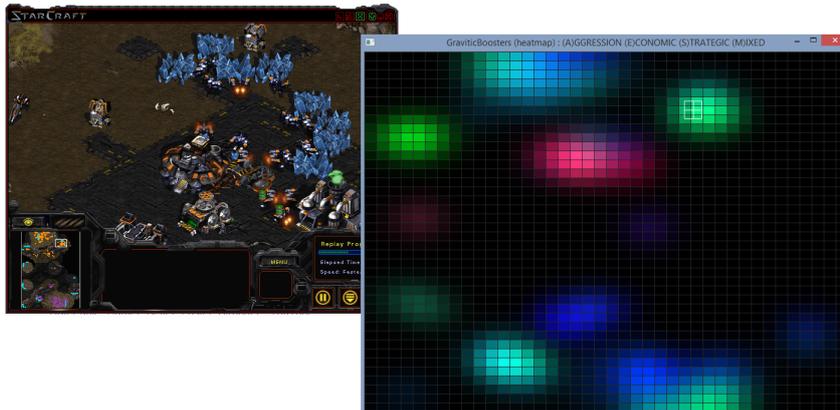


Automatic Observer For Real-Time Strategy Games

Emmanuel HADOUX Thomas HURAUX
emmanuel.hadoux@lip6.fr thomas.huroux@lip6.fr

LIP6, CNRS UMR 7606,
Université Pierre et Marie Curie, Paris, France



1 Context

Electronic sport (*e-sport*) is a not-so-young discipline slowly growing in western countries. It took root in South-Korea about 15 years ago and is currently experiencing an explosion of interest. The first game entering the closed category of e-sport was StarCraft 1. This fast-paced real-time strategy game (*RTS*) was released in 1998. Since then RTS games challenge many research works in AI such as planning in resource allocation, force deployment, and battle tactics [7, 4]; spatial reasoning [3] or opponent modeling [6].

During battles between armies, players can reach peaks of 450 actions-per-minutes. For this obvious reason, viewers cannot follow the game by watching the screens of the players. Competitions thus require an external observer following the game and showing important actions. However, being an observer requires a comprehensive knowledge of the game as well as anticipation and entertainment faculties. Few persons are able to gather those skills, making them more famous than some professional players.

In this work, we propose an automatic observer able to catch every piece of action to assist or replace the human observer. The following section briefly describes our approach to answer this problem.

2 Overview of the method

We set ourselves two constraints to the realization of an automatic observer: on one hand real time calculation to be used in the context of a game with pro players, and on the other hand, to use only standard information that can be found in any RTS (position, unit price, damage, ...). As far as we know, no previous research works have been made on this topic in the video game context. Nevertheless, we found on TeamLiquid (a famous website about StarCraft) an attempt to automatically product Youtube videos from finished game logs [2]. However, this solution is based on simple *ad-hoc* rules and thus is only applicable to StarCraft 1. Our objective is to remove this assumption even though few games can be linked to user code due to legal issues.

In our model, each entity has 3 potentials computed in real time:

AGGRESSIVE POTENTIAL allows to anticipate the interest of an upcoming fight by taking into account the distance to the enemy and the damage per second

ECONOMIC POTENTIAL informs on base creation, workers training, etc.

STRATEGIC POTENTIAL shows information-taking units (called *scoot*)

Those 3 features are the most important sides

of a competitive RTS game. Indeed, players need workers and bases to gather resources allowing them to build an army. However, as they cannot train workers and units at the same time, they need to choose to go aggressive or economic based on gathered strategic information. The purpose of this model is to show where and when those choices are made and thus complementing the work of the casters (the commentators).

To that end, the map is discretized so that each cell aggregates the potentials of all entities composing it. The camera is then positioned at the maximum value of the resulting matrix. To adjust the positioning to the action, we smooth the potential matrix using a gaussian kernel. This allows, for instance, to position the camera between two armies facing each other. We also added two dynamic parameters to the camera: *commitment* and *weariness* in order to limit both too-frequent disturbing changes and over-attention to the same area.

3 Implementation

Based on this method, we develop an automatic observer called *Gravitic Boosters*¹. We tested our approach on StarCraft BW² using the BWAPI API [1] which enables to retrieve all live information and take control of both the units and the game interface. As a side note, this API has been used for 5 years in several StarCraft AI Competitions, supported by the University of Alberta³.

The figure above shows a StarCraft game and a corresponding visualization of the 3 potentials as a heat map. RGB colors are used to encode the 3 components: red represents the aggressive potential, green the economic potential and blue the strategic one.

4 Perspectives

We consider several perspectives in the context of this work in order to increase the accuracy of our observer. Firstly, we plan to take into account the spaciality of the events (unit creations, skirmishes, etc.). Self-Organizing Maps seem very adapted for such

a purpose. Secondly, we could use clusterization to group similar game sequences. This will allow the automatic observer to not only follow the action, but also explain what it is going on. Those two perspectives are works in progress.

The video game industry continues to expand while almost no AI has been involved in this expansion [5]. Unfortunately, AI seems not to be a priority of publishers compared to *e.g.* graphics with a better marketing value. Nevertheless, we believe that AI can bring more to this media. We try to illustrate this point with our work presented in this paper. Recent competitions with over 10.000.000\$ in prize money and stadiums under construction are among indicators that video games and e-sport especially provide real life applied research opportunities. We argue that such research should be encouraged to take advantage of this recent expansion.

References

- [1] BWAPI. An api for interacting with starcraft: Brood war, 2015. <http://github.com/BWAPI/BWAPI>.
- [2] djsherman. Replay->vod tool, 2010. <http://www.teamliquid.net/forum/brood-war/142828-replay-vod-tool>.
- [3] Kenneth D Forbus, James V Mahoney, and Kevin Dill. How qualitative spatial reasoning can improve strategy game ais. *IEEE Intelligent Systems*, 17(4):25–30, 2002.
- [4] Joshua McCoy and Michael Mateas. An integrated agent for playing real-time strategy games. In *AAAI*, volume 8, pages 1313–1318, 2008.
- [5] Risto Miikkulainen. Creating intelligent agents in games. In *Frontiers of Engineering:: Reports on Leading-Edge Engineering from the 2006 Symposium*, page 15. National Academies Press, 2007.
- [6] Frederik Schadd, Sander Bakkes, and Pieter Spronck. Opponent modeling in real-time strategy games. In *GAMEON*, pages 61–70, 2007.
- [7] Ben George Weber and Michael Mateas. Case-based reasoning for build order in real-time strategy games. In *AIIDE*, 2009.

¹<http://github.com/EHadoux/GraviticBoosters>

²<http://us.blizzard.com/en-us/games/sc/>

³<http://webdocs.cs.ualberta.ca/~cdavid/starCraftaicomp/>