Starcraft 2 Cost-Benefit Analysis

Author: Daniel Newman

Proposal:
Create a Multi-Brain agent that has been trained on video replay data using image processing.

Constraints:
Use Reinforcement Learning
Multi-Agent Architecture
Needs to be trained on Starcraft 2 Replay Data
Requires Convolutional Neural Network for Image processing and analysis
Needs to implement some form of Imitation Learning

Time Costs

Task	Time Cost
Gathering the libraries required for environment	4hr
Gathering the libraries & dependencies for reinforcement learning	4hr
Gathering a suitable training data set	6hr
Gathering a suitable test data set	1hr
Research into RL Models	20hr
Deciding which RL model to apply	2hr
Implementing the chosen models	8hr
Creating a suitable SC2 framework for multiple model implementations	12hr
Research into appropriate training methodologies for chosen RL models	12hr
Choosing training regimen for each model	2hr
Creating a suitable training framework for multiple model & training combinations	8hr
Training each RL model using chosen regimen	24hr
Building test environment	6hr
Run RL models through testing data to ensure correctness	6hr
Test RL models against each other	1hr
Configure test environment for 1v1	2hr
Run human testing	24hr
Expand test environment to scale up encounters	6hr
Run expanded testing (FFA, 1vX, 2vX)	24hr
TOTAL	206hr (17 weeks @ 12hrs per week)

Equipment Costs:
Rental of GPU Bank/Server $8-20 a month (based on paperspace.com)
Justification: Need a hardware specification suitable for processing the large amount of replay data for the initial model training process. Plus also needs to house the AI during the self-play training process to begin reaching competence.

Benefits
Gain an understanding of how Reinforcement Learning algorithms operate
Gain an understanding of how RL algorithms can be applied to non-deterministic problems
Apply & Implement different RL algorithms and training methods
Learn how the training method influences the final result of the RL algorithm
Learn the limitations of current RL algorithms and possible ways to address or mitigate them
Allow a flexible architecture to be applied to training RL algorithms
Analyse how a problem domain can be broken down into subsets for training.
Demonstrate the capacity for stochastic problem-solving with RL algorithms.
Demonstrate how a flexible approach can be utilised to train multiple RL agents and understand what decisions can be derived each model for contrast and comparison.

Conclusion:
Creating a bot within the remaining timeframe that applies for this subject is not feasible. It is projected to take a minimum of 17 Weeks work based upon a 12hr workweek plus equipment costs for training to produce a viable result.