Proceedings of the 7th International Conference on Live Coding (ICLC 2023)
April 19th - 23rd – Utrecht, Netherlands

Live Coding and Education. A Practical Experience.

Francesco Corvi
Institute of Sonology
ffra237@gmail.com

Giovanni Mori
Tempo Reale
giova19828@gmail.com

Giovanni Nulli
Indire
g.nulli@indire.it

ABSTRACT

This paper presents an explorative research on the introduction of live coding into the Italian music curriculum in middle school with the aim to offer teachers a multidisciplinary didactic tool that can enhance the learning experience of students. The research focuses on the challenges that music teachers face when introducing live coding into their curricular activity and the type of training that can prepare them for it. The paper covers three main points: the training of teachers, the pedagogical strategies employed in the classrooms, and an overview of the results of the first year of experimentation. Overall, the paper aims to offer insight into the potential benefits and challenges of using live coding as a teaching tool in music education and highlights its potential for multidisciplinary applications, while also showing possible paths for future work and improvement.

1 Introduction

1.1 Research background and participants description

The project presented in this paper was promoted by Indire[1] and Tempo Reale[2] (as scientific partner) and it started during the 2021 – 2022 Italian school year[3]. It is designed to be an explorative research (Collins E. *et al.* 2020) project which plans, among its main purposes, to introduce live coding into ordinary curricular activities in middle schools in Italy (11 – 14 years old students). The proponents chose to introduce live coding to this school level as it is the only one that has music as curricular subject[4] in the Italian school system. The project will last for three years and it involves at least one class for each school included in this experimentation from the first to the third (and last) grade. Essentially the research question is: "Is it possible to introduce live coding into the music curriculum? If yes, how?". Indire's institutional mission is to promote sustainable innovation in schools and pedagogical approaches. Therefore, its projects strive to strike a balance between innovation and existing systems while carefully registering and analysing potential difficulties. Differently from Christopher Petrie (2021 and 2022) who adopted an experimental method to analyse the impact of Sonic Pi on students, to achieve our research goals, we employed a participatory research paradigm (Bargold and Thomas 2012 and Mortari 2007). A team was assembled, composed of five music teachers - each representing one of the participating schools - a musicologist specialised in live coding, a pedagogy researcher, and a musician with both a traditional background and an extensive experience in live coding. Their unique skills and expertise were utilised not only to provide guidance and support, but also to engage in meaningful discussions with the teachers, allowing them to articulate their ideas and challenges, and collaboratively develop solutions.

This project is part of a larger group of initiatives led by Indire aimed at promoting computational thinking as a cognitive and methodological tool, capable of unifying a wide range of topics. While the specific focus of this project is to assess the feasibility of incorporating live coding into the music curriculum, its broader objective is to explore the definitions and possibilities of computational thinking, as well as its potential integration into diverse academic subjects within the school curriculum.

[1] INDIRE is an Italian research institute which manages, promotes and documents projects for innovation in the Italian school system. More info on INDIRE https://www.indire.it/ .

[2] Tempo Reale is a centre for experimentation, education and production in new music headquartered in Florence (IT). More info on Tempo Reale are available at www.temporeale.it

[3] The Italian school year lasts from around middle September until mid June of the next solar year. This duration may vary depending on decisions from local administrations.

[4] Except for the musical Lyceum, that is a secondary school. However, there are just a few Music Lyceum in Italy. For this reason we preferred involving just middle schools considering that the subject of music education is among the curricular teaching subjects for this grade. That way, we can reach more students.

