



Strategy Card Game AI Competition

Jakub Kowalski, Radosław Miernik
Institute of Computer Science, University of Wrocław, Poland
Correspondence: jko@cs.uni.wroc.pl



The Challenge

- Large branching factor, multi-action turns
- Deckbuilding phase
- Lot of hidden information
- Huge amount of nondeterminism
- Yet the gameplay is still very strategical



Competitions

One lane version

- Jul 2018 – CodinGame Sprint Contest (24h) – 742 participants
- Sep 2018 – CodinGame Marathon (30 days) – 2,174 participants

Two lanes version

- CEC 2019 – 6 entries (+2 baselines)
- COG 2019 – 9 entries (+2 baselines)



The Game

- Designed especially for AI playing.
- Simpler than Hearthstone, but the same set of challenges.
- `stdio`-based interface, supporting any programming language.
- Well defined set of rules, easy to code the simulation engine.
- Fair (symmetrical) arena mode instead of constructed.
- All plays during a turn are deterministic.
- Extendable cardset.
- Online testing arena supported by CodinGame.com platform.
- Based on The Elder Scrolls: Legends.



Cards

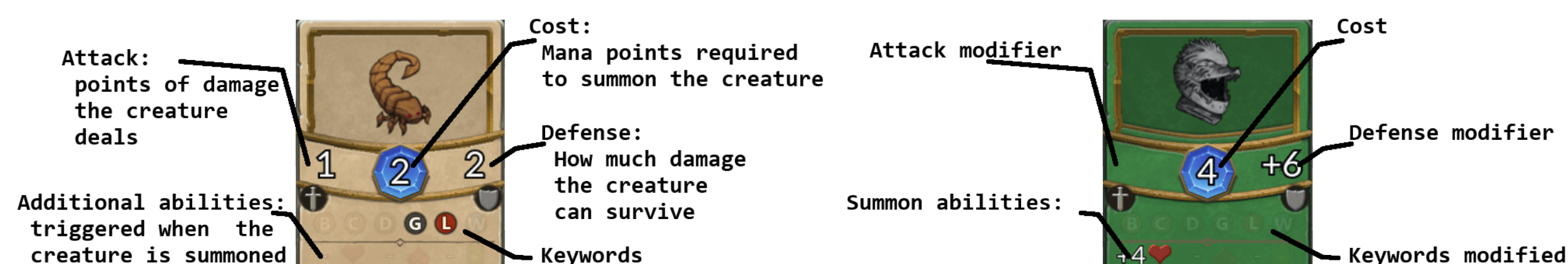
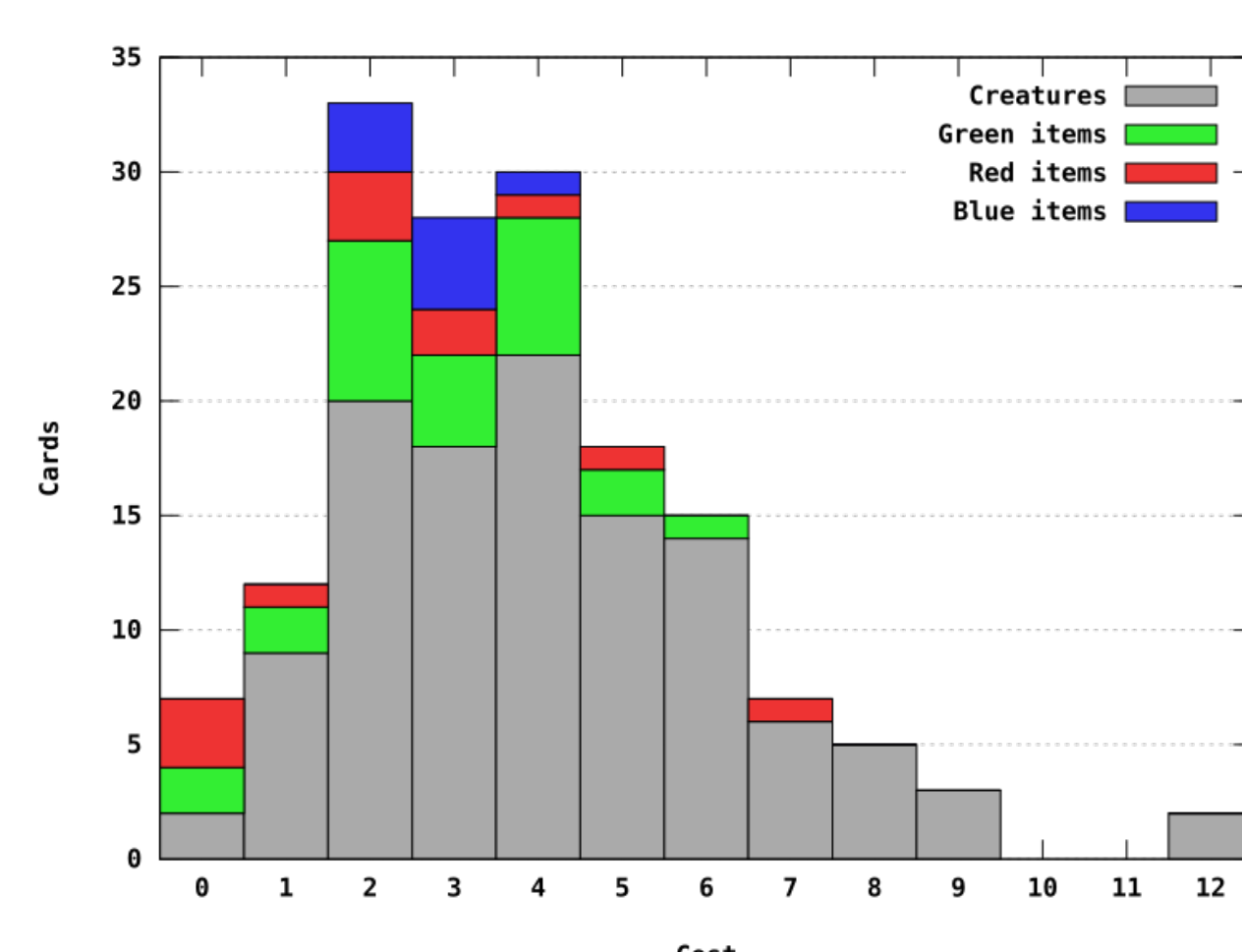
The game consists of 160 cards, including
116 creatures,
44 items (24 green, 12 red, and 8 blue).

