Evolving L-Systems with Musical Notes

Ana Rodrigues(✉), Ernesto Costa, Amílcar Cardoso, Penousal Machado, and Tiago Cruz

CISUC, Deparment of Informatics Engineering, University of Coimbra, Coimbra, Portugal
{anatr,ernesto,amilcar,machado,tjcruz}@dei.uc.pt

Abstract. Over the years researchers have been interested in devising computational approaches for music and image generation. Some of the approaches rely on generative rewriting systems like L-systems. More recently, some authors questioned the interplay of music and images, that is, how we can use one type to drive the other. In this paper we present a new method for the algorithmic generations of images that are the result of a visual interpretation of an L-system. The main novelty of our approach is based on the fact that the L-system itself is the result of an evolutionary process guided by musical elements. Musical notes are decomposed into elements – pitch, duration and volume in the current implementation – and each of them is mapped into corresponding parameters of the L-system – currently line length, width, color and turning angle. We describe the architecture of our system, based on a multi-agent simulation environment, and show the results of some experiments that provide support to our approach.

Keywords: Evolutionary environment · Generative music · Interactive genetic algorithms · L-systems · Sound visualization

1 Introduction

It is a truism to say that we live in a world of increasing complexity. This is not because the natural world (physical, biological) has changed, but rather because our comprehension of that same world is deeper. On the other hand, as human beings, our artificial constructions and expressions, be them economic, social, cultural or artistic, are also becoming more complex. With the appearance of the computers, the pace of complexification of our world is increasing, and we face today new fascinating challenges. Computers also gave us a new kind of tool for apprehending and harnessing our world (either natural or artificial) through the lens of computational models and simulations. In particular, it is possible to use the computer as an instrument to interactively create, explore and share new constructs and the ideas behind them.

Music is a complex art, universally appreciated, whose study has been an object of interest over the years. Since the ancient days, humans have developed a natural tendency to translate non-visual objects, like music, into visual codes,

C. Johnson et al. (Eds.): EvoMUSART 2016, LNCS 9596, pp. 186–201, 2016.
DOI: 10.1007/978-3-319-31008-4_13

i.e., images, as a way to better understand those artistic creations. More recently, some authors have tried to translate images into sounds using a wide variety of techniques. Although there is still a lot of work to be done in the field of cross-modal relationships between sound and image [1–7], the achievements made so far in the devising of audio-visual mappings show that this approach may contribute to the understanding of music.

In this work we are interested in using computers to explore the links between visual and musical expressions. For that purpose we develop an evolutionary audiovisual environment that engages the user in an exploratory process of discovery.

Many researchers have been interested in devising computational approaches for music and image generation. Some of these approaches rely on generative rewriting systems like L-systems. More recently, some authors questioned the interplay of music and images, that is, how can we use one type to drive the other. Although we can find many examples of L-systems used to algorithmic music generation [1,5,7–9], it is not so common to find generation of L-systems with music. Even less common is to find attempts to have it working in both ways.

We present a new method for the algorithmic generation of images that are the result of a standard visual interpretation of an L-system.

A novel aspect of our approach is the fact that the L-system itself is the result of an evolutionary process guided by musical elements. Musical notes are decomposed into elements – pitch, duration and volume in the current implementation – and each of them is mapped into corresponding parameters of the L-system – currently line length, width, color and turning angle.

The evolution of the visual expressions and music sequences occurs in a multi-agent system scenario, where the L-systems are the agents inhabiting a world populated with MIDI (Musical Instrument Digital Interface) musical notes, which are resources that these agents seek to absorb. The sequence of notes collected by the agent, while walking randomly in the environment, constitutes a melody that is visually expressed based on the current interpretation of the agent's L-system. We use an Evolutionary Algorithm (EA) to evolve the sequence of notes and, as a consequence, the corresponding L-system. The EA is interactive, so the user is responsible for assigning a fitness value to the melodies [10].

The visual expression provided by the L-system aims to offer visual clues of specific musical characteristics of the sequence, to facilitate comparisons between individuals. We rely on tools such as Max/Msp to interpret the melodies generated and Processing to build the mechanisms behind the interactive tool and the respective visual representations.

Even if the main focus is to investigate the L-systems growth with musical notes, we also try to balance art and technology in a meaningful way. More specifically, we explore ways of modeling the growth and development of visual constructs with music, as well as musical content selection based only on the visualization of the constructs.

Moreover, we are also interested in understanding in which ways this visual representation of music will allow the user to associate certain kinds of visual patterns to specific characteristics of the corresponding music (e.g., its pleasantness). The experiments made and the results achieved so far provide support to our approach.

