

# MaxPiccolo: a prototype for an early computerbased music education in schools

Alessandro Anatrini

Hochschule für Musik und Theater Hamburg

al.anatrini@gmail.com

## ABSTRACT

Pedagogical approaches to computer music in primary and secondary schools are often discarded in favour of a perspective often rooted within the realm of higher education alone. Nonetheless today's increasing availability of free music pieces of software and music contents suggests the necessity of a serious discussion concerning the development of a virtuosic electroacoustic practice within music education curricula in schools. In this context a new learning environment is suggested, *MaxPiccolo*, goals and strategies are outlined. The article then presents some GUI design principles which have been taken in account in the development of this tool. A technical description of the prototype is given, followed by a comparison amid commercial pieces of software with similar features pointing out to possible future developments.

## 1. INTRODUCTION

Some of the most interesting developments in music education in the last 20 years are related to the fact that computers and mobile devices have acquired capabilities to make, record, store and spread music, while becoming available to a greater number of people. Computer science and engineering have been started to be marketed to younger children through low-cost computer hardware such as Raspberry Pi<sup>1</sup>, and electronic inventor kit such as Kano<sup>2</sup>, littleBits<sup>3</sup> and MaKey MaKey<sup>4</sup>.

Copyright: © 2017 Alessandro Anatrini. This is an open-access article distributed under the terms of the [Creative Commons Attribution License 3.0 Unported](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

<sup>1</sup> <http://www.raspberrypi.org/education/>

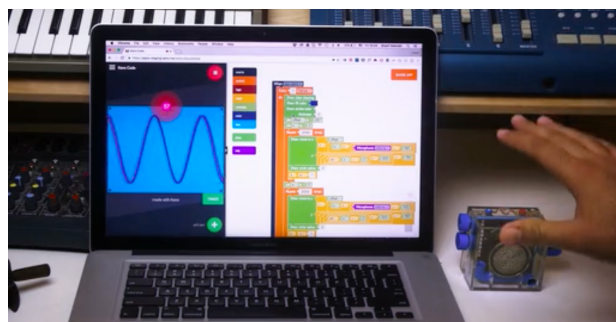
<sup>2</sup> <http://www.kickstarter.com/projects/alexklein/creative-computing-for-all>

<sup>3</sup> <http://littlebits.cc/kits/synth-kit>

<sup>4</sup> <http://www.makeymakey.com>

<sup>5</sup> Other relevant projects not included in the above mentioned article are for example: *Digitópia* [3], *Sound, Electronics and Music* [1].

<sup>6</sup> I refer to the pieces of software presented in [2] DSP, E-Lab / Live\_8, Compose with Sounds not anymore developed from 2013, 2009 and 2013 respectively. Other pieces of software such as the Max-patches developed for the Sound, Electronics and Music project are not publicly available or they have been designed for different purposes such as for example *Digitópia*, *A band is born*, *MUCCA*, *AtelierAudioIrcam*, *ML-Maquette* / *Music Lab* 2.



**Figure 1.** A sketch of Kano patching environment with a Speaker Kit on the right side.

With the advent of touch-screen technology for mobile phones and tablets, many children are becoming engaged at a very young age [1]. This leads to a question: how the pedagogical knowledge related to music should be passed down in the world of the information and communication technology (ICT)?

I have already presented three of the most relevant historic projects in this field as well as discussed why the medium to fully actualise this practice should mostly be comprised of pieces of software specifically created for educational purposes<sup>5</sup> [2]. Even if these environments are effective at enabling students to repeat trial and error by getting an immediate feedback from the system with respect to the operation, on the other hand they are no more available<sup>6</sup>, and they are all technically designed for the solo user, even though they will be used in a classroom context. In order to overtake these limitations I start from 2016 the development of the prototype *MaxPiccolo*.

*MaxPiccolo* is a Max-standalone modular learning environment addressed to students of 10-15 yrs. and designed

to assist them through each phase of the electroacoustic composition process in educational contexts.

## 2. OBJECTIVES AND STRATEGIES

The prototype is devised to be used in the perspective of a *praxial* music pedagogy approach [4] in order to assure an authentic practice<sup>7</sup> [5].

The main goal is the student's refining of the sensibility and attention for the sound and for the relation between different sounds, improving the ability to explore the morphological characters and control the musical form. Previous successfully run projects [6] demonstrate how an *extended* experience of musical creation is the way in which the sensibility and the taste for the sound may be developed. *Extended* means an experience that consists in all the phases of the creative process.

- Conception of the idea.
- Creating or finding the raw acoustic material.
- Selection, editing and treatment of the sounds.
- Formal construction according with the employed material.
- Performance<sup>8</sup>.

