ALife for Real and Virtual Audio-Video Performances

Luigi Pagliarini
Centre for Playware, Technical University of Denmark,
2800 Kgs. Lyngby, Denmark
E-mail: luigi@playware.dtu.dk
www.playware.dk

Henrik Hautop Lund
Centre for Playware, Technical University of Denmark,
2800 Kgs. Lyngby, Denmark
E-mail: hhl@playware.dtu.dk
www.playware.dk

Abstract

MAG (an Italian acronym which stands for Musical Genetic Algorithms) is an electronic art piece in which a multifaceted software attempts to "translate" musical expression into a corresponding static or animated graphical expressions. The mechanism at the base of such "translation" consists in a quite complex and articulated algorithm that, in short, is based on artificial learning. Indeed, MAG implements different learning techniques to allow artificial agents to learn about music flow by developing an adaptive behaviour. In our specific case, such a technique consists of a population of neural networks – one dimensional artificial agents that populate their two dimensional artificial world, and which are served by a simple input output control system – that can use both genetic and reinforcement learning algorithms to evolve appropriate behavioural answers to an impressively large shapes of inputs, through both a fitness formula based genetic pressure, and, eventually, a user-machine based feedbacks. More closely, in the first version of MAG algorithm the agents' control system is a perceptron; the world of the agents is a two dimensional grid that changes its dimensions accordingly to the host-screen; the most important input artificial agents get (i.e. not necessarily the only one) is the musical wave that any given musical file produces, run-time; the output is the behavioural answer that agents produce by moving, and thereby drawing on to a computer screen, therefore graphical. The combination of artificial evolution and the flows of a repeated song or different musical tunes make it possible for the software to obtain a special relationship between sound waves and the aesthetics of consequent graphical results. Further, we started to explore the concept of run-time creation of both music and graphical expression. Recently, we developed a software by which it is possible to allow any user to create new song versions of popular music with the MusicTiles app simply by connecting musical building blocks. This creation of musical expression can happen as a performance (i.e. run-time). When connecting the MusicTiles app to the MAG software, we provide the connection and the possibility to melt both musical expression and graphical expression in parallel and at run-time, and therefore creating an audio-video performance that is always unique.

Keywords: Playware, Art, Music, Graphics, Artificial Life.

1. Introduction

In this paper we present the "MAG"[1] application, an ALife software based on and derived from our previous work on the ALife models and work. MAG (i.e. Musical Genetic Algorithm, in Italian) is an electronic art piece that throughout an automatism translates sound expressions into graphical ones. The mechanism behind such "translation" is a quite complex algorithm that,

synthetically, relies on artificial learning. Indeed, the MAG software implements a population of artificial agents – living in an artificial world and furnished with a simple sensory-motor system – that genetically evolves to coherently respond to different inputs, thanks to the most classical selective pressure.[2]

In the MAG software the agents control system is a Perceptron,[3] the simplest version of a Neural Network[4]; The world in which such perceptrons are placed is a two dimensional grid of adaptive dimensions and with a toroidal shape; The main input which the artificial organisms respond to is a sound wave; the artificial agents' answer to inputs is a behavioural one where the behavior is traced and graphically impressed onto the computer screen.

The combination of artificial evolution and the execution of a song or many different songs make it possible to achieve at any software execution a special and unique relationship between incoming sound waves and the consequent emerging graphical expression.

After having made all possible efforts to reach highly aesthetical results, and once obtained such a relationship between music and graphics, we developed the concept of applying the MAG algorithm to our MusicTiles application[5, 6] - a software conceived to open up music remixes and composition to ordinary people and professional users. All of that, because our final intention is to research for a mechanism by which any kind of user could use a tool for creativity to live performing both music and graphics, sound and images.

2. Technology

The underlying technologies of the MAG algorithm belong to the classic simulative logic with its roots in Artificial Life[7], and which we already used for artistic means[8, 9]. Within an artificial planet (i.e. a two dimension matrix, a chessboard with a toroidal spatial geometry) a population of artificial organisms - whose N shapes according to the screen size – starts evolving and by doing so draws on an "electronic canvas" a continuously-transforming image.

