# ARTIFICIAL LIFE CONTEST:
## *a tool for informal teaching of artificial intelligence*

Diego Milone
*CONICET, sinc(i)-FICH-UNL, Ciudad Universitaria, 3000, Santa Fe (Argentina)*
*d.milone@ieee.org*

Georgina Stegmayer
*CONICET, CIDISI-FRSF-UTN, Lavaise 610, 3000, Santa Fe (Argentina)*
*gstegmayer@santafe-conicet.gov.ar*

Daniel Beber
*FI-UNER, Ruta 11 Km. 10, Oro Verde, Entre Rios (Argentina)*
*dbeber@enersa.com.ar*

Keywords: informal teaching, students motivation, artificial life, game-based learning.

Abstract: This work reports an experience in using an Artificial Life competitive game that simulates an artificial life environment for unstructured and informal Artificial Intelligence (AI) teaching to students from computer science engineering careers. The game consists of a simulated Petri dish where two colonies of microorganisms –software agents– must struggle to survive. To achieve this goal, the participants must implement surviving strategies for their agents, which include fighting strategies and basic reproduction rules to prevail over all the artificial environment. The technical bases of the contest as well as a description of the artificial life model are explained in detail. The pedagogical experience acquired in the contest development is discussed, as well as the resulting learning experience, which generated students enthusiasm and has helped them to develop mental models of possible AI algorithms.

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
D. H. Milone, G. Stegmayer & D. Beber; "ARTIFICIAL LIFE CONTEST: a tool for informal teaching of artificial intelligence"
International Conference on Computer Supported Education (CSEDU 2009), pp. 119-124, Mar, 2009.

## 1 INTRODUCTION

Artificial Life (AL) has a special attractive to computer science and engineering students. The promotion and foster of creativity in problem solving and the stimulation of spontaneous finding of solutions to previously unknown problems are important challenges while teaching. Given the impact of modern computer games on teenagers and young students, the time they spend on those games and the skills they develop for abstraction and generation of creative solutions, a game-like competition can encourage students to individual learning and the development of these skills by themselves. Thus, computer games may provide a great source of motivation when teaching artificial intelligence (AI) and other topics in computer science. Furthermore, an informal learning methodology gives the possibility to freely act in unknown situations and self learning without any obligations (based on the free choice of interests), which promotes the development of responsibility and self management.

The initial model, previous to the contest, was motivated by a simple and direct connection between a biological cell and a software object. After that, a model to reproduce the output of the well-known prey-predator system modeled by the Lotka-Volterra equations (Lotka, 1925)(Volterra, 1926) was developed. Some years later, based on this population model two new simulations for teaching topics related with object oriented programming, AI and biological models have been created.

The first model was a homework requiring the simulation of microorganisms having an internal artificial neural network model that sensed the neighborhood for nutrients and decided the next movement. In this simulation, a sower planted traces of nutrients and the microorganisms evolutionarily learnt the weights of the neural networks to follow these traces. After several iterations, all dead microorganisms are replaced by the offspring, obtained by crossover and mutations of the alive ones.

The second model was the Artificial Life Contest[1], where students should develop their own strategies of survival, with their own ideas about microorganisms modelling. The created microorganisms must compete for survival in a common environment (with reduced nutrients availability). In this competition, they have to design and implement an artificial

[1] http://alifecontest.sourceforge.net/

life form (a software agent), applying their AI and programming knowledge in a different, perhaps more attractive way than in a structured class. The simulation environment and its interactions with the agents are provided, therefore the students may only focus on defining the agent strategies for survival and how it decides what action to execute in each interaction with the environment.

For the contest development, object-oriented programming in C++ language is used (Milone et al., 2003). Students can access the environment source code (although they cannot modify it) and several examples of simple agents. Tournaments are organized periodically using the original environment (provided by the organizers) and the source code of the students agents.

