A Music Programming Course for Undergraduate Music Conservatory Students: Evaluation and Lessons Learnt

Marcella Mandanici[1] [a] and Simone Spagnol[2] [b]

[1] *Music Conservatory "L. Marenzio", Brescia, Italy*

[2] *Iuav University of Venice, Venice, Italy*

Keywords: Computational Thinking, Music Programming, Technology Integration.

Abstract: This paper introduces the content and organisation of a music programming course offered to undergraduate Conservatory students in the spring of 2022. A number of evaluation procedures, including pre- and post-course questionnaires and exercises, and a final assignment have been administered by the teacher. Results indicate an increased confidence in the use of computers and programming, although some aspects of creativity and computational thinking need further revision. The authors examine the course content in light of the results obtained, discuss the followed approach, and make assumptions for the improvement of both course content and assessment methods.

1 INTRODUCTION

According to G. R. Skuse (Skuse et al., 2017), algorithmic thinking can benefit scholars and students of any discipline. If it is true that computer technology has pervaded nearly every field of knowledge (Bailey and Stefaniak, 2002), the skill of solving a problem, of reducing a process into different steps, and the connected rational reasoning are fundamental abilities to be achieved during the training process of the XXI century learners. This means that computational thinking must also be introduced in the curricula of arts and humanities students. Music benefits from more than a century of research and artistic productions involving first electrical and then electronic technologies (Collins et al., 2013). Moreover - well before the birth of electronic music - the basic musical structures such as intervals, scales, and tuning systems, originated from mathematical and scientific reasoning. Just think of Pythagoras, Kepler, and Euler, to cite only a few (Fauvel et al., 2006).

However, in spite of this meaningful background, it is still difficult to explain why having a short, limited but still important experience of music programming is significant for music Conservatory undergraduate students. This is due to many factors.

Firstly, electronic music has been present in Italian music Conservatoires for more than 40 years, as one of the earliest chairs of electronic music has been assigned to Teresa Rampazzi[1] at the Conservatory of Padua in 1972 (Zattra, 2000). But the heritage of knowledge and the evolution of the taste in sound linked to the practices of electronic music have not managed to leave a small circle of adepts, thus bypassing the great potential for renewal of these technologies. The flaws of technology integration have not reached the core of professional musical training yet, drawing a strong separation between music technologies and all the other sectors of professional music education (Bauer and Kenton, 2005).

Secondly, music editing software such as Soundation,[2] PreSonus,[3] or Reaper[4] have long since come into common use among musicians both for music production (Burgess, 2013) and pedagogical practice (Tobias, 2013). These software usually include tens of loops, patterns and pre-recorded music chunks that make music creation easy and fun. However, the strong stylistic and commercial biases connected to their use make these environments more fit for standard music production rather than for a reflection on the nature of musical structures and on the way they

[a] https://orcid.org/0000-0003-1863-4178

[b] https://orcid.org/0000-0002-9309-0871

[1] Teresa Rampazzi (1914-2001) composer, pianist and researcher, is one of the pioneers in electronic music in Italy.

[2] https://soundation.com/

[3] https://www.presonus.com/products/studio-one/

[4] https://www.reaper.fm/

Mandanici, M. and Spagnol, S.
A Music Programming Course for Undergraduate Music Conservatory Students: Evaluation and Lessons Learnt.
DOI: 10.5220/0012056900003470
In *Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023) - Volume 1*, pages 387-396
ISBN: 978-989-758-641-5; ISSN: 2184-5026

can be generated by an algorithm. While music editors imply a certain knowledge of music technology, this is not aimed at content transformation (Hamilton et al., 2016) but rather at the optimisation of the production process. On the contrary, mastering a music algorithm may lead to unprecedented music creation which depends on the understanding and manipulation of music elements from scratch.

Thirdly, professional musicians usually consider themselves extraneous to the processes of technology integration or – at best – employ technology only if it can better realise an already well known task, such as recording a performance or printing a score. Moreover, with more than 40% of the curriculum devoted to individual instrumental practice,[5] the acoustic instrument learner of an Italian conservatoire has little time left for developing abilities such as creative and critical thinking, collaboration, and communication.

2 THE *MUSIC INFORMATICS* COURSE

For these reasons the content of the *Music Informatics* course is devoted to learning some of the fundamental processes of music programming. The course is held online on the Google Classroom[6] platform of the Conservatory "L. Marenzio", Brescia (Italy). It is subdivided into 8 lessons, each lasting 3 hours. The course gives 3 ECTS[7], which are obtained without grading but through an internal verification.

2.1 Objectives

The course aims at offering a very basic music programming experience to young musicians who have possibly never seen computer programming in their previous educational curriculum. There are several positive values connected to this choice.

