Developing a StarCraft: Brood War agent

Carsten Nielsen

DTU

Summary (English)

The goal of the thesis is to ...

Summary (Danish)

Målet for denne afhandling er at ...

Preface

This thesis was prepared at DTU Compute in fulfilment of the requirements for acquiring an M.Sc. in Engineering.

The thesis deals with ...

The thesis consists of ...

Lyngby, 30-June-2015

Carsten Nielsen

Acknowledgements

I would like to thank my....

Contents

Introduction

Goals, challenges, programming language.

Other projects and agents.

1.1 The "separate document"

Original project plan.

Revised project plan.

Brief self-evaluation.

1.2 Starcraft: Brood War

Game description: premise, goals, races, units, challenges.

1.3   BWAPI

BWAPI description.

BWTA and SparCraft description

1.4   AI Competitions

Competition rules: Setup, limits, goals.

CHAPTER 2

Implementation

Separated in modules.

2.1 Resource Management

In early- and mid-game, *economic advantage* is the most important aspect of StarCraft. This is when you have a higher resource input than your opponent, and happens when you focus more on resources than them, or they sacrifice long term advantage over short term. The point of this is if you eclipse your opponent in resources, you can perform worse in combat and still end up on top. You can send more units, recover them faster and expand economy quicker, retaining your advantage.

There are three resources in StarCraft. Minerals are mined from mineral fields which usually are in clusters of 8-12 by each base location. Everything you produce costs minerals, so it is the most important resource. They are harvested by your workers, which bring it back to a nearby resource depot in batches of six. Usually there are 2-3 workers on each field depending on the distance from the depot. *Vespene Gas* or simply gas is harvested from *refineries*. These must be built upon *Vespene Geysers* of which there are one or two by placed by

each base location. It is harvested in batches of eight with a max of three workers per refinery at optimal depot distance. Advanced structures, units and all technologies depend on gas, and a decision with major impact is when a player starts harvesting gas. The immediate cost of the refinery and lost mineral gathering is a liability against fast rushing strategies. The last resource, supply, is reserved by units. Every unit reserves some amount of supply which is released upon their destruction. The only way to secure more is building *supply depots*, *pylons* or *overlords* respectively for Terran, Protoss and Zerg. Each of these add some amount of supply, which is removed if they are destroyed. A player can happen to use more supply than they have, but will be unable to build more units until more supply is acquired.

Minerals, gas and supply.

2.1.1   Harvesting

Simple implementation: no assignment.

Advanced implementation: assigning.

Gas and minerals ratio.

Other implementations:

2.1.2   Supply

A player can have a maximum of 200 supply. The simplest supply implementation is to build supply units whenever the supply limit has been reached while it is below the maximum threshold. This will however throttle production while the supply structure is being constructed, which is inefficient. The optimal solution would be predicting supply needed in the near future and interlace supply production with unit production such that all orders are completed the earliest. Observe however that this is alike the *Job shop scheduling* problem but more complex with the additional factor of resources. As such the problem is NP-complete, and even then resource gathering rates must be predicted for optimal solutions.

A simple solution which is implemented in the current bot is to build supply units when released supply is below a set threshold. This threshold is somewhat arbitrarily set according to testing. A more dynamic solution would be to set the

threshold to the amount of supply used if all production facilities built a unit, assuming the unit type can be predicted. This solution will build supply units that are completed before they are needed, but might not result the optimal amount of units completed at any given time (impacting economy with relation to workers). The worst problem, production throttling, is avoided however.

2.2 Expansions

When to expand.

Where to expand.

Building expansions.

Base management.

2.3 Scouting

Scouting is important throughout the entire duration of the game. Early on, players need to know where the opponent is, what faction they are playing and which strategy they are employing. Later on, observing the enemy army size, unit types, base expansions and tech level is important to counter strategies.

Enemy opening -> opponent modeling.