The remainder of the paper is organized as follows. In Sect. 2, we present some background concepts needed to understand our proposal. In Sect. 3, we describe some work related with the problem of music and image relationship. In Sect. 4 we specify the system's architecture and development, which includes describing the audiovisual mappings and the evolutionary algorithm we use. We continue in Sect. 5 with the presentation of the results. Lastly, in Sect. 6, we present our main conclusions, achieved goals and future improvements.

2 Background

In this section we briefly refer to the main concepts involved in the three basic elements of our approach: L-systems, evolutionary algorithms and music.

2.1 L-Systems

Lindenmayer Systems, or L-systems, are parallel rewriting systems operating on strings of symbols, first proposed by Aristid Lindenmayer to study the development processes that occur in multicellular organisms like plants [6]. Formally, an L-system is a tuple $G = (V, \omega, P)$, where V is a non-empty set of symbols, ω is a special sequence of symbols of V called axiom, and P is a set of productions, also called rewrite rules, of the form $LHS \rightarrow RHS$. LHS is a non-empty sequence of symbols of V and the RHS a sequence of symbols of V. An example of L-systems is:

$$G = (\{F, [,], +\}, F, \{F \rightarrow F[F][+F]\})$$

As a generative system, a L-system works by, starting with the axiom, iteratively rewriting in parallel all symbols that appear in a string using the production rules. Using the previous example of L-system we obtain the following rewritings:

$$F \xrightarrow{1} F[F][+F] \xrightarrow{2} F[F][+F][F[F][+F]][+F[F][+F]] \xrightarrow{3} \ldots$$

After **n** rewritings we say we obtain a string of level **n**. The axiom is considered the string of level 0.

In order to be useful as a model, the symbols that appear in the string must be interpreted as elements of a certain structure. A classical interpretation, that we will use here, is the turtle interpretation, first proposed by Prusinkiewicz [11]. The symbols of a string are commands for a turtle that is moving in a 2D world. The state of the turtle is defined by two attributes: position (x, y) and orientation α. The commands change these attributes, eventually with side-effects (e.g., drawing a line). In Table 1 we show this interpretation.

Table 1. Turtle interpretation of an L-system

Symbol	Interpretation
F	Go forward and draw a line
f	Go forward without drawing
+	Turn counter-clockwise
-	Turn counter-clockwise
[	Push turtle's state
]	Pop turtle's state

Fig. 1. Example of a visual interpretation of the string at level 5.

Using this interpretation the visual expression of the string of level 5, of the given L-system is presented in Fig. 1. Notice that the user has to define two parameters: the step size of the forward movement and the turn angle.

Over the years L-systems were extended and their domains of application, both theoretical and practical, was broadened [1,5,7–9]. Some L-systems are context-free (the LHS of each production has at most one symbol), while others are context-sensitive (the production have the form $xAy \rightarrow xzy$, with $A \in V$ and $x, y, z \in V^+$. Some L-systems are said to be determinist (at most one production rule with the same left hand side) while others are stochastic. Some L-systems are linear while others, like the one above, are bracketed. The latter are used to generate tree-like structures. Yet some other L-systems are said to be parametric, i.e., when a parameter is attached to each symbol in a production rule whose application depends on the value of the parameter. Finally, some L-systems are called open, when they communicate with the environment and may change as a consequence of that communication [4].

2.2 Evolutionary Algorithms

Evolutionary Algorithms (EA) are stochastic search procedures inspired by the principle of natural selection and in genetics, that have been successfully applied in problems of optimization, design and learning [12]. They work by iteratively improving a set of candidate solutions, called individuals, each one initially generated at random positions. At each evolving step, or generation, a subset of promising solutions, called parents, is selected according to a fitness function for reproduction with stochastic variation operators, like mutation and crossover.

Mutation involves stochastic modifications of some components of one individual, while crossover creates new individuals by recombining two or more. The result of these manipulations is a new subset of candidate solutions, called offspring. From the parents and the offspring we select a new set of promising solutions, the survivors. The process is repeated until a certain termination criterion is met (e.g., a fixed number of generations). Usually the algorithm does not manipulate directly the solutions but, instead, a representation of those solutions, called the genotype. To determine the quality of the genotypes they must be mapped into a form that is amenable for the assessment by the fitness function, called phenotype.

2.3 Musical Concepts

Notes, or pitched sounds, are the basic elements of most music. Three of the most important features that characterise them are: pitch, duration and volume. *Pitch* is a perceptual property of sound that determines its highness or lowness. *Duration* refers to how long or short a musical note is. *Volume* relates to the loudness or intensity of a tone.