- Learning a programming language means entering a world where everything is governed by logical reasoning. This may be a complete novelty for a student of humanities or arts.
- Music programming requires logical reasoning about musical structures. This implies self-awareness, critical thinking, and analytical abilities. At the end it can produce demystification of musical concepts and a deeper knowledge about them.
- This experience may contribute to increase digital literacy and computer familiarisation in students who are usually not required to use digital devices in their learning process.
- Using a computer for music programming may help future processes of technology integration, i.e., the use of digital technologies for musical activities.
- The creative activities included in the course may complement a curriculum primarily devoted to score reading and music performance.

In line with the concepts expressed above, the course aims at realising the following objectives.

1. Introducing the students to computational thinking, i.e. the ability of " *...formulating problems so their solutions can be represented as computational steps and algorithms"* (Aho, 2012). Computational thinking also implies problem solving, analysis, and pattern recognition (Wing, 2011).
2. Offering them the possibility of experiencing critical thinking, creativity, collaboration and communication (also known as the four C's[8]) as an important complement to their academic curriculum.
3. Stimulating their creativity and musicality in relation to computer programming.

2.2 Content

The course employs Pure Data[9] as music programming software. Pure Data has been created in 1996 by Miller Puckette, one of the fathers of computer music. It is a multi-platform and free software and for these reasons it is particularly suitable for use in the course. Pure Data is also a graphical program, where the various functions are represented by objects. The algorithm is built by linking the objects through cords in a tree structure that governs the flow of data.

2.2.1 Inner Organisation

Table 1 shows the course content subdivided into lectures, workshops, and homework. During lectures the concepts are explained frontally through multimodal presentations and patch examples. In workshops the same content is presented through a set of exercises proposed to the students. One of the students in turn

[5] See this study plan from the Brescia Music Conservatory as an example, https://www.consbs.it/content/uploads/2022/02/PDS-SINT-TRI-ARPA.pdf

[6] https://edu.google.com.au/workspace-for-education/classroom/

[7] European Credit Transfer System

[8] https://www.aeseducation.com/blog/four-cs-21st-century-skills

[9] http://puredata.info/

Table 1: Course organisation and content.

Week	Lecture	Workshop	Homework
1	Introduction Computational thinking The “Four C’s”	Introduction to Pure Data Controlling numerical flow (`random`, `moses`, `clip`)	Pre-course questionnaire lab1_assignment (logic)
2	Audio and MIDI MIDI events	MIDI production (`noteout`, `makenote`) MIDI controls (modulation, volume, pan and sustain)	lab2_assignment (exploration)
3	Counters (iteration, reset, loops)	The `counter` abstraction A pattern of a musical form	lab3_assignment (musical form)
4	Scales (MIDI notes and arrays)	Major and minor scales Scale transposition	lab4_assignment (scale fragments)
5	Musical structures (patterns, polyphony, chords)	Pattern repetition (cycles), polyphony	lab5_assignment (chords)
6	Melodies	Augmentation, musical streams	lab6_assignment (creativity)
7	Patch modules	Bands, clusters and lines	Final group assignment
8	Group assignment discussion	Group assignment discussion	Delivery of the final project Post-course questionnaire

tries to solve one exercise with the help of other students. The teacher comments the solutions proposed by the students and possibly offers alternative ways. During the workshop, homework can be discussed upon request. In their homework students are required to solve problems closely related to the content of the current lecture and workshop, but not identical to the examples/exercises (i.e., they must demonstrate not only to have understood the content of the lesson, but also to be able to elaborate upon it).

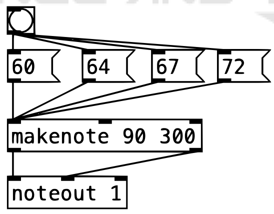

Figure 1: A basic algorithm for chord production.

2.2.2 Topics

Following the approach already proposed by V. J. Manzo (Manzo, 2016) the course starts from the algorithmic implementation of basic musical structures such as melodies, scales, and chords. The main reason for this choice is that starting with concepts that students already know may ease the process of approaching computational thinking. All these musical structures are built following the same principle:

1. first show the simplest implementation of the structure, for instance a 4-part C major chord (see Figure 1);
2. secondly, consider that this is a rigid algorithm because it requires to write down all the intervals each time a new chord is needed;
3. finally, apply the principles of computational thinking (decomposition, abstraction, and pattern recognition) to produce the most flexible structure possible. As can be seen from Figure 2, the fundamental of the chord has been separated from the chord pattern (decomposition), allowing an easy transposition of the same chord (pattern recognition). The whole algorithm represents an abstraction (i.e., a general solution) with respect to the basic version (i.e., a particular solution).