1.2 Live coding

The project promoters thought that live coding could be an excellent means for teaching different subjects at the same time because of its multidisciplinary nature. In fact, even if it can be defined as a "new improvisational practice tool" (Mori 2021), live coding fuses in the same practice the activity of music making, coding, improvisation, composition, borrowing concepts from math, English, technology and obviously music. And, most importantly, all these subjects are concentrated in a stimulating, involving, and fun activity. Many teachers have surely faced problems in teaching subjects which are considered complicated, annoying and sometimes even useless by the students. Putting the concepts taught into practice can be a great stimulus for pupils and can greatly ease their school duties while learning how these concepts can be applied in the real world. So, with these ideas in mind, and following other important experiences as the ones developed by Pam Burnard at Cambridge University (Burnhard et al. 2016) and another one brought ahead by Mehackit in Sweden[5], course promoters thought that training music teachers to live coding can be the best option to test the efficacy of what we can call the "live coding approach to learn": if you have fun and you work along with your peers, it is easier to learn any subject. Teaching music with the aid of live coding can connect students with the world of new technologies and make them aware that programming can be an easy, creative and enjoyable activity. Additionally, pairing music teaching with live coding combines notions from math, English, and technology, without requiring an excessive effort and, most importantly, by having fun! For all these reasons, the course organisers decided to dedicate the course to music teachers who manifested an interest in the use of new technologies in their classes as means to create a multidisciplinary teaching experience. To have strongly motivated participants has been deemed particularly important as learning new technologies and experimenting with their use in the classroom can be a difficult challenge for an adult educator with an already established teaching practice.

The promoted project involves five schools distributed in various parts of Italy: for each of these schools, a teacher of music participated in the live coding and education course led by an expert live coder. The five teachers included in the experimentation, before being involved in this project, already had a strong inclination toward the use of technology in their teaching approach in order to build an innovative educational path. Before starting to ask teachers to experiment teaching live coding in their classes, we opted for focusing at least a part of the masterclass on computational aspects. This approach aimed at making the teachers aware of the expressive potential of live coding practice, and also at how to use programming concepts, such as arrays and functions, as expressive tools.

1.3 Reason why we chose Sonic Pi

One of the reasons that pushed us to choose Sonic Pi as reference software for live coding in this course is that it is considered an easy language to learn, considering that it has been employed with young students in many cases all around the world[6]. Additionally, three of the involved teachers were already practising Sonic Pi, and they have already proposed the software to children during some workshops, even if not during their official lessons. Another reason for choosing Sonic Pi has been a very detailed tutorial, containing numerous examples at different grades of complexity, which makes this software a good option for starting to learn live coding and music from scratch. Moreover, Sonic Pi has a quite large community of users, with a quite active base of teachers from all around the world. In this way, teachers who use Sonic Pi in schools can freely exchange their impressions, approaches and ideas with their colleagues from different backgrounds, using online spaces like the Sonic Pi forum[7]. Finally, the last reason for choosing Sonic Pi has been that it is multiplatform and with a free and open source license that makes this software easy and cheap to install, without requiring additional expenses from schools and families.

2 The teacher training

2.1 Course structure

The masterclass began in October 2021 and lasted until the end of the school year in June 2022. In total it consisted of 50 hours of collective online lessons. In addition to these online classes, the trainer provided continuous support to the teachers using a group chat that also served to coordinate the project and to share ideas, materials and feelings between the group. The course started with an introductory meeting, in order to have an overview of the teachers' prior experience and understand what kind of expectations they had with regards to the course. During the meeting, the trainer outlined a number of possible topics that could have been addressed during the course. The main point stressed in this presentation was that the purpose of the course was not just limited to learning Sonic Pi, but to support

[5] See https://sonic-pi.mehackit.org/ for more information.

[6] See for example (Burnard et al 2016) and Mehackit website mentioned in footnote 5.

[7] https://in-thread.sonic-pi.net/

teachers' experimentation in using live coding as a teaching tool to employ in their classrooms. Therefore, the structure of the course could not be predetermined or rigid, but flexible to the various needs of teachers. For this reason, after the presentation all the participants shared feedback and ideas on the various topics presented and agreed with an overall common path of work, with the assumption that everything could have been adjusted as they went along.

We can loosely divide the course in two main phases, even though the split between the two is not neat. In the same way there was not always a clear separation between the various topics addressed by the lessons which often overlapped and intertwined during the explanations and the practical exercises. The split between theoretical and practical lessons was also not clearly defined. It happened that the trainer presented new concepts, functions and processes in Sonic Pi during two subsequent meetings, then he dedicated the next one to consolidate the notions explained and after he bounced back to new notions during the next lesson. This was very much dependent on the needs expressed by the teachers during the course. Below, we present an a posteriori account of the topics covered during the course and a structure of the order in which they were actually introduced.