Most of the western music is *tonal*, i.e., melody and harmony are organised under a prominent tonal center, the tonality, which is the root of a major or minor scale.

When a central tone is not present in a music, it is said to be *atonal*.

Even though the concepts of harmony and progression do not apply in an atonal context, the quality of the sounding of two or more tones usually strongly depends on formal and harmonical musical contexts in which it occurs. This quality is usually classified as *consonance*. Consonance is a context-dependent concept that refers to two or more simultaneous sounds combined in a pleasant/agreeable unity of sound. On the other side, dissonance describes tension in sound, as if sounds or pitches did not blend together, and remain separate auditive entities [13]. Anyway, consonance is a relative concept: there are several levels of consonance/dissonance. Although consonance refers to simultaneous sounds, it may also be applied to two successive sounds due to the memorial retention of the first sound while the second is heard.

The difference between two pitches is called *interval*. Intervals may be harmonic (two simultaneous tones) or melodic (two successive tones). In tonal music theory, intervals are classified as perfect consonants (perfect unison and perfect 4^{th}, 5^{th} and 8^{th} intervals), imperfect consonants (major and minor 3^{rd} and 6^{th}) and dissonants (all the others) [13].

Our system produces melodic sequences of notes in the C Major Scale. However, we do not constrain the system to produce tonal sequences or even consonant pairs of sounds.

3 Related Work

This section presents some of the most relevant references to the development of our work.

We can see in the following examples, that science and music have a long common history of mutual interactions. As Guéret et al. [14] say, music can lead to new perceptions in scientific realities and science can improve music performance, composition and understanding.

Music has a huge structural diversity and complexity. Algorithms that resemble and simulate natural phenomena are rich in geometric and dynamic properties. Computer models of L-systems are an example of such algorithms, and therefore can be helpful in automatic music composition [8,15].

Early work on L-systems and Music includes Keller and Měch et al. [4,16]. Many authors have described techniques to extract musical scores from Strings produced by L-systems [9,11,16]. One of the first works on the field of music generation and L-systems belongs to Prusinkiewicz [1]. He described a technique to extract music from a graphical interpretation of an L-system string. The length of each note was interpreted as the length of the branch, and the note pitch was interpreted as the vertical coordinate of the note. Graphical and musical interpretation were synchronized [11]. A survey on the evolution of L-systems in artistic domains includes McCormack work [2].

This mapping of sound parameters into something that usually is not considered audible data is called sonification. Many had an interest in exploring sonification in a way of understand scientific data the same way visualization is able to do [7].

There have been some efforts to create evolutionary systems for the automatic generation of music. A remarkable example is the EA proposed by Biles [17]. In this work he uses a GA to produce jazz solos over a given chord progression. In recent years several new approaches have emerged based not only on GA, but on other techniques such as Ant Colony [3,14,17–19].

4 System's Architecture

To explore the interplay between music and visual expressions by L-systems we construct a 2D world. In this section we describe this world, the entities that live and interact in it and evolve under the user guidance.

4.1 General Overview

In our world there are two types of entities: agents and notes. Notes have immutable attributes: their position and value. They do not die or evolve over time. Agents are entities with two components: (1) an L-system that drives its visual expression and (2) a sequence of notes that define the L-system's parameters at each level of rewriting (see Fig. 2). Agents move in the world by random walk, looking for notes that they copy internally and append to their sequence. These notes change over time through an Interactive Genetic Algorithm (IGA) [10].

The environment begins with a non-evolved individual (level 0) that wanders in the environment catching notes. In this case, its growth is determined by the musical notes that it catches, creating a sequence of notes. The first note caught makes it evolve to level 1, the second note to level 2 and so forth Fig. 3.[1]

[1] It is possible to catch some note that has been previously caught.

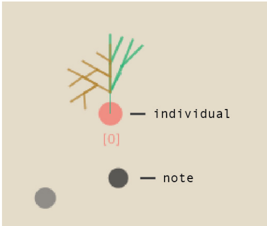

Fig. 2. Environment's elements. (Best viewed in color, see color figure online)

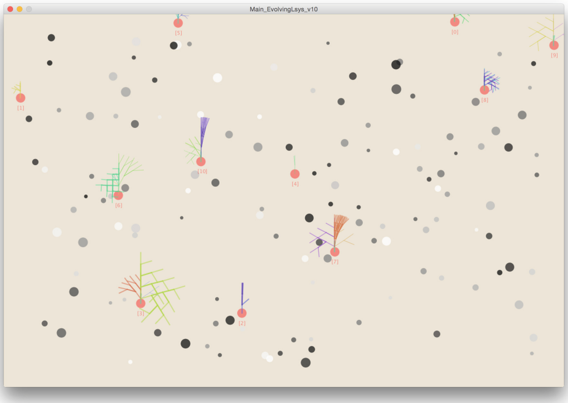

Fig. 3. Environment overview. (Best viewed in color, see color figure online)

A new individual can be generated from the current one through two different processes: mutation and crossover. A more detailed description of this can be found in Sect. 4.3.