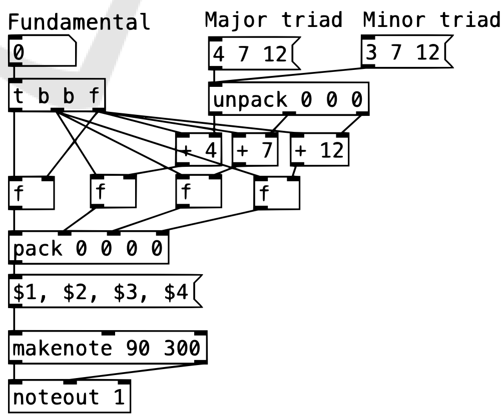

Figure 2: A more flexible algorithm for chord production.

The principles of decomposition, abstraction, and pattern recognition are introduced in the first lesson as theoretical assumptions. Later in week 7, the various objects presented during the course are summarised and grouped together according to a number of patch modules, each performing a specific function.

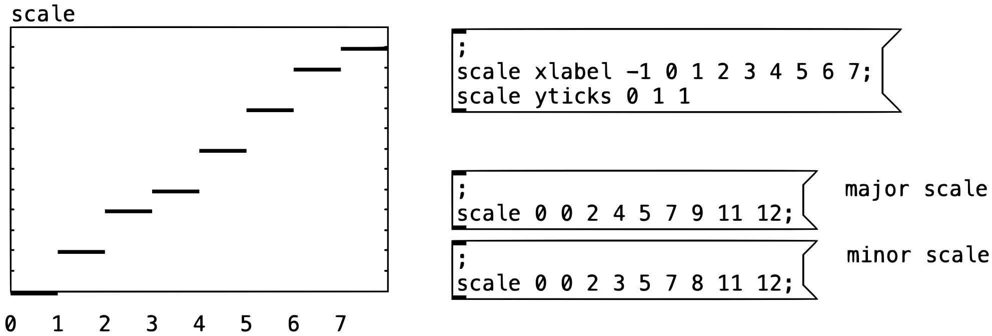

Figure 3: An array containing the sequence of intervals of a major scale. A minor scale can be obtained by clicking on the message beneath.

These are:

1. **Data storage.** This function is performed by the `array` object, which can store sequences of intervals that can be changed dynamically with a message (see Figure 3).
2. **Timing and Initialisation.** The timing function is performed by the `metro` object. The `trigger` object allows to simultaneously start `metro` and to set the initial conditions of the patch (e.g. counters to 0, note duration, fundamental pitch, and so on).
3. **Array data scrolling and loop.** The main abstraction here is the `counter` which is the true engine of the patch. In conjunction with the `module` object it allows to iteratively scroll the array performing repetitions, loops, and transpositions.
4. **Musical form.** This can be controlled by a `selector` object, which keeps track of the numbers produced by the `counter` and allows to stop the performance, to change musical parameters such as timbre, velocity and duration, or to change the content of the array.
5. **Sound production.** This module groups all the necessary MIDI objects for sound production such as `makenote` and `noteout` .

In order to realise a melody composed of a number of patterns, repetitions, and transpositions it is enough to store the patterns in the array and control the range of the scroll by adding an offset to the indices or by changing the fundamental. But the modules and objects presented during the course can be used not only to reproduce scales and melodies but also to create new musical forms. This possibility is exemplified in the week 4 assignment, where starting from an array containing the intervals of a major scale a new musical form composed of scale fragments scattered upon various fundamental pitches is generated.

The potentialities of the various objects for composition of musical forms are described at the end of the week 7 lecture. For instance, the `random` object is a powerful tool for the randomisation of musical parameters (e.g. randomising the argument of the `metro` object can make the musical performance irregular and more interesting in some contexts). The `spigot` object can govern streams of notes, making them appear and disappear at random intervals of time, while `pipe` can provide polyphonic events at various time delays. Moreover, in the week 7 workshop various examples of musical structures such as bands of sounds of different direction and amplitude, clusters and lines are presented with the aim of showing different cases of non-traditional music elements. For their final assignment, the students are required to form groups of maximum 6 people and to produce a musical project based on a Pure Data patch with a written description of project analysis, aims, and limitations.

3 ASSESSMENT

The assessment of the *Music Informatics* course aims at measuring the progress of the students according to the objectives outlined in Section 2.1. For the assessment we employed both quantitative and qualitative methods. Quantitative data are collected through pre- and post-course questionnaires and exercises, while qualitative data are obtained from the analysis of the final projects of the course.

3.1 Participants

The participants who completed the course and all the assessments are 43 in total (23 females), mean age 23.6 years, std = 8.35. The majority of the students (69.7%) were attending the second year of their bachelor's degree course, mainly in piano (18.6%), guitar (13.9%), and singing (11.6%).

Table 2: The pre- and post-course questionnaire.