The opponents faction is important when deciding on opening strategies. The factions behave very different early in the game. Openings are a decisively slower when one has to defend against every possible attack. Only if the opponent picked random as a faction will it be hidden. No units are shared across factions, so the first unit discovered will reveal it, which usually is their scout or main-base.

Base positions + expansions

2.3.1 Initial Scouting

The map terrain, along with resource locations, are completely revealed at match start. This includes all possible player start locations. Scouting for the opponent base is then just going through all other start locations. Maps used in competitive matches are usually no larger than four-player size, meaning there are four possible start locations. A player then has to take into account that they might have to scout up to two bases before knowing the enemy start location. Usually a player wants to know more than where the opponent starts, so even when it is deductible where the opponent is, the player wants to scout the base itself.

Enemy opening.

Once the enemy base has been revealed, and with it his opening strategy, the scout is not useful any longer. It is a long trip back to gathering resources, and the scout might be followed, revealing your own location. Harassing enemy workers can put a dent in the enemy economy, even if none of them die. Simply by attacking enemy workers forces the opponent to pull two from gathering to defending. From there, the harassing scout can retreat until it is no longer chased, or lead chasing workers around while attacking passive gatherers. If the scout manages to kill a worker the opponent will fall behind in economy. However, compared to competitive human players, bots are usually too inefficient to take full advantage of this, but every bit helps.

There are additional uses to the scout however compared to harassing. These must all be considered in the grand strategy, as they involve spending resources. These strategies include proxy bunker, photon cannon, barracks or gateway, manner pylons or gas stealing. Proxy structures involve building right below the enemy ramp or even inside their base. The usual distinction here is whether the structures keep the opponent in with defensive towers or rush attack with front line troop producers. Manner pylons are used in conjunction with these as Protoss, where the required pylon for forward bases are placed within the enemy mineral line, blocking and possibly caging enemy workers. Gas stealing involves building a refinery on the enemy geyser, blocking gas harvesting and forcing tier one unit use. While proxy troop production has not seen much use in bots, both gas steal and proxy towers has been used for varying effect.

2.3.1.1 Implementation

Usually build orders include specifically when to send out a scout. In case of
Protoss, the scout is often a worker that has just warped in a structure. This
would however require a build-order system capable of containing other elements
than builds. A simpler solution which is used here is sending a scout when a
specific supply limit has been reached. This is noted in build orders, allowing
for only slightly inaccurate build-order implementations.

When picking a scout, the bot searches through different worker groups. First
the idle, then the mineral miners and finally the gas harvesters. When it comes
across a worker not currently carrying any resources, it assigns it as a scout,
removing it from the Task Master in the mean time. Contrary to building or
defending, scouts are assumed to not return, so it can safely be removed from
the local worker pool and harvesting. If the scouting manager is unsuccessful in
finding a worker, it retries next frame.

While we have a scout and do not know the opponents position, we pick an
unexplored base location and move the scout there. A tile is unexplored if it
has never been revealed for the duration of the game. Thus, our home base will
not be considered for scouting, and once the scout reaches the destination the tile
will be explored, removing it from the possible scouting locations. Exploration
is handled by BWAPI.

This implementation scouts as long as no enemy buildings are known, extending
its use into late game. Usually fast or cloaked units are used to scout later in
the game, but it is not necessary.

2.3.2 Intelligence Management

Locating positions, types.

Geysers <-> refineries

Possible Additions: Faction

2.4 Attacking

When to attack.

Where to attack.

2.4.1 Combat

Predictions.

Grouping.

Prioties.

Retreating.

2.5 Defending

Current: anyone in vicinity

Worker militia.

Anti-scouts / anti-harassment

Worker relocate.

Others: Standing army, Pullback, Vicinity prediction and need.

2.6 Strategies

Main strategies: Rush, Turtle, Boom.

Economic advantage v short term advantage.

Niche strategies: all-in, wall-in, proxies.

2.6.1 Build-orders

Planning.

Openings.

CHAPTER 3

Conclusion

Agent status, results.

Competitions.

Further study.

APPENDIX A

Stuff

This appendix is full of stuff ...