A Web Prototype to Teach Music and Computational Thinking Through Building Blocks

Adriano Baratè
Laboratorio di Informatica Musicale
Department of Computer Science
University of Milan
adriano.barate@unimi.it

Luca A. Ludovico
Laboratorio di Informatica Musicale
Department of Computer Science
University of Milan
luca.ludovico@unimi.it

Davide A. Mauro
Department of Computer &
Information Technology
Marshall University
maurod@marshall.edu

ABSTRACT

This paper presents the recent evolution of a Web prototype originally conceived to teach music and computational thinking to preschool and primary school learners through a gamification approach. The software tool, called *Legato*, is based on the metaphor of building blocks, whose characteristics (e.g., position in space, shape, and color) can be associated with basic music parameters (e.g., pitch, rhythm, and timbre). *Legato* is a Web app written using standard languages, such as HTML5, CSS and JavaScript; besides, it adopts the Web MIDI API to produce sounds. The prototype is made publicly available for evaluation and use in an educational context.

CCS CONCEPTS

• **Social and professional topics → Computational thinking**; • **Applied computing → Interactive learning environments**; *Computer games.*

KEYWORDS

music education, computational thinking, building blocks, web audio

ACM Reference Format:
Adriano Baratè, Luca A. Ludovico, and Davide A. Mauro. 2019. A Web Prototype to Teach Music and Computational Thinking Through Building Blocks. In *Audio Mostly (AM'19), September 18–20, 2019, Nottingham, United Kingdom.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3356590.3356625

1 INTRODUCTION

Even if we are moving toward the so called *digital-natives* generation, construction toys are still fairly popular. The use of bricks in education is not a novel idea. Just to give few significant examples, [Church et al. 2010] and [Danahy et al. 2014] explore their application to robotics, [McNally et al. 2006] and – more recently – [Pinto-Llorente et al. 2016] discuss their use in Computer Science education. There are noticeable examples in the field of music learning and composition, too, such as [Oestermeier et al. 2015].

Construction blocks, e.g. well-known LEGO©bricks, are very familiar to young people, and their tangible nature can be profitably used in combination with computer applications that dematerialize them in order to design effective educational activities.

In this context, the present work stems from the latest evolution of the theories proposed in [Baratè et al. 2013] and applied, in particular, to the didactic applications described in [Baratè et al. 2017] and [Ludovico et al. 2017]. This work is posed in continuity with the mentioned experiences expanding on some of the limitations that were first discovered in those prototypes.

The paper will continue introducing the rationals behind the music-to-brick mappings proposed by the new tool, the Web application itself and the design principles that inspired it, and finally some remarks on the current limitations and further activities needed to improve the prototype.

2 MAPPING MUSIC PARAMETERS ONTO BRICKS' CHARACTERISTICS

The educational goal of *Legato* is letting young learners understand music principles and create simple compositions through the use of construction bricks. In order to achieve this goal, we have selected some basic music parameters to be mapped onto simple bricks' characteristics, giving the possibility to create customized associations between the two groups.

Specifically, the user can control four parameters of note events: (i) pitch; (ii) rhythm, i.e. duration and position in time; (iii) dynamics; (iv) timbre.

Concerning bricks' characteristics, for the sake of simplicity we decided to support only squared and rectangular pieces. Both their dimensions range from 1 to 4 units, thus originating block sizes from 1×1 to 4×4. The other feature currently managed by the interface is brick color, which can be picked from a 16-element palette. After selecting a shape and a color, bricks can be placed on a 2D game plate; their horizontal and vertical positions are additional characteristics that can be associated with musical parameters.

In the interface (see Figure 1), the bricks representing musical events can be combined so as to produce simple tunes and chords. As in most performance-oriented formats, rests (i.e. the absence of sound) are not explicitly encoded, but they emerge from the absence of events.

The Web interface, described in detail in Section 3, allows to map each of the mentioned music parameters onto different brick characteristics, supporting also void associations. Let us mention a simple use case, mimicking the well-known piano roll representation in use in digital audio workstations and sequencers. In this case, the horizontal dimension of the game plate represents time aspects, the vertical dimension encodes pitch, and colors can be associated with different instruments.