Group	#	Pre-Course	Post-Course
Expectations and outcomes	Q01	Learn something more about the integration of computers and music	Now I understand something more about the integration of computers and music
	Q02	Learn something more about computers and programming	I gained more knowledge about computers and programming
	Q03	Learn something more about music and composition	As a result of this course I would like to learn something more about music and composition
	Q04	Collaborate with others in my major	I learned how to collaborate with others in my major
	Q05	Learn to be more creative	I learned to think more creatively
	Q06	Learn to be more communicative	I learned to be more communicative
	Q07	Become more willing to take risks	I am more willing to take risks
Self-perception	Q08	Technical/artistic	Technical/artistic
	Q09	Uncreative/creative	Uncreative/creative
	Q10	Do not take risks/Take risks	Do not take risks/Take risks
	Q11	Rules/insights	Rules/insights
	Q12	Big picture/Detail	Big picture/Detail
Personal opinions	Q13	I know what it means to work with computers	Now I know better what it means working with computers
	Q14	I enjoy working with computers	As a result I am more comfortable working with computers
	Q15	I know what it means to create music	Now I know better what it means to create music
	Q16	I enjoy creating my own music	I enjoyed projects where we had to create music
	Q17	I am confident using a computer language to accomplish a complex task	Now I am more confident using a computer language to accomplish complex tasks
	Q18	I am confident in my ability to express myself through music	Now I am more confident in my ability to express myself through music
	Q19	I am good at breaking a large problem down into its components and attacking those one at a time to solve the bigger problem	As a result of this course I am better at breaking a large problem down into its components and attacking those one at a time to solve the bigger problem
	Q20	I am good at diagnosing problems and formulating solutions	As a result of this course I am better at diagnosing problems and formulating solutions
	Q21	Computers can be used to create cool music	Computers can be used to create cool music
	Q22	I enjoy working on group projects	Now I appreciate the benefits of working on group projects
	Q23	There is a deep connection between music and computer science	Now I am seeing a deep connection between music and computer science
	Q24	Computer programming is fun	Computer programming is now more fun to me
	Q25	Standard music notation is a form of code	Now I see the connection between standard musical notation and code
	Q26	Writing music down is similar to writing a computer program	Writing music down is similar to writing a computer program

3.2 Quantitative Assessment

During the first lecture, the teacher administered a pre-course questionnaire aimed at evaluating the students' expectations on the course as well as personal attitudes and opinions on music, computing, and their interplay. Similarly, a post-course questionnaire (with matching questions) was administered at the end of the last lecture. The two questionnaires are elaborated on the basis of the *Sound thinking pre-course survey* and *Sound thinking post-course survey* by Greher and Heines (Greher and Heines, 2014), a pair of tools specifically targeted at assessing how a class learns and how attitudes change as a result of taking the class itself. Each of the two questionnaires is composed of twenty-six 5-point Likert scale questions, as reported in Table 2. The original 33 questions have been reduced to 26 by eliminating those items that did not pertain to the current course or did not match from pre- to post-course.

In addition, right after the pre-course questionnaire, the teacher administered 4 exercises on music analysis and computational thinking. In the first exercise (EX1), a melody was presented to the students with some examples of pattern repetition and pattern transposition; then, they were required to identify how many patterns they find in a new melody. In the second exercise (EX2), students had to identify pattern repetitions and transpositions in the same melodic excerpt. In the third exercise (EX3), students were presented with a flowchart and asked to identify its output when a short sequence of notes is used as input. Similar exercises were also administered right after the post-course questionnaire. A fourth exercise on patch logic was considered redundant and not administered in the post test. More details on the four exercises can be found in (Mandanici, 2022).

Table 3: The 9 final projects and their musical features.

#	Name	Timbres	Harmony/Elements	Model
1	Jazz Improvisation	Drums, piano, double bass, sax	Jazz harmony	Accompanied melody
2	Inno alla Gioia	Piano, organ, flute, double bass	Tonal harmony	Accompanied melody
3	Grandcanon	Harp, brasses, sax, guitar	Tonal harmony	Polyphony
4	Timbri in scala	Random timbres	C major scale	Scale
5	Pachelbel canon	Strings	Tonal harmony	Polyphony
6	Cum santo	Vibraphone, harp	Band, crazy harp, melody	Original musical form
7	Pachelbel canon	Trumpet, french horn, flute, harpsichord, piano	Tonal harmony	Accompanied melody
8	Halloween	Space voice, woodblock, ocarina, warm pad, vibraphone, square wave	Band, melody and line	Original musical form
9	Polovtsian Dances	Piano, random timbre	Twisted melody	Polyphony

3.3 Qualitative Assessment

Qualitative assessment is based upon the analysis of the final assignments of the course. The 43 students were subdivided into 9 groups with the aim of producing an original musical project composed of a commented Pure Data patch and written presentation. The aims of this final assignment were:

1. to foster creative thinking;
2. to assess if the students are able to master the possibilities offered by the patch modules;
3. to offer the possibility of experiencing collaboration and communication in a working group.