The organization of this paper is the following: Section 2 presents the environment and artificial life simulation of the contest. Section 3 explains the software agents that must survive and interact in the artificial environment. Section 4 presents some details regarding winner definition and organization of the tournaments. Finally, the conclusions and future work are presented in Section 5.

## 2 MOTIVATION

There is a historically close relationship between Artificial Intelligence (AI) and games, such as chess, backgammon or poker, which have provided challenge problems for AI research. The use of AI in games presents an opportunity for AI educators to motivate students to learn about AI technologies through interactive learning and simulations. Playing any game well requires a player to choose a course of action taking into account the environment (the game situation) and the likely actions of other competitors or enemies (oppponents), so as to maximise opportunities for achieving goals (winning the game). Computer games as educational tools also have an intrinsic motivational factor that encourages curiosity and creates the impression that the students are in control of their own learning (Hingston et al., 2006).

There have been reported in literature several examples of AI teaching with games. In (Chiang, 2007) the traditional Pacman game has been adapted to provide student learning motivation for case-based reasoning AI technique learning. In (Kim, 2006) a virtual agent platform is presented for teaching agent systems design through a tournament game. This work aims at teaching agents in the first year of a computer science career and, differently from our unstructured learning approach, the authors use this tool as part of a formal course and for students evaluation. In fact, all of these proposals use the game for teaching inside a formal class environment.

Our approach, instead, aims at providing an informal learning methodology to senior students. This methodology has provided them an interesting opportunity to exercise the use of their imagination to solve previously unknown problems through self-learning, without a formal obligation, which also promotes self management. In general, the students that participate in the contest could organize their duties and manage their available time, exercising responsibility by themselves, without it being imposed from external pressions nor formal structures. In contrast to the classic approach to teaching AI with an objectivist approach, and similarly to (Pantic et al., 2005), we are focused in providing a simulation environment and a problem that needs to be solved through some AI technique. Like stated by this author, following a constructivist approach, an authentic real-world environment is provided in which students apply and test their knowledge and skills.

According to a recent analysis of game-based teaching and training systems (Martens Alke and Steffen, 2008) games have a high motivational character, often missing in traditional computer-based training systems. According to this analysis, this work, regarding learning theory, is an example of training with a microworld, a small world with a closed environment which functions based on its own artificial rules.

## 3 Artificial Life Contest: the environment

The Artificial Life Contest is a competition of software agents that have to be designed with the objective of surviving in an artificial environment. The agents must move in order to get food, they can reproduce themselves, they can decide fighting against another species or, if they have not enough energy, to run away to avoid dying in a battle.

In the Artificial Life Contest, the environment represents a Petri dish which contains Agar (with nutrients) and two different microorganism (MOs) colonies, which must fight for survival inside this environment (see Figure 1). The competition model includes these four classes: *Petri*, *Agar*, *Colony* and *Microorganism*, shown in Figure 2. The figure also shows the classes *MO_1* and *MO_2*, which are two different microorganisms developed by competitors, from which colonies will be built. The energy and food for the MOs are managed by the Petri and Agar

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
D. H. Milone, G. Stegmayer & D. Beber; "ARTIFICIAL LIFE CONTEST: a tool for informal teaching of artificial intelligence"
International Conference on Computer Supported Education (CSEDU 2009), pp. 119-124, Mar, 2009.

classes, which will be explained in detail in the following subsections.

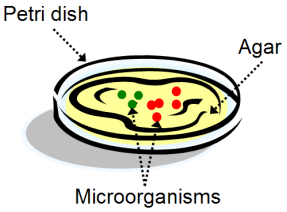


Figure 1: The "real" life environment, then simulated in the artificial life contest.

