



UNIVERSITY OF BIRMINGHAM

Artificial Intelligence in StarCraft 2

Project Proposal

School of Computer Science

Sotiris Loizou - 1966658

Supervisor: Dr. Eliseo Ferrante

Project Category/Topic: AI

21st June 2019

Declaration

I confirm that I have read and understood the University's definitions of plagiarism and collusion from the Code of Practice on Assessment. I confirm that we have neither committed plagiarism in the completion of this work nor have we colluded with any other party in the preparation and production of this work. The work presented here is my own and in my own words except where I have clearly indicated and acknowledged that I have quoted or used figures from published or unpublished sources (including the web). I understand the consequences of engaging in plagiarism and collusion as described in the Code of Practice on Academic Integrity (Appendix A).

Project Aim

The main aspect of the project is to explore exact facets of artificial intelligence. Specifically, how can they be applied to a complex environment which resembles to some degree the complexity of the real world. The primary candidate solution is nature inspired solutions that have already been developed over the years by nature to tackle various tasks. For instance, swarm intelligence is one of type of collective behaviour which is decentralised and self-contained within a population of agents. More specifically, the aim is to apply swarm intelligence into a game which has specific tasks/goal. At a further stage is to create different types of swarms with different characteristics/rules to see how they perform in regards with the underlying tasks/goals.

The game selected for this experimentation is StarCraft 2. The importance of the project derives from the complexity of the multi-variable environments. StarCraft is known to be one of the games that has been optimised over the years started in 1998 with StarCraft 1 and 2010 with StarCraft 2. The multiplayer aspect of the game allowed millions of people to play and discover bugs as well as balancing issues between factions and environments. The competitive esports scene managed to drive high level players to find even harder issues that a casual player may have not even notice. At the moment only reinforcement learning has been substantially used to create intelligent agents in StarCraft. Various other categories of artificial intelligence can be tested or slightly modified so as to yield progress to the contemporary approaches.

Related Work

APIs

- Official C++ SC2 library created by Blizzard
source: <https://github.com/Blizzard/s2client-api>
- Python intermediary implementations are
 - PySC2: <https://github.com/deepmind/pysc2>
 - Example of PySC2: <https://github.com/chris-chris/pysc2-examples>
 - Python-sc2: <https://github.com/Dentosal/python-sc2>
- Java intermediary implementation: <https://github.com/ocraft/ocraft-s2client>

Agents

- AlphaStar : <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/>
 - Publication page: <https://deepmind.com/research/publications/>
 - StarCraft 2 Reinforcement Learning : <https://deepmind.com/documents/110/sc2le.pdf>
- Pysc2-tutorial simple agents: <https://github.com/skjb/pysc2-tutorial>
- Dota 2 agents by OpenAIFive: <https://github.com/openai>

Types of AI

- Swarm Intelligence:
 - Swarm Intelligence in Swarm Robotics: <https://hal.archives-ouvertes.fr/hal-01405919/document>
 - Self-organized flocking with a mobile robot swarm <https://hal.archives-ouvertes.fr/hal-01406049/document>
- Reinforcement Learning:
 - <https://deepai.org/machine-learning-glossary-and-terms/neural-network>
 - <https://www.oreilly.com/ideas/neuroevolution-a-different-kind-of-deep-learning>
 - <https://arxiv.org/abs/1712.06567>
 - <https://ieeexplore.ieee.org/document/6756960>

Project Objectives/Deliverables

- The programmatical structure of the flock
 - This is a complete representation of the flock and its available actions that can be carried out at any given time.
- Production of a simple solution for a set of units that have to reach a beacon and another for a fight (random movement and attacking).
 - Reward regarding time spend doing the task
 - Reward for the survival of more units when fighting
- Creating the initial rules for the swarm.
 - Initially, an assignment must occur with the starting rules for a swarm which will change overtime.
- Creation, adjustment and analysis of specific scenarios.
 - Success can be measured by actually having a set of scenarios which have been created with an increasing difficulty or versatility
 - Basic scenario where the flock must reach a beacon.
 - A scenario where the flock must reach decide which beacon to reach given there are multiple available at a time.
 - In a scenario where the flock reach the beacon but avoid turrets that cause harm to the flock.
 - The creation of a map which has a set of consecutives beacons that the flock has to visit creating a path.
 - A scenario where the flock must micro-manage the units in a fight.
- Analyse the swarm that has been created for the specific task
 - Critical analysis of the swarm internal representation. Consider the characteristics of the individual units (StarCraft2 units such as scvs, probes and drones). What can be attributed to the environment as well as the swarm
 - Comparing and Contrasting various approaches (random selection vs swarm intelligence)

- Setting and recording metrics
 - minimising the time spend on task
 - minimising the number of units lost due to fights or turrets.
 - collecting most beacons given a time frame
- Report writing

Methodology

Initially the creation of all the maps/environments are going to be based on the StarCraft 2 game edit tool. The tool would allow to create all the scenarios necessary for the swarm as well as the unit composition needed. The programming language selected for the representation of the swarm along with all its attributes is python which works with most third-party APIs, a secondary fall-back option is C++ which allows of a direct use of the official API published by blizzard. The experimentation schema is going to be used in order to be able to attribute the correct behaviour based on the changes that occurred internally in the swarm. The idea fits well with the current deliverables since most experiments have small alterations of the independent variables in order to observe the alterations in behaviour. All the scenarios are going to be developed separately but the swarm will be the same. A comparison is going to occur analysing the rules at the end of every scenario. The swarm at the end of every scenario will be evaluated based on some metrics stated in the deliverables which produces the overall performance of the swarm at the given task.

