ARENA and WOXBOT: First Steps Towards Virtual World Simulations

FÁBIO ROBERTO MIRANDA[1], JOÃO EDUARDO KÖGLER JR.[1]
MÁRCIO LOBO NETTO[2], EMÍLIO DEL MORAL HERNANDEZ[2]

Faculdade SENAC de Ciências Exatas e Tecnologia
Rua Galvão Bueno, 430, São Paulo, SP, Brasil
(fabio.rmiranda, kogler)@cei.sp.senac.br

[2]EPUSP- Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, 158, trav. 3, 05508-900 São Paulo, SP, Brasil
{ emilio, lobonett}@lsi.usp.br

Abstract. This paper reports new results of a project to build virtual worlds aimed at the graphic simulation of an arena where small mobile robots can perform requested tasks while behaving according to their own motivation and reasoning. Each robot is an intelligent agent that perceives the virtual environment via a simulated vision system and reacts translating or rotating its body by driving its own wheels. The conception and specification of the robots and the environment are being done very carefully to create an open distributed object architecture that could serve as a testbed freely available and ready to use for testing theories in some computational areas such as evolutionary computation, artificial life, pattern recognition, artificial intelligence, cognitive neurosciences and distributed objects architectures.

1 Introduction

Artificial life is a term originally intended to mean the simulation of macroscopic aspects of living beings behavior using microscopically simple components [1]. However, the term spanned to designate a wide variety of simulated living creatures, including virtual characters whose behavior emerges from hierarchical and functionally specialized complex structures like an animal body [2]. Both kinds of simulations are very interesting from the computational point of view: they offer very attractive means to computationally model complex behavior, a subject that had gained a special relevance under both the theoretical and the applied standpoints.

Artificial life worlds are computational simulations like virtual places where animated characters interact with the environment and with other virtual beings of the same or distinct categories. Different degrees of sophistication can be found in these virtual creatures, from unicellular life with minimalist models to complex animals with detailed biomechanical models. However, all of them display behavior emergent from the dynamics of a complex system (for instance, systems with learning, reasoning, ontologies, cognitive processes, etc).

The present paper presents recent new results of a project set to build an artificial environment for animated virtual creatures (ARENA) [13]. Although it was conceived to allow more than one robot or virtual creature to be easily introduced to co-exist in the same environment, its first version presented here, is a single-robot virtual world.

Our goal is the exploitation of diverse strategies used by the creatures to perform tasks while adapting themselves to the environment. This scenario can be useful for many purposes: for behavior modeling, as a laboratory of learning algorithms, for research on societies of virtual characters, in the study of populational collective dynamics, etc. At the present stage the ARENA robot – WoxBot (Wide Open Extensible Robot) – has a vision system consisting on a simulated camera and a neural network that classifies the visual patterns to provide input to a motor system controlled by a deterministic finite state machine. This automaton is obtained from an optimization procedure implemented by a genetic algorithm.

1530-1834/01 $10.00 © 2001 IEEE

As an example of application, a given research project can be targeting visual recognition methods, so it will be using WoxBot as a subject employing the methods under test, and will take ARENA as a laboratory for evaluation experiments. Another research project instead, can be targeting autonomous robot mobility algorithms and traveling strategies. In this case, the ARENA floor will be the field of traveling, and WoxBots will be simulating the autonomous vehicles. Yet the WoxBot could be re-shaped to have the aspect of a given car or truck model and the ARENA could be configured to resemble a street, in a project intended to analyse vehicle traffic conditions.

The ARENA is implemented as a distributed object environment and can run on single processor platform as well as can take advantage of parallelism or multithreading in high performance computer architectures. In this later case it could be used to handle highly sophisticated and complex applications, such as the simulation of a street with many cars. It could also be used to evaluate multi-agent distributed computational models using the WoxBots as the prototypical agents.

The organization of the paper is as follows: the next section gives an overview of the project, pointing its requirements and choices for solutions, addressing also some issues of ARENA. Sections 3 to 6 review several conceptual aspects involved in the various project components and give guidelines to be used in the implementation. Section 7 presents the results obtained. Section 8 foresees the further steps to be taken with the project.. A discussion of particular implementation issues, like the choice of Java3D as the graphical library, addresses the modeling of the character and the environment can be found in [13] and [15].