2.2 First part

The first part of the course was intended as a full immersion in the world of Sonic Pi. From the first lessons the aim was to immediately give the teachers the knowledge to start to experiment with the software and to translate their already solid musical skills into the domain of algorithmic music. For this purpose none of the topics addressed needed to be comprehensive. Instead they offered an overview of the creative musical possibilities allowed by the programming language. The trainer often introduced the code and possibilities of Sonic Pi making reference both to general programming concepts and to musical concepts in order to contextualise them in a broader context. In addition, most of the computational concepts were introduced directly in their musical milieu, explaining their practical application. This strategy proved effective as it fostered the emergence of similarities between musical and computational concepts and made the teachers familiar from the very beginning with the process of converting their musical ideas into code. In many cases, some programming notions were simplified, preferring a short but intuitive explanation, pointing out at more in-depth references in case teachers wanted to go deeper into the concepts. Throughout the lectures, a series of illustrative examples and materials of the theoretical concepts and their musical applications were distributed to the teachers. In this way, teachers could deepen their understanding of the topics covered and also find suggestions about how to structure their work in class. In addition, this first part of the course included extensive collective exercise and detailed explanations of all the musical examples shown by the trainer.

The topics covered in the first part of the course were the following:

2.2.1 Introduction to the software and to live coding

To begin with, an overview of the Sonic Pi interface was given, explaining its panel configuration and showing the various settings. This was necessary to prevent any problems in teachers' first experiences with the software, and to offer clear directions for them to explain to students in their classes how to operate with the interface. This phase was followed by an overview of the practice of live coding, giving references of its use in education as well as in music and visual arts and introducing its various communities.

2.2.2 Introducing the variable and the loop

The first programming concept introduced was the variable. This included explaining how to assign different kinds of information to it, how to change the content assigned to it, how to manipulate the content inside of it, how to use it to create a one-to-many mapping by recalling the same variable in multiple parts of the code, and finally how to update the value of a variable inside a loop.

The loop was introduced with an analogy to the concept of a music loop (repeating section of sound material), showing how it was possible to update the contents of a variable inside a loop. After this brief explanation, the live_loop function was introduced by providing a series of musical examples using synths and samples to create musical patterns

2.2.3 Data types

The different data types available in Sonic Pi were introduced (String, Integer and Float, Array and rings and List, Boolean, Symbol, Hash). This included how to inspect the data type using the class method and how to convert data types using the other available methods. This part of the course particularly focused on analogies between data types and musical abstractions, creating analogies between concepts such as the array or ring and the musical chord or scale.

Methods for manipulating collections such as arrays were introduced with analogies to counterpoint techniques such as transposition, inverse canon, and thematic variation. The musical possibilities of these techniques were explored in Sonic Pi, with references to their use by Baroque and Minimalist composers.

2.2.4 Alea

Starting with the random selection of an element from a collection using the choose method, the aleatoric functions available in Sonic Pi were introduced with references to stochastic and algorithmic music of the twentieth century. Beginning with simple patterns, more sophisticated applications of these principles have been shown such as complex stochastic patterns and tendency masks.

2.2.5 Time states

Time states had been introduced using the cue and sync commands available in Sonic Pi. Starting from some examples, various strategies for creating interactions between various live_loops were discussed. The possibility of sharing data between threads using the get and set commands was also discussed.

2.2.6 Functions

To provide teachers with the opportunity to expand the possibilities of the language, one lesson was devoted to explaining how to define new functions within Sonic Pi. The approach used was to start from some codes produced in previous lessons and generalise their operation. The teachers were interested in this possibility, but they also found its application difficult in the context of the classroom.