Users, like children, that do not have any previous specific musical and technical knowledge can encounter several difficulties in this kind of experience. In this sense the ICT becomes not only a support system in the context of the creative process but a mediator between the child and the sound as well. Working with sounds as a material - and using materials to make sounds - provides a non-preferential platform from which musical creations are made possible. The experience of sound itself - how it is perceived, understood, and talked about - can be considered without necessarily having to engage with the solfège system, rhythm analysis, and so on. Nonetheless this approach involving *epistemic* (learning) and *ludic* (playing) behaviours can promote relaxation, motivation, concentration, exploration of new skills and creativity [7].

Another crucial aspect the prototype wants to deal with is the actualisation of a collaborative scenario in the classroom. The notable advantage of a collaborative learning perspective in music education are well known and cannot be discussed here [8]; for this reason the prototype has been designed to actualise a shared composition practice and collaborative work is strongly encouraged. This should take the form of whole class collaboration, where decision on selecting samples to use, or how to sculpt a piece are made either through group discussion and/or turn-talking. Pupils who are receiving music lessons should be encouraged to bring their own instruments to the class so that they can employ and expand their skills<sup>9</sup>. Small group collaborations can also enable instrumentalists to work with the new-appointed *live electronic per-*

*formers* who would manipulate sounds made by their classmates through the prototype that could be operated swiftly using computer keyboard and trackpad.

In order to fit the needs of the students and actualise these scenarios, with their premises and goals, becomes crucial to be aware of the fact that the design principles of graphical user interface (GUI) created for adults cannot be easily scaled down and applied to children's products. A child is not a miniature adult: needs, skills and expectations are drastically different and goals cannot necessarily be met by adult tools [9].

## 3. DESIGNING GUI FOR CHILDREN

Chiasson and Gutwin [10] have already extensively debated the argument therefore the scope of this chapter will be limited to the key statements concerning the most significative aspects that I had to deal with in order to design an effective software targeted to children<sup>10</sup> [11]. Concerning our purposes design principles can be divided in three main topics.

### 3.1. Feedback and guidance

Child expects an immediate feedback, showing that his/her actions have had some effect, otherwise he/she will repeat actions until some outcome is perceived, possibly causing a chain of unexpected and unwanted events.

Being able to use a system without instruction or read a manual is crucial as well; interfaces should be intuitive as much as possible, ideally providing the current state of the system, whether it is busy processing or waiting for input from the user [9].

### 3.2. Mental development

Children's GUI need to take into account the fact that the child may not yet understand abstract concepts; young learners have difficulties approaching abstract concepts, and may not have the in-depth content knowledge required for navigating and categorising complex interfaces. For these reasons this kind of GUIs should not make use of extensive menus and submenus [12].

The usual approach of a child is trial-and-error: once they find a method that works fine, they are unlikely to look for a more efficient strategy or for advanced options. Furthermore their actions should map directly to the actions on the screen, because children are accustomed to direct manipulation interfaces [13].

<sup>7</sup> This argument has already been discussed by the author in [2] and more detailed in [4].

<sup>8</sup> I deliberately discard the graphical representation of the piece since this one is not one of the goals to be actualised within the prototype.

<sup>9</sup> From an experimental perspective they are going to be introduced to extended techniques, improvisation and electronic augmentation.

<sup>10</sup> The definition of the word *child* is that one given by Piaget [10]. Especially he refers to the different stages of child's cognitive development: 0-2 yrs. sensory-motor intelligence, 2-4 yrs. preconception thought, 4-7 yrs. intuitive thought, 7-11 yrs. concrete operations, 11-15 yrs. formal operation.

### 3.3. Motivation and engagement

Since children use ICT for educational and entertainment purposes - unlike adults which usually have a task in mind and the ICT becomes a tool to complete this task - in order to be effective, a software needs to keep their interest and attention focused. A way to address the engagement is to give children the power to make decisions by allowing them to learn about consequences of their actions. Researchers [14] have found that young students (9-14 yrs.) request an environment that let them be in control, set and achieve goals, and be part of the action. Therefore the activities that can be actualised within the software should be inherently interesting and challenging so children will want to do them for their own sake.

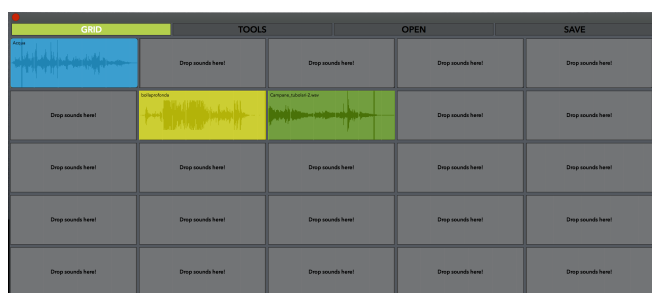
## 4. SYSTEM OVERVIEW

The prototype has started to be developed as Max<sup>11</sup> patch during the spring 2016. The GUI has been completely redesigned in autumn 2016. Currently it is running debug and compatibility tests in order to be available as stand-alone application from early spring 2017. *MaxPiccolo* is a modular learning and editing environment, one of its strong points is the possibility to switch between two different edit modes: the *grid* and the *tools*. This two modes allow a raw representation of the macro-form of the employed sonic material, as well as the possibility to modify the morphological characteristics of a sample at the same time, even during the performance.