Artificial organisms, that are evolved thanks to a Genetic Algorithm[10], occupy a Cartesian point of such world (i.e. are mono-dimensional) and their central nervous system consists of a Perceptron that allows to establish a specific relationship between a sensory input and a motor output.

The main input we chose to stimulate such perceptrons is a given portion (i.e. of parameterized dimensions) of a sound wave, extrapolated run-time from any file or direct sound running on the sound board. To such input, that numerically speaking is the main source of information to all the Perceptrons, the user could easily add a different input, such as the organism's distance from one or more landmarks. Such landmarks - which are positioned into the world either randomly or logically (i.e. at user choice) - create a referring point(s) around which the organism can navigate or maybe, as in our specific case, "dance".

To these, that under all circumstances can be considered real inputs, we added different simulation options and tunable algorithmic parameters (see Figure 1) that enrich the behavioral scenario and, consequently, the graphic one, transforming each single instance of the simulation.

Indeed, as we can easily imagine observing the Control Panel, shown in Figure 1*A*, there are quite many parameters that contribute to define the specificity and evolutionary characteristics of the running simulation. Some of them act on the agent's general behavior - for example the static or dynamics of the landmarks – and others influence the aesthetical result – for example the phenotypic expressions of the agents as the tail or the color can be.

```
GRAPHICS:
A = Drawing of Agents Head (false)
F = Colored Agents' Tails (false)
W = Switches Background/Foreground (true)
C = Agents Control the Screen Cleaning (false)
EVOLUTION, GENES AND FITNESS CRITERIA:
E = Uses Darwinism  (true)
 1 = Selects (<>) no-static (false)
 2 = Selects (<-) borders crashes (false)
 3 = Selects (<-) encounters (false)
 4 = Selects (->) high-dynamic (false)
 5 = Selects (<-) close to center (true)
 G = Usees Color Gene (true)
 S = Usees Agents Step Length Gene (false)
 T = Agents Elite Reproduction (false)
 Z = Freeze the bests genetic codes (false)
REINFORCEMENT LEARNING:
R = Crashes Reinforce Learn (false)
M = Meeting Reinforce Learn (false)
ENVIROMENTAL PARAMETERS:
B = World has Borders (false)
U = Attractors Dynamics (false)
X = Drawing Attractors (false)
SOUNDS:
L = Continuous File/s Loop (true)
O = Looping current File (false)
TOOLS:
H = Shows Help Window (true)
D = Shows InfoData Window (true)
Y = Shows Sound Input (true)
V = Init Video Images Storage Process (false)
I = Captures Current Image
P = Pause/Play Application (false)
ENTER = Save Simulation
SPACE = Load Saved Simulation
Q = Quit the Application
```

```
World Width  : 1280
World Height : 720
Population Size : 80
Plotting Samples : 6
Painting Scale X : 213
Painting Scale Y : 7
Playng File: 00 xluigi.wav
Volume : 6.9897

Generation : 1
Cycle : 360 (runs: 28800)
Audio Cycle : 51745
Percentage played file: 43 %
Borders Crashes : 23

Inputs : 14
Weights: 84
Outputs: 6

Digestion: 20
Using 3 attractors
Average Steps: MAX Steps
Tails : 50
Best Agent Fitness: 0
Average Fitness: 0

Time-lapse: 7 sec
```
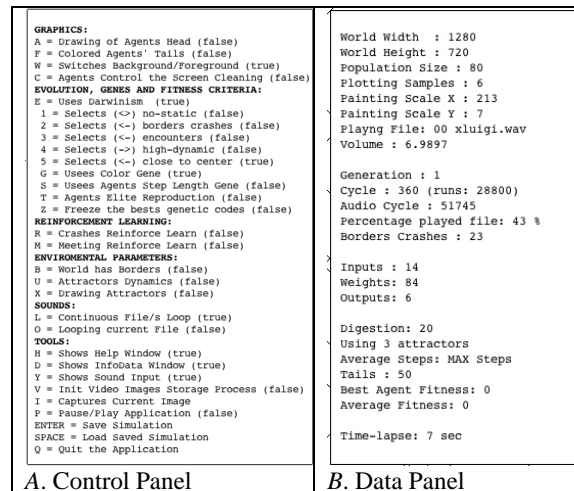
| *A*. Control Panel | *B*. Data Panel |

Fig. 1. The Control and Data Panels

All the variables shown in the Control Panel can be changed run-time, meaning that the user can shape each single simulation, making it more or less unrepeatable.

