PLAYING MUSIC, PLAYING WITH MUSIC
A PROPOSAL FOR MUSIC CODING IN PRIMARY SCHOOL

Adriano Baratè[1], Luca Andrea Ludovico[1], Giuseppina Rita Mangione[2] and Alessia Rosa[3]

[1]*LIM - Laboratorio di Informatica Musicale, Dipartimento di Informatica, Università degli Studi di Milano*
Via Comelico, 39/41 – 20135 Milano (MI) – ITALY
[2]*INDIRE - Istituto Nazionale di Documentazione, Innovazione e Ricerca Educativa, Sede territoriale Sud*
Via G. Melisurgo, 4 - 80133 Napoli
[3]*INDIRE - Istituto Nazionale di Documentazione, Innovazione e Ricerca Educativa, Sede territoriale Nord*
Corso Vittorio Emanuele II, 70 - 10121 Torino

ABSTRACT

In this work we will introduce the concept of *music coding*, namely a new discipline that employs basic music activities and simplified languages to teach the computational way of thinking to musically-untrained children who attend the primary school. In this context, music represents both a mean and a goal: in fact, from one side coding activities are based on music processes that are able to unveil algorithmic thinking; from the other side, such processes may stimulate creativity and collaborative learning, and their audio feedback is immediately perceivable. Consequently, music can represent a valid learning tool as well as an addictive reinforcement technique to approach coding. After describing a new formalism to graphically encode music operators, a Web prototype specifically designed for music coding will be presented and discussed.

KEYWORDS

Music, coding, web, application, primary school.

1. INTRODUCTION

Media and technology are nowadays characterized by a network system widely structured which encourages the development of phenomena with important consequences both in social and educational fields. Among them a prominent role is played by so-called *digital convergence*, i.e. the process by which different media are redefined in their peculiarities thanks to digital coding. This cannot be merely seen as a technical or technological development, as this process has changed irreversibly media industry and the way media products are enjoyed.

Within a convergent culture, meanings – even music meanings – circulate according unexpected routes, contaminate each other, undergo discontinuous processes of certification, finding only temporary consolidation and being subjected to constant revisions (Ardizzone et al., 2008). In these complex and heterogeneous universes, sound and music take a new and stronger value as languages integrated into broader and diversified contexts, each leading to a hyper-fruition of aural stimuli.

It is worth underlining another key aspect, represented by the new technological opportunities of interaction, with the overcoming of the idea of *audience* proper of the Gutenberg model and the development prosumers increasingly active (McLuhan & Nevitt, 1974). Through digital technologies and the great impulse of Web the sharp dichotomy between producers and consumers disappears, similarly to what happened to communication processes in the 90' (Ceri & Gallino, 1994).

Due to the deep and rapid changes in the world around us, occurring also through media and technologies, the concept of literacy has to be rethought by encompassing a multiplicity of languages. Some researchers state that, in order to achieve socio-educational integration, it is not possible to conceive models of educational relationship other than inclusion, multi-literacy and promotion of excellence (Napodano & Iandoli, 2008). This is a set of educational actions that enhance workshop activities, foster the ability to use

language, orality and heterogeneous systems for analog and digital encoding, and promote the variability of cognitive styles and communication registers according to Gardner's idea of *forma mentis*.

New challenges emerge in modern school, traditionally considered as the depositary of literacy processes: guaranteeing the growth of students, each one with his/her own peculiarities, and giving equal dignity to the languages they interact with in everyday life. As regards music language, the problem is not how to employ media technologies during traditional hours of music education, but completely rethink their value in order to train individuals along different axes: searching in sound and timbre collections, analyzing their role and meaning inside a complex media message or a soundscape, understanding the value of music and audio authorship, and much more.

For an actual acquisition of competence, intended here as the ability to act in – or react to – a given situation within a given context in order to achieve a performance (Le Boterf, 1994), it is necessary to develop education proposals able to integrate music analysis with music production. In this sense, coding can prove to be an effective proposal since it implies writing, namely making explicit, juxtaposing and processing concepts in a critical, aware and creative way.