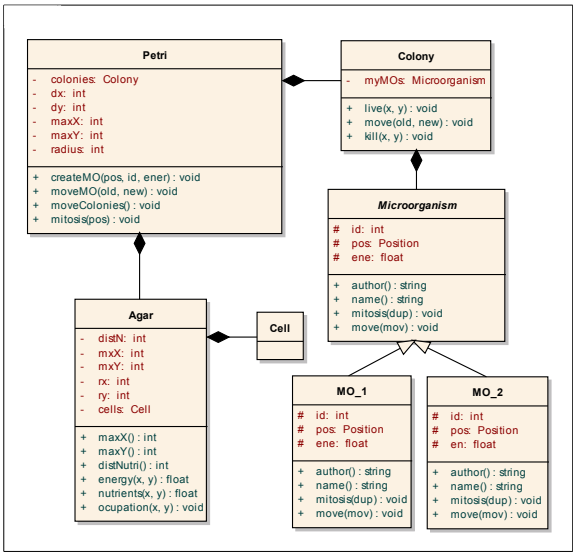


Figure 2: Artificial Life Contest model.

## 3.1 The Petri class

The MOs life takes place at the container class *Petri*, where the rules of life are applied. This set of rules verify the validity of the movements required by every MO. The MOs are randomly ordered and then, from each of them, an action is requested and whether they want to reproduce themselves or not. After each agent action, the Agar is updated. The MOs can "see" during all the simulation an updated version of the environment, which is totally accessible for them.

According to the rules of the artificial life simulation, the MOs may feed, fight or procreate. The main rules are the following:

- Initially, 50 MOs of each colony are created, with 1000 energy units each.
- The MOs are all randomly positioned inside the Petri dish.

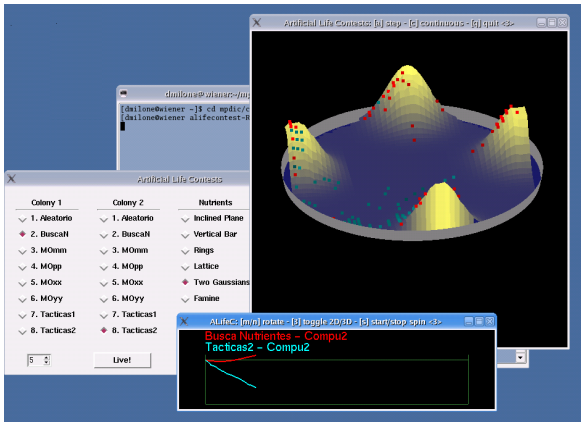


Figure 3: Artificial Life Contest: graphical user interface. The colonies are on the right window, inside the Petri dish, competing on a nutrient distribution given by two bidimensional Gaussians. The window at the bottom indicates the total energy of each colony.

- Every MO looses 5 energy units per time unit due to aging, and it wastes 10 energy units for each movement.
- Every time step, all MOs increase their energy by feeding with 1% of the nutrients existing in their current position.
- The initial nutrient distribution and the way it could vary during the competition is chosen by Petri. There are actually six food distributions and there is battle for each one of them. For example, a bidimensional gaussian mixture distribution could be selected

$$G(x,y) = \sum_i N_i e^{-\left(\frac{x-x_i}{\Delta_i}\right)^2 - \left(\frac{y-y_i}{\Delta_i}\right)^2},$$

where $N_i$ is the maximum for each gaussian component, $(x_i, y_i)$ are positions and $\Delta_i$ is related with each dispersion. This distribution can be seen in Figure 3 together with the graphical user interface of the competition. For this particular distribution case, a colony having a strategy to detect and remain in one of the energy peaks would certainly obtain an important vantage over the other colony.

- With respect to allowed agent actions, if a MO wants to move to a specific position, three possible situations are possible:
  1. The position is empty and it is inside the Petri dish: Petri allows it and moves the MO.
  2. The position is occupied by a MO belonging to the opposite colony: then the MOs fight and the one whose energy level is higher survives. The winner does not simply kills the looser, but

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
D. H. Milone, G. Stegmayer & D. Beber; "ARTIFICIAL LIFE CONTEST: a tool for informal teaching of artificial intelligence"
International Conference on Computer Supported Education (CSEDU 2009), pp. 119-124, Mar, 2009.

an amount of energy is taken which is equal to the difference of energy between them. If the looser remains with an energy level minor to zero, it then dies. After the fight, both MOs remain in the same previous position and the winner increments its energy in a 7.5% level regarding the looser.

