Complex Video Game AI Competitions at AIIDE’06

Michael Buro (mburo@cs.ualberta.ca)
David W. Aha (aha@aic.nrl.navy.mil)
Nathan Sturtevant (nathanst@cs.ualberta.ca)
Vincent Corruble (vincent.corruble@lip6.fr)

ABSTRACT

This document[1] describes our ideas for organizing a video game AI competition at AIIDE. We discuss our goals, software, issues, and tournament organization, and propose to hold the first such event focusing on real-time strategy games in 2006.

MOTIVATION AND GOALS

Complex video games are popular pastimes. They form a multi-billion dollar market and have begun to attract interest from the AI research community and developers of military training and simulation technology. (Wikipedia 2005) provides the following list of major video game genres: fighting, racing, role-playing, simulators, sports, strategy, first and third-person shooters, and city-building games. A common element of these game genres is fast-paced action that serious challenges to AI systems designed for playing these games well in the role of allies, adversaries, or neutral parties.

Game AI research has a long history in which it produced many successes in the area of classic two-person games, culminating for instance in strong chess, checkers, backgammon, and Othello programs. Game AI competitions have played a significant role in this development. Here we propose to organize an annual complex video game AI competition as part of a major AI conference. Our goal is to repeat this earlier success with classic games in an application area that currently lacks strong AI performance. While it is conceivable to organize competitions in all of the genres mentioned above, we restrict ourselves to real-time strategy (RTS) games for the first year’s event. For subsequent competitions we will consider broadening the scope.

RTS games, such as the popular Warcraft and Age of Empires series, model real-time multi-unit combat. In recent years research interest in RTS game AI has grown because these games pose many unique AI challenges that are also relevant to important other areas such as robot collaboration and military combat simulations. Example challenges concern spatial and temporal reasoning, acting under uncertainty in dynamic environments, the presence of a large number of interacting objects, and the need for abstraction and planning due to the predominance of actions with local effects. Most RTS games are highly configurable which allows us to tailor them towards particular research topics. For instance, we may at first want to partially restrict ourselves to simplified perfect information sub-games before addressing more complex research issues such as imperfect information.

Our main goals for this proposed RTS game AI competition are as follows:

- *Setting up an objective test-bed for measuring RTS game AI progress.* Currently, the significance of reported experimental research results is hard to judge, because no baselines have been established yet. RTS game competitions will help researchers to gauge the performance of their AI systems by means of direct and objective comparison with competing technologies. To help pinpoint progress, it will be necessary to organize competitions that focus on individual skills, such as pathfinding and local combat.
- *Boosting real-time AI research.* We argue that playing performance in RTS games hinges on several cognitive abilities that have only received comparatively little research attention (e.g. abstraction, spatial and temporal reasoning, and adversarial planning in real-time). By establishing RTS game competitions we hope to increase research interest in those areas to improve on the performance of classical heuristic search and planning approaches, which seem to be inadequate when applied to raw RTS game state representations. We believe that winning competitions will, in part, signify research excellence.
- *Increased technology transfer.* Improving RTS game AI strength is not the first priority for games companies; they traditionally focus on creating challenging and fun gaming environments in which AI fairness is not a requirement. Nevertheless, we think that commercial RTS games can also benefit from stronger AI systems which will result from competitions. Examples range from better pathfinding to delegating tactical tasks to AI modules to allow players concentrate more on strategic decisions. Similar technology transfer opportunities exist in the military sector, where smart training and simulation systems are in high demand.
- *Drawing additional attention to the AIIDE’06 conference and the topic area more generally.*

The target participants for the RTS game competitions are real-time AI researchers, game programmers, and experts on military simulators of computer generated forces.

[1] This document is based on minutes taken at a meeting of the authors at IJCAI’05.

WHY HOLD COMPETITIONS NOW AND WHAT IS NOVEL?

Although commercial RTS games have been around for at least ten years, no RTS game has permitted users to connect general AI systems to run RTS game AI competitions. By the advent of free software RTS game engines – most notably (Stratagus 2005) and (ORTS 2005) — this undesirable situation has recently changed. Both systems are licensed under the GNU General Public License which enables users to study source code and to contribute to the projects. Both engines are also mature and support the specification of new RTS games, which makes them especially interesting for competitions that are focusing on particular RTS game AI aspects.