The paper is organized as follows: Section 2 discusses the pedagogical aspects of coding in primary school curricula, Section 3 illustrates the relationship between music and coding starting both from the music and from the technological points of view, Section 4 presents a Web interface that implements music coding, and finally Section 5 proposes a number of case studies.

2. FROM CODING TO MUSIC CODING

In Computer Science, the term *coding* refers to the implementation of a program, namely a sequence of instructions that make a computer run a given set of actions. It is well known that computer programming is a base skill required to understand information technology processes. More surprisingly, in recent times coding has been proposed also as a tool to foster learning processes in young students. By facing coding problems, students activate many cognitive functions that go beyond the acquisition of technical expertise in a strict sense. Coding tasks should be independent from a specific programming language or software environment. Rather, the target is developing a digital competence through playful learning (Resnick, 2004).

The theoretical bases go back to the pedagogical approach of *constructivism* (Piaget et al., 2014). As regards the introduction of coding and computational thinking in primary school, it is worth citing the ideas of Papert: while each discipline claims to push students to think, computer science achieves this result in an operative and concrete way. Papert considers coding as a useful tool to learn how to think, not as a way to obtain improved reasoning skills. In fact, the intrinsic goal is not making children good programmers, but providing the knowledge to create contexts where they can explore advanced ideas (Papert, 1980). For example, young learners can approach differential geometry through the metaphor of a turtle or they can understand the concept of feedback thanks to LEGO robots. Papert adopts LOGO as the reference programming language for children's coding activities. An implementation of Papert's ideas requires a deep change in teaching, transforming the mere transmission of knowledge and skills into laboratory activities and structured projects aimed at encouraging collaboration and discussion.

Music education and training for children require ad hoc techniques and methods as well as a specific review of school curricula. Recent scientific works (Young, 2003) show that an integration of multi-modal experiences based on activities such as moving, creating, playing, reflecting support the development of a "symbolically fluent child" (Gromko, 1995). The learning environment should be able to represent activity-oriented musical experiences, where students – properly sustained by scaffold elements – are involved in a process of music construction/deconstruction. For example, scaffolding children's early musical experiences, investigations and engagement in the world of sound helps them establish strong, confident, vibrant, and creative identities in learning, communication, and performance (Tomlinson, 2013).

Currently a new music pedagogy based on an integrated approach is emerging. Recalling the fundamental concepts of pedagogical activism (Dewey, 2005), the goal is enhancing that educational cross-component able to influence key aspects of the growth such as expressiveness, autonomy and sociality.

Music is able to influence the construction of the child's personality because it promotes the integration of perceptual, motor, affective, social and cognitive dimensions (Willems, 2011) by relating basic aspects of

human life (e.g. physiological, emotional and mental spheres) with the basic elements of music (e.g. rhythm, melody and harmony).

The abilities of listening, exploration and analysis are fundamental for the development of general meta-cognitive skills of the child, such as attention, concentration, control. In this sense, music is both an opportunity and a crucial educational strategy. For example, through music young students can develop the aspects of analysis and synthesis, problematization, argumentation, evaluation and application of rules. In addition, as regards the ability to read and understand, children have the possibility to train their transcoding skills, moving from the musical domain to the verbal language in order to describe what they heard (Branca, 2012).

In the digital era, new technologies and computer-based approaches can influence music learning and teaching processes. A recent and comprehensive review of this subject can be found in (Finney & Burnard, 2010), a work that discusses a range of innovative practices in order to highlight the changing nature of schooling and the transformation of music education. Many researchers, experts and music teachers feel a pressing need to provide new ways of thinking about the application of music and technology in schools. It is necessary to explore teaching strategies and approaches able to stimulate different forms of musical experience, meaningful engagement, creativity, teacher-learner interactions, and so on.

The idea of the this work is applying the most recent pedagogical theories about coding and music teaching in primary school through a playful approach to music composition, conceived for musically untrained children and designed to encourage the computational way of thinking.

