Adaptive Real-Time Strategic Agent in StarCraft

Abdelrahman Elogeel

Dept of Computer Science and Engineering

University of Washington, Tacoma

[elogeel@u.washington.edu](mailto:elogeel@u.washington.edu)

**ABSTRACT**

Recent advances in computer hardware motivated the work on incorporating planning and learning in RTS games which has a direct impact on many industries like military, robotics, city building and lastly game industry. The current state of art lakes of having adaptive agents that reacts in real time so in this proposal, we present an adaptive real-time agent in that plays StarCraft (a popular RTS game) successfully on high strategic level. The evaluation at the end shows how the agent is capable being adaptive in real-time compared to static agents which don’t adapt to the current environment state.

# INTRODUCTION

Real-Time Strategy Games offer a wealth research environment by having an unstationary environment with thousands of objects interacting with each other in a real time with imperfect information and uncertainty challenges [1]. The research in RTS games impacts many military aspects like group formations, analyzing attacks and retreat timing and composing the army. On the peaceful side, RTS games help in proposing a real-time simulation of building a comfortable city given the population constraints. Most importantly, the current RTS games focus on the graphics and storyline side and have static behavior for most of its AI bots. In some scenarios, gamers want to make the AI bot mimic and learn their behavior so they can focus on other tasks.

Many research work have been done on this area included techniques that are adaptive but not real-time [2] or works in a reactive and good time manner (using for example FSM) but not adaptive on high-strategic level [3].

This proposal will be specific for high level strategic aspect of a playing agent. The high level strategy is responsible for determining the high level overview of how the agent will play the game from many areas like what’s the agent opening strategy is it rushing strategy (make a quick army and attack) or defensive (build a strong economy and defend the city) or something else? The high level strategy also determines how the agent will build its economy from high level aspect for example the engine can rely on heavily farming only or getting a hiring a constant refineries that get money? What’s the high level structure of the army? This is also considered in the high-level strategy aspect of the agent.

# Real-Time Strategy Games

Computer games have many genres one of them is called Real-Time Strategy (RTS) Games in which the player is supposed to build a complete city, manage its resources, builds a strong army for this city, gather resources, make aliases and attack enemies. The goal of a RTS game is to defeat all the enemies and destroy all their structures.

RTS games have some attributes that make them special genre:

* Real-Time: the game is played in real time with no delay not like chess where user does one move and wait for the other player to do a move. Moreover, the actions in RTS games are *durative* that means an action takes certain amount of time to finish not gets finish instantly
* The state in RTS games is huge where a number of plays can have different number of buildings or units at a time
* RTS games are partially observable so a player can’t see what the enemy is doing right now. This is called *fog-of-war*

In this proposed work, StarCraft [4] a popular RTS game developed by Blizzard Entertainment will be used as test bed. A screenshot from the game is shown in Figure 1.



Figure A screenshot of StarCraft: Brood War.

# Experiment Design

## Turing Test

A Turing test [5] will be applied to the game where a human player will play against the adaptive agent and he/she should feel that this is not a machine rather it acts like a human

## Measurements

A bunch of games played by the agent will be compared with others played by static agents. The graph will show how the agent’s actions are different based on the current environment where the static agent’s actions won’t adapt and will be same.

# Schedule

|  |  |
| --- | --- |
| Schedule | Task |
| W1 | Setup the development environment using BWAPI and do initial work |
| W2 | Read about decision trees and how to learn a decision tree from a given goal |
| W3 | Change the current retriever which uses Euclidian distance to use decision tree. Will do this for just one goal and then get feedback on the results |
| W4 | Improve the previous implemented work based on the given feedback from advisors and expand it to include all goals |
| W5 | Find the results and do the experiments |
| W6 | Document the final work and prepare the demo |

# Deliverables

To do: fill out deliverables

# References

|  |  |
| --- | --- |
| [1] | M. Buro, "Call for AI Research in RTS Games," in *American Association for Artificial Intelligence*, Palo Alto, California, 2004. |
| [2] | S. Ontanon, K. Mishra, N. Sugandh and a. A. Ram, "On-line case-based," *Computational Intelligence,* vol. 26, no. 1, pp. 84-119, 2010. |
| [3] | R. Houlette and D. Fu, "The ultimate guide to fsms in games," in *AI Game Programming Wisdom 2*, 2003. |
| [4] | Wikipedia, 2013. [Online]. Available: http://en.wikipedia.org/wiki/StarCraft:\_Brood\_War. |