When the user selects an individual there are two possible operations at the level of the interface: (i) Listen to the musical sequence of the individual in question note by note; (ii) Listen to the musical sequence of the individual in question in a simultaneous way (all notes together). Two other possibilities exist at the evolutionary level: (iii) Apply a mutation; (iv) Choose two different parents, and apply a crossover. (See Fig. 4)

4.2 Audiovisual Interpretation and Mappings

To have a qualitative criteria for auditory and visual mappings, we established a guide that formalises the relationship between these two domains:

1. Every auditory category admitted should have assigned a corresponding visual effect. We accomplish this by visualising the following parameters: (i) pitch, (ii) duration, (iii) volume, and (iv) notes interval.

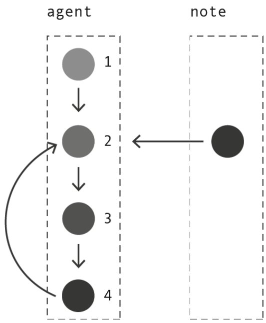

Fig. 4. System's architecture overview. (1) Agent is placed randomly in the environment; (2) Agent searches for a note and catches it; (3) Visual expression with an L-system of that note is made; (4) A GA can be applied.

2. As the work has a selection method based on an IGA, simplicity shall be maximised. When we have a high degree of complexity, the user often loses the ability to maintain sufficient visual control and perception over the results [20].

We divide the visual representation of music into two distinct parts: (i) the visual representation of the notes spread across the environment that individuals may catch, and (ii) the notes that the L-systems effectively catch. The first representation is static, because they are always a direct representation of the notes' parameters. The second one is dynamic, in the sense that different shapes are formed as new notes are caught.

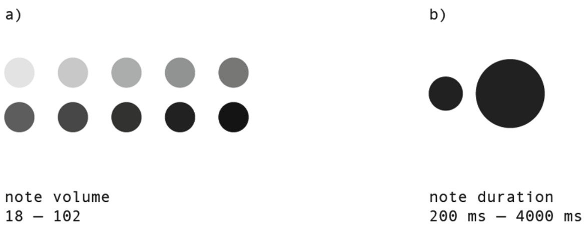

Fig. 5. Graphic interpretation of the notes spread across the environment: (a) The note volume is represented by color saturation. The higher the note volume is, the more intense is the object's color. (b) Size represents the note duration. The higher the duration is, the bigger is the object's size.

The static notes in the environment are circles in levels of grey, with saturation representing volume and size representing note duration (see Fig. 5). Pitch is not represented. Position in the environment is random.

For the L-system visual representation, authors who have made similar attempts have chosen to associate the L-system's vertical coordinates to note pitch, and the distance between branches to note duration. However, we are

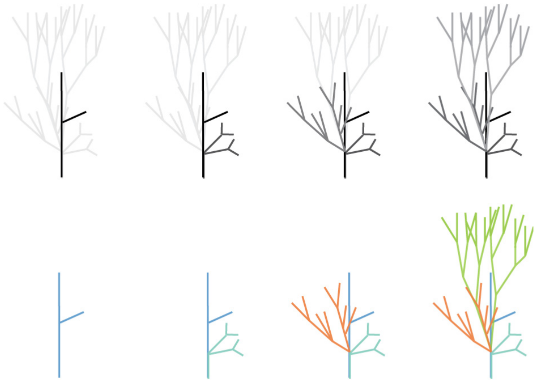

Fig. 6. Example of the L-systems growth process. (Best viewed in color) (Color figure online)

interested in comparing musical sequences in a qualitative way, considering the notion of consonance instead of absolute or relative pitch. Therefore, we had to adopt our own mappings between music and image to use in the L-system.

The L-systems presented in this work grow with the musical notes collected (see Fig. 6). Each note affects the L-system visual parameters at each level: (i) branch angle, (ii) branch length, (iii) branch weight, and (iv) color. Note duration maps into branch length, note volume into branch stroke (see Fig. 7), and consonance into branch color.