2 Project overview

Our long-term research line in artificial life is aimed to the development of complex virtual worlds with realistic creatures able to exhibit sophisticated behaviors supported by reasoning, learning and cognition. To achieve it, one is required to account for the following aspects:

- account for the wide variety of mathematical and computational models usually required to represent the complex aspects of living behavior;
- provide an efficient environment and platform with enough computational power to appreciate the details of the simulated creatures behavior in real time, including the ability to support live interactivity with persons;
- specify and build a system that embodies constant evolution and improvement in its constructs at the same time that enables or, indeed, promotes the reuse of well succeeded results and implementations.

Our choice was to fulfill these requirements building an application development environment as an architecture of distributed communicating objects mapped on a microcomputer cluster [3]. It is a non-expensive scalable parallel computing architecture with high performance capability that can be used with a non-homogeneous set of operating systems, like Linux and Windows NT running simultaneously on distinct nodes of the cluster. The use of the object-oriented paradigm promotes code reuse and modularity while giving great simplicity for maintenance and evolution. This is a capital aspect to cope with the necessity of testing many computational and mathematical models to account to the wide variety of behavioral theories. The ARENA implementation exploits parallelism via multithreading, so the program can run both on multi and single processor platforms.

The structure of ARENA is the following: it has a floor plan and walls encompassing the limits of the virtual world. Inside this scenario one can place objects of several kinds and functionalities like obstacles, barriers, traps, shelters, energy or food sources etc, serving as pitfalls or resorts for the creatures (figure 1). One or many characters will be introduced in this environment to perform certain tasks. In the first stage of the project the sole character will be the WoxBot (figure 2) whose task is to keep itself alive as long as it could. Section 8 presents the specific details of the present state of the project and section 9 will point the further improvements.

Our desire is to provide a very general environment to serve as a workbench for many applications focused in artificial life. Certain conceptual ingredients play specific roles on composing the artificial life scenario and thus should be present in this implementation: *sensing and perception, knowledge use,* and *evolution*. Sensing and perception is the way by which the creature can grab information from the environment essential to update its internal state when performing behavior regulation. This last concept refers to the tasks and purposes of the creature. The invariants detected by perceptual system can be used to construct or update a knowledge base or ontology. A knowledge base (or, more specifically, the ontology) constitutes stable and structured information that can be re-used by the creature to reason about the environment, to predict the effects of its own behavior and to perform task planning. Finally, evolution is the ingredient that enables a creature to change its adaptation to a variable environment. In the following sections we will examine these aspects.

3 Artificial Life

Artificial life has been associated to computer science by many ways, ranging from cellular automata theory [4] [5], to computer graphics animation [3][6][7]. Some researchers have been seeking models for the beginning and evolution of life in nature [4]. Graphics animators and scientists have been looking for physical models to give natural appearance and behavior to their simulated characters [7]. Although very different in nature, these works have been related to evolution and natural selection concepts.

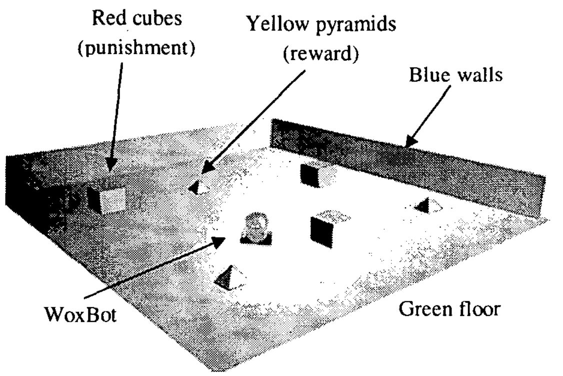

Figure 1: ARENA (Artificial Environment for Animats) – a virtual world for the development of artificial life projects. The character at the center is a robot that avoids the cubes and seeks the pyramids. A contact with a cube reduces the robot life, while the pyramids extend it. The robot goal is to prolong life as long as it can. An evolutionary computing scheme selects the better performing robots. The environment can support several robots.