3. Reproduction may be requested, by mitosis (a process of cell division, which results in the production of two children from a single parent cell). It could take place if there is a free adjacent place to put the new MO, and the father and the son remain with an energy equal to the 49% of the father energy.

### 3.2 The Agar class

The *Agar* class acts like a proxy between the agents and the artificial environment, providing the MOs, when requested, the perceptions they can use for internal decision making. It models the food where the MOs may feed from and, this way, increase their energy. Agar knows the exact position and energy of every MO in the Petri dish, and this information is constantly updated.

The MOs can also ask the Agar about any information they may need, that is, the nutrient distribution, the amount of nutrients on a specific position $(x,y)$ or the energy and species of a MO located at this point.

A very simple example of such kind of requests may be the following:

```
if (id <> agar.ocupation(myX+1,myY))
{ // to do something with a competitor
  // existing at the right of my position
}

if (ene > agar.energy(myX+1,myY))
{ // to take action according to energy
  // levels
}
```

where `id` and `ene` are the identifier and the energy of the MO, respectively.

## 4 Artificial Life Contest: the software agents

In this artificial life model, the software agents represent microorganisms (MOs) that must live inside a Petri dish, feeding, reproducing themselves or fighting for survival. Each contestant must develop the survival algorithm of their MOs that will compete with other colonies for the exclusivity of the environment. The contest provides two basic classes that model the MOs and their colonies: the *Colony* class and the *Microorganism* class.

The *Colony* class contains a group of MOs that belong to the same species (in this case, same participant). Its main responsibility is moving the MOs when Petri requires it. The *Microorganism* class defines the MOs basic behavior. Starting from this class, the students must develop their MOs by inheritance, with their own survival tactics.

The possible actions that a MO may perform to achieve its survival goal are: eating the food, reproducing itself though mitosis and fighting against an enemy colony. Each MO must provide a virtual method that returns the position where the MO wants to move to, and consequently making a choice about feeding and attacking. In addition, MOs must provide another virtual method to communicate their decisions about mitosis.

Figure 4 shows a very simple example of a competitor MO, named *SearchN*, because its strategy consists of simply looking for the nearest position with higher food level. Through `agar.nutrients(x,y)`, a perception regarding the amount of nutrients existing in any position is requested. In the example, a mitosis as simple as: "if the MO has more than 5000 units of energy then divide" is implemented. This MO is very simple because it only considers the positions where there are more nutrients, and does not takes into account enemies positions, nor those of its own colony.

More developed algorithms may make better use of the available information to achieve more effective strategies, such as making mitosis according to the amount of available nutrients, attacking the contrary MOs, seeking and remaining in regions having more food, among others.

## 5 Artificial Life Contest: discussion

The idea of converting the original artificial life model into a contest was a great motivation and a real challenge for the students. They arrive to the competition as contestants, that is to say, developers of microorganisms, as well as developers of the environment. Each year new contestants arrive and only a few of them participate more than 2 consecutive years.

About each 2 weeks we organize the "tournaments" where all the participants bring their new releases of microorganisms and "fight" in six "battles" for the survival in our common environment. Students may form groups to design a MO-agent. A tourna-