According to Sternberg and Kaufman (Sternberg and Kaufman, 2010) creativity is a relative concept, which depends on the interaction between the stimulus and the receiver. Moreover, the evaluation of creativity may be biased by cultural constraints, because a product can be thought to be creative in one historical era and insignificant in another. In the context of this course we started from the implementation of simple musical structures such as scales, melodies and polyphony, with explicit reference to the tonal language. At the same time, however, we have proposed computational structures which – in addition to implementing the aforementioned models – also have the power to undermine the perception of these same structures, transforming them into something new. This entails the application of *divergent thinking*, i.e., the ability of producing many alternative responses to the same task (Runco and Pritzker, 2020). Thus, in order to evaluate creativity in the musical projects presented by the students, we list the main musical features that characterise them, i.e., the timbres employed, harmony or musical structure, and the model that inspired the project. All these elements are summarised in Table 3.

4 RESULTS

In this section we analyse the results obtained from the applied methods. We present quantitative pre- and post-course data for questionnaires and exercises, and qualitative data for the final assignments of the course.

4.1 Quantitative Analysis

Figure 4 reports the median scores obtained for the 26 questions in each questionnaire. Notice that the large majority of these scores is stable around 4, denoting a clear agreement with the corresponding questions and therefore highlighting generally positive expectations, outcomes, and opinions on the course and on its potential to improve students' critical thinking, creativity, collaboration and communication skills. Given the typical distribution of Likert scale data, we chose to apply non-parametric tests to look for statistically significant differences between paired pre- and post-test data. Therefore, twenty-six separate Wilcoxon signed-rank tests, one per questionnaire item, were run.

In the *expectations and outcomes* question group, the post-course items Q02 and Q04 significantly ranked higher than the corresponding pre-course items ($Z = -2.068$, $p = .039$ and $Z = -2.095$, $p = .036$, respectively). For what concerns the *self-perception* question group, the tests highlighted no significant difference between the outcomes of the pre- and post-course items except for Q11, where scores were significantly higher in the post-course questionnaire ($Z = -3.166$, $p = .002$). The items that showed statistically significant differences in the *personal opinions* question group are Q13 ($Z = -2.118$, $p = .034$), Q16 ($Z = -3.082$, $p = .002$), Q17 ($Z = -4.709$, $p < .001$), Q19 ($Z = -3.089$, $p = .002$), Q22 ($Z = -2.054$, $p = .04$), Q24 ($Z = -2.586$, $p = .01$), Q25 ($Z = -2.228$, $p = .026$), and Q26 ($Z = -2.184$, $p = .029$). All the related scores were significantly

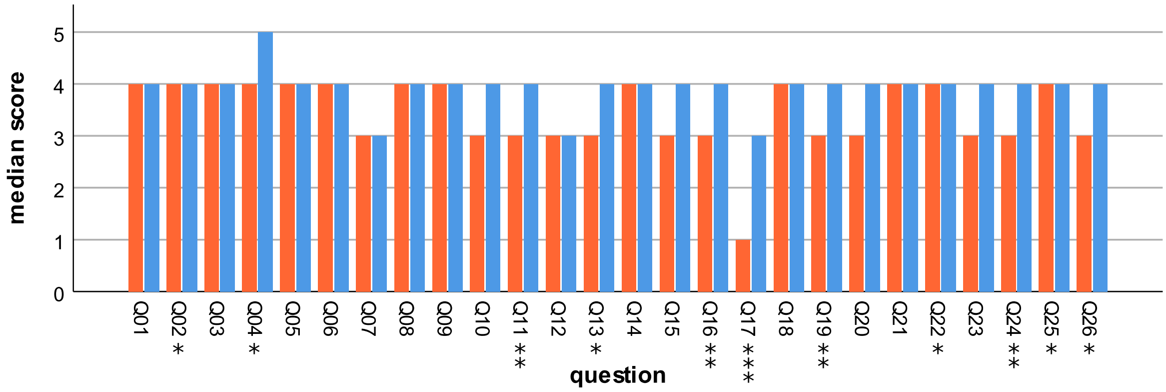

Figure 4: Median scores for the 26 questions in the pre-course (orange bars) and post-course (blue bars) questionnaires. Asterisks indicate statistically significant differences between matching pairs of pre- and post-course questions: $*p \leq 0.05$, $**p \leq 0.01$, $***p \leq 0.001$.

