STEPS TOWARD BUILDING A GOOD AI FOR COMPLEX WARGAME-TYPE SIMULATION GAMES

Vincent Corruble, Charles Madeira
Laboratoire d'Informatique de Paris 6 (LIP6)
Université Pierre et Marie Curie (Paris 6)
4 Place Jussieu
75252 Paris Cedex 05
France
{Vincent.Corruble,Charles.Madeira}@lip6.fr

Geber Ramalho
Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851, Cidade Universitária
50732-970 Recife, PE
Brazil
glr@cin.ufpe.br

KEYWORDS

Game AI in Wargames, Terrain Analysis, Representations for Learning

ABSTRACT

One of the key areas for the application of Artificial Intelligence to the game domain is in the design of challenging artificial opponents for human players. Complex simulations such as historical wargames can be seen as natural extensions of classical games where AI techniques such as planning or learning have already proved powerful. Yet the parallel nature of more recent games introduce new levels of complexity which can be tackled at various levels. This paper focuses on the question of finding good representations for the AI design, which implies finding relevant granularities for the various tasks involved, for a popular historical wargame. This work is based on the partially automated use of the rules of the game, as well as some common sense and historical military knowledge, to design relevant heuristics. The resulting gain in representation complexity will help the application of techniques such as Reinforcement Learning.

INTRODUCTION

A type of computer games that has been gaining significant popularity over the years lets 2 or more opponents confront each other via the manipulation of a number of units on a given terrain. Sounds familiar? Of course, this description is so general that it encompasses age old games such as chess. What we are interested in here are strategy games which range from real time action-oriented games such as Age of Empires (Microsoft) to intricate historical simulations and wargames such as Sid Meier's Gettysburg (Firaxis) or Talonsoft Battleground series. The innovation in this new type of games, from the point of view of AI and complexity, is both quantitative as well as qualitative.

Quantitatively, they show an increased complexity by letting players manipulate high numbers of units (typically in the hundreds, if not thousands). Additionally, these units have open to them a high number of possible actions, depending on their characteristics, which fall in various categories such as movement, combat, or building activities for some of them.

Moreover, the physical space on which they move is much larger. While chess has 64 positions, and backgammon 24, the new games we are interested in involve 2-dimensional grids which extend over hundreds of squares (or hexagons) in each direction, so the total number of positions is in the tens of thousands.

Despite the huge quantitative scale-up required to use existing AI techniques on these new problems, the main source of complexity is actually elsewhere. While traditional games usually let the player select one (or a very small number of) unit(s), and then select an action for it, large modern simulations replicate the parallel nature of the system they simulate: each turn, all of the units can be moved (or take other actions) simultaneously. Therefore, while the branching factor of most traditional games increases linearly with the number of units, its increase is exponential in our games. In practical terms, that means that the standard versions of popular AI techniques (such as planning or learning) [Newell & Simon, 1965, Samuel, 1959, Lenat 1983, Meyer et. al., 1997, Sutton & Barto, 1998] are rendered irrelevant because of the complexity involved here. In this paper, we investigate how a careful examination of a game, good choices of representation (as well as possible innovations in the algorithms themselves), can help to circumvent these limitations.

In the remaining of this paper we will focus on one specific commercial game series named Battleground (Talonsoft®, designer John Tiller). It is a turn-based game considered as one of the best simulations of Napoleonic battles. On this application, we will expose a number of research directions under investigation, all aiming at dealing with the complexity of the game so as to make it amenable to efficient AI techniques. This paper focuses mainly on the issue of finding good representations, using available sources of knowledge about the problem, Its intended impact is therefore on the initial stage of AI design. It can be seen as a complement to other active research directions in the field of machine learning which work on the learning algorithms themselves to deal with higher complexities, e.g. work on learning within a hierarchy of goals [Dietterich,

2000] or work on using function approximators such as multi-layer perceptrons to deal with large state spaces [Tesauro, 1995].