3. MUSIC OPERATORS AND COMPUTER PROGRAMMING

In this section we want to explore the historical, cultural, and theoretical relationships between two domains apparently far apart, i.e. music and computer programming. We will follow two approaches somehow complementary, highlighting how some concepts of music can find a counterpart in the constructs of programming languages, and how some concepts typical of algorithmic way of thinking can be intuitively represented through music operators.

3.1 From Music to Coding

In 1985 composer and music theorist Fred Lerdahl and linguist Ray Jackendoff conceived a *generative theory of tonal music* (Lerdahl & Jackendoff, 1985). They provided a "formal description of the musical intuitions of a listener who is experienced in a musical idiom" based on the unique human capacity for musical understanding. The purpose was uncovering a musical grammar that could explain the human musical mind in a scientific manner comparable to Noam Chomsky's transformational or generative grammar dating back to 1975 (Chomsky, 2002). The novelty of their approach was the investigation of the mental procedures under which the structure of individual compositions can be built and understood.

Even if the approach of the mentioned authors was original, many theorists, musicologists and composers in the past formalized or merely used algorithmic processes to conceive music. Examples range from late-Baroque complex counterpoint processes (see e.g. the fugues by Johann Sebastian Bach) to twelve-tone serialism (see e.g. the works by the Second Viennese School, composed by Alban Berg, Anton Webern, Hanns Eisler and Arnold Schoenberg). As regards theoretical works, it is worth citing (Hofstadter, 1980), where the concepts of self-reference and formal rules are applied to the domain of mathematics, art, and music, and (Bizzi, 1982), which unveils the way many historical canons have been composed through mathematical rules and tables.

The mentioned works demonstrate that a formal and algorithmic approach to music composition and analysis is possible. Consequently, we introduce a number of *music operators* that can be seen as single steps of a generative algorithm. Due to the goals of this initiative, the application domain is simplified:

- A score is composed by a number of melodies, one per staff, presenting no chords. Harmony-related aspects come from the coexistence of notes belonging to different instruments and staves;
- Every note is quantized according to the smallest rhythmical value allowed. Longer values can be obtained by tying quanta together. This approach is valid for children's songs and simple tunes, where a coarse quantization is sufficient, but it would fail with tuplets and more complex rhythmic layouts;

- Supported operators are simple to be used and understood. The goal is not covering all possible music-composition processes, but provide untrained children with an intuitive tool set to build basic music performances.

3.1.1 Melodic Operators

The first class of music operators is the one of Melodic Operators (MOs), namely the operators that work on note pitch. The proposed framework includes:

- *Play n* – This operator starts playing a *n*-pitched note. The execution is stopped at the end of the current quantum, unless a Continue operator (see Rhythmic Operators below) is invoked at the next quantum;
- *Clear* – This operator unsets the pitch of the current quantum. In a quantized environment, each note naturally takes a single quantum, nevertheless a quantum cannot be empty. In this sense, *Clear* is the default operator, and it corresponds to the music concept of rest. This behavior implies that *Clear* belongs both to Melodic Operators (since it unsets pitches) and to Rhythmic Operators (since it is a placeholder for a rest);
- *Transpose up/down (and play)* – This operator modifies the previous pitch according to a number of ascending or descending grades. In our implementation, we decided to deal with scale grades (referring to a diatonic scale) rather than halftones (referring to a chromatic scale) because most pitched instruments for children are mainly diatonic, such as toy xylophones and whistles.

3.1.2 Rhythmic Operators

Rhythmic Operators (ROs) work on rhythm-related aspects of notes. In this case, the framework includes:

- *Tie* – This operator extends the last note (or rest) for the duration of the current quantum;
- *Clear* – As mentioned above, this operator means absence of sound. So, even if a note is naturally stopped by the end of its quantum, *Clear* has rhythmic implications too.

3.1.3 Other Operators

In order to foster computational thinking, other operators which do not belong to the mentioned classes have been implemented. A first case is *Repeat*, which allows to choose a number of previous steps to be repeated for a given number of times. In music notation there are symbols that recall this function (e.g., n-bar repeat signs, repeat barlines and text indications such as "Da capo"). However, for the sake of clarity the implemented *Repeat* operator has a less flexible applicability.