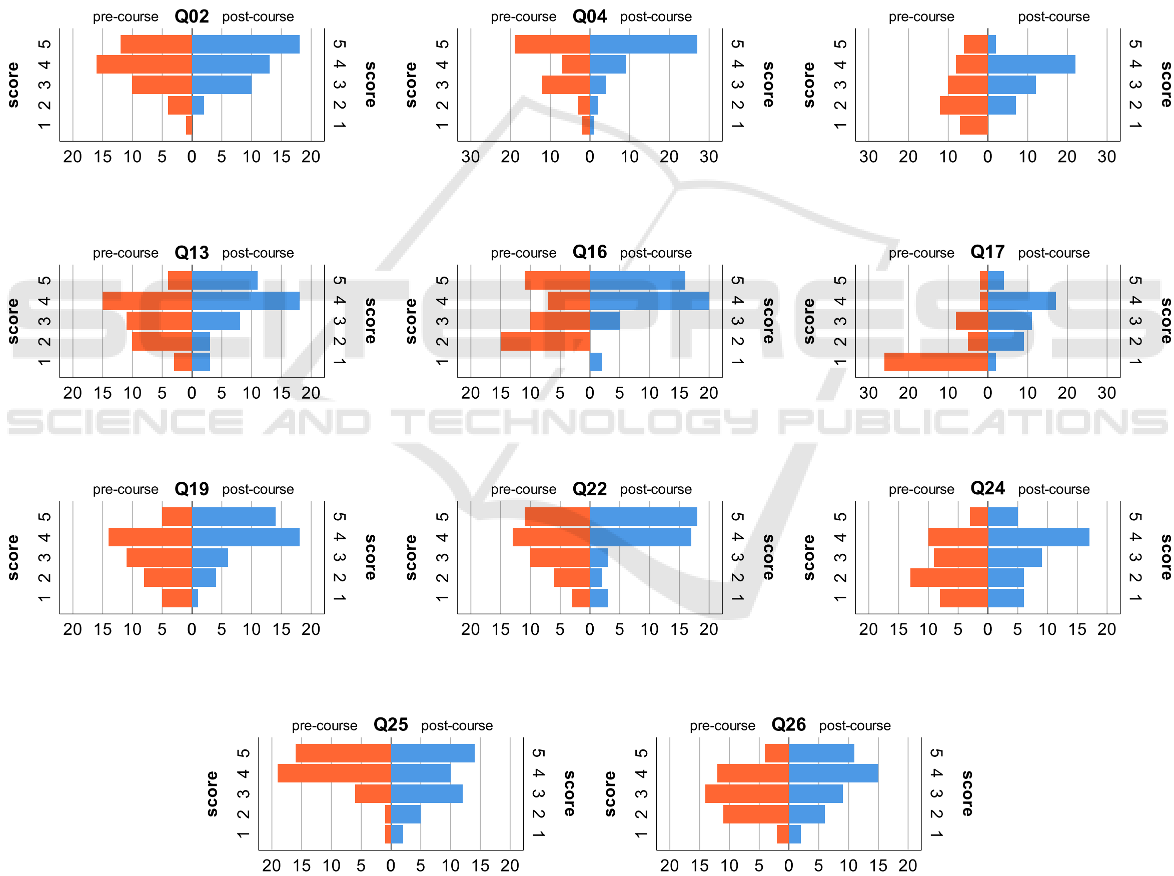

Figure 5: Histograms for the 11 items showing statistically significant differences between pre-course (orange bars) and post-course (blue bars) questionnaires.

higher in the post-course than in the pre-course questionnaire, except for Q25 where scores significantly decreased in the post-course questionnaire. Figure 5 reports all the histograms for the 11 questionnaire items where statistical significance was found.

Finally, we analysed the results of the three exercises. While the score in EX2 was obtained as the fraction of correct answers on the total number of beats, EX1 and EX3 could either have a right or wrong answer. Therefore, we applied a Wilcoxon

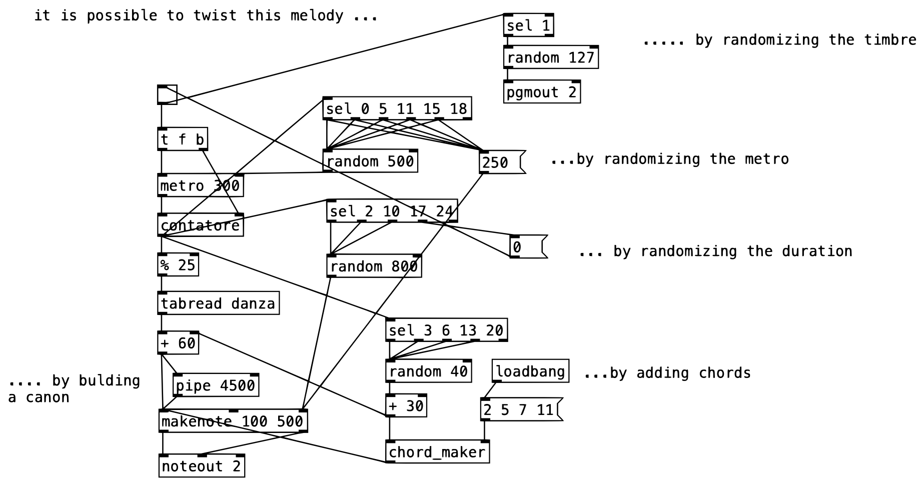

Figure 6: The melody taken from Borodin's Polovtsian Dances twisted by musical parameters randomisation and polyphony.