There are two computational issues that emerge in the subject of modeling life, a local and global one. The local aspect refers to the ability of an individual creature to regulate its own behavior. The global one concerns with the adaptation of a certain species. The global aspect emerges from the local in the sense that a species is well adapted to its environment if the corresponding creatures are also successful individually. These issues are manifest in the genotype and in the phenotype. The genotype carries the transmitted characteristics of a creature and as an equivalence class it represents the species. The phenotype refers to a specific individual. To design an artificial creature is to find an organism that can reach a specified regulation of behavior when performing some task. The organism can be modeled with some representational language, so let us say that it is given by some automaton. Structured design methodologies are very difficult to use in this field, even when a very good specification and description are provided. The main drawback is to cope with an evolving complex environment. Evolutionary computation is a modeling and design paradigm well suited to these cases and its main tool is the genetic algorithm.

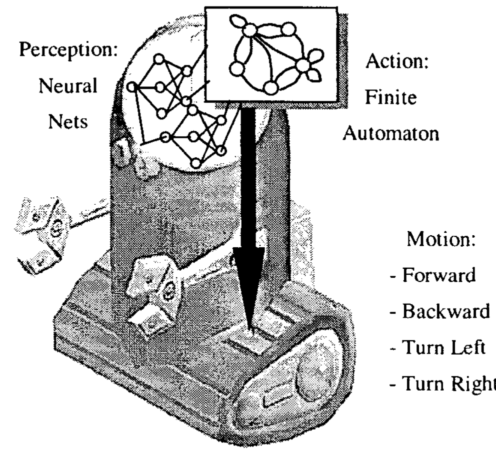

Figure 2: WoxBot (Wide Open Extensible Robot) – a virtual character that gets input from vision and reacts moving according to the decision taken by the finite state automaton. The automaton gets inputs from the pattern recognition module – neural nets trained to perform visual input classification. The WoxBot has a distributed object open architecture that enables changing the perception and action algorithms very easily.

Genetic algorithms are computational procedures to find the optimal solution in particularly hard problems [9][10][11]. Each creature is represented in this context by a character string that constitutes its own phenotype. A population of such creatures has a genotype that is the equivalence class of the phenotypes under a similarity relation that can characterize the species. The genetic algorithm used in the creature design consists in assisting some transformations carried in the genotypes of members of a population of such virtual creatures, allowing them to change from one generation to another and providing a selection of the best ones. The first generation starts with an arbitrary set of genotypes, *a priori* sub-optimal. Genetic mutation and combination (by reproduction) provide efficient ways to modify the creature's characteristics, as in natural life. Selection comes after, choosing those ones that, by some criteria, are the best suited, and therefore are allowed to survive and reproduce themselves.

In our work evolutionary processes are crucial components for our virtual character. Each WoxBot is controlled by a finite state automaton and the challenge is to design one that performs the WoxBot tasks efficiently to cope with complex inputs coming from the perceptual modules. The design procedure is carried with a genetic algorithm in which the automaton evolves through generations, based on genetic mutations and crossing-over. Each robot descendant arises with some innovative features that may result on a better or worse adaptation to the environment than those from previous generations. The individuals are selected with some given criteria (for instance, keeping greater energy, living more time, performing faster its tasks, etc) and the ones that achieve higher grades have higher chances to be selected to compose the next generation. This process tends to an optimum according to the efficiency of the problem specification and domain representation [9][10].

4 Intelligent Agents

The use of knowledge in the artificial life environment can be done under two aspects: (i) it can be present in the environment and in the conception of the creatures and, (ii) it can used by the creatures themselves when performing their actions. In both cases, the effective and rational use of the knowledge (about the world, the actions, the tasks) leads to what is called intelligent behavior. Intelligence can also be considered a property emergent from evolutionary systems [9]. Thus, the evolving characters can be modeled as intelligent agents performing tasks in the virtual environment.

Intelligent agents may be understood as computational entities that behave with autonomy in order to manipulate the information associated to its knowledge [12]. This autonomy must regard the agent design, the environment, the goals and motivations. They have the capability of reasoning about what is going on in the surrounding environment and about the consequences of their own actions, or to respond to some query conducted by other agents. As this are evolved many features of agents have been identified, as their capacity to communicate with other agent, their mobility (migrating from one computer to another through networks), their intelligence (how efficiently they can perform or use reasoning), and so forth.

The WoxBot was conceived starting from its specification as an intelligent agent. In this first implementation, it has been tested individually, without interaction with other Woxbots and without mobility over the processor cluster network, although ARENA was designed as a distributed system. This paper focuses only in aspects of the WoxBot design and training.