A completely different approach would consist in using colors for time position, the horizontal axis for pitches, and the vertical one for dynamics. In this case: the default sequence of colors provides timing (e.g., all yellow bricks will trigger a note event at beat $t_0 = 0$, all orange bricks will play at beat $t_1 = 1$, and so on); the horizontal placement (and size) of bricks determines pitches to be played;[1] the vertical placement produces variations in dynamics, from ***ppp*** to ***fff***.

3 THE WEB APPLICATION

This section provides some details about the Web application called *Legato*, publicly available at http://legato.lim.di.unimi.it/. Such a software tool has been developed by adopting standards promoted by the World Wide Web Consortium (W3C), in order to be compatible with most operating systems and browsers and freely available via the Web.

3.1 Design Principles

In the design and implementation of the prototype, we followed these principles:

- *Intuitiveness* – The application should be easy to use by young learners, thus it has to present a limited number of easy-to-understand controls and provide a clear feedback in terms of graphical and audio output;
- *Flexibility* – The application has to support any combination in the association of music parameters with the features of construction blocks, so as to foster different approaches to music encoding and encourage the development of abstraction skills;
- *Gamification* – Brick-based games are popular, and their virtualization through a Web interface is expected to engage young learners, even more so if coupled with manipulative activities with real construction blocks;
- *Availability* – The Web application is free, cross-platform, available through any HTML5-compatible browser for any device connected to the Internet.

We decided to demand the management of sound production to the Web MIDI API,[2] a specification currently under development by the Audio group of the W3C. The Web MIDI API provides support to the Musical Instrument Digital Interface (MIDI) protocol, thus enabling Web applications to interface with MIDI input and output devices on the client system and send and receive MIDI messages. The most noticeable reasons why this approach can be profitably used in a computer-based music educational environment are explained in [Ludovico 2017]. Among the advantages, it is worth citing that sound samples are completely managed by a MIDI synthesizer, with no need to pre-record a high number of audio samples with different pitch, duration, timbre, etc. Conversely, it is possible to use the full range of General MIDI patches for multi-timbral synthesizers. Besides, the client-server exchange of music data is very light and compact, since MIDI does not transmit audio signals, but it sends commands to trigger audio events. Finally, a Web applications based on the Web MIDI API can be embedded into a MIDI chain and connected to other MIDI real (hardware) or virtual (software) devices.

The adoption of the Web MIDI API unfortunately implies also some drawbacks. First, MIDI is not an audio format, and the quality of sound samples on different systems is unpredictable since it largely depends on the characteristics of the hardware or software synthesizer in use. Moreover, the Web MIDI API requires the presence of a MIDI synthesizer belonging to either a physical or a virtual MIDI chain (it could be, e.g., an external sound module attached to the client, or an ad-hoc software installed in the user's system). Finally, the Web MIDI API has not reached the status of standard yet, and, at the moment of writing, it is supported only by some browsers: Google Chrome (from version 43 on), Opera (from version 30 on), Chromium (from version 67 on), and other minor releases such as Opera Mobile, Chrome for Android, UC Browser for Android, Samsung Internet, and Baidu Browser; Microsoft Edge, Mozilla Firefox and Safari are still incompatible with the Web MIDI API.

3.2 The Interface

The interface of *Legato* can be roughly subdivided into 3 areas, as shown in Figure 1. The left column contains a number of panels to set the current size and color of bricks to be placed into the middle area, representing the game plate. The left column presents also controls to create associations between music parameters and brick features, and a simple media player to start, pause and rewind the computer-driven performance.

Some design choices have deep implications on the possibilities offered to users. First, the adoption of the Web MIDI API implies that the pitches covered by *Legato* are 128 (from 8.18 Hz C, corresponding to MIDI note number 0, to 13289.75 Hz G, MIDI note number 127), velocity is discretized into 128 values, and the available instruments are the 128 programs supported by the General MIDI standard. The horizontal and vertical dimensions of the game plate are fixed when a total amount of 128 elements has to be represented, in other case they can be indefinitely increased (typically, for tempo).

The width or the height of blocks, when associated with rhythm, allows 1, 2, 3 and 4-beats aggregations. Making the beat correspond, e.g., to an eighth note, the supported rhythmic values are those shown in Figure 2.