THE NAPOLECTRONIC PROJECT FOR AI DESIGN

The Battleground (Talonsoft®) series of wargames is a turn-based game considered as one of the best simulations of Napoleonic battles. It aims at a good historical accuracy, with detailed maps, orders of battles, combat resolution, etc. while retaining some gameplay value (though it would certainly not be a hit among "action oriented" players). The environment provided by this simulation constitutes the testbed of our *Napolectronic* project, an AI endeavour to provide human-level computer opponents to strategy/simulation game players (Corruble, 2000).

The battleground series reproduces the historical order of battle. It models units at the level of battalions, and organizes each game turn in two parts composed of a number of phases. The attacking side can move all of its units, then the defendant can fire defensively, the attacker fires, then the attacker's cavalry can charge, then the attacker can melee the defender's units which are in contact. Then for the second part of the turn, the attacker and defendant switch roles. Each scenario is defined by a fixed number of turns (10 to 52), each turn simulating 15 minutes of real time. The units move on a historical map composed of hexagons (each hexagon representing 100 meters) and can assume different tactical formations (line, column, limbered or unlimbered for artillery,…). Moreover, each unit is also characterized by its quality, fatigue level, ammunition level, etc.

Each sides aims at controlling a number of key positions by the end of the scenario. A number of points is associated with each one of these key positions, and the final scores are calculated based on these points and the losses suffered by each army.

The success of many simulation games results from the feeling of immersion into a complex world that they provide for the user. In order to obtain this feeling, the game designer must balance two notions which could seem contradictory. The player needs to have a lot of control on the evolution of the simulated world (so that he can feel engaged in it) yet he/she must be somewhat overwhelmed by its complexity and should be unable to grasp its entire depth all at once. This necessary combination of high controllability and richness/depth justifies the evolution toward highly complex simulations, which have also, for the player interested in history, the advantage of becoming more realistic. This highly complex modelling is a given of the game and a natural approach to the design of an AI for such games is to use this highly detailed model of the system being simulated as the basic representation to do some automated reasoning, some planning, or some learning. Yet, because of the complexity involved, typical methods (let's say for example Reinforcement Learning) cannot obtain satisfactory results based on this representation. So a first step in the design of the AI is to find a granularity of representation which suits well the task at hand. There are a number of difficult points to address in that respect:

- For a complex game, there are a number of tasks which involve reasoning at various levels (strategic vs. tactic; long-term resource management vs. short-term timing of low-level actions,…). A good AI should therefore have various representation granularities, each one adapted to the task at hand. This issue of representation is also directly linked to the issue of whether decisions should be taken centrally or in a distributed manner. We will not explore directly this issue in this paper.

- A representation with an appropriate granularity, needed for strategic (or "high-level") reasoning, has to be constructed automatically or semi-automatically, as an abstraction of the low level representation of the simulation. This is in itself a complex problem, maybe actually the central problem for the building of a complex AI for games. Fortunately, because of the historical simulation aspect of the game, we can use some knowledge about the domain (here military decision-making in Napoleonic times), a detailed analysis of the rules of the game, or indeed simple common-sense, to guide us toward that goal. In the next section, we will present briefly work done to partially automate this process of building a relevant abstract representation, both in the action space (what can be done?), and in the state space (what is the situation?).

CONSTRUCTING HIGH-LEVEL REPRESENTATIONS

Abstraction in the Action Space

As we saw earlier, most powerful AI techniques such as learning or planning are very sensitive to the size of the action space. Because the number of low-level actions available to each unit in our game is huge, one can naturally understand that any reasoning at a tactic or strategic level needs to be tackled at a higher, more abstract level. This is particularly true in the field of movement. A commander should not have to specify the exact path of every given unit. Instead it should be able to give a position as a goal, and to specify a mode for this movement reflecting the tactical situation. We carried out some experiments following this approach. The modes that we have experienced with are:

- Speed only: minimize the time taken to reach the goal

- Stealth: minimize the risk of being spotted and fired at by the enemy

- Safety: minimize the risk of being intercepted by the enemy

Speed is an easy problem to treat, since we know of the movement cost associated with each terrain type and unit type combination. The straight application of A* using the straight line distance as admissible heuristics, works perfectly well.