Organizing RTS game AI competitions now has also become possible by hardware improvements that facilitate the development of high-performance real-time AI software. Finally, the growing research, military, and game company interest in RTS game AI calls for objective algorithm comparisons. RTS game competitions are one such measure.

The number of AI competitions is growing, and should create new challenges that drive research to new heights. However, to establish a new type of competition, we must examine existing ones and clarify how (1) it will sufficiently differ and (2) how it will add significant value. Here we discuss in turn how the RTS game competition will differ from existing AI tournaments:

- RoboCup Soccer (Simulation League (RoboCup 2005)). Some features of the RTS game genre exceed those in soccer, including: a possibly large variety of significantly different object types, a variable and possibly large number of game objects (hundreds rather than dozens), the need for resource allocation, potentially complex map topologies, and imperfect information in terms of initially unknown terrain, a small fraction of visible terrain ("fog of war"), and unknown opponents' base locations. Furthermore, RTS games can be played by individuals as well as teams.
- The General Games Competition (GGC 2005) and RTS games competitions share a key attribute: they both use a generic framework for defining games to be played. However, it is currently not practical to encode rules of RTS games in a logic-based language. Also, the aim of the RTS game competition is not (yet) to play arbitrary RTS games well, after being presented with the game rules and a brief deliberation period. Rather, our focus is on increasing the playing strength in fully specified RTS games.
- The Trading Agent Competition (TAC 2005) is focused on bidding in auctions and not concerned with multi-agent motion planning and combat.
- AI competitions for classic abstract games — such as chess, go, and bridge — are being organized regularly to measure the playing strength of machines (e.g. Computer Olympiad). RTS games are sufficiently distinct from slow-paced turn-based games. New techniques are needed to play large-scale simultaneous move games with imperfect information well.

In summary, the time for RTS game AI competitions has come.

RTS GAME SOFTWARE

As mentioned before, there are currently two RTS game engines that could be used: ORTS and Stratagus. The major reasons for preferring ORTS over Stratagus for RTS game AI competitions are as follows:

- ORTS implements a server-client system in which players have only access to the part of the game state that is visible to them. So, whatever software players connect, client-side hacks are impossible. In contrast, Stratagus is based on P2P technology, which in the context of RTS games means that every player is running the entire simulation and therefore has access to all state information, including enemy locations. We think that the P2P mode is unsuitable for competitions because it is almost impossible to verify that competing programs use only the part of the game state visible to them.
- ORTS is being maintained by an AI research group at the University of Alberta that is devoted to the development of RTS game AI systems, whereas Stratagus is a volunteer hobbyist project. One implication is that resolving ORTS software issues should take less time.

The ORTS server and basic client code are functional and will be tested extensively in the upcoming months. Work continues on its documentation and on the graphics client, which for computer competitions is only used as visualization tool and has no effect on the actual game play. Daily snapshots can be downloaded from the ORTS webpage.

PRELIMINARY GAME SPECIFICATIONS

The plan is to have three competition disciplines:

1. Resource gathering (single-agent, no fog of war). Given k workers starting close to a control center and a distant gold mine on random terrain, gather as much gold as possible within a fixed time period. To make the task more challenging the terrain is densely populated with randomly moving objects.

   In this scenario, workers are commuting between a building and a distant resource location. The AI issues to be addressed are: finding shortest paths, avoiding collisions with co-workers and moving obstacles, and picking locations to access the gold mine and the building. Pathfinding in ORTS is not trivial because the world is described in terms of circles (mobile objects) and line segments (edges of buildings and plateau borders); there is no notion of tiles.
2. Small-scale combat (adversarial, no fog of war). k vs. k tanks and n control centers located on random terrain. The task is to destroy as many control centers as possible within a fixed time period. In case of an equal number

of destroyed buildings, the number of destroyed tanks decides the winner.

In this setup we gauge strategic and tactical AI combat strength without having to first develop and economy or scout the map. Players start with more than one control center to discourage draws (i.e. by remaining close to base). AI issues to address include whether and how to split up one's forces, where to move groups, what base to attack when (i.e. scheduling), and how to engage enemy units effectively by concentrating fire (i.e. tactics).

3. A simple RTS game (adversarial, fog of war). This will include three unit types: workers who gather gold and construct buildings, marines, and tanks. Three structure/building types will be available: control centers, barracks that train marines, and factories that build tanks. Starting with k workers close to a control center and a gold mine on an initially unknown random terrain, the task is to destroy all enemy buildings within a certain time period. Each game object has radar-like (360 degree) vision with a limited radius. As a tiebreaker we will use the total value of destroyed opponent objects.