Indeed, opening up such a set of variables to users' intervention corresponds to acting on the computed fitness gradient - since the fitness is the index by which the adaptation coefficient of an artificial agent is calculated. By modulating such computation we act on the reproductive power of each single organism and, therefore, we can change the effective history of the running simulation. Each simulation can be monitored through a Data Panel – see Figure 1*B* for an example.

Music is at the center of the MAG software, that is why each simulation starts as soon as an audio stream appears (e.g. at the beginning of a song) and produces a *generational transition* as soon as silence interrupts it (e.g. at the end of the song). By *generational transition* we mean that the ordinary GA's process of best individual's fitness selection and reproduction. In our algorithm, the reproduction routine applies both mutations and cross-over. All of that is repeated for each song (i.e. a high significant discontinuity in the amplitude of the streaming audio, or silence) and such process is reiterated up to when the user consider it sufficient, as an aesthetical result. Indeed, as the simulative process goes on, one can observe and more and more well defined behavioral expression of the artificial agents where, most of the times, agents learn to "dance" on the sound wave that goes across themselves. As shown in the Control Panel (Figure 1*A*), besides the inter-generational selective pressure, it is made possible to apply to agent behavior two different forces of Reinforcement Learning[12], since we were seeking for a force able to change the intra-generational behavior of our perceptrons.

The two Reinforcement Learning routines we implemented:

(i) *Crash.* We apply the RL each time an agent crashes against the world borders (i.e. we might reward agents that stand in the center of the "dance-floor");

(ii) *Meeting.* We apply the RL each time an agent crashes against another agent (i.e. we might reward agents that do "not smash other agents feet");

Both RLs are used and seem to work despite of the fact that agents are not always given the possibility of sensing the world contours or the mates' presence. This is also because the two behaviors described above start to be computed in to the Fitness Formula as soon as they are selected as operative.

In synthesis, to summarize the whole simulation flow:

(i) As soon the MAG is started, the users are allowed to choose amongst few basic and starting parameters (e.g. number of landmarks);

(ii) A population of N Perceptrons (according to screen size) is initiated;

(iii) As soon as the sound board is activated, agents population is started;

(iv) Agents, one after the other, receive as input a portion of run-time executed sound wave (i.e. plotting samples, which is itself a variable)

(v) For technical reasons, and since the sound process is always faster than simulative computation, each single agent is fed with a different portion of sound and, therefore, has a specific and unique input;

(vi) Agents calculate their behavioral reaction to sound and, eventually, other factors (e.g. distance to landmarks);

(vii) Agents calculates their potential movement in the Cartesian space and, therefore the fitness of such (RL takes place here, eventually);

(viii) All agents draws their movement;

(ix) The simulators opens up to any user interaction, either regarding functional parameters (i.e. Genetics, Fitness Formula, Reinforcement Learning, Environmental Parameters, Sound Input Choice and Analysis) and non-functional ones (i.e. Graphical Output, Feedback Tools, Run-Time Data Description, Simulation Controls, etc.);

(x) As soon as the sound board is deactivated (i.e. end of files or silence) agents population is submitted to a generational transition, which means selection, reproduction and mutation;

This process is re-iterated at the users' will.

3. Aesthetic Outcome

The simulation leader decides if, and to what extends, to control the graphical outcome of our population of agents, leave that decision to emerge from specific expressions of agents' artificial genes or, since the whole process can be modified run-time, both. For the graphical outcome we intend both the colour and the shapes of the drawings. The MAG genes act in such a way that part of the phenotypic expressions of the