2.3 Second part

After becoming familiar with the concepts introduced in the first part and their use in Sonic Pi, the teachers asked the trainer to focus more on practical applications rather than continuing to introduce more advanced programming concepts and software functionalities. This second part of the course focused on fostering a discussion among teachers and to offer support in developing personal algorithmic ideas on the software. This approach proved to be effective in stimulating the teachers' curiosity and creating an interest in them that continued even after the end of the course. At this stage, the introduction of new software features occurred only as a result of specific requests or needs expressed by the teachers. Some of the questions raised by the teachers concerned how to improve the "quality" of sound and of articulations, how to use Sonic Pi as an environment for exploring new compositional ideas, how to explore multimedia and multidisciplinary applications and how to use the computer as a performative instrument given its textual interface.

The topics and the activities covered in the second part of the course were the following:

2.3.1 Exercise and correction

The goal of these lessons was to push teachers as much as possible to experiment with the various possibilities of the software and to start thinking about what kind of music they would like to make with it. To make it easier to overcome the initial hurdle, the trainer prepared a series of guided exercises to accompany the teachers step by step in implementing ideas from scratch, starting from algorithms already seen during the lessons. These exercises have incremental complexity and were designed not to have a specific end goal, but to put teachers on the right track to create something of their own, that would go beyond the simple application of some functions or replicate some of the examples already provided. Each of these classes began with a collective correction of the exercises done during the week. Teachers were not required to bring working code, but to clearly present the kind of idea they had tried to implement. In this way the trainer could quickly correct syntax errors and help them convert their musical ideas into code.

2.3.2 Collaborative compositions

To stimulate teachers to collaborate with each other, the trainer suggested that they exchange the codes they were working on and create collaborative compositions. This method was also used to test the possible effectiveness of this method to understand the possibilities of its application in the classroom. The strategy of this method was to share a block of code containing a simple musical idea with another teacher by attaching a brief explanation of how it works.

The other teacher had to engage in creating a second block that could interact musically with the pre-existing materials by trying to create complementary ideas, for example, adding melodies to a rhythmic element or harmonising a pre-existing melody. Some teachers preferred to further develop pre-existing blocks instead of creating new blocks, and this strategy also proved to be valuable for improving skills and comparison among the group.

2.3.3 Sound effects

As a result of the demand for more appealing sound results, one lesson was devoted to sound effects that can be used within Sonic Pi. The reason why this topic had not been covered in the first part of the course is that applying effects in Sonic Pi without first knowing their functioning, at least in broad strokes, can be quite confusing. In order to introduce the use of effects, the trainer preferred to show their operation on a classic DAW, and then to get to the code in Sonic Pi only after the teachers had understood, at least on an intuitive level, the kind of sound manipulation was brought by the effect and the kind of change introduced by the various parameters.

2.3.4 Multimedia and Hydra

Following a specific request from one of the teachers and the interest expressed by the others in this direction, one lesson was devoted to the presentation of Hydra and the multimedia possibilities of live coding. This insight were used by some of the teachers that organised afternoon workshops for students and to evaluate multidisciplinary connections with art teachers.

2.3.5 Brainstorming

The brainstorming sessions proved very useful in getting teachers to discuss the eventual problems they may encounter during their activities about live coding difficulties they encountered in their classrooms with their students. During these sessions, an interesting critical discourse developed on the educational and artistic possibilities of live coding. The background of each of the teachers allowed for a collective discussion about possible contamination between live coding and more traditional methods, to understand when the use of software such as Sonic Pi can facilitate the understanding of musical concepts and extend its laboratory possibilities in the classroom.

2.4 Remarks

The course first part had a more frontal approach, mostly consisting in explanations of how the software and of algorithmic procedures are applied to music making. Since none of the teachers involved had extensive coding experience, this first part of the course was necessary to create a common basis for the group, and to show participants how the coding activity works. Additionally, in this phase the trainer introduced the various techniques and abstractions involved in live coding. The main purpose of this phase was to make the teachers aware of the code potential (and constraints), to make them accustomed to its use, and finally to work out how such knowledge could facilitate their teaching activities. The second phase instead was more focused on how to employ the acquired skills in expressing artistic ideas in relation to the teaching context. In this part, the participants were more involved in creating original musical ideas with code, making practical use of the notions learned during the first phase, and exploring the unique expressive possibility of these new strategies. Additionally, in the second part of the course the teachers were highly encouraged to collaborate between them and to share ideas and solutions adopted in the class. At the end of the school-year, when also the masterclass was going to an end, the trainer dedicated the last couple of lessons to discuss problems that emerged in the single classes in which the teachers proposed Sonic PI as musical activity. This has been a very important phase as it allowed teachers to think more about the overall work done during the year and to exchange their acquired knowledge and experience with the others. This phase proved particularly fruitful also to the trainer and the course organisers since it highlighted which are the main challenges to face in the approach we have adopted.