Every time a note is caught its pitch is compared to the previous note. From there, we calculate its consonance or dissonance. To the first note caught by an individual (level 1) is attributed a pitch color corresponding to its pitch height (see Fig. 8). If the sequence of notes is consonant then it is applied a tonality based on the color of the previous note caught. In case it is dissonant, a random color tonality is applied. Looking at Fig. 8 we can realize that consonance can be distinguished by its subtle change of color. On the contrary, a dissonant melody will produce changes of color and color tonalities with bigger steps.

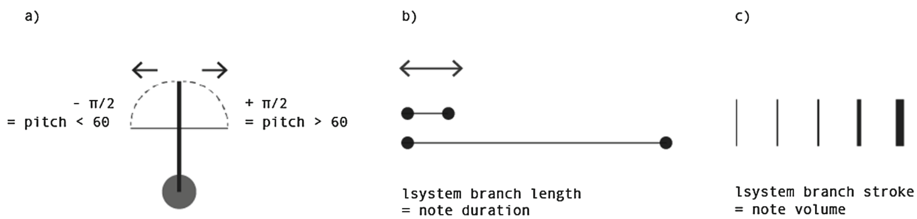

Fig. 7. Mapping process of the note's characteristics (a) pitch, (b) duration, (c) volume, into L-system graphic representation.

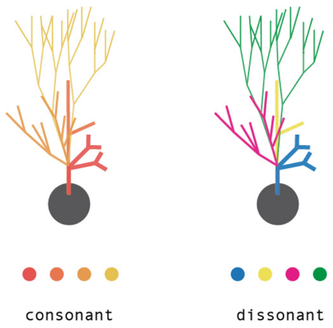

Fig. 8. Consonant and dissonant visual representations. (Best viewed in color) (Color figure online)

Furthermore, since there is no term of comparison to other notes when the L-system catches its first note, the color assigned corresponds to the pitch (see Fig. 9) of the caught tone. To the other notes color is assigned accordingly to the classification of consonant or dissonant depending on the note that has been previously caught.

Our environment is stochastic in the sense that agents walk randomly through the system. Furthermore the own process of note's modification implies a chance of being chose or not a note to apply these modifications. Stochastic systems can have different strings derived from the same string at any step, and they may produce a high diversity of sequences [7].

The number of possible outcomes for both sound and visual combinations is dependent on the number of possible values for the notes[2] pitch (127), duration (3800) and volume (82), in addition to the number of notes that we set up for each individual (4). Although we set the latter value to 4 in our experiments, it is not a limitation of our system.

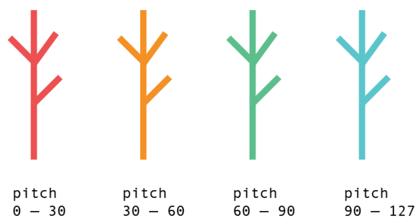

Fig. 9. Color association for pitch. Warm colors correspond to lower pitches, and cold colors to higher pitches. (Best viewed in color) (Color figure online)

[2] Each note parameters were interpreted as a MIDI note: (i) pitch range: 0 – 127 (ii) volume range: 20 – 102 (iii) duration range 200 – 4000 ms (iv) timbre – piano (0).

4.3 The Evolutionary Algorithm

Controlled evolution in the environment was a solution that we adopted to allow the creation of a large variety of complex entities that remain user directed and simple to interact with. Most organisms evolve by means of two primary processes: natural selection and sexual reproduction. The first determined which members of the population would survive to reproduce, and the second ensured mixing and recombination [21].

An IGA is used to assign the quality of a given candidate solution. The solutions favoured by the user have a better chance of prevailing in the gene pool, since they are able to reproduce in higher amount.

The musical sequence caught by an individual consists in its genotype, and its phenotype is composed of sound and image, i.e., L-system (see Fig. 10). The order of the genotype is defined by the order in which notes are caught.

Selection: Computationally, the measurement of the quality of a chromosome is achieved through a fitness function. In this work, this process is done interactively and is provided by a human observer. The use of an IGA, based in this case on the user visual and auditory perceptions, allows the user to direct evolutions in preferred directions. With this approach, the user gives real-time feedback. The expected output is a computer program that evolves in ways that resemble natural selection.

Offspring is born based on selected individuals, and to it a mutation process is applied. This replication of the preferred individual feeds up the probabilities of growing up more individuals that the user enjoys.