On the contrary, conditional expressions and constructs do not have a music definition, but they have been introduced to foster algorithmic thinking (see Section 3.2). In the proposed framework, comparison operators such as *lower than*, *higher than*, *lower or equal to*, *higher or equal to*, *equal to*, and *different from* can be invoked on pitches.

3.2 From Coding to Music

In Section 3.1 we have listed a number of music operators that are implicitly or explicitly used by composers when they create a music piece (Schoenberg & Stein, 1970). We have shown how these operators can be converted into atomic steps of an algorithm. On the contrary, in this section we start from some key concepts typical of computer programming and try to map them onto the music case study.

In our approach, programming constructs intentionally are not presented in an explicit way to children. In other words, the proposed framework hides their presence, but makes the underlying concepts emerge.

First, let us consider the standard *structure of a program*. Most programs have the following pattern: i) statements to mark the start of the program, ii) variable declaration, and iii) program statements. This basic schema can be mapped on: i) choice of the basic parameters (tempo, score length) and creation of instruments, ii) choice of the instruments to play, and iii) for each instrument, selection of a set of actions to be performed. Code lines are usually written and read from top to bottom, whereas music code – in accordance with score notation – should be interpreted from left to right.

In computer programming, *data typing* implies a classification identifying one of various types of data (e.g. integer, real, boolean, etc.) and defines allowed values and available operations on such values. In our proposal numeric, character and boolean values can be selected to perform specific operations, such as setting pitches or testing conditions. However, the novel approach is mapping the concept of data type onto

the choice of an instrument type to play. If instruments can be seen as instances of variables, then instrument types correspond to data types. For the sake of clarity, we will present the case of a xylophone vs a drum, which are two of the available instruments supported by the interface described in the following. Please note that they are both percussion instruments, but the former is pitched whereas the latter unpitched. The xylophone allows to define the exact note to be played. This implies the availability of operators such as *Transpose up* or *Compare pitches*. Besides, in accordance with computer programming where values are limited to a range (e.g. in C integer values have to fall in the [−32768,+32767] range, being 16 bits in size), pitch values present limits that depend on the chosen instrument. Needless to say, for an unpitched percussion instrument such as a drum the concept of pitch has no meaning, consequently only rhythmic operators are supported.

The concept of data typing can be further detailed. In computer programming, there is a dichotomy about *strong vs weak typing*. In our proposal, this aspect can be found in the independent use of instruments against the interdependence of notes across different instruments.

As regards operations on variables, *assignment* can be employed at different stages. For example, deciding which actions should take place at a given time is a form of assignment. Besides, some of the supported operators allow to set additional values, such as the pitch to be played or the number of repetitions to be performed. A complete list of operators for this proposal will be provided in the following.

As regards basic *control flow*, most programming languages support unconditional branch (e.g. goto, break, return, etc.), conditional branch (e.g. if-then-else, switch, etc.) and loops (e.g. while-do, do-while, for, etc.). In order to apply conditional statements to the music case, we introduce comparison operators to check for conditions and consequently choose between two alternative execution paths. For example: "If the current note of *Instrument A* is lower than the current note of *Instrument B*, then follow the upper path, otherwise choose the lower one. In order to keep things simple, braches always present the else clause and have the same length.

Loops and iterative constructs are exemplified through the repetition of the last n steps for k times. Please note that these statements introduce a number of implementation issues that in standard coding activities do not arise. The last n steps to be repeated do not have to generate ambiguous or illegal situations, pointing for example to a step where different branches are present (at the moment of coding, the system is not aware of conditional choices which may depend on run-time decisions). Similarly, a standard while-do loop would be difficult to introduce since the exit condition – and consequently the loop duration - cannot be predicted.