The WoxBot was specified having a vision sensor and all input from the exterior will arrive from this port. It supports a knowledge base or an ontology, although it was not yet included. And it regulates its behavior in response to the visual inputs and its own purposes with a finite state automaton. This architecture can support also reasoning, but we will incorporate latter this feature. The communication and mobility issues will also be incorporated in the next release of ARENA and WoxBot.

By now we are interested in the perception and behavior regulation issues. The perceptual module and the finite state machine have been designed for these tasks, and we made use of learning and evolution as the ingredients responsible to provide a somehow complex agent. The robot tasks now seem simple: it should look for energy sources, while avoiding traps. However, with a single agent, the possible situations are still many and the tasks could be performed following a sophisticated plan, considering that the performance criteria could be set to include more constraints, like minimal time, minimum use of energy, etc. As explained ahead, based on visual information, the robot plans its way to get closer to the energy sources, and this planning is in fact conducted by the finite state machine that evolved through generations.

On next versions of ARENA we will explore multi-agent interaction, and WoxBot and other derived creatures will be able to display sociability skills, which is an important feature to promote a better adaptability. Furthermore, they will also implement mobility features, since we plan to have the multiple creatures living in a virtual environment, supported by the distributed computer cluster, the SPADE.

5 Perceptual modules and learning

One of the tasks that WoxBot has to perform in this first project is to be aware of nutrients (yellow pyramids) and hurting entities (red cubes) that are present in ARENA (figure 1). The visual information is gathered from the 3D surrounding scene by projecting it to a viewport with 3 color channels, namely R (red), G (green) and B (blue). Figure 3 depicts a typical input, which at the present state is taken in low resolution, since there are no textures and the accuracy is not yet affected. To differentiate and spatially locate these entities, two specialized neural networks interpret the visual information received by the robot, one of them targeting nutrients and the other targeting hurting entities. These networks are named here ANN-I and ANN-II.

ARENA and WoxBot

The network ANN-I is trained for the identification of yellow pyramids as well as for a general evaluation (classification) of the position occupied by the principal yellow pyramid in the robot's visual field. To help in this task, a simple filter for the yellow color is implemented as a pre-processing stage. This is done by combining the primary sensory channels RGB so to obtain a gray scale image where yellow is mapped into high values of gray and the remaining colors are mapped into low values of gray.

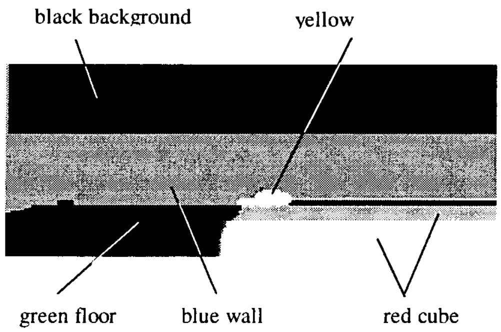

Figure 3: A typical image of the ARENA environment as seen by the WoxBot vision. The actual image has the depicted colors with shading resulting of the illumination. This image is the input of the neural nets that perform perception. It is also possible to add textures to the 3D scene and make yet more complex patterns.

The monochromatic image obtained in this way is then used as the input to the ANN-I, the one that targets nutrients. After training, this network is able to activate one of 4 outputs (see Table I): output 1, indicating that no yellow prism is present in the visual field; output 2, indicating that the principal yellow prism is on the robot's left, output 3, when the principal yellow prism is on the center of the visual field, and output 4, when the principal yellow prism is on the right. In this way, ANN-I makes the robot aware of the presence and location of nutrients.

ANN-I	Scene
0	No yellow pyramids
1	Biggest pyramid at left
2	Biggest pyramid at center
3	Bigest pyramid at right

Table I: ANN-I outputs

In the same lines as ANN-I, the second neural network, ANN-II, performs a similar function for the identification and position evaluation of the red cubes representing the hurting entities (see Table II). For this second network, however, the gray images that are sensed by the neural network inputs are obtained by using a different filter, which combines the channels RGB so to favor the red color.

