The Second Annual Real-Time Strategy Game AI Competition

Michael Buro, Marc Lanctot, and Sterling Orsten
Department of Computing Science
University of Alberta, Edmonton, Alberta, Canada
{mburo|lanctot|sorsten}@cs.ualberta.ca

KEYWORDS
Real-time strategy games, ORTS, real-time AI systems

ABSTRACT

Real-time strategy (RTS) games are complex decision domains which require quick reactions as well as strategic planning and adversarial reasoning. In this paper we describe the second RTS game AI tournament, which was held in June 2007, the competition entries that participated, and plans for next year's tournament.

Introduction

Creating computer-controlled agents for Real-Time Strategy (RTS) games that can play on par with skilled human players is a challenging task. Modern game AI programmers face many obstacles when developing practical algorithms for decision-making in this domain: limited computational resources, real-time constraints, many units and unit types acting simultaneously, and hidden state variables. Furthermore, market realities usually place limits on the time and manpower that can be spent on AI in a commercial game.

Two common ways to circumvent these problems are simply to "cheat": give the AI agent access to more information than the human players are given, and to encode pre-processed human knowledge in the form of scripts. Aside from the obvious issue of unfairness, there are other problems with this solution. The scripts created for the AI hard-codes behaviors, i.e., responses to pre-determined observations, result in predictable decisions. In addition, there is little, if any, long-term planning done during the game.

Today, gamers have higher demands and expectations. They prefer to play online against other humans but not all gamers have access to high-speed Internet connections, and some prefer to play alone. Creating good RTS Game AI is therefore an interesting, challenging, and worthwhile research venture.

There are many scientific motivations for and expectations of research in RTS game AI [2]. Researchers are now developing learning and planning algorithms in this domain [6, 5, 4, 7]. However, different researchers focus on different specific subproblems. The relative quality of the techniques is difficult to assess because they are not exposed to the same empirical evaluations. The spirit of the annual RTS Game AI competition is to encourage the development of these techniques in a common and competitive environment. The quality of various methods can by judged by comparatively measuring the performance of their implementations. Consequently, general conclusions can be drawn from the outcome of the tournament.

Competitions have proved to be excellent way to encourage advancement in AI research, such as when IBM's Deep Blue defeated reigning chess champion Garry Kasparov [3]. Other examples include the RoboCup competition, which has improved techniques in the fields of robotics and multi-agent learning, a computer Go server on which the World's strongest 9x9 Go programs compete, and the AAAI General Game-Playing competition. The purpose of the annual RTS game AI competition is to drive the same progress in real-time AI research.

RTS Games and ORTS

RTS games are tactical simulations involving two or more players, each in control of a growing army of units and bases, with the same goal of conquering the region. Players are faced with a multitude of difficult problems: controlling potentially hundreds of units, limited terrain visibility, resource management, combat tactics, and long-term planning, all of which must be handled simultaneously while considering what the opponent might be doing. Building an AI agent is certainly an ambitious undertaking.

At the very least, a planning agent must control units whose actions lead to potentially varying circumstances. Different unit types can have a variety of different abilities. Coordinating these units both spatially and temporally, in such a way as to maximize their effectiveness, is a nontrivial task. A further complication is that the agent may be subjected to, and have to compensate for, imperfect information (the so-called "fog of war").

A key problem in RTS games is that many decisions necessary for victory (expanding, launching an attack, etc.) carry significant risk. Even smaller decisions such as how to deploy resources and forces can have great consequences. Making these decisions in the wrong circumstances could lead to certain failure.

ORTS

The Open RTS game engine ORTS, available from `www.cs.ualberta.ca/~mburo/orts`, provides a flexible frame-

work for studying AI problems in the context of RTS games. The ORTS engine is scriptable, which allows for game parameters to be easily changed, and new types of games, or subsets of existing games, to be defined.
Units in ORTS are simple geometric primitives located on a fine grid. Map terrain is specified by a grid of tiles of different types and heights. Objects may travel at an arbitrary heading, with collisions handled on the server. Unit vision is tile-based, with different units having a sight range that determines how many tiles away they can see. The vision model also supports "cloaked" units which can only be seen by "detectors".
All ORTS components are free and open-source. Along with the server-client framework, this allows users to create their own AI components capable of acting autonomously or to augment a human player.

The AIIDE RTS Game Competition

The second AIIDE RTS Game competition took place between May 7th and June 1st, 2007. Tournament games were classified into four main categories: cooperative pathfinding, strategic combat, mini RTS game, and tactical combat. Competitors submit different entries for each category. Tournament games are run between the competition entries in the corresponding categories only. In turn, results are classified by game category.