Reproduction: We apply both crossover and mutation in our system for evolution to progress with diversity. While crossover allows a global search on the

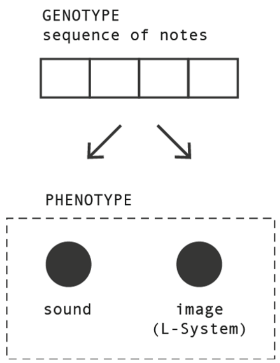

Fig. 10. The genotype (sequence of notes) is translated into sound and image (phenotype). Although sound has a direct mapping to MIDI notes, the image is interpreted with an L-system.

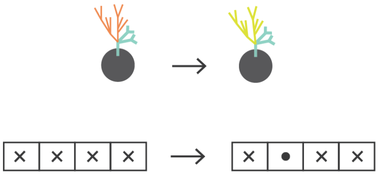

Fig. 11. Mutation example. One note of the original sequence of notes was chosen to be modified.

solutions space, mutation allows a local search. Each element has a chance (probability) of being mutated. Implementing these algorithms, we intend the evolution of L-systems with musical material through genetic transmission.

Offspring resulting from mutations or crossover are incrementally inserted into the current population and original chromosomes are kept. According to Sims [20], "Mutating and mating parameter sets allow a user to explore and combine samples in a given parameter space".

<u>Mutation</u>: Mutation takes a chromosome of an individual and randomly changes part of it [19]. It allows to change pitch, duration and volume in this case. Our mutation mechanism receives two parameters: the sequence of notes that will be modified and the probability of mutation of each note in the genotype. The probability of mutation will decide which note(s) collected by that individual will be modified. Each element in the sequence of notes caught by the individual has equal chance of being chosen (uniform probability). To each chosen note for mutation, the following parameters are changed randomly: pitch, duration and volume (see Fig. 11).

<u>Crossover</u>: Crossover allows the exchange of information between two or more chromosomes in the population [19]. This mixing allows creatures to evolve much more rapidly than they would if each offspring simply contained a copy of the genes of a single parent, modified occasionally by mutation [21]. In this case, it is possible to select only two parents which will give birth to two children.

We start by selecting random cut points on each parent, and then we give birth to the children based on these cut points (see Fig. 12). The resulting size of each child is variable since the cut points made in the parents are random.

4.4 Auxiliary Tools

To interpret sound we use Max/Msp. It is a graphical environment for creating computer music and multimedia works and uses a paradigm of graphical modules and connections. It reveals to be very helpful in sound interpretation and manipulation. For the grammatical construction and visual interpretation of L-systems we did rely on Processing [22]. Processing is a visual programming tool, suitable for designers and computer artists.

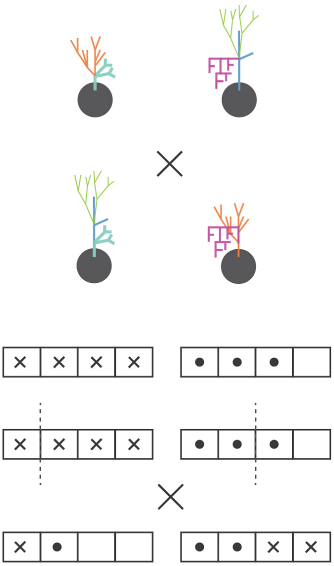

Fig. 12. 1 point crossover example. Two parents are crossed and give birth to two different children.

5 Experimental Results and Discussion

Music can be a very complex thing itself. When we add more complexity to it by using GAs and graphical interpretations of L-systems, if we are not careful, the perception and interaction of the system can easily get out of control. Given the experimental nature in this work, many of our decisions relied on simple concepts so that a full understanding of the system behavior would be possible.

According to Lourenço et al. [1], L-systems wouldn't be a perfect fit for this case because if the rendering techniques are too simple the resulting melody will probably end up with the same motif over and over again. Our solution to increase variability was to implement a generative solution and use some operators from GAs.

It is in fact far from trivial to conciliate both musical and pleasant aesthetic results with L-systems due to the small level of control of the structure. We have tried to solve this problem by providing the user the chance to interactively choose the survival chance of individuals. Although this system has been mostly guided through user interaction, we must question ourselves if it is possible to reach the same quality of results without user guidance.

Since all the parameters present on each L-system were translated into some kind of mapping, it had a direct impact on their developmental process. The resulting individuals revealed to have a lot of visual diversity and express well what we listen to as pleasant or not. Even though we work with simple musical inputs, a big variety of images and melodies (audiovisual experience) was produced as well (see Fig. 13).

