

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/338137808>

# EvoMan: Game-playing Competition

Preprint · December 2019

CITATIONS

0

READS

502

5 authors, including:



[Fabricio Olivetti De França](#)

Universidade Federal do ABC (UFABC)

97 PUBLICATIONS 1,027 CITATIONS

SEE PROFILE



[Denis Gustavo Fantinato](#)

University of Campinas

68 PUBLICATIONS 134 CITATIONS

SEE PROFILE



[Karine Miras](#)

Vrije Universiteit Amsterdam

22 PUBLICATIONS 165 CITATIONS

SEE PROFILE



[A. E. Eiben](#)

Vrije Universiteit Amsterdam

440 PUBLICATIONS 19,743 CITATIONS

SEE PROFILE

# EvoMan: Game-playing Competition

Fabricio Olivetti de Franca, Denis Fantinato  
*Heuristics, Analysis and Learning Laboratory (HAL)*  
*Center of Mathematics, Computing and Cognition (CMCC)*  
*Federal University of ABC*  
 Santo Andre, Brazil  
 {folivetti, denis.fantinato}@ufabc.edu.br

Karine Miras, A.E. Eiben  
*Department of Computer Science Edinburgh Centre for Robotics*  
*Vrije Universiteit*  
 Amsterdam, Netherlands  
 karine.smiras@gmail.com

Patricia A. Vargas  
*Heriot-Watt University*  
 Edinburgh, UK  
 P.A.Vargas@hw.ac.uk

**Abstract**—This paper describes a competition proposal for evolving Intelligent Agents for the game-playing framework called EvoMan. The framework is based on the boss fights of the game called Mega Man II developed by Capcom. For this particular competition, the main goal is to beat all of the eight bosses using a generalist strategy. In other words, the competitors should train the agent to beat a set of the bosses and then the agent will be evaluated by its performance against all eight bosses. At the end of this paper, the competitors are provided with baseline results so that they can have an intuition on how good their results are.

**Index Terms**—game-playing agent, artificial intelligence, Evo-Man

## I. INTRODUCTION

EvoMan [1] is a framework for testing competitive game-playing agents in a number of distinct challenges such as:

- Learning how to win a match against a single enemy
- Generalizing the agent to win the matches against the entire set of enemies
- Coevolving both the agent and the enemies to create intelligent enemies with increasing difficulties.

This framework is inspired on the boss levels of the game Mega Man II [8] created by Capcom in which the player controls a robot equipped with a simple arm cannon and must beat 8 Robot Masters equipped with different weapons.

In the game, every time Mega Man defeats a Robot Master, it acquires its weapon making it easier to defeat the remainders bosses. This game is considered to have a high degree of difficulty among skilled players. As a personal challenge, some skilled players try to beat all Robot Masters using only the default arm cannon.

With this competition we propose the following question: can a fully automated intelligent agent defeat each one of the Robot Masters using only the default arm cannon? In more details, we challenge the competitors to evolve an intelligent agent by allowing it to train against four (4) of the Robot Masters but with the ultimate goal of defeating all eight (8) of them.

## II. THE CHALLENGE

In this challenge, the contestants should train their agent on a set of four enemies (defined by the contestant) and evaluate how general is their learned strategy when fighting against the whole set of enemies.

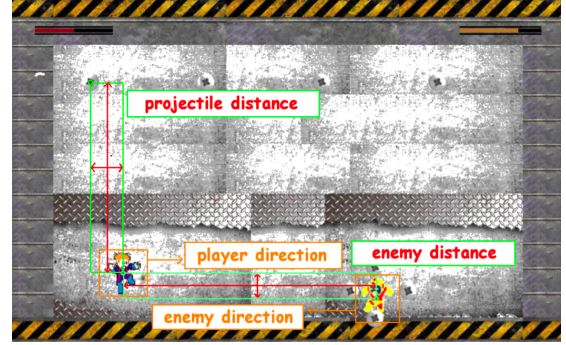


Fig. 1. Sensors available for the competition.

Since each enemy behavior greatly differs from each other, the player should learn how to identify and react to general patterns like avoiding being shot or shoot at the direction of the enemy. Learning a general strategy capable of winning over the entire enemies set can be very challenging [1], [2].

The agent will have a total of 20 sensors, with 16 of them corresponding for horizontal and vertical distance to 8 different bullets (maximum allowed), 2 to the horizontal and vertical distance to the enemy, and 2 describing the direction the player and the enemy is facing. The sensors are illustrated in Fig. 1.

The framework is freely available<sup>1</sup> and it is currently compatible with Python 3.6 and 3.7 (Python 3.8 is not compatible at the moment). There is also an extensive documentation available<sup>2</sup>.

## III. EVALUATION CRITERIA

Both the agent and the enemies start the game with 100 energy points. Every time the player or the enemy gets hit, they lose one point. Whoever reaches 0 points loses the match. The final performance of the agent after the end of a match is calculated by the energy gain, as a maximization problem, calculated by the difference between the player and the enemy energy:

$$Gain = 100.01 + ep - ee,$$

<sup>1</sup>[https://github.com/karinemiras/evoman\\_framework](https://github.com/karinemiras/evoman_framework)

<sup>2</sup>[https://github.com/karinemiras/evoman\\_framework/blob/master/evoman1.0-doc.pdf](https://github.com/karinemiras/evoman_framework/blob/master/evoman1.0-doc.pdf)

where  $ee$  and  $ep$  are the final amount of energy of the enemy and the player, respectively. The value of 100.01 is added so that the harmonic mean always produces a valid result.