Game 1: Cooperative Pathfinding

In Game 1 the goal is to gather as many resources as possible in a fixed amount of time. The player starts the game with a single base and twenty workers. The workers must travel to mineral patches distributed randomly about the map, mine from them, and return minerals to the main base.
The entire map, which includes the locations of mineral patches, is given to the player as the game starts. The map includes obstacles such as impassible terrain (plateaus) and indestructible roaming "sheep" to complicate the task. No information is hidden from the player, but since simultaneous actions get resolved in random order, there could still be some unpredictable consequences. The challenge then is to efficiently coordinate the paths taken by the agents, which involves avoiding both congestion and planning lag.

Game 2: Strategic Combat

In Game 2, the goal is to destroy as many as of the opponent's bases as possible in a fixed amount of time. Players start with five randomly positioned bases, with ten tanks around each. If all of one player's bases are lost, the game ends and assigns a loss to that player.
As in Game 1, no information is hidden from the player. Plateaus block line-of-sight attacks from tanks, and indestructible sheep roam the map randomly. The challenge in this game is to find attack strategies and formations that maximize the offensive advantage while minimizing the defensive disadvantage. This must be done in a scenario with spatial constraints, so planning when to concentrate or split forces appropriately is key.

Figure 1: Screenshot of Game 1

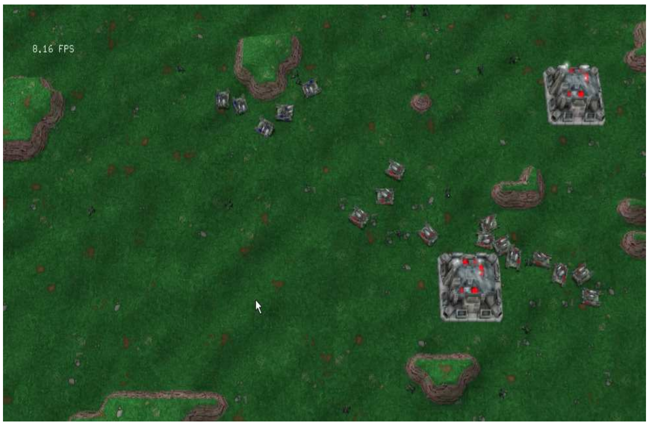

Figure 2: Screenshot of Game 2

Game 3: Mini RTS

Game 3 is a reduced version of a full RTS game. Players start with a single base and a few workers located next to a mineral patch. The only part of the map that the player knows about is what is currently observable by all of the units. The player must use minerals mined by the workers to construct barracks and/or factories, which are used to create marines and tanks. Tanks have greater attack range and power than marines, but also cost more to build.
The goal in this game is to get more points than the opponent. Points are awarded for gathering resources, constructing buildings, creating units, and destroying enemy units and buildings. A player wins automatically if the all of the opponent's buildings are destroyed.

Game 4: Tactical Combat

In Game 4 the goal is simply to eliminate as many of the opponent's units as possible. Players start the game

Figure 3: Screenshot of game 3

Figure 4: Screenshot of game 4

with 50 marines in random but symmetrically opposed locations. As in games 1 and 2, the entire map and positions of the opponents are known to both players. The map contains no minerals and no terrain obstructions other than roaming sheep. The objective is to find the best combat tactics to defeat the opposing force.

Tournament Setup

On May 7th, competitors were given access to the computers which were to be used during the tournament. The tournament itself was run between May 28th and June 1st. Results were announced June 1st; final results and videos were posted by June 5th.
Each machine was equipped with an Intel Pentium-4 2.4 GHz CPU and 512 MB RAM running 32-bit GNU/Linux with kernel version 2.4.33 and gcc 3.3.6. The tournament management software, which was developed last year, was reused to run a large number of games for each game category.
Authors had access to the tournament computers on which they could upload their programs to test them in individual protected accounts which were frozen just before the tournament commenced. Each participant was asked to send a "random" integer to a member of the independent systems group which also set up the tournament accounts. These numbers were then exclusive-or combined to form the seed of the random number generator used for creating all starting positions. This way, no participant was able to know beforehand what games would be played.
In what follows we present the tournament results and briefly describe the best entries in each category. Videos and more detailed program descriptions are available from the tournament web site [1].