output are controlled by the single agent, while other ones (e.g. the background color) might be controlled by all agents together. For example, a single agent's gene controls the length of its stepping, while in each cycle the algorithm weights up the outputs of the whole population and determines the background color (i.e. when a specific option has been activated by the users). Therefore, the chromatic basis of the agents' "drawings" can either be determined by the user or rely on a genetic mechanism. The resulting aesthetical outcomes perform a quite impressive range of graphical solution and, moreover, is capable of big variations even within one single simulation run (see Figure 2 for few examples).
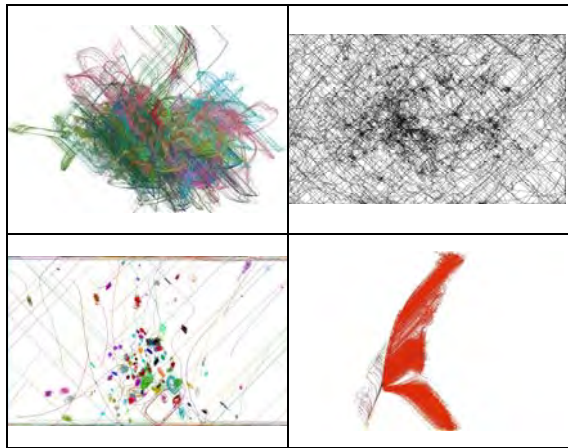
Fig. 2. Some still images from different evolutions of different songs.

Aesthetics is (or can be) modified across the time and changes accordingly to the interactive process enacted the users with genetic components, landmarks, reinforcement learning and the other variables but, overall, it is influenced and transfigurated by the choice of different music flows, that stands for the user decision of the sound input which can be determined at any given moment along the evolutionary paths initiated by the population of artificial agents.

The MAG software has already found its artistic positioning under the form of a preassembled video (Gabriele 2012) whose projection rights have been sold to different galleries and festivals, all over the world. Despite of that, the software itself is under constant technical evolution, and in the early future we will release an advanced version where the agents population will:

(i) Incorporate three layers Neural Networks, instead of the actual Perceptrons (i.e. to reach much more complex and articulated movements);

(ii) From simple input-output organisms agents' brain will be furnished with a neural memory mechanism (i.e. to better follow the bit);

(iii) Each single population will be split into different "races", or different subsets of populations (i.e. to push agents specialization on different frequencies of sound analysis).

4. MusicTiles Interface

The MusicTiles[5] (former MagicTiles[6]) music app, derived from the RoboMusic concept[13] and that can be used by users of different musical competencies - from precipitants to musician to expert composers - to both remix songs or experiment with new tunes.

The MusicTiles app that runs on iPads and iPhones, is made of a small number (i.e. more than 3 less than 10) of virtual music tiles. Each single tile embodies a single instrument and each instrument-tile is capable of expressing seven different musical solution (i.e. loops) or variation of the same one. Every instrument of a MusicTiles song is wrapped inside a single tile and, therefore, we can have a tile for vocals, one for the rhythm guitar, one for the bass, one for the piano, etc.
At the start, the user is presented with all the tile-instruments of a given song inside the tiles board with the tile-instruments being disconnected from each other (see Fig. 3).

Fig. 3. MagicTiles for iPad.

Since a tile is disconnected from other tiles, it is not active and will not play. Once the user drags and

attaches a tile to another (see Fig. 5 left) the two tiles get synchronized and are able to play together.

To actually start the music, the user has to press the play/stop icon, positioned at the bottom-right side of the screen (see Fig. 3). At this point, the music will keep playing and the remaining music tiles can be connected to the former two enriching the music scenario of the song running. In the same way, by dragging the tiles around and rearranging their geometry, the user changes the music sequence and flow.

Fig. 4. Left: Connecting two tiles. Right: 2-sides connection example.

Every tile has 7 music loops, one for each side (north side, east side, south side and west side) plus 3 extra music loops which will play when a tile is connected on, respectively, two, three or four sides. In this way, as in Fig. 5 left, the Drum tile will play the East Side loop and the Guitar tile will play the West Side loop, and as in Fig. 5 right, the Drum tile will play the *2 Sides* loop, the Guitar tile will play the West Side loop and the Choir tile will play the North Side loop.

Likewise, as in Fig. 5 the Drum tile will play *3 Sides* loop, the Choir tile the North Side loop, the Guitar tile will play the West Side loop, and the Bass tile the East one.
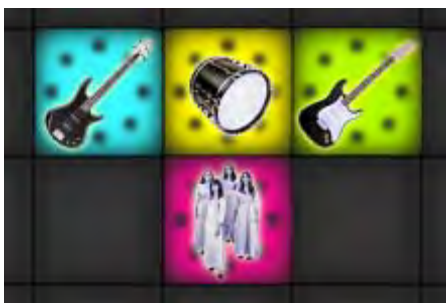

Fig. 5. A *3 Sides* connection example.