   In this simplified RTS game scenario we measure a player's ability to develop an economy, allocate resources, actively gather information by scouting, and engage in combat. Unlike the first two disciplines, here the game state is only partially observable.

TOURNAMENT ORGANIZATION

Logistics

The competitions are planned to be held at the conference site before and during the conference, preferably in time slots not occupied by presentations so that conference participants can watch the games.

If possible, each participant shall get sole access to identical computers with at least 1 GB of RAM. Three machines will be needed for each concurrently played match. One of them runs the ORTS server and displays the current game state. It therefore needs a descent graphics card ($\geq$ NVIDIA 6600). It may be possible to run two servers and display two games on one machine. At the very least we would need a router and three machines if the games were to be run sequentially. Having access to more computers, however, permits us to run matches in parallel to greatly speed up the tournament.

For this event we are looking for sponsors who are willing to lend us the required hardware and/or contribute prizes.

Competition entries consist of executable files which will be run by scripts in software sandboxes. This prevents programs from learning in between rounds which allows us to run multiple games using identical maps to ensure fairness. Running the tournaments in batch mode also speeds up the competition. We considered to accept entries created for Linux/GNU, Windows, or Mac-OS and came to the conclusion to only use the Linux/GNU operating system running on Intel x86 compatible PCs. This decision is based on the advantage of using a single operating system, which simplifies running tournament games in batch mode, and on the availability of free Linux/GNU on ubiquitous hardware which correlates with having access to more computers during the tournament.

Each team can submit at most one executable file for each RTS game discipline. Entry fees will pay for prizes and may depend on whether authors are willing to make their code open source after the competition and on whether authors are present at the competition site. A complete set of tournament rules will be established in the coming weeks. We need an independent tournament director or committee for resolving conflicts (such as alleged program plagiarism) and to decide whether we permit entries whose authors will not attend AIIDE'06.

At the end of the competition the teams shall get the opportunity to give short presentations about their entries and their underlying technology. We also propose to organize a workshop on complex video game AI alongside this competition in which each competitor can have a publication, which could facilitate their ability to attend.

Preliminary Tournament Rules

The tournament mode (groups, round-robin, swiss system, etc.) will depend on the number of participants. Each encounter of two players in discipline 2 and 3 consists of $2k$ games of fixed maximal duration (e.g. 5 to 10 minutes). Games are based on k randomly generated maps that will be created at the start of the tournament. Each participant will use the same set of maps. In the first discipline, $2k$ fixed but randomized single-agent scenarios will be played. These will be drawn immediately before the games are played. In each category, the player with the highest total score will be declared as the winner. The tiebreaker will be the head-to-head result or, if time permits, the result of an additional longer match.

SCHEDULE

Sep-7, 2005	Send proposal to AIIDE organizers
Oct-7, 2005	AIIDE'06 decision anticipated If accepted, web announcement Make available: – ORTS software – Game draft specifications
Dec-30, 2005	Make available: – Extensive ORTS documentation – Final game specifications – Competition rules – Software: ORTS, reference players, TIELT (TIELT 2005) (if ready) General announcement request to AIIDE

FUTURE EVENTS

If successful, we propose to organize subsequent RTS game AI competitions with increased game complexity featuring team-play, more complex technology-trees, and generalized setups in which parts of the game specifications are not fixed. In addition, competitions in other genres such as first-person-shooters will be considered.

ACKNOWLEDGMENTS

We would like to thank Jonathan Schaeffer and Charles Tarun for valuable feedback on an earlier draft of this document.

REFERENCES

GGC. 2005. General Games Competition Homepage at games.stanford.edu.

ORTS. 2005. Free Server-Client RTS Game Engine at www.cs.ualberta.ca/~mburo/orts.

RoboCup. 2005. RoboCup Homepage at www.robocup.org.

Stratagus. 2005. Free Peer-to-Peer RTS Game Engine at stratagus.sourceforge.net.

TAC. 2005. Trading Agent Competition Homepage at www.sics.se/tac.

TIELT. 2005. Test-bed for integrating and evaluating machine learning techniques in real-time games at nrlsat.ittid.com.

Wikipedia. 2005. en.wikipedia.org/wiki/Computer_and_video_game_genres.