ANN-II	
0	No red cubes
1	Biggest cube at left
2	Biggest cube ahead
3	Biggest cube at right

Table II: ANN-II outputs

The architecture chosen for ANN-I and ANN-II is the standard multi layer perceptron with one single hidden layer of moderate size and learning through the error back- -propagation algorithm. Only eight hidden nodes were enough for this recognition task. Notice that this low number is consistent with the fact that we are dealing with a low number of output classes, i.e., only four.

Among the observations that we did during the training of the neural networks we want to mention that the detection and position classification of visual targets was facilitated by the use of samples (pyramids and cubes) of similar sizes. In other words, ANN-I and ANN-II don't have good abilities to implement scale invariance. That was in fact expected, since the complexity of the network is relatively limited, and its architecture does not have any specialized organization proper to implement scale invariance. This indicates that some kind of size invariance mechanism could be added in the pre-processing stage. Another possibility is to use a larger and more complex neural architecture so to make possible the interpretation of targets positioned at very discrepant distances. Again, WoxBot was conceived to allow these improvements and others to be easily implemented, granted by its modularity and object-oriented open architecture.

An important issue to consider in the context of the neural networks of WoxBot is adaptability. In these initial experiments with the robot, the training of ANN-I and

ANN-II was done in an isolated phase, which was performed previously to the exploration of ARENA. In other words, only after WoxBot was able to do the differentiation between nutrients and hurting entities, he started his exploration in ARENA, and then, no further adaptation of the visual recognition system was allowed. In future phases of this project, we want the recognition system to be able to deal with unpredicted changing conditions such as change of illumination, and we will include the change of the features of the nutrients targeted by WoxBot (shape and color for example). Of course, this depends strongly on the ability of the visual recognition system to adapt, since the target recognition prototypes change with new experiences and new needs of the robot, as its life and exploration of ARENA evolves. At that point of the project, the intrinsic adaptive nature of neural networks will be crucial, and it will be fully explored.

6 Action and Behavior

The WoxBot character is an intelligent agent with a simulated visual sensor to pick images of the environment from its point of observation. The two neural networks inside the agent analyze these images classifying the visual patterns. These networks outputs are tokens to the agent to make a decision about what to do. Based on this information it can decide to go ahead, turn right or left in order to reach a pyramid or avoid a cube (see Table III).

Code	Action
0	Turn left
1	Go ahead
2	Turn right
3	Go backward

Table III: Action encoding

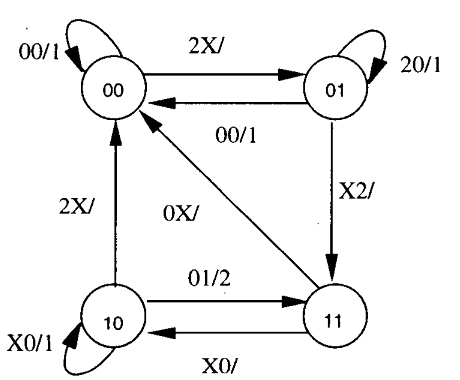

Figure 4: A sample finite state automaton used by WoxBot to control its reactions to the visual stimuli.

The motion control is performed by a finite state machine. This automaton was designed without specifying how the inter state transitions should occur. It is set initially with a random structure that is improved changing based on evolutionary computation concepts (see figure 4 for an example).

The state machine is coded into a string of bits representing its states, inputs and actions. This string is named the robot's chromosome. The 16 possible inputs are the combinations of the outputs of the two ANN's as shown in Tables I and II, coded in octal in the example of figure 4. As an example, figure 5 depicts a section of a sample chromosome, corresponding to the inputs 00 and 20 occurring at the state 01 of the automaton of figure 4.

State	...	01	01	...
Transition	...	00	01	...
Action	...	1	1	...
Input	...	00	20	...

Figure 5: Sample chromosome corresponding to the automaton of figure 4. Actions and inputs are coded in octal.