signed-rank test to EX2 scores and two McNemar's tests to EX1 and EX3 scores. The tests highlighted no significant difference between the outcomes of the pre- and post-course exercises except for EX2, where scores were significantly higher in the post-course questionnaire ($Z = -4.014$, $p < .001$).

4.2 Qualitative Analysis

The first aim of the qualitative assessment is the evaluation of the amount of divergent thinking applied in the realization of the projects (see Section 3.3). Most of the projects in Table 3 employ tonal or jazz harmony as the main reference musical structures. If we compare this with the musical models and timbres chosen, on the whole we can affirm that six projects out of nine (i.e., projects 1, 2, 3, 4, 5, and 7) essentially follow musical models belonging to the students' musical experience. Since group 1 was composed of jazz students, they explicitly declared to stick to jazz harmony and improvisation form. Group 2 employed the famous melody from "Ode to Joy",[10] while the students of group 3 composed an original canon to be implemented in Pure Data. This choice explicitly assigns to the program the function of a performer rather than that of a creator/transformer of the musical product. Finally, groups 5 and 7 employed Pachelbel's Canon,[11] one in a polyphonic form and the other in a harmonised form. These projects do not show a high degree of divergent thinking because they basically employ Pure Data as a MIDI sequencer, ignoring the creative potentialities of music programming.

On the other hand, projects 6, 8, and 9 contain elements such as *crazy harp* (taken from the assignment of week 4), and bands and lines elaborated on the examples of week 7. Even though tonal melodies are still present in all three cases, they are presented in the context of an original musical form. For instance, group 9 proposed a melody taken from Borodin's Polovtsian Dances[12] that is first played as it is and then completely twisted by introducing random timbres, metro, and note durations (see Figure 6). Chords and a canon are also added to the melody, obtaining a completely different effect every time the process is activated. In all three cases students showed to be able to manipulate Pure Data objects in a creative way, offering original solutions in the use of the objects previously presented in a tonal context.

5 DISCUSSION AND CONCLUSIONS

From the analysis of those questions that obtained a statistically significant difference from pre- to post-test, the following conclusions can be drawn:

1. Computers and programming: there is a clear belief among the students that their abilities in managing computers and computer programming have increased (Q02, Q13, Q17, Q19, and Q24);
2. Collaboration: students think they have learnt how to work in a group and better appreciate this activity (Q04 and Q22);

[10] Written on the words of German poet F. Schiller in 1785, the "Ode to Joy" is the fourth movement of Beethoven's Ninth Symphony, completed in 1824.

[11] Written by the German composer Johann Pachelbel, born in 1653.

[12] These Dances belong to a scene from Act 2 of Alexander Borodin's "Prince Igor", remained incomplete upon the composer's death in 1887.

3. Self-perception: they rely more on insights rather than rules (Q11);
4. Music: they tend to enjoy more music creation (Q16) and to think that writing music is similar to computer programming (Q26). However, they are more negative about recognising standard music notation as a form of code (Q25).
5. Communication: there is no significant result for this ability.

According to points 1 and 2, undergraduate students who are very seldom required to use electronic devices in their academic career met computing programming with a positive attitude and seem to have become more confident with it, making progress in computational thinking too. They enjoyed working in groups and music creation activities, but seem to draw a line between standard music notation and coding (point 4). They still prefer to follow insights rather than rules (point 3), as to say "we are artists and not computer scientists". Moreover, the students' perception on their communication abilities has not changed (point 5), possibly because of their little coverage during the course and/or presence in the questionnaire.

As far as concerns the exercises, one would expect a similar result for EX1 and EX2 since they target the same analytical ability, i.e., recognising the repetition and transposition of the same pattern in a melody. Alas, only EX2 showed a statistically significant improvement in the results. Actually, while in EX1 students were asked for a synthetic answer, EX2 is more analytical. Since it would probably have made more sense to first identify the patterns and then count them, perhaps this procedure has misled many students, causing a difference in favour of the analytical procedure. EX3 was designed for assessing if the logic represented in a flowchart could be better interpreted by the students after the course. Results clearly show this is not the case. However, it has to be acknowledged that flowcharts have not been addressed during the course, and this result shows that insight is not sufficient to obtain advances in this ability.