### 4.1. The Grid

The software opens automatically in the *grid* mode (Fig. 2). Basically it acts as a sequencer and it consists of a grid with 25 buttons (5x5). The following actions can be performed on each single button, each action has a graphical feedback:

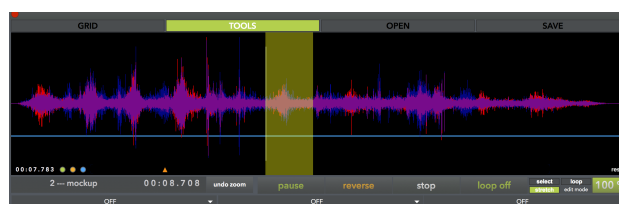
- Get new samples by dragging and dropping audio files<sup>12</sup>. Once the sample is displayed in the button, it is automatically copied in the "samples" folder and normalised as well. The button becomes yellow displaying the sample's waveform.



**Figure 2.** The *grid* view: first sample is paused, the yellow one is stopped, the green is playing, the scrollbar is displayed. Other slots are empty.

- Playback and pause (**click**). Each sample is going to play as loop by default. The colour of the button swirls constantly among an interpolation of green and red corresponding to the gain level of the sample. A scrollbar representing the playback position is displayed over the waveform.
- Access the *tools* mode through the button where the sample is in (**cmd** or **ctrl** + **click**). The button turns to black and the display switches to *tools* mode. After to have switch to *tools* mode, the other samples that are playing simultaneously are filtered in order to focus on the sample that is going to be edited. This feature can be disabled during the performance.
- Stop the sample (**double click**). The button becomes yellow.
- Create new samples by recording sound in real time (**R** + **click**). The colour swirls like for the playback mode, at the same time the waveform is designed. Once the recording is ended, the sample is going to be automatically stored<sup>13</sup> in the "samples" folder and normalised as well.
- Delete the sample from the button (**canc** + **click**). The button is displayed as empty (grey colour), the waveform is deleted, "Drop sounds here!" appears.

Due to the CPU limitation of the computers that are usually available in the schools the possibility to playback audio files from the *grid* is limited to five samples simultaneously. Acting as a sequencer, the *grid* is designed to represent the time-line of the piece during the performance. It is up to the teacher and the students, according with the nature of the project that is going to be realised, to decide how to represent it: the samples can be disposed vertically one after each other, or horizontally in a Live-like style<sup>14</sup>.



**Figure 3.** A screenshot of the *tools* view: a portion of the waveform is selected (yellow band) therefore the edit operation will affect only this part of the sample. On the bottom left of the waveform the pitching commands are shown and active. For this reason the blue line over the waveform as well as the cursor close to the commands are displayed. Currently we are pitching and slowing

<sup>11</sup> Max is a visual programming language for music and multimedia developed by Cycling'74, <http://cycling74.com>.

<sup>12</sup> Uncompressed formats such as (.aiff, .wav) and mp3 are supported.

<sup>13</sup> The recording is stored with a generated name that looks like "My\_Recording\_no.x".

<sup>14</sup> The reference is to Ableton Live *session view* <http://www.ableton.com/en/manual/session-view/>. Here the vertical axis corresponds to all the sample that have to be triggered simultaneously whereas the horizontal axis represents the different sections (scenes) of the piece and so ideally the temporal progression.

down (blue line and the cursor are not centred). Below them we can see the playback commands.

At the very bottom of the picture there are three menus from which is possible to select the devices. These letter ones are going to be displayed below each menu.

## 4.2. The Tools

If the *grid* view ideally relates to the macro-formal processes, the *tools* mode (Fig. 3) represents the “detail” and the way through which each sample can be edited according with the different available techniques. *Tools* can be accessed through the buttons (**cmd** or **ctrl** + **click**) and it consists of two main parts.

- The waveform display, containing also the commands to select how to pitch the sample.
- An editing area on which is possible to insert through the menus up to three different devices to treat the sound.

### 4.2.1. Waveform display

The waveform is shown as an over-imposition of two waveforms of different colours one for each stereo channel. On the bottom left the duration of the currently selected portion of the sample is displayed. Close to it there are three buttons: green, orange and blue. Through them it is possible to pitch the sample according with two different techniques.