There are 6 keywords:

Breakthrough, Charge, Drain,
Guard, Lethal, Ward.

And 3 possible additional abilities:

Modify player's health.
Modify opponent's health.
Increase next turn draw.



Actions

Draft phase:

- PICK nb

Battle phase:

- SUMMON id lane
- ATTACK idAttacker idTarget
- USE idCard idTarget
- Several actions per turn: SUMMON 3 1; ATTACK 4 5; ATTACK 8 -1; USE 7 2.



Results

1. Coac, Victor Le (C++)

Predefined card evaluation function for the draft phase.
Minimax-like search of depth max 3 with alpha pruning.
Handmade evaluation function (statistics of onboard cards, players' health).
Random search for enemy turn evaluation.

2. Prophet Coac, Ronaldo Vieira (C++)

Added prediction of the opponent's hand.

3. Marasbot, Jakub Maras (C++)

Handmade card evaluation function for the draft phase.
Simulation-based search with random card play (no blue items).
Handmade board evaluation heuristic.

4. UJIAgent2, Alejandro Juan, Arturo Barbosa, Diego Villabrille, Sergio Ferreras, Daniel Delgado, Raúl Montoliu (Python 3)

Probability-based card selection:
takes into account cost buckets and already taken cards.
Sampling combinations of predefined strategies
(9 for attacking, 35 for summon/use).
Handmade board evaluation function.