Project Plan

Feasibility

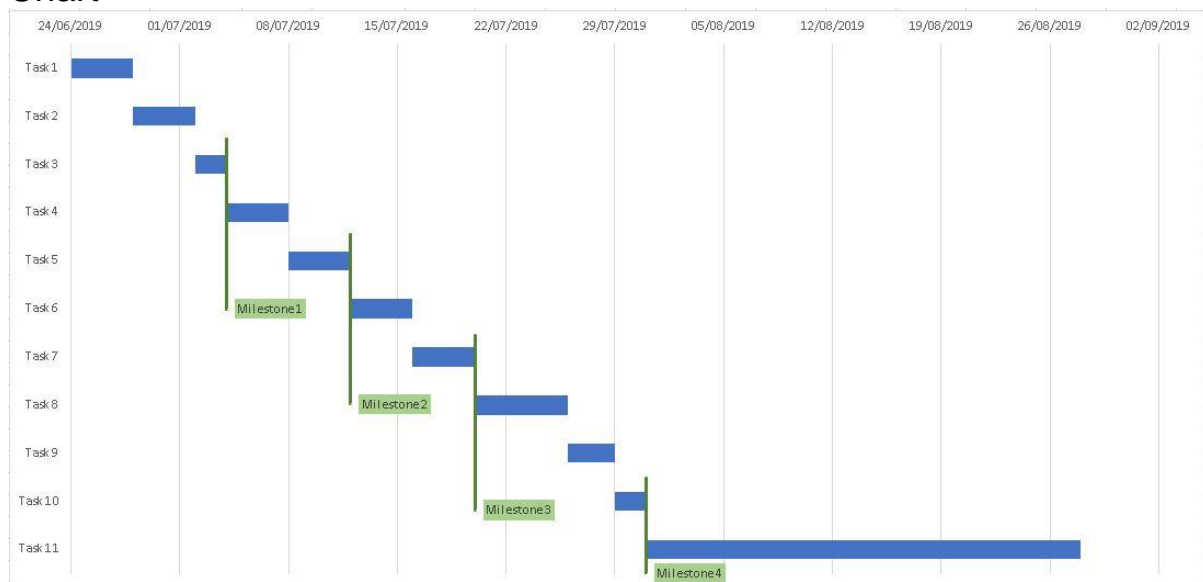
The initial problem that must be faced in the development of an AI project is the comprehension of the game. I used to play StarCraft 2 to a medium level which grants me a small advantage compared to someone that has never played StarCraft 2. This is of a great importance especially in a game which is defined as a real time strategy game. I have already programmed for around 4 years in multiple programming languages. I feel confident using python or C++ for this project. Resource wise I have a desktop computer that will serve as the platform to create all the source code. Video demos of progress can be recorded in OBS to show the current state of the project. The only concern is the AI category that I have picked which is swarm intelligence. I have no previous experience but only a fundamental understanding of the concept. This is the main area of expertise of my supervisor who I assume will guide me. This is a great opportunity for me to further my understanding in a field which is intriguing and mysterious.

Gantt chart

Tasks & Milestones

1. Software architecture of swarm
2. Simple solution (agent) for beacon task
3. Initial rules of the swarm(template) swarm **(Set Milestone 1)**
4. Basic scenario 1 beacon
5. Scenario with multiple beacons to choose **(Set Milestone 2)**
6. Scenario with turret which causes damage
7. Scenario of path (multiple consecutive beacons)
8. Simple battle **(Set Milestone 3)**
9. Comparison of swarm with simple agent
10. Record performance metrics **(Set Milestone 4)**
11. Write Report

Chart



Risks and Contingency plan

The most unreliable aspect is the game itself which tends to get new updates on a regular basis that may interfere with many of the deliverables stated. There is a possibility to use older version of the game or at least one which is stable. For instance, the created maps may need regularly updating or rebuilding to run as a map in the game. The initial aim is to create and develop all scenarios as stated in the deliverables but in case of time limitations the set regarding the advance battles may be disregarded and not implemented. Furthermore, the API available maybe restrictive and lack specific functionality which is needed to implement the swarm behaviour. In such occasion, an alternative option is to try co-evolution or even reinforcement learning. What is more the base API can be substituted with the C++ API which is officially provided by blizzard. The other APIs are third party created that communicate with the C++ API with the ability to use a different programming language rather than C++.