to some of the most recent agents, like HaAI [8]. It remains one of the most complete implementations of a Diplomacy agent with reasonable performance in both strategic analysis and negotiation and capable of winning games with press against human competition. It uses a dynamic multi-process layer-based architecture. Each process is responsible for a part in a specific aspect of the game (negotiations, movements, etc). It supports a decent protocol of communication that allows for basic alliance/peace negotiations as well as more detailed order negotiations. The strategy plan includes a one-season movement analysis that considers the gain of a set of moves and current or future alliances or agreements with other powers. Some emphasis on variable personality was included to prevent a deterministic form of play.

The use of personality traits that influence decisions on how the agent system acted when facing the same problems is also mentioned but the details about this method are not specified.

3.2 The Bordeaux Diplomat

Bordeaux Diplomat [5], by Loeb, like the Israeli Diplomat also makes use of modules to divide the game's complexity. It has a negotiation module and a

strategy planning module. The negotiation module deals only with communication with other powers and the strategic core tries to find the best moves for each power ignoring the power it represents. The strategic module uses an improved best-first search algorithm, Refined Evolutionary Search, to find the best moves for one season starting with a seed of movements and them mutating it until it contains the best set of moves for each power or alliance of powers. In this search, Nash Equilibria, when found, are not lost since each alliance (be it one single power or a group of allied powers) has no profit in mutating its moves. For modelling the loyalty of each ally it uses a Friendliness Matrix. This matrix holds a minimum gain value for each power that represents the likeliness of a certain power to back-stab another if at any given time it can achieve that gain in the field. For builds and retreats it uses a complete search space algorithm since the number of combinations is not overwhelming in terms of computational effort.

3.3 LA Diplomat

Shapiro, Fuchs and Leginson's LA Diplomat [9], works by learning new moves in consecutive plays with itself. It uses a hierarchy of pattern-weights to represent partial positions on the world map or specific moves. It then uses time difference learning for the evaluation of each pattern, thus giving certain powers more value than others. This approach has some interesting results, with the agent actually learning some book openings by itself. However some unrealistic move patterns end up with high value.

3.4 HaAI bot

HaAI [8], by F. Haard and S. Johansson, is a multi-agent approach where each unit is modelled by a sub-agent. HaAI proposes what it believes is the best movement and the needed supports to a common pool of orders. To select the season orders it uses an adapted Contract Net to coordinate sub-agents, where each sub-agent is both a manager and a contractor. The supports being the contracts that can be bid on by other agents.

The game board is modelled using a weighted world map where only supply center provinces have values. A province's total value, however, is derived from its base value (in case it is a supply center) and a fraction of the value of the adjacent provinces. This is then used by unit sub-agents to determine the best movement considering the closest maximum profit. Unit building is handled by a new unit sub-agent and uses an army/fleet ratio. This bot has proven rather successful against other DAIDE projects prior to 2005.

3.5 Diplominator bot

Diplominator [10] is a recent bot and produces interesting results. It is capable of basic peace negotiations, partially supporting Level 10 of the DAIDE Diplomacy

Protocol, and shows, to some extent, how negotiation, even in its basic level, influences an agent's results. It was tested only against DumbBot. When not using negotiation it falls slightly behind DumbBot, but when negotiation is active it performs slightly better.

In terms of strategic analysis, Diplominator attempts to maximise a unit profit one at a time, thus using a monolithic system. For profit evaluation it uses province values and value propagation to determine the most profitable destination. To avoid deterministic factor it uses a random selection of the most profitable provinces.

3.6 DumbBot

This is a bot developed by David Norman that belongs to DAIDE Community bots [11]. Despite being a simple agent, as described by the author, it performs well and some times is able to beat other more sophisticated bots as well as un-experienced human players in no-press diplomacy. Movement is determined by calculating weights for each province and moving the units to more valuable provinces. It is a widely used agent for comparison and almost all bots use it in their tests.

