# Introduction

Despite the amount of effort put in AI research to develop a program capable of playing real-time strategy (RTS) video games and recent successes of programs such as AlphGo [6] and DeepStack [7], professional players in complex RTS games remain unchallenged by an expert level AIs. Making the game more fun to play for humans by developing advanced AI is not the only one benefit of AI research in this area. Tasks carried out in hardly predictable environments of those games, resemble real-world military scenario which demands to solve many complex issues quickly and satisfactory. Most of the decision making in real-world has similar nature. Therefore one can expect that techniques used in AI able to master an RTS game could perform well in other domains.

## Motivation

There are many challenges to overcome by AI researchers to be able for their bots to match human expertise in RTS games. In our work we would like to contribute to following ones:

* **Adaptive planning**: most bots are not able to adapt their strategy to counter the opponent. Instead, they use some hard-coded triggers to select between sets of strategies or mix between them based on performance or by random.
* **Domain Knowledge integration**: this is related to lack of ability to adapt as many bots creators struggle to incorporate various forms of domain knowledge to their bots. Excellent sources of domain knowledge for the game are demonstrations of human play. Observing other players is a natural way for a human to learn to play the game but it does not hold for bots.
* **Integration of techniques to unified architecture**: playing RTS is multiscale AI problem. Bots creators know that and decompose the problem to sub-problem. However, approach how they do that varies from bot to bot.

The test-bench for our research is RTS game Starcraft: Brood War where all three challenges and the ones described in {chapter link} remain valid. Researchers have been working on techniques to solve the issues of bots in many RTS games over a decade. However, StarCraft: Brood War has become the first choice for this kind of research in recent years. It has one significant advantage over other similar games in a number of competitions for AI StarCraft bots (organized by AIIDE, CIG, and SSCAIT [1]) so one can easily see how created bot is doing against others. Another advantage is that players’ community around the game is still active even though the game was released back in 1998.

## Objectives

Our primary goal of this work is to develop a bot for StarCraft Brood War which learns its decision-making processes from the demonstration. To meet this aim we set up following guideline:

1. Review current approaches for building expert level real time strategy game-playing AI.
2. Identify decision-making processes realized by a human player in the expert level gameplay of real-time strategy game StarCraft: Brood War.
3. Implement game-playing agent for StarCraft Brood War. To reduce complexity and amount of required expert knowledge, design the game-playing AI as a highly decentralized multiagent system with the ability to learn from demonstration (watching replays of StarCraft: Brood War games played by humans).
4. Use Markov decision processes and Inverse Reinforcement Learning to train decision modules for individual MAS agents.
5. Evaluate and discuss the capabilities of game-playing AI and compare it to state of the art AIs.

## Contribution

By meeting our objectives, we deliver following contributions:

* By learning decision-making in a different situation of the game by observing experts, we eliminate most of the hard-coded behavior in our bot in the sake of better adaptability. It is also a good example how to encode part of the domain knowledge which is otherwise hard to get on the highest level of abstraction.
* Using Markov decision processes and Inverse Reinforcement Learning is way how to explain/decide behavior in given situation. Approaching problem trough utility based planning has some advantages over goal-oriented planning used in most current bots (more on differences in {link to chapter}).
* Decomposition of the problem is inevitable as hardly only one technique can be used to solve all subproblems (which is in alignment with results of famous No Free Lunch Theorems [3]), we developed a prototype of unified architecture in the form of an opinionated framework based on the multi-agent system. Not only our StarCraft bot project can benefit from this as our framework is domain independent.

## Outline

In {chapter 2} we give a broader introduction to the domain of RTS games, StarCraft and analyze decision making involved to identify decision-making processes. {chapter 3} is a continuation of {chapter 2} and presents challenges and techniques in RTS game AI development. Another purpose of that chapter is to introduce ways how bots are being developed. All of this is useful in our attempt to build our bot. In {chapter 4} we provide a description of Inverse Reinforcement Learning technique which we use to learn to make decisions in the same way as humans do it in given situation. {chapter 5} presents an explanation of what we understand under term agent as we see our bot as an agent. An important part of this chapter is to explain types of the agents to give some classification of them as our bot is partially concerning utility in contrast to common bots. We also provide a description of the multi-agent system as we use it for task decomposition in the form of the opinionated framework described in {chapter 6}. {chapter 6} is also a place where we bring everything together as we present way how we integrated decision making to our bot and how other techniques can be incorporated. In {chapter 7} we present our initial version of the bot with examples of the framework usage and IRL integration. This chapter also evaluates its current performance and gives analysis of current issues. In last {chapter 8}, we give a recapitulation of our work and discuss directions for future work.

# Bibliografie

|  |  |
| --- | --- |
| [1] | SILVER, David, Aja HUANG, Chris MADDISON et al. Mastering the game of Go with deep neural networks and tree search. *Nature* [online]. 2016, **529**(7587), 484-489 [cit. 2017-05-03]. DOI: 10.1038/nature16961. ISSN 0028-0836. |
| [2] | MORAVČÍK, Matej, Martin SCHMID, Neil BURCH et al. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* [online]. b.r., , 6960- [cit. 2017-05-03]. DOI: 10.1126/science.aam6960. ISSN 0036-8075. Dostupné z: http://www.sciencemag.org/lookup/doi/10.1126/science.aam6960 |
| [3] | ONTANON, Santiago, Gabriel SYNNAEVE, Alberto URIARTE, Florian RICHOUX, David CHURCHILL a Mike PREUSS. A Survey of Real-Time Strategy Game AI Research and Competition in StarCraft. *IEEE Transactions on Computational Intelligence and AI in Games* [online]. 2013, **5**(4), 293-311 [cit. 2017-05-05]. DOI: 10.1109/TCIAIG.2013.2286295. ISSN 1943-068x. Dostupné z: http://ieeexplore.ieee.org/document/6637024/ |
| [4] | BEN G. WEBER., . *Integrating learning in a multi-scale agent* [online]. 2012. Santa Cruz: University of California, 2012 [cit. 2017-01-11]. ISBN 14-776-1473-7. Dostupné z: http://alumni.soe.ucsc.edu/~bweber/bweber-dissertation.pdf |
| [5] | No Free Lunch Theorems. *No Free Lunch Theorems* [online]. b.r. [cit. 2017-05-03]. Dostupné z: http://www.no-free-lunch.org/ |
| [6] | CHURCHILL, David, Mike PREUSS, Florian RICHOUX, Gabriel SYNNAEVE, Alberto URIARTE, Santiago ONTAñNÓN a Michal ČERTICKÝ. StarCraft Bots and Competitions. *Encyclopedia of Computer Graphics and Games* [online]. Cham: Springer International Publishing, 2016, , 1 [cit. 2017-05-03]. DOI: 10.1007/978-3-319-08234-9\_18-1. ISBN 978-3-319-08234-9. Dostupné z: http://link.springer.com/10.1007/978-3-319-08234-9\_18-1 |
| [7] | CHURCHILL, David. *Heuristic Search Techniques for Real-Time Strategy Games*. University of Alberta, 2016. |