An interesting challenge here is for the AI to discover how to implement the other movement modes. This has been done first by characterizing the static version of these concepts, then by applying A*, with a heuristic function that covers both the geographic distance the static cost of the mode.

Stealth is obtained when a unit moves through locations which are out of sight of enemy units. Therefore knowing for sure whether a potential path guarantees that a unit will be stealthy would require that all enemy units are visible. The fog-of-war option of the game, which makes for a much more realistic simulation, has as a result that this is not the case. Therefore, we defined statically that there is a heuristic stealth cost associated with each location which is proportional to the number of other locations from which it can be seen (these are susceptible to be occupied by enemy units). For example, going through a forest is very stealthy since there a unit can only be spotted, or fired at, from adjacent positions.

Safety is obtained by keeping a distance from enemy units. The bigger distance the less likely this enemy unit is from moving to intercept. Moreover, the cost associated takes into account the strength of the threatening units, because the stronger units are more likely to attack, and more likely to cause serious problems if they do.

Lastly, initial work has been done to combine these various modes of movement. So far this has been done simply by proposing a heuristic function which is a linear combination of the previous ones. Later, we envisage using some more subtle combination of modes, which would be the product of strategic reasoning and consider the motion of unit as a multi-objective decision problem.

In the examples of Figure 1, one can see the paths suggested for a single basic movement order, but with distinct movement modes. The speed mode favors a direct movement avoiding the forest hexagons, the stealth mode encourages motion through the relatively hidden valley, the safety mode favors remaining away from the enemy units and going through the forest, and the combined mode encourages an even wider circle going through the forest and using another valley for stealth.

Abstraction in the State Space

Any significant and tractable tactical or strategical reasoning needs to be able to refer to locations or situations at an appropriate level of abstraction. Hence a leader should be able to tell a subordinate to "take his troops to Village V using the road going through forest F to the south of the body of enemy troops E. To facilitate this process, we have used simple algorithms inspired from the field of Artificial Vision to automatically define relevant regions, which are group of adjacent hexagons which share a relevant

Figure 1: Paths obtained for the same basic movement order (initial and goal positions), first with the 3 basic modes, then with a simple combination mode. The initial position is circled in red. The proposed path is given by the numbers appearing on some hexagons. Each number shows the cumulative cost associated.

property, such as terrain type (e.g. forest), altitude, or in tactical terms (group of friendly, or enemy troops, waiting or moving together). Figure 2 show an example of tactical regions symbolizing the zones of control of the French troops (in blue) and of the Russian troops (in green).

Additionally, these abstract regions can be used to carry out some intelligent reporting describing in high level terms the major events and the evolution of the situation at each turn. This is the first application of this work that we are now developing.

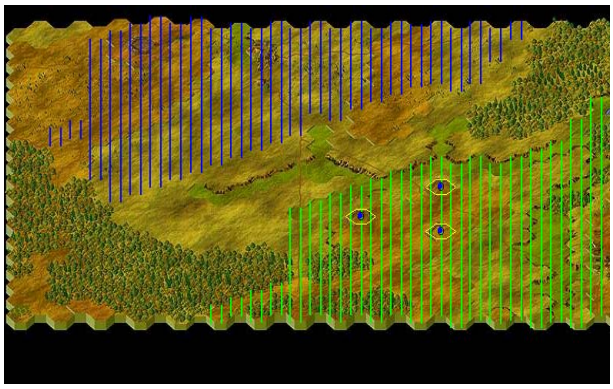

Figure 2: Inferring each side's zones of control

Going further than the description of a situation, it is interesting to go deeper into the automated terrain analysis with the idea of discovering interesting tactical concepts. We have first focused on an important subproblem: where one should locate artillery units for defence or attack. This is particularly crucial since artillery movement is very limited and cannot therefore be easily readjusted while in the thick of the action.