Game 1 Entries

Entries were judged on the average number of minerals gathered after ten minutes over two hundred fifty games. The following table summarizes the results; units are amount of minerals mined.

Entry	Minerals
Warsaw University, Poland (Team B)	6837.5
University of Michigan, USA	6784.9
University of Alberta, Canada	6651.6
Gábor Balázs	5935.6
Naval Postgraduate School, USA	5425.5
Warsaw University, Poland (Team A)	2609.7
Universidad Carlos III de Madrid, Spain	2444.4

Warsaw University (Team B)
Team Leader: Michal Brzozowksi

This entry assigns workers to accessible corners of the closest minerals. Pathfinding is done by searching on a graph based on the terrain. This graph is modified to have one-way edges in key locations. This allows for the formation of lanes between mineral patches and the command center, reducing collisions and allowing this entry to gather more minerals on average than any other entry.

University of Michigan
Team Leader: John Laird

This entry mainly focuses on low-level systems described in [8]. It uses a mining coordinator to assign workers to mineral patches, a pathfinder with heuristics to assist in cooperative pathfinding, and a movement FSM with reactive rules to avoid local collisions with other workers or dynamic obstacles.

University of Alberta
Team Leader: Michael Buro

The entry enumerates routes between the command center and mineral access points, which are locations close enough to at least one mineral to mine. It assigns workers to routes, prioritizing shortest routes. This tends to cause short round trip times but high congestion which has to be resolved by local obstacle avoidance.

Game 2 Entries

Entries were judged via a round robin tournament with 40 games played per entry pair (320 games played per entry). The following table summarizes the results; the shown percentage is the proportion of wins out of the 320 games played by the entry.

Entry	Wins
National University of Singapore	98%
Warsaw University, Poland (Team B)	78%
University of British Columbia, Canada	75%
University of Alberta, Canada	64%
University of Alberta, Canada [1]	46%
Blekinge Institute of Technology, Sweden	32%
Warsaw University, Poland (Team A)	30%
University of Maastricht, The Netherlands [1]	18%
University of Michigan, USA	6%

National University of Singapore
Team Leader: Lim Yew Jin

This entry makes extensive use of influence maps to represent the strategic state of the map. It intelligently splits its forces into groups based on the situation. These groups attempt to hunt weaker enemy groups, and prioritise taking down units before buildings. Combat efficiency is maximized by lining units up at firing range from the convex hull of enemy groups.

Warsaw University (Team B)
Team Leader: Michal Brzozowski

This entry attempts to fire the most shots and be hit as little as possible by keeping units at the maximum distance from the enemy while still inside their firing range. Units do not advance further until their area is clean, however, they will "rotate" around the enemy position, making room for other allied units to enter firing range. Over time, this can encircle an enemy position and destroy it easily.

University of British Columbia
Team Leader: Zephyr Zhangbo Liu

This entry splits its forces into five squads, and assigns them various targets, such as command centers, enemy groups, or areas. The squads can change targets based on the situation, but not too frequently. Nearby squads can merge if they are not currently occupied. The entry uses clustering to analyse enemy positions, and can rescue its own command centers if they are under attack.

[1] 2006 Entry

Game 3 Entries

Entries were judged by playing 200 games. The results are summarized in the following table. Note that the performance of the University of Michigan's entry suffered from software problems which led to many automatic forfeits.

Entry	Wins
University of Alberta, Canada	89%
University of Michigan, USA	11%

University of Alberta
Team Leader: Michael Buro

This entry uses a hierarchical system of "commanders". Each commander controls multiple units and attempts to complete a specific goal. Commanders can spawn sub-commanders and operate at a specific level of granularity. The entry prioritizes aggressive exploration, expansion and monopolization of the map's resources, so as to inevitably produce marines and tanks faster than the enemy is capable of, and win via sheer numeric strength.

University of Michigan
Team Leader: John Laird

This entry uses a software layer to abstract both the information available and the actions that can be taken, to allow ORTS to be played by a SOAR agent [8]. The agent for this entry followed a plan of building up a force of marines, scouting, and attacking in groups. It is also capable of robustly altering its strategy to compensate for emergencies or unexpected situations.

Game 4 Entries

Entries were judged via a round robin tournament with 100 games played per entry pair (700 games played per entry). The following table summarizes the results.