4 Nomenclature

Before describing our model, we must formalise the description of the map. The map is composed of provinces that are represented by set $\mathcal{P}$. The ones that are supply centers are represented by set $\mathcal{P}_c$, and the ones that are home supply centers by set $\mathcal{P}_h$, that is to say, $\mathcal{P}_h \subset \mathcal{P}_c \subset \mathcal{P}$. These symbols are used to refer to supply centres owned by a power. Regarding province connectivity, we define $\mathcal{V}_p$ as the set of neighbouring provinces of province p that a unit of some type can move to.

5 Agent Description

5.1 Architecture

Darkblade is a multi-agent system with agents organised in a 2-layer hierarchy. Each owned unit is assigned an agent, named *General*, that is responsible for proposing orders for its unit. These orders are submitted to the agent at the top layer, named *President*, that is responsible for calculating the most profitable order combination. The architecture is organised in three modules. The Strategic module contains the President, the Generals and the data structure where generals put their orders and the president fetches them. The Knowledge and Belief Base module only contains data about board state, personality parameters and diplomatic relations. This data is used by both types of agents. A third module is responsible for communication with the DAIDE server.

Figure 2 shows the architecture of our agent with threads pictured as ovals and data structures as rectangles with round corners and solid lines. Arrows indicate information flow.

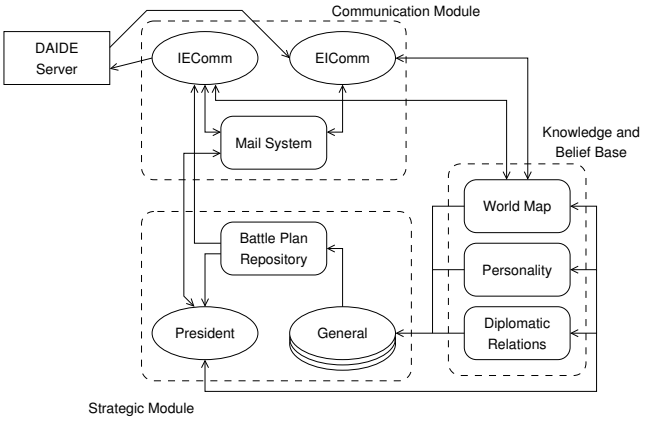

Fig. 2. Darkblade architecture.

5.2 Personality Traits

Personality traits applied to a Diplomat were first suggested by Kraus in [4] and later [12]. The idea behind using personality traits is to implement a set of parameters that condition the agent's behaviour, i.e., how the agent feels and acts about what it perceives. These parameters can be used to assess the threat of an unit moving towards us, to decide whom to attack or ally with, etc.

The model chosen for personality traits is Norman's Five Factor Model [13]. Norman's model is simple and efficient. It has also been used as an Artificial Intelligence personality model [14]. Application of Norman's taxonomy of personality attributes to Darkblade yields the following traits:

Aggressiveness[3] The desire to conquer new territory. A personality with low aggressiveness tends to be more defensive and conquer only when it is sure of success and never at the expense of a possible defensive loss.

Conscientiousness Methodical and reliable behaviour. A conscientious agent follows a defined plan and doesn't change its plan at the first obstacle. After choosing a target it tends to focus on it instead of constantly changing its goals.

Agreeableness Capacity for cooperation with others, trusting. A disagreeable agent tends to trust only himself and be suspicious of others.

Extroversion Capacity to create friends and enemies faster. A high extrovert agent quickly makes a friend of someone that has moved away from their common front, but just as quickly will be viewed as an enemy if he gets closer. On the other hand, an introvert agent is slower at making both friends and enemies.

Neuroticism Associated with uneasiness, neurotic and paranoid behavior. A neurotic personality will try to react to the simplest sign of threat from an opponent.

5.3 Strategic Evaluation

In order to tackle the problem of searching a good set of orders for a turn, two types of values are used: province value and unit threat. Province values are represented in two dimensions: one that distinguishes land provinces from sea provinces, and another that distinguishes between owned provinces from not-owned provinces. Unit threat history keeps track of unit movement. Province values of $\{p\} \cup \mathcal{V}_p$ influence the probability of a unit in p to move or to hold. These probabilities add up to form the opposing strength, S_p, of a power in province p.