We have taken the approach of applying local heuristics directly making use of the rules of the game. In a fashion similar to the one used at the beginning of this paper, each heuristic function reflects a different concern or goal a leader should have in locating his artillery, the most important idea is that the chosen location must balance the effectiveness of the artillery fire and the protection to the unit. In Figure 3, we show a part of the map where each hexagon is covered by a coloured dot. The "warmer" the colour, the better field of fire the location has. Hence the inside of forests are in blue (bad field of fire) while the top of a hill is red (excellent field of fire). It is important that the colour (the heuristic function) is calculated directly through the application of the rules of the games: the system was not given any *a priori* military knowledge.

Figure 4 shows the same type of picture but the colour represents the amount of protection offered by the local terrain. Here ridges appear as good locations because their elevated position offer defence bonuses according to the combat rules, and moreover, infantry units can be placed in front of these positions, where they can protect the artillery. According to this heuristic function, hexagons inside forests would be good locations for artillery (because they are indeed well protected from enemy fire).

In Figure 5, basic heuristics (including the 2 previously presented) are combined to provide a global evaluation showing which positions are interesting candidates for locating artillery. We can see with the green dots that ridges are always sensible locations. This is a very interesting result because it is perfectly consistent with well known military knowledge. So we can expect that applying the same approach to other subproblems will let us find automatically some other tactical concepts relevant to this simulation.