Entry	Wins
National University of Singapore	99%
University of British Columbia, Canada	75%
Warsaw University, Poland (Team B)	64%
Warsaw University, Poland (Team A)	63%
University of Alberta, Canada	55%
Blekinge Institute of Technology, Sweden	28%
Naval Postgraduate School, USA	15%
University of Michigan, USA	0%

National University of Singapore
Team Leader: Lim Yew Jin

This entry lines up its forces just on the edge of firing range of the convex hull of the set of enemy units. This ensures that a maximum number of units can attack, while a minimum number of enemies can return fire. A complicated set of rules allows it to efficiently form a tight line formation around an enemy group. This entry is notable for its ability to quickly encircle and destroy enemy squads.

University of British Columbia
Team Leader: Zephyr Zhangbo Liu

This entry uses several small squads to attack the corners of the convex hulls of enemy groups. This lets several units come into range to attack a single enemy, while staying out of the firing range of other enemies. Rules ensure that squads are assigned to attack hull corners in intelligent ways. This entry is also able to quickly break down and destroy many types of enemy formations.

Plans for the 2008 Competition

There are many potential improvements that can be made to the annual RTS game AI competition. This coming year, we plan to address a few in particular:

Simplified Client Software Interface.
Recently, the AI system was restructured as a hierarchy of separate components. The commander interface currently issues commands to each component in a hierarchy, which in turn sends commands to lower level components. Many of the lower level components in the standard ORTS clients need simplification and refactoring. This way, all typical AI functions can be consolidated in one interface and complex behaviors can be compositions of these primary operations.

Opponent Modeling. We are considering adding game categories that allow entries to maintain data on disk across games. Any files created by the entries will be preserved for some proportion of the total number of games in a series against two players. This will allow for learning AI systems to adapt to their opponents, but not to the terrain.

Varying Game Parameters. In the current setting all game parameters such as unit speed and attack range are fixed. To encourage the development of more robust AI solutions we plan to add game parameter randomization, which at game start draws parameter values from specific distributions and therefore forces AI systems to adjust their strategy accordingly.

ORTS Development Roadmap

Several related items within ORTS are also scheduled to be implemented. One addition will be a tweakable graphical user interface. High-level AI behaviors will be attached to graphical components such as buttons and keys, allowing human players to send intelligent commands to a group of units. For example, to execute a spread attack with a group of units, the player will be able to add a customized command which will instruct a group of troops to do so very quickly, without the need to micro-manage their units. Ultimately, we plan to expose human players to the competition entries. Then, we will be able to compare the relative strengths of strategies used by human players versus the strategies employed by the competition entries.
The ORTS project will soon be following a regular release schedule. ORTS will be available in packaged form making it somewhat easier to install and manage. There will be more and better documentation; in particular, a comprehensive, instructional competition guide will be provided to next year's participants. Finally, we hope to provide better support for development and usage of ORTS under Microsoft Windows.

Conclusion

In this paper, we presented the software environment used for the 2007 RTS game AI competition, the results of the tournament, and plans for the future. Many interesting techniques and strategies were implemented and there has been a noticeable improvement in quality of AI techniques in these entries compared to last year's. There has also been more than a two-fold increase in teams and entries than the first competition in 2006. This development is encouraging and we hope the annual RTS game AI competition will continue to attract researchers to this fascinating and complex field.

REFERENCES

[1] The 2007 ORTS RTS Game AI Competition. `http://www.cs.ualberta.ca/~mburo/orts/AIIDE07/`.

[2] M. Buro. Real-time strategy games: A new AI research challenge. In *IJCAI*, pages 1534–1535, 2003.

[3] M. Newborn. *Deep Blue: An Artificial Intelligence Milestone*. Springer, 2003.

[4] S. Ontañón, K. Mishra, N. Sugandh, and A. Ram. Case-based planning and execution for real-time strategy games. In *Proceedings of the Seventh International Conference on Case-Based Reasoning (ICCBR)*, 2007.

[5] M. Ponsen, P. Spronck, H. Munoz-Avila, and D.W. Aha. Automatically generating game tactics through evolutionary learning. *AI Magazine*, 27(3):75–84, 2006.

[6] F. Sailer, M. Buro, and M. Lanctot. Adversarial planning through strategy simulation. In *IEEE Symposium on Computational Intelligence and Games (CIG)*, 2007.

[7] M. Sharma, M. Holmes, J.C. Santamaria, A. Irani, C.L. Isbell Jr., and A. Ram. Transfer learning in real-time strategy games using hybrid CBR/RL. In *IJCAI*, pages 1041–1046, 2007.

[8] S. Wintermute, J. Xu, and J. E. Laird. SORTS: A human-level approach to real-time strategy AI. In *AI-IDE*, 2007.