This approach is similar to swarm systems that are based on the collective behaviour exhibited by social insects [15]. Not-owned provinces emanate a pheromone that attracts an agent units, while owned provinces emanate a pheromone that attracts units of a defensive agent. Likewise, the threat of adversary units, also attracts units of an attacking agent. The propensity to either attack or defend is a function of Darkblade personality.

Province Value Province values, v_p, are calculated using the following equation:

$$v_{p_i} = \sum_{\{p_j : d(p_i,p_j) \leq 2\}} \frac{b_{p_j}}{a^{d(p_i,p_j)}} \tag{1}$$

where $d(p_i, p_j)$ is a function that returns the length of the shortest path between provinces p_i and p_j. Parameter a influences the maximum value of v_p. Constant b_p represents the base value of a province. Since home supply centres have a strategic value superior to normal supply center (recall that they are the ones where the agent may build new units), their base value is the highest. We limit propagation to provinces that are at most 2 nodes from the origin, otherwise, province values would be homogeneous. A province base value is defined as follows:

$$b_p = \begin{cases} 2 & \text{if} \quad p \in \mathcal{P}_h \\ 1.5 & \text{if} \quad p \in \mathcal{P}_c \setminus \mathcal{P}_h \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Province values depend on two parameters: 1) ownership; 2) unit type. Owned provinces inside the territory should have a lower value than borderline provinces. On the other hand, not-owned provinces should be valued higher and Darkblade must calculate a set of orders to gain them. Recall that armies can only be located in inland or coastal provinces, while fleets can be located in coastal and sea provinces. Therefore some provinces have no value for some units. The bottom line is we have four distinct v_p values for each p.

Each v_p corresponds to a potential field and higher potentials attract Darkblade units. This field decays with the distance other provinces are from province p. Therefore, parameter a must be greater than 1. This parameter also represents the fact that units that are far away have more provinces to move to. In the present case we used $a = 4$.

Unit Threat Unit threat in a province is also calculated through a similar equation to (1) in that the base value of a unit is propagated to adjacent provinces. However a threat of a fleet does not propagate to inland provinces and, likewise, the threat of an army does not propagate to sea provinces. Propagation is done only to provinces a unit can move to. Since units have the same strength, 1 is used as the base value. Unit threat represents the strength a power has in moving towards or holding a certain province.

Hold or Move Probability The probability of a unit u holding or moving to province p is linearly proportional to the province value as seen in the following equation:

$$S_{u,p} = \frac{v_p - \underline{v_p}}{\overline{v_p} - \underline{v_p}} 0.5 + 0.3 \tag{3}$$

where $\underline{v_p}$ and $\overline{v_p}$ are, respectively, the lowest and highest province value of $\{p\} \cup \mathcal{V}_p$.

Opposing Strength in a Province This value is calculated by adding the probability of each unit of a power[4]:

$$S_p = \sum_u S_{u,p}. \tag{4}$$

This sum represents the most probable attack or defensive strength of a certain power over a certain province. Note that support to hold or to move requires the supporting unit being able to move to, respectively, the unit's province or the unit's destination. Therefore this equation allows us to consider *support* orders. Units that, despite being adjacent to the province, are unable to reach it (e.g. a fleet in a coastal province can not reach an inland province) are not considered when determining a power's opposing strength.

This method has some disadvantages and obviously doesn't take into account all information acquired by the agent's perception. For example, threat variation could be used to better assess the probability of a power moving its units towards a certain province. Despite this, the function itself produced satisfactory results guessing some moves not only by other Darkblade instances but also from other agents.

Province Profit Province profit is made of two values: defensive, π_d, and offensive, π_o. The first represents how valuable an owned province is, and thus how much effort should be spent in order to keep it. The second represents the value of gaining one more province. These values are a function of unit threat, opposing strength (see (4)), and personality parameters. Due to its complexity we refer to [16] for more details.

[4] Recall that all units have the same strategic value in the game.

Success Rate When a *General* proposes a set of orders for its unit (either to hold or to move combined with support orders for adjacent units) it calculates a success rate. This is based on the number of units, n, supporting its unit and S_p:

$$\begin{cases} \frac{n}{S_p} & \text{if holding} \\ \frac{n}{1+S_p} & \text{if moving.} \end{cases} \tag{5}$$

Recall that whenever two units are ordered to move to the same province, the unit with the highest support wins.

