

AI design for real-time strategy games

Firas Safadi, Raphael Fonteneau, Damien Ernst

Real-time strategy

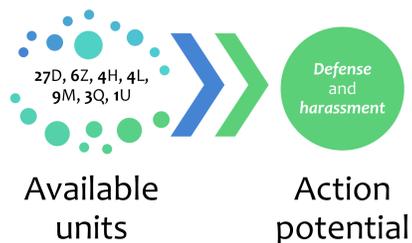
- * **Fast-paced, real-time multifaceted** action
 - * Complex battlefield
 - * Multitude of units
 - * Economy management
 - * Technological development
 - * Partial observability

Objective

- * Improve current agent performance in real-time strategy games
 - * Design **generic** and **efficient** agent models
 - * Add **learning capabilities** to generic models
 - * Offer developers a framework to **seamlessly integrate challenging, adaptive AI**

Overcoming complexity

- * Abstraction
 - * **Simplify environment**
 - * Game state
 - * Action potential
 - * **Simplify objectives**
- * Efficiently manage complexity



Modular and hierarchical design

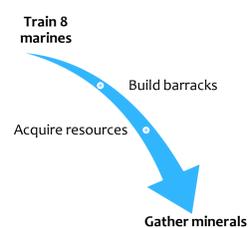
Design overview

Orders travel from the strategy manager down to individual units. They start as **abstract** orders and are processed by the intermediate managers, eventually **translating** into **concrete** unit commands.



Order processing

Production Order

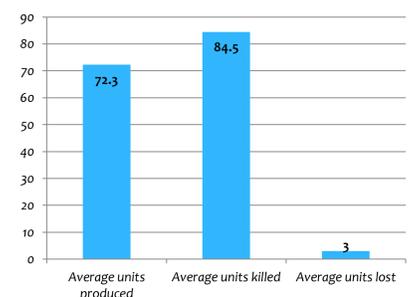


Combat Order



Results and limitations

- * Test setting
 - * **StarCraft: Brood War**
 - * Terran vs Zerg opponent
 - * **Play against default AI**
 - * 10 games in 1 map
- * Efficient model
- * **No learning** means agent is vulnerable to **flaw abuse**



Next steps

- * Adding **learning capabilities** to the agent design
 - * At the strategic level (i.e., build orders)
 - * At the tactical level (i.e., unit grouping)
 - * At the operational level (i.e., target selection)
- * **Build a framework** for the design of intelligent agents for real-time strategy games

Acknowledgments



Raphael Fonteneau is a postdoctoral researcher of the FRS-FNRS from which he acknowledges the financial support. This poster presents research results of the **Belgian Network DYSCO** funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office.