3 Challenges in the class

In this section we will elaborate on the main challenges that have been faced during the activity led by the teachers in the classes. Due to the COVID-19 related restrictions, we were forced to observe the class activity from an external and indirect point of view through online debriefing discussions with the teachers regarding what was happening in classrooms. For this purpose Indire led single interviews with the various teachers to check for problems with the course, the classes and the schools.

Below, we list the 3 main challenges encountered in the class:

1. A lack of pre-existent knowledge by the main part of the students. Most of them had a lack of both musical and coding knowledge. While the Italian school system provides a quite articulated program for music teaching, it does not provide any introduction to computer programming, even though this discipline gained a bit of attention recently. Despite this, we faced the problem of providing a solid background for live coding activities and to apply them to music teaching. Students and teachers are often not aware of what live coding is and what kind of skills it entails. The trainer and the teachers discussed how to cope with this problem: how can someone improve the teaching of musical concepts using a programming language if the student does not have any prior experience with coding?

2. When teachers stimulate their students to use software in a creative way, they find themselves to face questions that fall outside their expertise. This puts teachers in an unfamiliar condition to them as they are not always able to follow up with students and answer possible questions. Additionally, it is hard to foresee the kinds of questions the students may ask, as they have a very lively imagination and, additionally, the software opens up potentially infinite possibilities.

3. The most practical question the teachers have to face is that while schools usually have a lab provided with enough PCs for every single student of the class, there are a number of students that do not have a computer at their home so they are unable to practise the notions learned during lessons. There is a sensible number of students that do not have any familiarity with the PC as a tool. Actually less and less families own a personal computer in their homes nowadays, and if so PCs are parents' tools for work or "serious" activities which do not attract young generations attention. Students from middle school ages are more familiar instead with touch screens as they find devices provided with this kind of user interface almost everywhere. Usually they own and use a smartphone from a very young age. Additionally, schools sometimes equip their labs with tablet computers or Chromebooks instead of "traditional" PCs, because they are cheaper and easier to use for students and allow all the tasks required for this school level. As a consequence, many students, if not most of them, are not skilled at all with the mouse and keyboard interface.

Below we list some observations and solutions emerged from the experimentation:

1. The various Sonic Pi commands have been introduced to the students incrementally starting from simple tasks declined in a very elementary way. Consequently, many practical examples and exercises have been used in order to introduce any single function and its context. What emerged during the work in the class is that while some musical concepts can be taught directly by relating them to programming concepts, others need a separate explanation as the textual interface is not always clearly readable for the students. In these latter cases, making examples with code can further enrich the explanation of the given concept and highlight new perspectives. An aspect to take into consideration is that it is important to introduce new programming concepts starting from a practical example of how it will be used in a musical application. This can make musical concepts easy to understand even if the programming concepts involved are not entirely clear. This was one of the main goals of the training course provided to the teachers and one of the main reasons why we searched for an experienced live coding performer as a trainer instead of a programmer.

2. This aspect may be tricky for the teacher at first, but it can also be very stimulating for both students and teachers. Facing new challenges together with the students leads to lessons which are less frontal and classic and more horizontal and collective. In this approach students feel free to try things out, to speak their mind and to experiment outside the main scope of the class. By going beyond the pre-established goals and by finding problems and solutions as a group they develop together with the teacher new strategies and find new ways of learning, also developing a good team. In addition, demonstrating that there are problems and that they can be addressed together helps to create a good and inclusive team spirit in the class instead of creating competition among individual students.