Hold and Move Orders Whenever a *General* submits a set of orders, it assigns it a value that is based on defensive and offensive profit and the aggressiveness personality trait, O:

$$\pi_d(1-O) + \pi_o(1+O). \tag{6}$$

In order to decrease computational requirement of finding the best combination of orders, there is a threshold on the success rate of the submitted set of orders. This threshold increases as the agent gains more units. The *President* selects the combination of set orders that maximises the profit.

Retreat Orders When considering where to retreat to, we consider the province with the highest value, v_p.

Build Orders In *Fall* phase, when it is time to build or to remove units, we choose the type of unit to build or to remove based on two ratios: 1) number of army units versus number of fleets; 2) land versus sea provinces adjacent to home supply centres. The type of unit is chosen in order to keep the two ratios equal. This tries to model the fact that some powers are more prone to sea expansion while others by land.

6 Experiments

6.1 Setup

We ran different experiments with Darkblade against itself and other agents designed to work with DAIDE Server. When running the experiments a few problems arose so most tests had to be launched one at a time to make sure all were executed properly. Some of these problems include random disconnects, messages getting lost or, less frequently, clients failing to establish a session with the server, usually when more than one client started a connection at the same time. Game duration was a factor that limited the amount of tests performed. Some games, due to Darkblade's nature of play, lasted past year 1950 (100 game turns) and, on later stages of the game, the amount of units in the board resulted

in a bot taking as long as 15 minutes to issue its orders. These issues limited the number of tests done.

We chose two of the agents introduced in Sec. 3: Diplominator and DumbBot. Diplominator for being one of the latest agents and using similar evaluation processes as Darkblade. DumbBot was chosen since it is used by other agents developed for DAIDE, such as HaAI [8] and Shaheed's Diplomat [17], and thus allows us to indirectly evaluate Darkblade's performance against other agents which were unavailable for tests. Another factor that made us choose these two agents is the fact that neither of them supports Convoys, just like Darkblade. Despite the use of Convoys being sporadic, for some powers it is one of the most profitable moves in the long term. For example, England, when it establishes itself in the north of Europe, convoying armies into the main land will allow it a faster win than relying exclusively on conquering coastal supply centres. This is the reason why England is one of the powers that produces most wins in Human vs Human matches [18].

Bot fitness was measured using a point system defined as:

10 Won the game
7 Survived with 10 or more supply centres
4 Survived with more than its initial supply centres
1 Survived with the same or less than its initial supply centres
0 Was eliminated before the end of the game

For each group of bots being evaluated, a certain number of games were performed assuring that each bot played with all powers at least once.

6.2 Results

The first set of experiments consisted in searching the set of personality parameters that maximised Darkblade fitness. In these experiments only Darkblade bots where involved. After this phase, the two best Darkblade configurations played with DumbBot and Diplominator. Bot instances were 1 DB44, 2 DB42, 2 DumbBot and 2 Diplominator. A total of 23 games were performed with this bot setting. Selected Darkblade personality parameters are presented in Tab. 1. Average points per game obtained and power played by each bot are shown in Tab. 2.

From these results it is clear that Darkblade outperforms some recent bots. As for bots that were not available, but results against DumbBot are available, Tab. 3 shows the win ratio between these bots using DumbBot as benchmark. A value of x means that the corresponding bot has x times more victories compared to DumbBot. For instance BD42 has many victories as DumbBot, while the number of victories BD44 has is 50% superior to the victories achieved by DumbBot.

	DB42	DB44
Aggr	0.70	0.80
Cons	0.70	0.80
Agre	0.60	0.80
Extr	0.50	0.80
Neur	0.30	0.80

Table 1. Personality parameters of Darkblade instances.

	DB42	DB44	DumbBot	Diplominator
AUS	1.13	3.67	0.17	0.00
ENG	4.50	5.50	4.71	2.57
FRA	6.14	7.70	1.71	3.43
GER	2.43	3.50	0.00	0.20
ITA	3.00	3.00	1.43	0.57
RUS	5.86	3.50	3.33	3.00
TUR	4.50	4.00	1.67	3.14
Average	3.89	4.17	1.87	1.96

Table 2. Results from games between Darkblade and selected bots. Average points per power for each bot.