The number of elements in the palette colors has been set to 16 in order to support an extended ensemble when colors are associated with timbre, and the possibility to build a 4-measure tune in 4-beat tempo when colors are associated to rhythm (in this case, note duration is fixed to 1 beat). The behavior for pitch and dynamics is the logic consequence: covered pitches range from 261.63 Hz C (MIDI note number 60) to 622.25 Hz D$\sharp$ (MIDI note number 75); dynamics has been discretized into 16 values spanning over the full interval of MIDI velocity.

3.3 Main Changes

Legato presents many visual similarities with the previous release discussed in [Baratè et al. 2017]. Nevertheless, the prototype has

[1] Please note that, in this case, bricks sized $x \times y$ with $x > 1$ generate a so-called cluster.

[2] https://webaudio.github.io/web-midi-api/

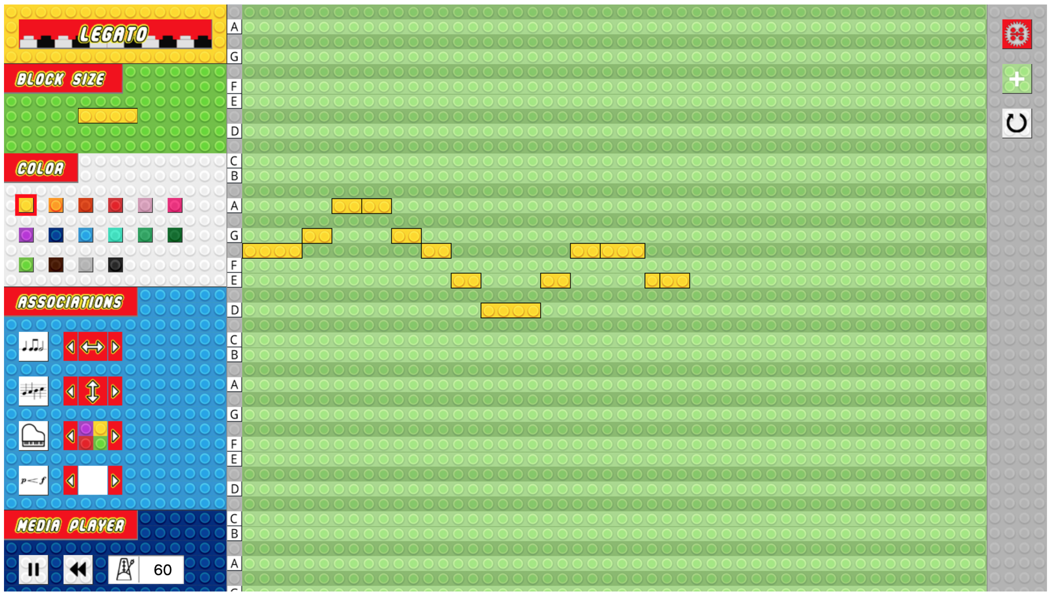

Figure 1: The interface of *Legato*. In this example, rhythm is represented on the horizontal axis, pitch on the vertical one, and timbre is associated with color. Blocks are placed so as to encode the *incipit* of the "Ode to Joy" theme by Ludwig Van Beethoven.

been refined in a number of aspects, thus extending its potential from both an expressive and an educational point of view.

First, we decided to switch the left and right columns in the Associations panel. This apparently little adjustment in the layout implies a radical change of view: in the previous version, bricks characteristics had to be associated with music parameters, so the stress was on the appearance of construction blocks; now, conversely, young learners are invited to focus on music parameters and link them to specific brick features. An important implication occurs when students have to change such associations in order to develop abstraction skills: the same music parameters, fixed in the left column, are mapped onto a new selection in the right column.

Another important change concerns a new music parameter that can be controlled through the interface, namely dynamics. In the previous experience, user could potentially associate 3 block features with 3 music parameters plus void values, thus originating a maximum of $3^3 = 27$ combinations. Actually, since rhythm and pitch had to be present in order to enable playback and rhythm could be assigned only to either the horizontal or the vertical axis, sounding combinations were only 8. The new prototype greatly extends the number of options, introducing a fourth music parameter and allowing $4^4 = 256$ combinations, where only the case with all associations set to null does not produce any sound.