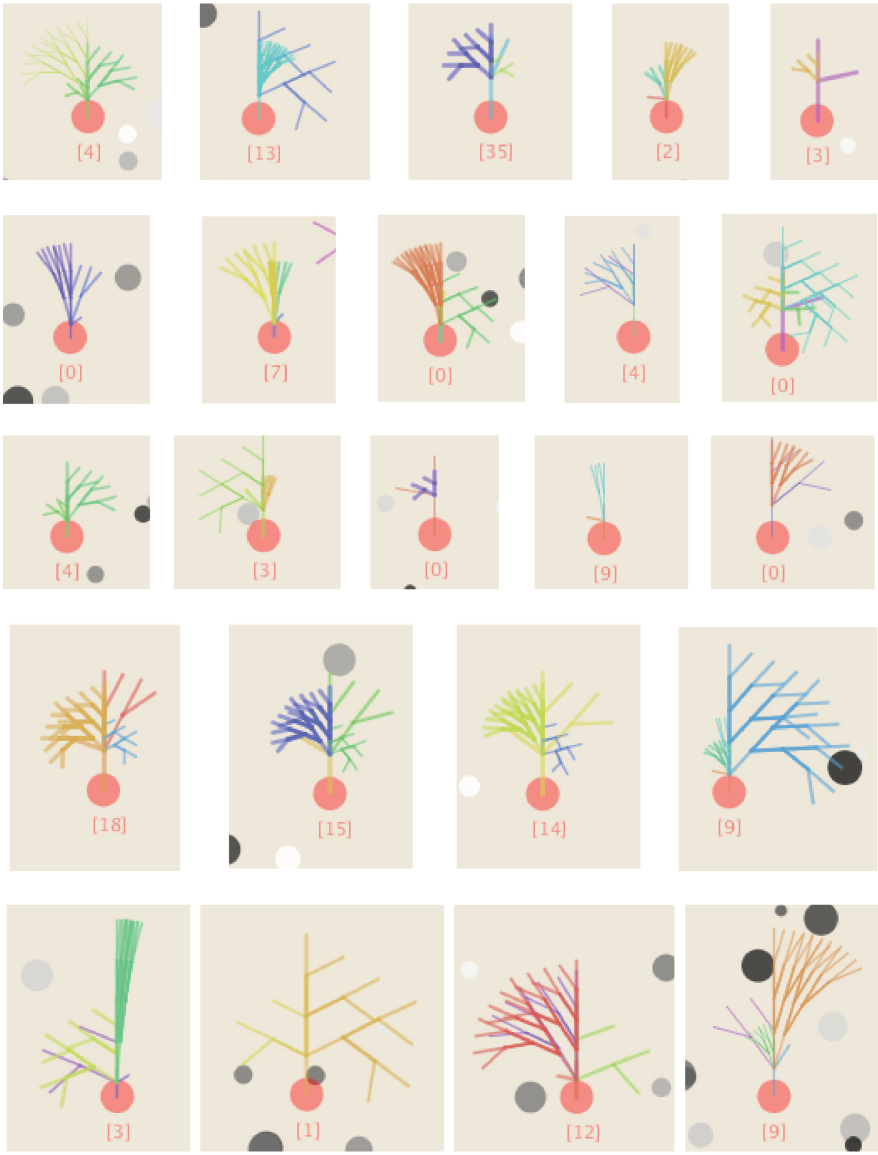

Fig. 13. Example of multiple individuals generated by the system. (Best viewed in color) (Color figure online)

In sum, this audiovisual environment provides the user with a visual representation to a sequence of notes and visual pattern association to the musical contents which can be identified as pleasant or not pleasant. This also means that the user does not have to listen to every individual present in the environment to understand its musical relevance.

A demonstration video can be found at the following link: https://goo.gl/mrbhYa.

6 Conclusion and Further Work

The key idea that makes our approach different from others studies is the concern of mapping sound into image and image into sound. More specifically, our L-systems develop and grow according to the musical notes that were collected by them. At the same time, visual patterns aim to reflect musical melodies built in this process.

For the system evolution and subjective evaluation we have implemented a GA inspired in EC. Stronger individuals had higher probability to survive and reproduce while the weaker did disappear from the environment much faster. The use of an IGA allowed the user to work interactively and in novel ways, meaning that he/she would not be able to reach some results if the implemented computer generative system did not exist. Overall, the system hereby presented is an audiovisual environment that offers a rich and enticing user experience. This provides the user a clear and intuitive visual experience, which is something that we need to have into account since it is a system that is guided by the user.