3. A possible solution to this problem may be using applications that can run on smartphones and tablets. While not all kids have access to PCs, almost everyone has access to smartphones. By using a bluetooth keyboard that can be connected to the device, the coding process can also resemble the one of the laptop. In order to do so we should produce or find a web based live coding environment or a portable application similar to Sonic Pi in terms of simplicity of language and richness in documentation. During the faculty course various alternatives were evaluated, but no candidate seemed to be better than Sonic PI. Moreover, this issue emerged at an already advanced stage of the course where dropping Sonic Pi would have meant starting over on many aspects related to the syntax and interface of the software.

4 Documentation from the classes

During the overall course duration, teachers who have participated in the activity frequently discussed how to find the best way to introduce live coding to students inside their standard school activities. As a consequence of this discussion, it has been necessary to rethink the way in which some tasks have to be performed using Sonic Pi. At the same time, organisers and teachers faced the problem of how to introduce coding in the classroom reality: students need, at the beginning, to know how the software interface works (that is very easy perfortune) and to learn how to write code without errors. On the other hand, teachers introduced Sonic Pi as a new instrument to play. In this way, it was less hard to make Sonic PI play as they refer their activities to the practical part of music instead of the theoretical one.

In this first year of experimentation, teachers mainly employed Sonic Pi as an interactive compositional environment. This was necessary because before introducing Sonic Pi as a instrument in the laboratory activity, it was useful to use the software to introduce students to the musical and computational concepts needed to engage in the performative tasks

Teachers used different strategies to introduce Sonic Pi in their music lessons. Some of them used it to explain part of the notions contained in the curriculum: the notes, timbres, parameters and so on. Others just started teaching it as a new music instrument to play.

In Agostini *et al* (2022), p. 412 - 416, four progressive examples are shown.

The first example shows the use of `live_loop` instruction:

```
# Nicolò
    live_loop:batteria do
    sleep 0.25
    sample :drum_cymbal_open
    sleep 1
    sample :drum_snare_hard
    sleep 0.5
    sample :drum_heavy_kick
    sleep 0.25
    sample :drum_snare_soft
    sleep 0.5
    sample :drum_cymbal_closed
end

live_loop:basso do
    use_synth :dull_bell
    play 45
    sleep 1
    play 40
    sleep 0.5
end
```

This command is introduced creating a parallel between the code and the different instruments playing in a band, so each `live_loop` code block becomes a different instrument. As we can see in the previous example, each `live_loop` represents a different instrument ("batteria" means drum, "basso" means bass guitar, mimicking a rock music band). In this way students learn the means of `live_loop` through the metaphor of "the band". As we can hear[8], the students who created the following code, created a polyrhythmic song, so the use of the code lets the students experiment with unconventional music structures.

In the second example, the teacher introduces the use of arrays, as a "more economic way" to write an algorithm and at the same time a way to write a melody:

```
#Coded by Federico (°-°)
#Ottudols
use_bpm 120
# ho messo la velocità della pulsazione a 120
live_loop :patt do
```

[8] Sound example 1: https://drive.google.com/file/d/192etPYvnV5bsyU4Q6mHIruz51uvwmSs6/view

```
    use_synth :fm
    #ho usato un sintetizzatore che fa un rumore elettronico
    play [:f, :e, :d, :e, :g].choose
    #ho fatto scegliere al sistema tra quattro note di un play
    sleep [0.5, 1, 1, 0.5].choose
# ho fatto scegliere al sistema 2 tipologie di sleep
end

live_loop :energy do
    use_synth :mod_fm
    #ho usato un sintetizzatore (anch'esso elettronico)
    play_pattern_timed [:f, :e, :f, :d, :g], [0.5, 1, 1, 0.5].choose
    #ho fatto una lista di note con gli sleep che vengono scelti dal sistema
end

live_loop :ok do
    sample :drum_bass_hard,amp: 8
    # ho usato il campione di una grancassa
    sleep 1
    # ho usato uno sleep 1 in modo da staccare per fare una melodia
    sample :drum_tom_hi_hard
    # ho usato un campione di un piatto della batteria
    sleep 0.5
    # ho usato uno sleep 1 in modo da staccare per fare una melodia
    sample :drum_tom_hi_hard
    #ho usato un campione di un piatto della batteria
    sleep 0.5
    # ho usato uno sleep 1 in modo da staccare per fare una melodia
end
```

The array, as a list of items, can easily be explained as a list of pitches, and a list of pitches can be used to compose a melody. In this way, the concept of melody, that is more intuitive and practical, can be used as a key to make the concept of an array more accessible and highlight a possible practical use. As we can see in the second example just a few lines above, there are two arrays used: one for pitches, one for durations. These two arrays allow the students to create a melody in a single line (line 19) [9].