Qualitative assessment aimed at verifying the musical creativity of the students in relation to the patch modules and musical forms presented during the course. Three creative projects versus six based on the persistence of already known musical models show that the potentialities of computer programming have not been sufficiently implemented by the students. It seems instead that they need to take inspiration from well established musical models and that they cannot get over the limits of a musical training too tied to a single musical style.

5.1 Lessons Learnt

The results obtained in the *Music Informatics* 2022 course have been employed to improve the content and assessment methods of the 2023 course.

1. The main changes concern the questionnaire. Because of the different formulation of most questions in the post-course questionnaire 2022, many questions may have produced unreliable results. Therefore, in the 2023 version, the same questions are being used for the pre- and post-course questionnaire. Moreover, to balance the questionnaire in relation to the four C's abilities, more questions about collaboration and communication have been added.
2. Another change is about the use of flowcharts to represent the inner working of an algorithm, based on the observation that flowcharts are able to outline the logic of a music patch. For this reason, during lectures, examples of flowcharts have been coupled to Pure Data patches to stimulate the students' computational reasoning.
3. A final but very important change has been introduced to aim at offering models of algorithmic composition to the students to improve their creativity. Among these the *Illiac Suite*, by L. Hiller and L. Isaacson (Hiller and Isaacson, 1979), and short musical excerpts by A. Webern[13] and B. Bartòk[14].

5.2 Conclusions

The role and importance of a music programming course in the curriculum integration of undergraduate music conservatory students has been carefully evaluated and motivated in light of the development of computational thinking and the four C's. Quantitative and qualitative methods have been tested for the assessment of these abilities. The results of the assessment of the 2022 course have been used to improve the content and assessment of the 2023 course, which is still running at the time of submission of this article. The authors hope that the iterative verification and update of its contents can improve the efficiency of the course, thus contributing to a more complete training in line with the needs of digital culture.

[13] Anton Webern (1883–1945) is one of the members of the "School of Wien" together with A. Schönberg and A. Berg.

[14] Béla Bartòk (1881–1945) is a Hungarian composer who derived his musical style from his country's popular music.

REFERENCES

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7):832–835.

Bailey, J. L. and Stefaniak, G. (2002). Preparing the information technology workforce for the new millennium. *ACM SIGCPR Computer Personnel*, 20(4):4–15.

Bauer, J. and Kenton, J. (2005). Toward technology integration in the schools: Why it isn't happening. *Journal of Technology and Teacher Education*, 13(4):519–546.

Burgess, R. J. (2013). *The art of music production: The theory and practice*. Oxford University Press.

Collins, N., Schedel, M., and Wilson, S. (2013). *Electronic Music*. Cambridge University Press.

Fauvel, J., Flood, R., and Wilson, R. J. (2006). *Music and mathematics: From Pythagoras to fractals*. Oxford University Press on Demand.

Greher, G. R. and Heines, J. M. (2014). *Computational thinking in sound: Teaching the art and science of music and technology*. Oxford University Press.

Hamilton, E. R., Rosenberg, J. M., and Akcaoglu, M. (2016). The substitution augmentation modification redefinition (SAMR) model: A critical review and suggestions for its use. *TechTrends*, 60:433–441.

Hiller, L. A. and Isaacson, L. M. (1979). *Experimental Music; Composition with an electronic computer*. Greenwood Publishing Group Inc.

Mandanici, M. (2022). Fostering computational thinking in undergraduated music conservatory students. In *Proc. 14th Int. Conf. on Computer Supported Education (CSEDU 2022)*, pages 449–457.

Manzo, V. J. (2016). *Max/MSP/Jitter for music: A practical guide to developing interactive music systems for education and more*. Oxford University Press.

Runco, M. A. and Pritzker, S. R. (2020). *Encyclopedia of creativity*. Academic press.

Skuse, G. R., Walzer, D. A., Tomasek, K., Baldwin, D., and Bailey, M. (2017). Computer science and the liberal arts: Hidden synergies and boundless opportunities. In *New Directions for Computing Education: Embedding Computing Across Disciplines*, pages 45–61. Springer.

Sternberg, R. J. and Kaufman, J. C. (2010). Constraints on creativity. *The Cambridge handbook of creativity*, pages 467–482.

Tobias, E. S. (2013). Composing, songwriting, and producing: Informing popular music pedagogy. *Research Studies in Music Education*, 35(2):213–237.

Wing, J. (2011). Research notebook: Computational thinking—what and why. *The Link Magazine*, 6:20–23.

Zattra, L. (2000). Da Teresa Rampazzi al Centro di Sonologia Computazionale (CSC): La stagione della musica elettronica a Padova. *Master thesis, Università di Padova*.