In future work we would like to make an attempt implementing an ant colony behaviour for notes collection in the environment. It would also be important to investigate a more sophisticated process of music composition, including some rules of harmonisation and chord progression as well as the possibility to introduce more than one timbre in the system. Departing from a tonal system we could have then a set of musical rules that could lead to a fitness evaluation with more values. We have interest as well in exploring these audiovisual mappings at a perceptual level, i.e., using emotions provoked by music as a basis to guide the visual representations. Other future explorations could include L-system with a major diversity of expression or even the use of other biological organisms.

Acknowledgments. This research is partially funded by the project ConCreTe. Project ConCreTe acknowledges financial support of the Future and Emerging Technologies (FET) programme with the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

References

1. Lourenço, B.F., Ralha, J.C., Brandao, M.C.: L-systems, scores, and evolutionary techniques. In: Proceedings of the SMC 2009–6th Sound and Music Computing Conference, pp. 113–118 (2009)
2. McCormack, J.: Aesthetic evolution of L-systems revisited. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2004. LNCS, vol. 3005, pp. 477–488. Springer, Heidelberg (2004)

3. Moroni, A., Manzolli, J., Von Zuben, F., Gudwin, R.: Vox populi: an interactive evolutionary system for algorithmic music composition. Leonardo Music J. **10**, 49–54 (2000)
4. Měch, R., Prusinkiewicz, P.: Visual models of plants interacting with their environment. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH 1996, PP. 397–410. ACM, New York (1996)
5. Pestana, P.: Lindenmayer systems and the harmony of fractals. Chaotic Model. Simul. **1**(1), 91–99 (2012)
6. Prusinkiewicz, P., Lindenmayer, A.: The algorithmic beauty of plants. Springer, New York (1990)
7. Soddell, F., Soddell, J.: Microbes and music. In: Mizoguchi, R., Slaney, J.K. (eds.) PRICAI 2000. LNCS, vol. 1886. Springer, Heidelberg (2000)
8. Kaliakatsos-Papakostas, M.A., Floros, A., Vrahatis, M.N.: Intelligent generation of rhythmic sequences using finite l-systems. In: Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), pp. 424–427. IEEE (2012)
9. Nelson, G.L.: Real time transformation of musical material with fractal algorithms. Comput. Math. Appl. **32**(1), 109–116 (1996)
10. Sims, K.: Interactive evolution of dynamical systems. In: Toward a practice of autonomous systems, Proceedings of the First European Conference on Artificial Life, pp. 171–178 (1992)
11. Prusinkiewicz, P.: Score Generation with L-Systems. MPublishing, University of Michigan Library, Ann Arbor (1986)
12. Eiben, A., Smith, J.E.: Introduction to Evolutionary Computation. Natural Computing Series, 2nd edn. Springer, Heidelberg (2015)
13. van Dillen, O.: Consonance and dissonance (2014). http://www.oscarvandillen.com/outline_of_basic_music_theory/consonance_and_dissonance/. Accessed 01 November 2015
14. Guéret, C., Monmarché, N., Slimane, M.: Ants can play music. In: Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T. (eds.) ANTS 2004. LNCS, vol. 3172, pp. 310–317. Springer, Heidelberg (2004)
15. Manousakis, S.: Musical l-systems. Koninklijk Conservatorium. (master thesis), The Hague (2006)
16. Keller, R.M., Morrison, D.R.: A grammatical approach to automatic improvisation. In: Proceedings, Fourth Sound and Music Conference, Lefkada, Greece, July 2007, Mostof the soloists at Birdland had to wait for Parker's next record in order to and out what to play next. What will they do now (2007)
17. Biles, J.: Genjam: A genetic algorithm for generating jazz solos. In: Proceedings of the International Computer Music Conference, International Computer Music Association, pp. 131–131 (1994)
18. Todd, P., Werner, G.: Frankensteinian methods for evolutionary music composition. In: Griffith, N., Todd, P.M. (eds.) Musical Networks: Parallel Distributed Perception and Performance, pp. 313–339. MIT Press/Bradford Books, Cambridge (1999)
19. Wiggins, G., Papadopoulos, G., Phon-Amnuaisuk, S., Tuson, A.: Evolutionary methods for musical composition. Dai Research Paper (1998)
20. Sims, K.: Artificial evolution for computer graphics. Comput. Graphics **25**(4), 319–328 (1991)
21. Holland, J.H.: Genetic algorithms. Sci. Am. **267**(1), 66–72 (1992)
22. Shiffman, D.: Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction. Morgan Kaufmann, Amsterdam (2009)