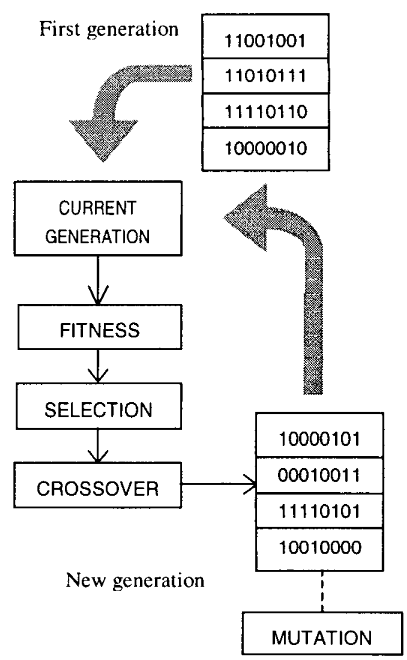

Figure 6: The evolutionary process. It starts with a generation of WoxBots with arbitrary automata. These undergo successive genetic optimization steps to produce the best adapted automaton that gives WoxBot an optimal performance.

At the start of the robot life the contents of this string are set at random, resulting in an arbitrary state machine. The optimization procedure via genetic programming begins with a set of robots initially created with arbitrary state machines (see figure 6). The optimization criterion is set to accomplish the goal of maximizing the duration of the robot life. Then each of these initial robots are put to live individually in the ARENA, and their lives duration are recorded. The best-suited, i.e. the ones who lived more, are then preserved. They can generate descendants by reproduction making the crossover of the chromosomes producing a new breed. The newer generation is then tested in the same manner, until no expressive changes in the robot life duration could occur. Then, a random mutation is produced in some of the chromosomes, to induce variety and to proportionate a new strategy. Based on this concept we expect to be able to see after many generations the survival of the most adapted creatures.

7 Simulation and Results

A genetic algorithm was used to evolve a population of 12 individuals through 39 generations. The first generation was randomly generated. Every following generation had one half its members randomly generated, and the other half created by a crossover of genotypes of the two most fit individuals of the previous generations. The crossover point was picked at random on the parent's chromosomes. After a child had been generated, each of its bits had a liability of 4% of being mutated.

Simulated robots used a state machine with 4 states, which resulted in genotype of 258 bits (4 times 64 bits – the length of the description of each state/transition pair, plus 2 additional bits to indicate the initial state of the machine).

Generations are tested at the arena sequentially, and, within each generation, one of its members is tested after another. Robots are given an initial lifetime, that can be extended or shortened based on its behavior. Each time the robot collides against an pyramid, it receives extra 8 seconds of life, if it collides against a cube, there is a loss of 7 seconds o life. Both cubes and pyramids disappear upon contact, and reappear at a random place after 2 seconds.

The results discussed here are based on two configurations of robot evolutions, both using parameters described above, but differing in the initial amount of time given to robots: in the first configuration, robots were given 60 seconds of time to live; in the second configuration, robots received 120 seconds. Each configuration was simulated 3 times.

The results of simulating both configurations show an increasing average fitness of robot population. Figure 7 shows the average fitness of robot generations simulated using the first configuration, and figure 8 shows the average results of simulation using second configuration.

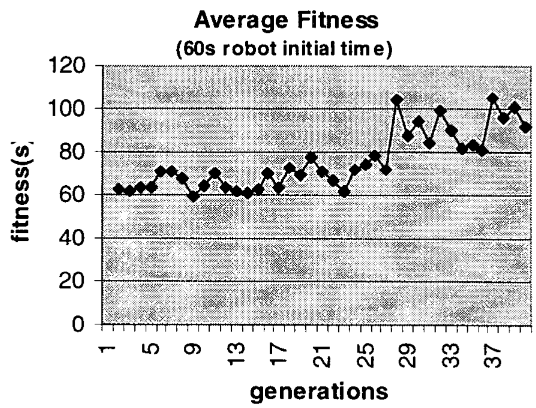

Figure 7: Average Fitness for simulations where virtual robots were given initial 60s to live.

Besides the observed increase in average fitness in both simulations, we could also observe very successful individuals that scored fitnesses noticeably above average. The most successful individual belonged to 28th generation of one of the simulations using 60s as robot initial time to live, this particular individual lasted 428 seconds.