Some non-trivial programming features are supported too. First, data types are not limited to simple ones. The concept of *array* can be intuitively represented by a particular set of instruments (an array of mixed variables). If we define the array K by grouping a subset of instruments, operators can refer to single-element as well as aggregated values. For example, in conditional statements we can check conditions such as: "If the current note is higher than all current notes in K..." or "If current note of *Instrument K_1* is higher than the one played by K_3...". In this way, children can get familiar with the access to array elements by index (e.g. $K[1]$) or by key (e.g. *K["drum"]*).

Finally, the specific implementation of some operators recalls advanced concepts such as *overloading*, namely a type of polymorphism where different functions with the same name are invoked based on the data types or the parameters passed. For instance, the operator *Play* can be invoked with no parameters – i.e. *Play()*, which repeats the last pitch if present, and performs the default one otherwise – or with one parameter – i.e. *Play(n)*, which explicitly sets the pitch to be performed. Similarly, *Repeat()* means "play the last music symbol once again", *Repeat(x)* means "play last x music symbols once again", and *Repeat(x,y)* means "play last x music symbols y times". These are simple but effective examples of polymorphism.

4. WEB PROTOTYPE

In order to make music coding appealing, amusing and accessible to children, we have implemented an experimental Web interface adopting a gamification approach. The use of standard Web technologies and languages such as HTML5 and Javascript, developed and promulgated by W3C,[1] let us release a cross-platform multimedia environment, available on all browser-equipped and network-attached devices. The interface is shown in Figure 1.
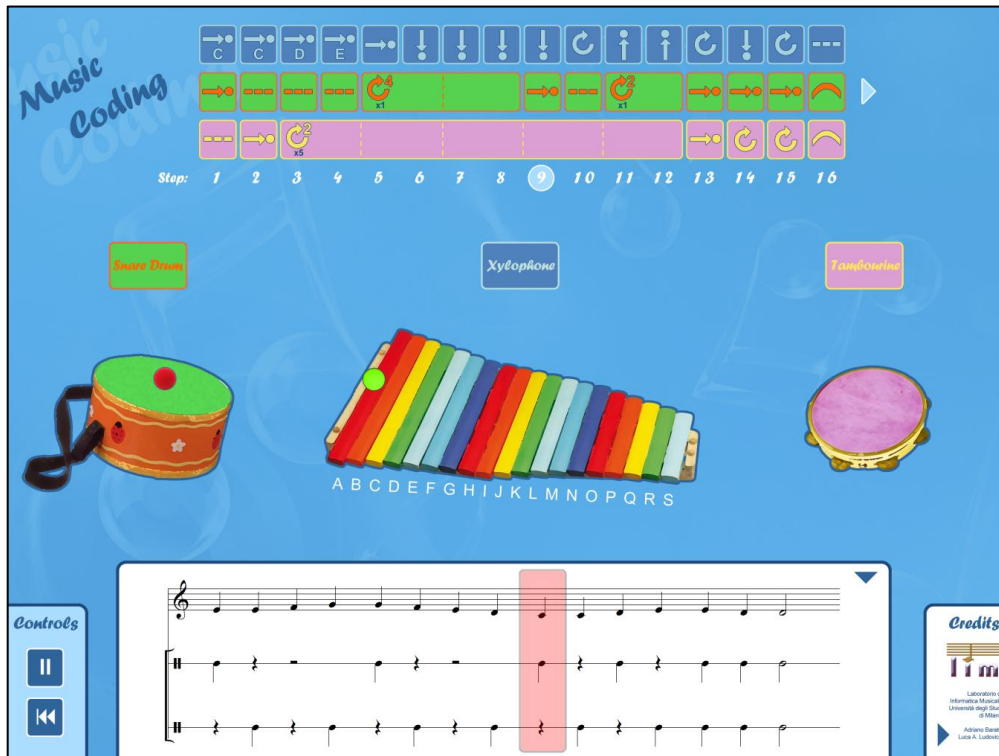
---

[1] World Wide Web Consortium

Figure 1. The Web interface designed for music coding.

4.1 The Gameplay

The educational activity is composed by two different steps, named *score coding* and *interactive listening* respectively.