The main goal of this competition is that a given agent perform equally good for every boss. So, each contestant agent will be tested against all of the enemies, and they will be ranked by the harmonic mean of the performance over the different bosses.

#### IV. PARTICIPATING IN THE COMPETITION

The initial code, manual and every other needed resources are available at a Github repository<sup>3</sup>. The competitors should pay attention to the following directions:

- Follow the installation instructions in the file `evoman1.0-doc.pdf`.
- Run the demo script `controller_specialist_demo.py` to test if the framework is working.
- Play the game using your own keyboard to understand the difficulties of the problem. Use the script `human_demo.py` for that.
- The agent should be trained using the Individual Evolution and Multi-objective modes with the goal of beating each one of the four adversaries chosen for training.

#### V. CURRENT RESULTS

In [3] different learning strategies were tested for the individual evolution mode. In this mode the algorithm creates one agent for each boss, thus generating specialist agents. The algorithms used for this test was variants of neuroevolution [5] strategies with 1-layer perceptron and 2-layers perceptron with 10 and 50 neurons for the hidden layer. The weights of the Neural Network was adjusted by means of a Genetic Algorithm [6] (GAP, GA10, GA50) and LinkedOpt algorithm [4] (LOP, LO10, LO50). The other strategy was the evolution of a Neural Network topology with their weights by means of the NEAT algorithm [7].

Table I show the obtained results so far for the individual mode. These results serve as an upper bound for the proposed competition.

From this table we can see that NEAT provided the best overall results followed by the two-layer neural networks with their weights adjusted through a Genetic Algorithm.

Notice that since these results were obtained with Individual Mode they will most likely serve as an upper bound of the results obtained in this competition.

#### VI. CONCLUSION

This paper introduces the EvoMan competition as a new general game playing challenge based on the Mega Man II game by CAPCOM. The main goal of the competition is to evolve an intelligent agent capable of defeating each one of the eight available bosses while being able to train against a smaller subset of those enemies.

TABLE I  
GAINS OBTAINED BY EACH TESTED ALGORITHM REPORTED IN [?]. THE *mean* ROW IS THE HARMONIC MEAN OF THE RESULTS.

| Boss | NEAT   | GAP    | GA10   | GA50   | LOP    | LO10   | LO50   |
|------|--------|--------|--------|--------|--------|--------|--------|
| 1    | 190.01 | 190.01 | 190.01 | 190.01 | 0.01   | 196.01 | 80.01  |
| 2    | 194.01 | 190.01 | 182.01 | 178.01 | 190.01 | 182.01 | 188.01 |
| 3    | 180.01 | 158.01 | 158.01 | 136.01 | 124.01 | 70.51  | 116.01 |
| 4    | 194.01 | 93.51  | 118.01 | 169.01 | 73.01  | 36.51  | 119.01 |
| 5    | 194.01 | 180.01 | 188.01 | 179.01 | 178.01 | 181.01 | 188.01 |
| 6    | 173.01 | 79.01  | 77.51  | 103.01 | 139.01 | 128.01 | 20.01  |
| 7    | 177.01 | 170.01 | 156.01 | 118.01 | 186.01 | 169.01 | 190.01 |
| 8    | 186.01 | 177.01 | 183.01 | 178.01 | 0.01   | 182.01 | 183.01 |
| Mean | 185.67 | 139.64 | 143.74 | 149.43 | 0.04   | 104.01 | 79.32  |

The winner of this competition will be the one agent that performs equally well on each one of the eight bosses, hopefully defeating them all.

In order to help the competitors, we provided a table with the upper bounds of the gains for each boss, obtained with the help of specialist agents.

#### REFERENCES

- [1] de Arajo, Karine da Silva Miras, and Fabrcio Olivetti de Frana. An electronic-game framework for evaluating coevolutionary algorithms. arXiv preprint arXiv:1604.00644 (2016).
- [2] de Araujo, Karine da Silva Miras, and Fabrcio Olivetti de Franca. Evolving a generalized strategy for an action-platformer video game framework. 2016 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2016.
- [3] de Araujo, Karine da Silva Miras Neuroevoluo para a construo de uma estrategia genrica com o ambiente EvoMan. 2016.
- [4] de Franca, Fabrcio Olivetti. "Maximization of a dissimilarity measure for multimodal optimization." 2015 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2015.
- [5] Floreano, Dario, Peter Drr, and Claudio Mattiussi. "Neuroevolution: from architectures to learning." Evolutionary intelligence 1.1 (2008): 47-62.
- [6] Goldberg, David (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley Professional. ISBN 978-0201157673.
- [7] Stanley, Kenneth O., and Risto Miikkulainen. "Evolving neural networks through augmenting topologies." Evolutionary computation 10.2 (2002): 99-127.
- [8] M. MEGA, Produced by capcom, distributed by capcom, 1987, System: NES.

<sup>3</sup>[https://github.com/karinemiras/evoman\\_framework](https://github.com/karinemiras/evoman_framework)