The third and following example uses arrays for both pitch and durations as well [10]. Declaring the arrays at the beginning of the code allows the students to use the .tick and .choose methods in a more elegant way:

```
#Nicola & Nicolò & Thiago song
dormita = [0.75, 0.25, 0.5]
audio = [:d5, :e5, :g5]
live_loop :melodia do
    play audio.choose
    sleep 0.75
end

live_loop :piercarlosulbeat do
    sample :bd_boom
    sleep 1
end

live_loop :michelechitarra do
    sample :guit_e_fifths
    sleep 0.75
end

liveloop :chitarrathigo do
    play scale(:c, chinese).tick
    sleep dormita.choose
end
```

[9] Sound example 2: https://drive.google.com/file/d/1JoGO4BQHBL2-zsQZwpeVSUkH7UnnAsrB/view

[10] Sound example 3: https://drive.google.com/file/d/1e1M3Qu3nn6PgwMiXXy8ZJ5EojnBRL4iL/view

The first command (`.tick`) implements a modulo operation. Since an array in Sonic Pi is considered as a ring, after the last listed item the software automatically loops back to the first item of the array. In this way the student acquires an intuitive understanding of the modulo operation by listening to the looped sequence of pitches. The second command, instead, introduces the concept of randomness: each array's item is picked randomly from the array.

In the final example that was produced in the last part of the year, the same pitch array is used in three different parts of the code:

```
#Alice
note = [:e, :c, :f, :g]
#note contiene una lista di note che verrà utilizzata in tre live_loop diversi
live_loop :melodia do
    use_synth :piano
    sleep 1
    play note.choose #scelta casuale dalla lista "note"
    sleep 1
    play :d
    sleep 0.5
    play :a
    sleep 1
    sample ambi_piano
end

live_loop :suonicauti do
    use_synth :hollow
    play note.tick #suona le note in ordine
    use_octave 2 #suona due ottave sopra
    sleep 1
    play :d, amp:4
    sleep 2
    sample :elec_pop
end

live_loop :batteria do
    sample :elec_flip
    sleep 1
    play note.choose #scelta casuale dalla lista "note"
    sample :elec_wood
    sleep 0.5
    use_octave 1
    play :a
    sleep 2
end
```

This shows how students learnt to use arrays in more flexible and versatile ways, producing different musical output from the same data structure. Using only one array to drive different loops is a very synthetic and efficient way to code that will be useful once students will engage in live coding performance, because they can easily change the whole harmonic field of the different live_loops by modifying a single line of code.

5 Conclusions

At the end of our first year of experimentation we can say that teachers involved in the project successfully introduced Sonic Pi into their lessons, facing the challenges we described in section 3.