During the first phase, a score is produced by disposing a number of cards (corresponding to music operators) on the game board. Available operators are chosen by clicking the instrument name, which opens a contextual toolset, since different instruments may present different sets of operators. The score length is not fixed, on the contrary it is set on-the-fly while dragging cards. The available space in the interface can be extended thanks to horizontal scrolling. As explained below, during this process the corresponding "traditional" music score is produced as well. The coding process can be performed collaboratively by many children together, for instance by assigning to each student either an instrument or a step, like in turn-based games.

The interactive listening phase starts when the Play button is pushed causing playback to occur. When score steps are parsed, colored balls begin to jump over the instrument interfaces. Graphical and audio materials are fully synchronized, thus allowing a multi-layer score-following experience. For children it is possible to look at the ball moves, follow music over the graphical as well as the traditional score, and listen to the corresponding sounds. This process somehow resembles *debugging*, since actions are parsed one by one to see if the desired result is being achieved. Music performance can be paused, stopped, and rewound.

The second phase has been defined interactive for a number of reasons. First, music performance can be influenced by adjusting and rearranging music operators: at any moment users can modify the score, thus going back to phase 1. Second, children can passively enjoy the results of coding but they can also interact by playing their own instruments.

8

4.2 The Interface

As regards the interface, the main area of the screen contains a number of pitched and unpitched music instruments. The available orchestra can be configured during the game setup. For music playback, the underlying metaphor is a jumping ball that strikes the sensitive areas of each instrument to make it play. Needless to say, jumps are controlled by the score.

The upper panel represents the game board, namely the area where cards corresponding to music operators are placed. Rounded squares constitute the quantized timeline, and their colors refer to a given music instrument. Icons like those shown in Figure 2 are employed to confer an intuitive graphical representation to music operators.

Figure 2. The icons for Play, Transpose down, Transpose up, Tie, Repeat and Clear music operators.

The bottom panel contains a score transcription following Common Western Notation (CWN) rules. Music contents are automatically generated over the staff system while cards are being placed on top of the game board. This "traditional" score representation has not been conceived for children, nevertheless, its presence can be important for a number of reasons:

- It can be checked by musically-trained students and teachers to validate the use of music operators;
- It can be directly read by young musician to create a live performance or an accompaniment;
- Thanks to full synchronization among all graphical and audio contents (*score following*), it can be an intuitive way to learn the rudiments of music notation.

An inverse exercise based on score notation is trying to reconstruct a simple tune whose score is known through the operators provided by the game. In this case, coding is involved as a problem-solving technique, similarly to standard computer programming: given a goal, find the self-contained step-by-step set of operations to be performed in order to achieve it.

5. EXAMPLES

Now let us refer again to the screenshot shown in Figure 1. This time we will focus on cards over the game board and the corresponding music contents.

The upper staff refers to a pitched instrument, consequently all music operators are available. From the point of view of rhythm, no steps but the last one have been extended through ad hoc operators. The first 4 steps are explicit assignments indicating the pitch to be performed. The 5th step is an overloaded version of the assignment operator, presenting no indication about the pitch: in this case, the last pitch is repeated. Other in use operators are simple transpositions (1 grade up or down) and simple repetitions, i.e. single repetitions of the last step. Clearly the same result could be achieved through different combinations of operators: for instance, the 3rd and the 4th step could be obtained through transpositions. This is another aspect of music coding close to computer programming, where different algorithms can be employed to achieve the same result.

The second and third staves encode parts to be played by unpitched instruments. For an untrained child, starting from a simplified set of operators can be more intuitive. Besides, a rhythmic feedback is easier to recognize.

In this example, we have used a number of Repeat operators, thus originating aggregations of steps different in length and number of repetitions. The notation at the 3rd step of the bottom staff is particularly efficient in representing a huge area of the score. In computer terminology, we would call it *compression*.

6. CONCLUSION

The concept of music coding presented in this work constitutes a novel approach to the computational way of thinking. Actually, playing music is a stimulating and amusing activity for children, which fosters not only technical and theoretical abilities, but also creativity, adaptivity, socialization and cooperation with others.