Another limitation that *Legato* overcomes is the boundary constituted by the window size, often depending on screen dimensions

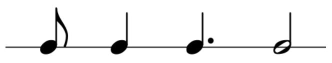

Figure 2: The rhythmic values that can be associated with brick width or height.

and resolution. In fact, when rhythm was represented on either the horizontal or vertical axis, the number of units (i.e. 1×1 blocks) displayed in the main area along that direction was the hard limit for the end of the music tune. Now the game plate can be indefinitely extended along the temporal axis (with the exception of colors), thus allowing the representation of long music pieces.

A more consistent adoption of the Web MIDI API has extended the available pitches and timbres, and introduced dynamics.

Finally, a new dialog window allows to match colors with specific values along the associated dimension. For example, when musical instruments are mapped onto colors, the window lists default program values and lets the user change them, also linking many colors to the same instrument.

4 FROM MUSICAL EXPRESSION TO COMPUTATIONAL THINKING

Legato can present multiple applications in the field of music education. Its more obvious goal is to provide young learners with an alternative form of music notation, often referred to as LEGO©Music Notation (LMN) in the scientific literature. Through *Legato*, students who have not developed the ability to read and write music in Common Western Notation (CWN) can compose or rebuild tunes by using bricks only.

The rationale behind *Legato* is two-fold. On one side, young learners can freely express their creativity by using blocks in an unmediated and intuitive way, and understand the musical meaning of their creation in a second step, after defining associations and listening to the (either human or automatic) performance of the LMN score. On the other side, learners could be asked to try to reconstruct a given music tune by using bricks and defining their own associations; in this case, the educational activity focuses on analytical skills. In both contexts, the presence of a facilitator can improve the efficacy of the experience and prevent the sense of frustration in learners. Such a gamification approach should motivate users through engagement.

Moreover, it is worth noting that *Legato* is expected to foster not only music abilities, but also the development of computational-thinking skills. The concept of *computational thinking* in education can be traced back to [Papert 1980]. Different from computer literacy, computer programming, and computer applications, such a term includes core concepts from the discipline of computer science, such as abstraction, decomposition, pattern generalization, visualization, problem-solving, and algorithmic thinking. For example, in [Furber 2012] computational thinking is defined as "the process of recognizing aspects of computation in the world that surrounds us, and applying tools and techniques from computer science to understand and reason about both natural and artificial systems and processes". Even if there is not one commonly-accepted definition of computational thinking, most researchers highlight the elements of abstraction, generalization, decomposition, algorithmic thinking, and debugging [Angeli et al. 2016].

Legato supports all these aspects in the context of a music-oriented educational experience. *Abstraction* is fostered both in the process of association of brick characteristics to music parameters and in the possibility to modify these associations in real time, e.g., turning the game plate or changing the musical meaning of already-placed blocks. *Generalization* occurs when students are able to grab a musical concept by inference from specific cases, e.g., by experimenting with bricks and listening to the audio result. *Decomposition* skills are developed by breaking music parameters apart, turning a unique music tune in a set of simpler symbols represented by blocks, and pushing this process further by decomposing single music events into pitch, duration, intensity, and timbre parameters associated to different block characteristics. *Algorithmic thinking* is encouraged both in the analytical attempt to reconstruct a known music theme with blocks and in response to changes in associations; this aspect could be fostered also by ad hoc music games conceived to be solved through algorithmic approaches. Finally, *debugging* (i.e. the detection and correction of errors) is implemented through the audio feedback and the possibility to modify brick layout accordingly, even during the performance.

5 CONCLUSIONS AND FUTURE WORK

The Web prototype has intentionally limited bricks' shapes and colors to those commonly available on the marketplace, so as to simplify the "mixed" approach implying also manipulative activities to be performed with real construction blocks. Nevertheless, it is worth citing the opportunities offered by the fabrication of bricks via 3D printing. First, this would let young learners use additional shapes and colors, or self-produce the ones missing in the available construction set. From an educational point of view, 3D printing would add a further step in the creation process: students using construction blocks to design and create music sequences would also design and create the bricks for their activity. Advantages range from the development of practical abilities (e.g., mastering a 3D CAD design tool) to the acquisition of abstraction skills in order to understand the characteristics that self-produced building blocks have to present. For a more detailed discussion about the multiple applications of 3D printing in music education, please refer to [Avanzini et al. 2019].