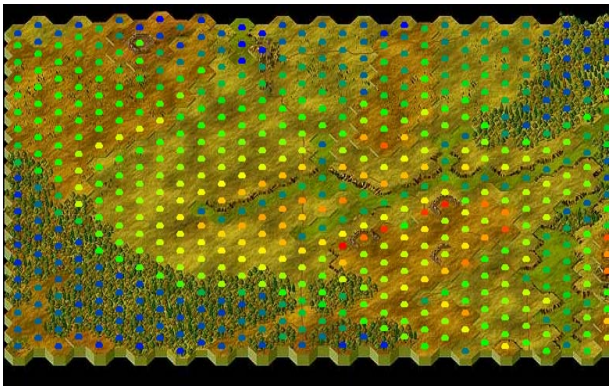

Figure 3: Hexagons in the field of fire

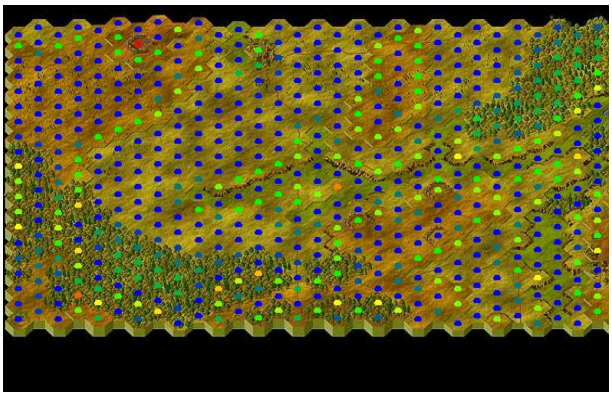

Figure 4: Protection offered by local terrain

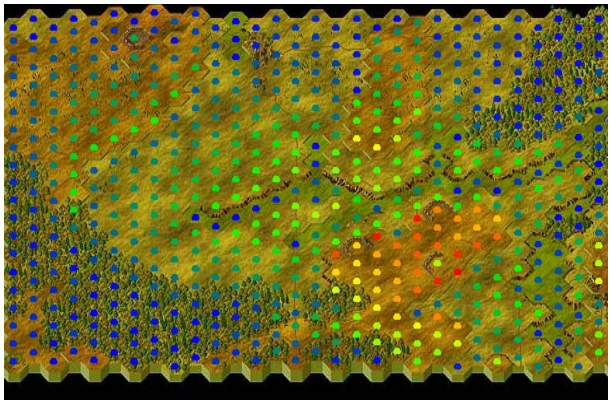

Figure 5: Combination of basic heuristics

CONCLUSION

In this paper, we have presented some experimental work aiming at finding good representations for a strategy game simulating Napoleonic battles. This is seen as an essential step to be able to use mainstream AI techniques, such as learning and planning, for the design of a human – level AI opponent. We have explored how abstraction in the representation can be carried out along the dimension of the action space, so that leaders can give high-level, tactically meaningful orders, and along the dimension of the state space, so as to be able to describe situations in concise and meaningful terms. We are now completing this work on representation before we start on applying and adapting techniques such as Reinforcement Learning and Planning. In parallel, we work on abstraction along the temporal dimension, so as to provide meaningful game summaries, and to obtain insights on the key events and tactical turning points of a scenario.

ACKNOWLEDGEMENTS

Charles Madeira's PhD work is funded by a scholarship from CAPES, Brazil.

A number of CS students of University *Pierre et Marie Curie* played active parts in this project. They include in particular for the experiments described in this paper Jean-Claude Thiout and Arlindo Dos Santos 4th year students in 2001, and Master students in 2002.

AUTHOR BIOGRAPHY

Vincent Corruble was born in Rouen, France, and obtained graduate degrees in Engineering, Systems Engineering, and Artificial Intelligence from the Ecole Centrale de Lille, the University of Virginia, and the University Pierre et Marie Curie (Paris 6) respectively. He is currently Assistant Professor at the LIP6, the Computer Science laboratory of the University Pierre et Marie Curie. His past and current research covers areas such as pattern recognition, data-mining, machine learning and machine discovery. His main application areas for knowledge discovery are medical research, web user modelling, and computer games.

Charles Madeira was born in Natal, Brazil. He obtained a Master's degree in Computer Science from the Federal University of Pernambuco (UFPE), Brazil and is now pursuing a PhD at University Pierre et Marie Curie (Paris 6) on the topic of AI and computer games.

Geber Ramalho was born in João Pessoa, Brazil. He obtained graduate degrees in Computer Science from the University of Brasilia (UNB), and from University Pierre et Marie Curie (Paris 6). He is currently Assistant Professor at the Federal University of Pernambuco (UFPE), Brazil, where he carries out research in the areas of symbolic artificial intelligence, autonomous agents, multiagent systems, computer music and computer games.

REFERENCES

Corruble, V. 2000. "AI approaches to developing strategies for wargame type simulations". AAAI Fall Symposium on Simulating Human Agents. Cape Cod, USA.

Dietterich, T. G. Hierarchical reinforcement learning with the MAXQ value function decomposition. Journal of Artificial Intelligence Research, 13:227-303, 2000.

Lenat, D. B. 1983. "Eurisko: A program which learns new heuristics and domain concepts". Artificial Intelligence, 21.

Meyer, C., J.-G. Ganascia, and J.-D. Zucker. 1997. "Learning Strategies in Games by Anticipation". International Joint Conference on Artificial Intelligence (IJCAI), Nagoya, Japan.

Newell, A., & Simon, H.A. 1965. "An example of human chess play in the light of chess playing programs". In N. Wiener and J.P. Schade (Eds.), Progress in biocybernetics (Vol. 2, pp. 19-75). Amsterdam: Elsevier.

Samuel, A. 1959. "Some studies in machine learning using the game of checkers". IBM Journal of Research and Development, 3(3):211-229.

Sutton, R. S., and Barto, A.G. 1998. "Reinforcement Learning, An Introduction". MIT Press.

Tesauro, G. 1995. Temporal Difference Learning and TD-Gammon. Communications of the ACM, March 1995 / Vol. 38, No. 3