The interface and the underlying language presented above address primary school children without a specific music background. They have been designed to allow easy interaction with music contents as well as intuitive manipulation through ad hoc operators. Nevertheless, young musicians and music teachers can be involved in these activities as well. From a passive perspective, they can enjoy an alternative and formal representation of music language, quite far from common notation but close to the composer's way of thinking. From an active perspective, musically-trained users can join the playback phase, acting the role of additional instrumentalists and arrangers.

It is worth underlining that the provided framework can be seen as a tool both for *music synthesis* ("choose cards, set parameters and listen to the music produced") and for *music analysis* ("analyze an already available tune and try to reconstruct it through a suitable combination of cards and parameters").

At the moment of writing, the software application has been implemented and is publicly available over the Web, but it has not been extensively tested. As regards future work, a relevant goal is to conduct a test phase in Italian primary schools, in order to understand user acceptance rates and to validate the pedagogical approach. For instance, we are planning assessment activities in order to test if a music-based approach to problem solving can be more effective than a traditional one, by proposing the same algorithmic problem in different forms. An example is reaching a target point by walking in an open space versus composing a simple tune, two clearly different activities that could be described through the same sequence of algorithmic steps. Another kind of exercise is recognizing algorithmic structures in a given music piece.

A number of open problems must be faced before the final release of an effective learning system. These problems suggest a variety of research directions in the fields of pedagogy, technology-aided teaching and learning, computer science, music, and musicology.

REFERENCES

Ardizzone, P., Vandenplas-Holber, C., & Rivoltella, P. C. (2008). *Media e tecnologia per la didattica*. Vita e pensiero.

Bizzi, G. (1982). *Specchi invisibili dei suoni. La costruzione dei canoni: risposta a un enigma*. Centro internazionale di studi per la divulgazione della musica italiana, Roma.

Branca, D. (2012). L'importanza dell'educazione musicale: risvolti pedagogici del fare bene musica insieme. *Studi sulla formazione, 15*(1), 85-102.

Ceri, P., & Gallino, L. (1994). *Manuale di sociologia*. UTET libreria.

Chomsky, N. (2002). *Syntactic structures*. Walter de Gruyter.

Dewey, J. (2005). *Art as experience*. Penguin.

Finney, J., & Burnard, P. (Eds.). (2010). *Music education with digital technology*. Bloomsbury Publishing.

Gromko, J. E. (1994). Children's invented notations as measures of musical understanding. *Psychology of Music, 22*(2), 136-147.

Hofstadter, D. H. (1980). *Gödel, Escher, Bach: An Eternal Golden Braid*. Penguin Books.

Le Boterf, G. (1994). *De la compétence*. Paris: Les éditions d'organisation.

Lerdahl, F., & Jackendoff, R. (1985). *A generative theory of tonal music*. MIT press.

McLuhan, M., & Nevitt, B. (1974). Medium Meaning Message. *Communication*.

Napodano, M., & Iandoli, M. N. (2008). *Socrate in classe* (Vol. 7). Morlacchi Editore.

Papert, S (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, Inc.

Piaget, J., Beth, E. W., & Brousseau, G. (2014). Learning: Creation or re-creation? From Constructivism to the Theory of Didactical Situations. *A Critique of Creativity and Complexity: Deconstructing Clichés*, 25, 19.

Resnick, M. (2004). Edutainment? No thanks. I prefer playful learning. *Associazione Civita Report on Edutainment*, 14.

Schoenberg, A., & Stein, L. (1970). *Fundamentals of musical composition*. G. Strang (Ed.). Faber & Faber.

Tomlinson, M. M. (2013). Literacy and Music in Early Childhood Multimodal Learning and Design. *SAGE Open, 3*(3).

Willems, E. (2011). *Las bases psicológicas de la educación musical*. Editorial Paidós.

Young, S. (2003). Time–space structuring in spontaneous play on educational percussion instruments among three-and four-year-olds. *British Journal of Music Education*, 20(1), 45-59.