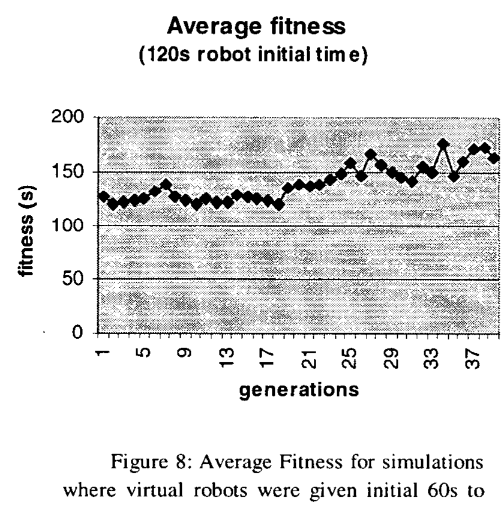

Figure 8: Average Fitness for simulations where virtual robots were given initial 60s to live.

8 Conclusion and Future Prospects

Results presented here provide us inspiration to keep evolving the ARENA to make it suitable for more kinds of experiments.

There is a number of ways to evolve work presented here. One of them heads to variation of parameters, like we varied initial robot time in this work, trying to understand better what makes some configurations of the arena result in more successful evolutions than others, what is the effect of the number of robot states in the quality of emerging behavior, how the size of the arena and values of reward and punishment affects the result of evolution, how variations in the number of objects reflects at behaviors, among others.

The internal structure the robot uses for action selection could also be improved, for example by adding a memory to allow the robot to remember past actions and a mechanism to somehow make this memory helpful to the agent. Steps toward making the robot learn during his lifetime (the learning Woxbot has now is *species* learning) need also to be taken.

Improvements can and will be added to the simulation engine to allow more than a robot, and different kinds of agents being simulated at the same time, and communicating with each other. The environment where agents live can grow in geometric and visual detail, creating more challenges to the agent's visual system, that would have to be improved.

References

[1] E.W. Bonabeau and G. Therulaz, "Why do we need artificial life?", *in Artificial Life an Overview,* edited by C.G. Langton, MIT Press, 1995.

[2] D.Terzopoulos (org.), *Artificial Life for Graphics, Animation, Multimedia and Virtual Reality,* SIGGRAPH 98 Course Notes 22 (1998).

[3] Volnys Bernal *et al*, "PAD Cluster: An Open Modular, and Low Cost High Performance Computing System", *in Proceedings of 11^{th} Symposium on Computer Architecture and High Performance Computing* Natal, RN.Brazil, Sep 29^{th}-Oct 2^{nd} (1999), 215--222.

[4] C. Adami, *Introduction to Artificial Life*, Springer Verlag, 1998

[5] A. Dorin, 'Physically Based Self-Organizing Cellular Automata", *in Multi-Agent Systems – Theories, Languages and Applications*, C. Zhang and D. Lukose (eds), Lecture Notes in Artificial Intelligence v. 1544, Springer Verlag, 1998

[6] C.Phillips and N.I. Badler, "Jack: A toolkit for manipulating articulated figures", *in ACM/SIGGRAPH Symposium on User Interface Software,* Banff, Canada, Oct. (1988).

[7] N.I. Badler, C.B. Philips and B.L. Webber, *Simulating Humans: Computer Graphics, Animation and Control*, Oxford University Press, 1992

[8] P. J. Bentley (ed.) , *Evolutionary Design by Computers*, Morgan Kaufmann, 1999

[9] D.B. Fogel, *Evolutionary Computation - Toward a new philosophy of machine intelligence*, IEEE Press, 1995

[10] J.H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1992

[11] J. Koza, *Genetic Programming,* MIT press. (1992).

[12] S. Russell and P. Norvig, *Artificial Intelligence – A modern approach*, Prentice-Hall, 1995

[13] M .L. Netto, J. E. Kogler Jr.,E. M. Hernandez, F. R. Miranda , "WoxBOT – A wide Open Extensible Robot for Virtual World Simulations", *in Proceedings of the GRAPHICON '2000 – 10th. International Conference in Computer Graphics and Vision*, Moscow, Russia, Aug 2000.

[14] R.D. Beer, H.J. Chiel and L.S. Sterling, "A biological perspective on autonomous agent design", *in Designing Autonomous Agents*, P. Maes (ed.), MIT Press, 1990

[15] F.R. Miranda, J. E. Kogler Jr., M. L. Netto, E. M. Hernandez - "ARENA and WoxBot: Some steps towards the animation of cognitive characters actin in a virtual world" - SENAC College of Computer Science and Technology Technical Report – RT 04 / 2001 – May 2001.