Concerning future work, the space of musical parameters currently managed could be expanded to integrate, e.g., articulation, expression, and agogics. Similarly, the customizable characteristics of building blocks could include odd shapes, opacity, and thickness. In particular, supporting the third dimension for blocks could be used to add one more mapping parameter, where having multiple blocks in the same place would increase the intensity of the sound. One drawback of this approach is that it would make the software tool harder to use by young learners, and it would imply a complete revision of the graphical interface.

An important aspect still missing in our experimentation concerns user acceptance and educational assessment. Once completed in all its aspects, *Legato* has to be extensively tested with young learners, so as to evaluate its educational efficacy by measuring the results achieved by the experimental and the control group. The assessment phase can involve the Web prototype alone, as well as the combined use of manipulative and computer-based activities.

REFERENCES

Charoula Angeli, Joke Voogt, Andrew Fluck, Mary Webb, Margaret Cox, Joyce Malyn-Smith, and Jason Zagami. 2016. A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society* 19, 3 (2016), 47–57.

Federico Avanzini, Adriano Baratè, and Luca Andrea Ludovico. 2019. 3D Printing in Preschool Music Education: Opportunities and Challenges. *Qwerty - Open and Interdisciplinary Journal of Technology, Culture and Education* 14, 1 Special issue – Digital Fabrication: 3D Printing in Pre-School Education (2019), 71–92.

Adriano Baratè, Mattia Giuseppe Bergomi, and Luca Andrea Ludovico. 2013. Development of Serious Games for Music Education. *Journal of e-Learning and Knowledge Society* 9, 2 (2013), 93–108.

Adriano Baratè, Luca Andrea Ludovico, and Dario Malchiodi. 2017. Fostering Computational Thinking in Primary School through a LEGO®-based Music Notation. *Procedia Computer Science. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 21st International Conference, KES 2017, 6-8 September 2017, Marseille, France* 112 (2017), 1334–1344. https://doi.org/10.1016/j.procs.2017.08.018

William Joseph Church, Tony Ford, Natasha Perova, and Chris Rogers. 2010. Physics with robotics—using LEGO MINDSTORMS in high school education. In *2010 AAAI Spring Symposium Series*.

Ethan Danahy, Eric Wang, Jay Brockman, Adam Carberry, Ben Shapiro, and Chris B Rogers. 2014. Lego-based robotics in higher education: 15 years of student creativity. *International Journal of Advanced Robotic Systems* 11, 2 (2014), 27.

Steve Furber. 2012. Making the case for digital literacy, Computer Science and Information Technology. In *Shut down or restart? The way forward for computing in UK schools*. The Royal Society, Chapter 3, 24–31.

Luca Andrea Ludovico. 2017. The Web MIDI API in On-Line Applications for Music Education. In *Proceedings of the Ninth International Conference on Mobile, Hybrid, and On-line Learning (eLmL 2017)*, Luca Andrea Ludovico and Ahmed Mohamed Fahmy Yousef (Eds.). IARIA XPS, 72–77.

Luca Andrea Ludovico, Dario Malchiodi, and Luisa Zecca. 2017. A Multimodal LEGO®-based Learning Activity Mixing Musical Notation and Computer Programming. In *MIE 2017: Proceedings of the 1st ACM SIGCHI International Workshop on Multimodal Interaction for Education*. ACM, New York, NY, 44–48. https://doi.org/10.1145/3139513.3139519

Myles McNally, Michael Goldweber, Barry Fagin, and Frank Klassner. 2006. Do lego mindstorms robots have a future in CS education?. In *ACM SIGCSE Bulletin*, Vol. 38. ACM, 61–62.

Uwe Oestermeier, Philipp Mock, Jörg Edelmann, and Peter Gerjets. 2015. LEGO music: learning composition with bricks. In *Proceedings of the 14th International Conference on Interaction Design and Children*. ACM, 283–286.

Seymour Papert. 1980. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Ana M Pinto-Llorente, Sonia Casillas Martín, Marcos Cabezas González, and Francisco José García-Peñalvo. 2016. Developing computational thinking via the visual programming tool: Lego Education WeDo. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*. ACM, 45–50.