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
D. H. Milone, G. Stegmayer & D. Beber; "ARTIFICIAL LIFE CONTEST: a tool for informal teaching of artificial intelligence"
International Conference on Computer Supported Education (CSEDU 2009), pp. 119-124, Mar, 2009.

```
//------------------------------------------------------
class SearchN: public Microorganism { public:
  virtual string name();
  virtual string author();
  virtual void move(Movement & mov);
  virtual void mitosis(bool & dup);
};
//------------------------------------------------------
string SearchN::name() { return("Search for Nutrients"); }
//------------------------------------------------------
string SearchN::author() { return("DGD"); }
//------------------------------------------------------
void SearchN::move(Movement & mov)
{ // the MO will move to any
  // of the 8 neighborhood positions having more food

  int x_rel, y_rel;  // relative testing position
  int x_max, y_max;  // relative position having more
                     // nutrients
  x_max=0;
  y_max=0;

  for (x_rel=-1; x_rel<2; x_rel++)
    for (y_rel=-1; y_rel<2; y_rel++)
      if (agar.nutrients(pos.x+x_rel,pos.y+y_rel) >
          agar.nutrients(pos.x+x_max,pos.y+y_max))
      { x_max=x_rel;
        y_max=y_rel;
      };

  mov.dx=x_max;
  mov.dy=y_max;
}
// -----------------------------------------------------
void SearchN::mitosis(bool & dup)
{ if (ene>5000) dup=true;
  else dup=false;
}
// -----------------------------------------------------
```

Figure 4: Example of a simple MO-agent code.

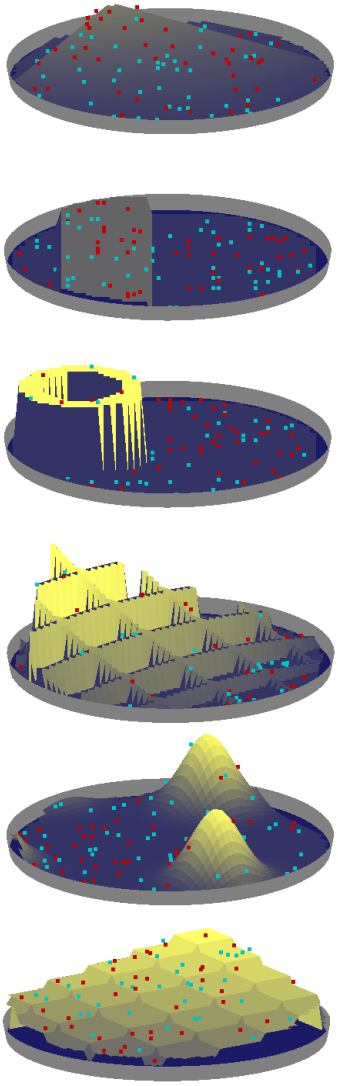

Figure 5: The six nutrient distributions used in the Artificial Life Contest battles. From top to bottom: inclinated plane, vertical bar or uniform distribution, ring, lattice, bimodal gaussian and five steps stair.

ment begins and ends with only one source code per group, but it can be improved during the weeks between two tournaments.

Each battle is performed on six different food distributions (shown in Figure 5), such as: an inclinated plane, a vertical bar or uniform distribution, a ring, a lattice, a bimodal gaussian and a five steps stair distribution. The colony that dominates the other ones and accumulates the major living energy is the winner of the tournament and wins 3 points for the global ranking; the second one obtains 2 points and 1 point is for the third one.

Open source is a very important philosophical framework behind all development about computer science, both for teaching as for academic research (Sonnenburg et al., 2007). Thus, we promote these ideas not only by publishing the source code of the environment but also by uploading the source code of the winner of each tournament (with public access for all the other participants). At the end of the championship, the participant with more points is awarded with some gift from the organizing Universities and all the participants explain their strategies to the others in a common presentation.

To offer an environment as much unstructured or informal as possible, we have minimized the rules of the competition. If there is any situation outside of what would be expected, the same participants are the ones who discuss and decide, guided by the teachers, about how to solve the problem. For example,

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
D. H. Milone, G. Stegmayer & D. Beber; "ARTIFICIAL LIFE CONTEST: a tool for informal teaching of artificial intelligence"
International Conference on Computer Supported Education (CSEDU 2009), pp. 119-124, Mar, 2009.

in principle it would not be allowed something like "hacking" the Petri class. However, if the idea is original enough that the remaining participants may learn from this code and accept it, the method can become a valid strategy for future tournaments.