In the next year we would like to diversify the scope of the project including different subjects to see if this increases the efficacy of the pedagogical activity and the degree of student enjoyment. Our primary objective is to foster an interdisciplinary approach, which would allow us to incorporate more subjects in a singular activity. Along with music, we can develop the students' skills in maths, performing arts, foreign language studies, theatre and technology. One of the most complex challenges for this project is to evaluate the impact of skills that students acquire with musical live coding on other subjects. For instance, our current insights suggest that there may not be a direct correlation between the ability to write complex algorithms and the enhancement of mathematical or logical skills. On the other hand this outcome may be attained through the combined efforts of teachers who can raise students' awareness of the interrelationship between subjects and connect skills from different areas. For this reason it is necessary to analyse the work done so far with teachers from other disciplines to enable them to illustrate to students how the knowledge they have acquired can be employed in other subject areas. According Lodi&Martini (2021) [6] p. 886 and 887 this high level skill transfers can occur mainly between similar subjects such as mathematics and science, while there are many more difficulties if the subjects are far apart. On the same line, Di Sessa (2017) criticises a strict definition of computational thinking highlighting the difficulties when transferring high level skills. If direct transfer of higher skills such as computational thinking is not directly provable, we think (Nulli *et al.* 2022) that coding and computational thinking can be very useful as meta-language between different subjects. Using coding both in music and maths classes could be a good way to reinforce similar skills and discuss them with the students, making them underline similarities and differences. Although we anticipate some difficulty in recruiting teachers from different subjects, we are quite optimistic as this experience resulted in interesting outcomes, and we will stress these positive perspectives on how merging subjects together can yield promising results and further enhance the efficacy of the project. Another aspect we will try to develop is to organise more workshops and activities outside school hours. We believe that working both inside and outside schools can be proficient for students' skill development and can give them a stronger motivation for going on exploring live coding and consequently increase their knowledge and enthusiasm. Additionally, students who participate in external labs may help their peers during school hours, activating a peer tutoring experience. Finally, we would like to find more ways of connecting teachers to the live coding community and share the knowledge they developed during their experiences in class.

References

Agostini R., Izzo L. and G. Nulli. 2022. "Spiegare il 'caso' e l'array nel live coding musicale." *Atti Convegno Nazionale Didamatica 2022*: 410 – 420. Aica, Milano.

Bergold, J., and S. Thomas. 2012. "Participatory Research Methods: A Methodological Approach in Motion." *Historical Social Research / Historische Sozialforschung*, *37*(4 (142)): 191–222. http://www.jstor.org/stable/41756482.

Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M.A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V. and G. Stupurienė. 2022. "Reviewing Computational Thinking in Compulsory Education." *Publications Office of the European Union*, Luxembourg, doi:10.2760/126955,JRC128347.

Burnard, P., Florack, F., Blackwell, A., Aaron, S., and C. Philbin. 2016. "Learning from Live Coding." In King, A. Routledge, *The Routledge Companion to Music Technology and Education.* [Book chapter]. https://doi.org/10.17863/CAM.10951.

Colin E., Gerring J., Mahoney J. 2020., "The production of knowledge enhancing progress in social sciences." *Cambridge University Press*, Cambridge.

Disessa A. 2018. "Computational Literacy and"The Big Picture" Concerning Computers in Mathematics Education." *Mathematical Thinking and Learning*, 20(1):3-3. DOI: 10.1080/10986065.2018.1403544.

Lodi, M., Martini, S. 2021 "Computational Thinking, Between Papert and Wing." *Sci & Educ 30*, 883–908. https://doi.org/10.1007/s11191-021-00202-5.

Miur. 2012. "Indicazioni Nazionali per il curricolo." https://www.miur.gov.it/documents/20182/51310/DM+254_2012.pdf/1f967360-0ca6-48fb-95e9-c15d49f18831?version=1.0&t=1480418494262.

Miur. 2015. "Piano Nazionale Scuola Digitale." https://www.miur.gov.it/documents/20182/50615/Piano+nazionale+scuola+digitale.pdf/5b1a7e34-b678-40c5-8d26-e7b646708d70?version=1.1&t=1496170125686.

Miur. 2018. "Indicazioni Nazionali e Nuovi Scenari." https://www.miur.gov.it/documents/20182/0/Indicazioni+nazionali+e+nuovi+scenari/.

Mori G. 2021. "Live coding? What does it mean?" Aracne. Rome.

Mortari L. 2007. "Cultura della ricerca e pedagogia." Carocci. Rome.

Nulli G., Miotti B., Di Stasio M. 2022. "Robotica educativa e coding: strumenti per la trasformazione del curricolo." Carocci. Rome.

Petrie C. 2021. "Interdisciplinary computational thinking with music and programming: a case study on algorithmic music composition with Sonic Pi." *Computer Science Education.* DOI: 10.1080/08993408.2021.1935603

Petrie C. 2022. "Programming music with Sonic Pi promotes positive attitudes for beginners." *Computers and Education*, vol. 179, 104409. https://doi.org/10.1016/j.compedu.2021.104409