5. AntiSquid, Sergiu Lucea (Python 3)

6. Fabbiamo, Shioda Masahiro (C++)

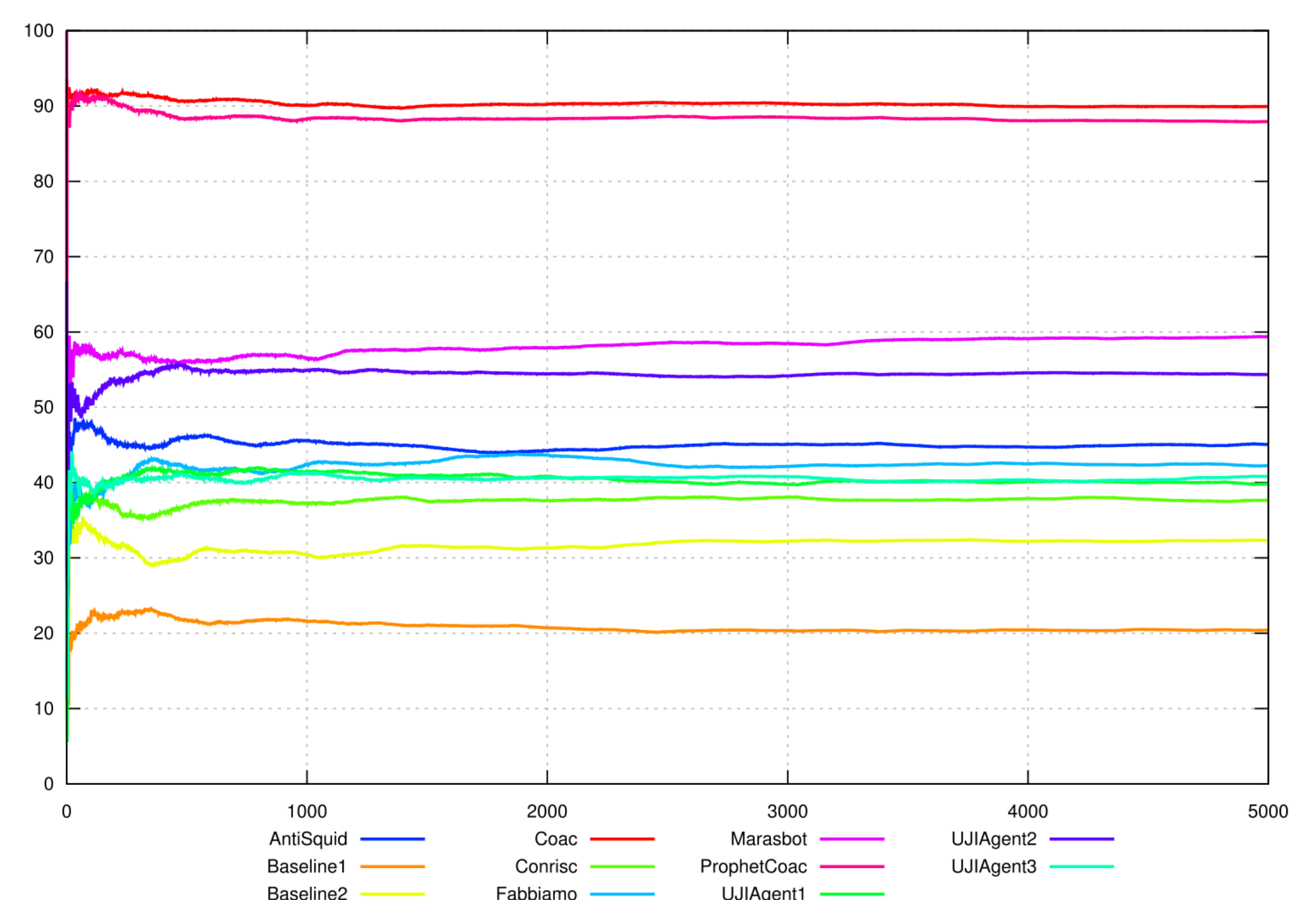
7. UJIAgent3, Raúl Montoliu, Daniel Delgado, Arturo Barbosa, Diego Villabrille, Sergio Ferreras, Alejandro Juan (Python 3)

8. UJIAgent1, Arturo Barbosa, Diego Villabrille, Sergio Ferreras, Alejandro Juan, Daniel Delgado, Raúl Montoliu (Python 3)

9. Conrisc, Konrad Cielecki (JavaScript)

10. Baseline2 (Python 3)

11. Baseline1 (Python 3)



Future of the competition

- Procedurally generated sets of cards.
- IEEE CEC 2020 & IEEE COG 2020



Acknowledgments

Prizes for this competition are sponsored by IEEE CIS.
LOCM crated in cooperation with CodinGame.com.



JAKUBKOWALSKI.TECH/PROJECTS/LOCM