Another example of an unexpected situation was a winner that had encrypted all of his source code, changing identifiers by sequences of "_" characters. This way, when his code was made public and available to all other contestants, nobody could know his strategies. This situation was submitted to the opinion of all other participants and it was decided that, if the competitor wanted to continue participating, he should provide a completely documented source code.

# 6 CONCLUSIONS

This paper has described an Artificial Life competition based on a computer program that simulates an artificial life environment: a Petri dish where two colonies of microorganisms must survive. The contest model has been explained and exemplified, and the competition rules have been presented as well.

This didactic tool has generated great enthusiasm regarding programming and AI techniques among students. We noted important improvements along the competitions in many different aspects: abstraction, codification skills, team work, dedication, innovation, creativity when designing and implementing software agents; something that does not generally happen when teaching with traditional structured methods.

The publication of the winners source code was a big motivation for improving other competitors codes, learning by themselves from the winner strategies. Some people were more motivated to improve the artificial simulation environment while others preferred to focus on more competitive algorithms. Many of them tried to applied advanced AI techniques, such as artificial neural networks or genetic algorithms, reading and asking about these subjects to teachers, but without a formal structure.

We consider that this kind of informal learning methodology has provided an interesting opportunity to students to exercise the use of their imagination to solve previously unknown problems through self-learning, without a formal obligation, which also promoted self management. In general, the students that participate in the contest could organize their duties and manage their available time, exercising responsibility by themselves, without it being imposed from external pressions nor formal structures.

Among future work we can cite the development of a new model design, that could provide higher flexibility (through plugins) to generate different kinds of competitions, for example distinguishing between beginners and advanced students. Also, we are working to provide a new version of the contest in Java, which could simplify the multi-platform programming and would allow to also incorporate pre-compiled C++ MOs, with a more compact and integrated graphical interface.

# REFERENCES


Chiang, A. (2007). Motivate ai class with interactive computer game. *Proc. of IEEE Int. Workshop on Digital Game and Intelligent Toy Enhanced Learning*, 1(1).

Hingston, P., Combes, B., and Masek, M. (2006). Teaching an undergraduate ai course with games and simulation. *LNCS*, 3942(1):494–506.

Kim, I.-C. (2006). 3d interactive computer games as a pedagogical tool. *LNCS*, 4270(1):536–544.

Lotka, A. J. (1925). *Elements of physical biology*. Williams & Wilkins Co., Baltimore.

Martens Alke, D. H. and Steffen, M. (2008). *Transactions on Edutainment I - Game-Based Learning with Computers Learning, Simulations, and Games*. Springer.

Milone, D., Beber, D., and Biurrun, J. (2003). Artificial Life Contests: Encouraging Creativity. In Doblaré, M., Cerrolaza, M., and Rodríguez, H., editors, *Proceedings of the International Congress on Computational Bioengineering*, Zaragoza, España.

Pantic, M., Zwitserloot, R., and Grootjans, R. J. (2005). Teaching introductory artificial intelligence using a simple agent framework. *IEEE Transactions on Education*, 48(3):382–390.

Sonnenburg, S., Braun, M. L., Ong, C. S., Bengio, S., Bottou, L., Holmes, G., LeCun, Y., Mller, K.-R., Pereira, F., Rasmussen, C. E., Rtsch, G., Schlkopf, B., Smola, A., Vincent, P., Weston, J., and Williamson, R. (2007). The need for open source software in machine learning. *J. Mach. Learn. Res.*, 8:2443–2466.

Volterra, V. (1926). Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. *Mem. R. Accad. Naz. dei Lincei. Ser. VI*, 2.



sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
D. H. Milone, G. Stegmayer & D. Beber; "ARTIFICIAL LIFE CONTEST: a tool for informal teaching of artificial intelligence"
International Conference on Computer Supported Education (CSEDU 2009), pp. 119-124, Mar, 2009.