Agent	**Ratio**
DB44	1.53
HaAI Berserk	1.25
DB42	1.00
HaAI Vanilla	0.93
Diplominator	0.85

Table 3. Bot win ratio comparison.

7 Conclusions and Future Work

We have presented a multi-agent system capable of playing the game of Diplomacy, through the DAIDE environment. Strategic evaluation was based not only on province value, but also on unit threat and threat history, an innovation compared to previous approaches. There are already some results using unit threat but not with threat history which has been also used to detect who are the most attacking or defending adversaries. Games with available bots [10] and comparative results with other bots [8] showed the merits of our approach. Still, there are some shortcomings, as *convoy* orders were not considered. Diplomatic relations lack in our work. As such, future work must include all types of orders and tackle the different diplomatic message levels that DAIDE provides. These should be thoroughly tested instead of the few tests reported in [12].

Regarding the applicability to other games, the potential field, that we have designed for each province, can be used in games where territory is an important factor. As an example we have the game of Risk which is also a n-player game.

References

1. Schaeffer, J., Burch, N., Björnsson, Y., Kishimoto, A., Müller, M., Lake, R., Lu, P., Sutphen, S.: Checkers is solved. Science **317**(5844) (2007) 1518–1522
2. Loeb, D.E.: Stable winning coalitions. In Nowakowski, R.J., ed.: Games of No Chance. Cambridge (1996) 451–471

3. Duffy, J., Feltovich, N.: Do actions speak louder than words? An experimental comparison of observation and cheap talk. Games and Economic Behaviour **39** (2002) 1–27
4. Kraus, S., Lehmann, D.: Designing and building a negotiating automated agent. Computational Intelligence **11**(1) (1995) 132–171
5. Loeb, D.E.: Challenges in multi-player gaming by computers: A treatise on the diplomacy programming project. http://www.diplom.org/Zine/S1995M/Loeb/Project.html (1995)
6. Hill, A.: The rules of diplomacy Available online at http://www.wizards.com/default.asp?x=ah/prod/diplomacy.
7. Kraus, S., Lehmann, D.: Diplomat, an agent in a multi agent environment: An overview. In: Computers and Communications, 1988. Conference Proceedings., Seventh Annual International Phoenix Conference on. (1988) 434–438
8. Håård, F.: Multi-agent diplomacy: Tactical planning using cooperative distributed problem solving. Master's thesis, Blekinge Institute of Technology (2004)
9. Shapiro, A., Fuchs, G., Levinson, R.: Learning a game strategy using pattern-weights and self-play. In Schaeffer, J., Müller, M., Björnsson, Y., eds.: Computers and Games. Volume 2883 of Lecture Notes in Computer Science., Springer (2002) 42–60
10. Webb, A., Chin, J., Wilkins, T., Payce, J., Dedoyard, V.: Automated negotiation in the game of diplomacy. http://www.doc.ic.ac.uk/project/2007/362/g0736203/TheDiplominator/ (2008)
11. Norman, D.: David's diplomacy ai page. http://www.ellought.demon.co.uk/dipai/ last checked March 2008.
12. Kraus, S.: Negotiation and cooperation in multi-agent environments. Artificial Intelligence **94**(1-2) (1997) 79–97
13. McCrae, R.R., John, O.P.: An introduction to the five-factor model and its applications. Journal of Personality **60**(2) (1992) 175–215
14. Talman, S.: The adaptive multi-personality agent. Master's thesis, Bar Ilan University, Ramat Gan, Israel (2004)
15. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Santa Fe Institute Studies in the Sciences of Complexity. Oxford University Press (1999)
16. Ribeiro, J.: Darkblade - an artificial inteligent agent for diplomacy. Master's thesis, University of Aveiro (2008)
17. Shaheed, J.: Creating a diplomat. Master's thesis, Imperial College, UK (2004)
18. Windsor, P.D.: Geography is destiny, how the standard map dictates fortunes and strategies. http://devel.diplom.org/Zine/F1999R/Windsor/dipmap.html (1999)