Growing the number of instruments/tiles impressively increase the number of possible combinations and out of these simple rules, one obtains an extraordinary number of musical possibilities.

5. MAG and MusicTiles App Integration

Besides all of that, we are at the beginning of a process of integration of MAG software with MusicTiles music app to explore the concept of run-time composition of both music and graphical expression. The idea is to allow any user to create musical expression through the MusicTiles (i.e. which can happen run-time as in a performance run-time) and, when connecting the MusicTiles app to the MAG software, providing he/she the possibility to melt both musical expression and graphical expression in parallel therefore creating an audio-video performance that is always unique.

6. Discussion and Conclusion

In this paper, we presented both a new application – MAG, which aims to create graphical expression by following musical paths - and a previous one - MusicTiles, which aims at exploring a new approach to music for run-time composition and/or remixing. Through such applications any user could create and perform both Graphics and Music, on one side, by manipulating simulation parameters and, on the other, by assembling a number of interactive virtual tiles on the screen. After our past efforts in building physical interfaces, we decided to conceive a cross-media platform to head to cross-modal and cross-sensory artistic result. Further evolution of such an approach might lead to a combination of both virtual interface (i.e. MAG and MusicTiles) with a physical interface. In short, it seems that we instantiated a research process in which we are trying to combine different input and output methodologies, systems and tools that might lead to a broader vision of software and robotic systems with an articulated, fluid and bidirectional flow between the physical and virtual environment. Indeed, we believe that such a fluid and multifaceted representation of a single activity may widely enhance the user immersion into a one reality that combines multi-sensory activation on a physical and a virtual level.

Acknowledgements

References

1. L. Pagliarini, *Algoritmi Genetici Musicali (MAG).* In Rivista di Psicologia dell'Arte, Nuova Serie, Anno XXXIII, n° 23, (December 2012) pp. 103-108.
2. D. E. Goldberg, *Genetic Algorithm in search, optimization and machine learning* (New York, Addison-Wesley, 1998).
3. F. Rosenblatt, The Perceptron--a perceiving and recognizing automaton. Report 85-460-1 (Cornell Aeronautical Laboratory, 1957).
4. D. E. Rumelhart, J. L. McClelland, *Parallel Distributed Processing. Explorations in the Microstructure of Cognition* (MIT Press, Cambridge, MA. 1986).
5. Peter Gabriel's Latest App is So 2.0, *Fast Company*, 7 (January 2013): http://www.fastcompany.com/3004537/peter-gabriels-latest-app-musictiles-so-20
6. L. Pagliarini and H.H. Lund, MagicTiles. ALife for real and virtual Robot Music, in *Proceedings of the 17th International Symposium on Artificial Life and Robotics* (AROB'12, ISAROB, Japan, 19-21 Jan, 2012).
7. C. G. Langton, *Artificial Life*. In C. G. Langton, editor. *Artificial Life*, Volume VI of *SFI Studies in the Sciences of Complexity* (Addison-Wesley, Redwood City, CA, 1989) pp. 1-47.
8. A. Spina, L. Pagliarini, Globalization: tra arte scienza e società. In *Rivista di Psicologia dell'Arte* Anno XII, n.12, (Roma, Dec. 2001) 87-92.
9. L. Pagliarini, Una Tela Viva. Rivista di Psicologia dell'Arte, Nuova Serie Anno XXIII, n°13, (December 2002) 33-36
10. M. Mitchell, *An introduction to genetic algorithms* (MIT Press, 1997).
11. A. Gabriele, Ogni Dove Edizioni, 2012 http://www.ognidove.eu/ode2/.
12. R. S. Sutton, Temporal Credit Assignment in Reinforcement Learning, Ph.D. thesis (University of Massachusetts, Amherst, MA, 1984)
13. H. H. Lund, and M. Ottesen, RoboMusic – A Behavior-Based Approach, *Artificial Life and Robotics Journal, 12: 1-2* (2008) 18-23.