- **Green**: it allows the manipulation of pitch and duration simultaneously by moving up or down the green vertical lines that appears on the display. It acts also as the main switch in order to enable both pitching techniques: if the button is unselected it is impossible to select the other two buttons as well.
- **Orange and blue**: they allow to decouple pitch from time so each can be adjusted independently. By selecting the orange button a horizontal cursor appears on the bottom of the waveform: it can be used to change the speed of the playback just by moving it left or right. If the cursor is centred the speed is the original one of the sample. By selecting the blue button a blue line pops up: through it is possible to modify the pitch of the selected portion of the sample without affecting its speed.

Below the waveform a display shows the number of the button of the grid on which we are acting on, the name of the sample and the playback position. It is possible to zoom on a portion of the waveform by clicking on it at least 2 sec. and undo the zoom through the appropriate button. Close to it there are the “classical” playback

commands: play/pause, reverse/pause, stop, loop on/off and a small table menu through which is possible to select the way the mouse interacts on the waveform. The percentage number sets the amount of the waveform that is displayed and played back. By default is 100% that means that the sample is normalised at 0 db, therefore by reducing it the gain is going to be reduced as well.

### 4.2.2. Editing area

Though the menus below the playback section is possible to select up to three devices through which the sample, or a portion of it, is going to be processed. Each device can be used only once for each sample: if a device has already been used it cannot be selected anymore from the menus. The devices are linked to each other, that means the output of a device is routed into the input of the next one and so on. Therefore the order in which they are used have to be taken carefully in consideration. Currently six devices are available<sup>15</sup>:

- **Delay**: up to 16 voices delay with feedback and a pan control over each voice.
- **Echo**: based on the delay technique with control over the amount of echo in ms. and feedback for left/right channel. Rate and depth of the echo can be set as well.
- **Filter**: a 256 bands filter, based on the FFT analysis and resynthesis. Each band can also be set and triggered randomly.
- **Freeze**: a tool for freezing and explore the incoming sound, based on the granular synthesis.
- **Harmoniser**: a 16 voices harmoniser with gain and pan control over each voice.
- **Reverb**: to control the reverb effect.

Furthermore each device has a dry/wet control, in order to control how much of the original sound is going to be processed.

## 4.3. Open/Save

It is possible to use the *save* command to save the current project. A json<sup>16</sup> file is created and stored according with the location that has been chosen and it can be recalled whenever necessary, through the *open* command. The option to export the output of a sample or of the entire project as an audio file is at the moment not implemented.

## 5. COMPARISON

At this point is quite obvious to ask ourself: was it really

<sup>15</sup> Due to the length limitation of this article a more detailed explanation of each single device cannot be discussed here. Currently six devices are implemented, more devices are already planned to be developed after the prototype will be tested in a classroom context according with the students’ needs.

<sup>16</sup> Acronym for JavaScript Object Notation is an open standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs.

necessary to create a prototype of a new digital audio workstation? Why the available pieces of software with similar functionalities cannot fit educational needs?

In the Tab. 1 I provide a comparison of some significant characteristics of two commercial pieces of software<sup>17</sup> that entail a similar conceptual approach to *MaxPiccolo*: they allow a control over the macro-formal processes and over the detail of the sample editing at the same time. Therefore they can be useful to compose, perform, and improvise even at one time.

At a first glance *MaxPiccolo* appears like the “little brother” of Live<sup>18</sup> and Bitwig<sup>19</sup>, with a very limited palette of available features, and indeed it is what it seems!

The limited features are effective not only to avoid to leave the students free to find their own way through the copious possibilities enabled with the software package [2], but they also allow to educators to teach how the software works without having to deal with a good knowledge of audio processes or the study of a manual to fully understand the possibilities of the program.

From the point of view of the user interface design these limited features as well as the simplified mouse interactions are useful to meet the young students needs for at least three reasons.

- The tasks to be accomplished may be forgotten, if they require several steps [14].
- Young students may lack the fine motor control needed to target small items such as icons and buttons used in traditional interfaces [9]. The provided on-screen items within the prototype are large enough to compensate for some inaccuracy in targeting. Buttons are spaced on the screen to minimise the chance that children accidentally press the wrong one.
- Multiple selection tasks are avoided: children have difficult to use traditional technique such as drag-selection. [12].

|                             | <b>MaxPiccolo</b>                       | <b>Ableton</b><br>(Intro version) | <b>Bitwig</b>                     |
|-----------------------------|---|-----------------------------------|-----------------------------------|
| <b>Target</b>               | children and educators                  | professional and higher education | professional and higher education |
| <b>Usability</b>            | learning by doing                       | manual and/or tutorial video      | manual and/or tutorial video      |
| <b>Devices</b>              | 6 audio-effect                          | 3 instruments<br>27 audio-effect  | 11 instruments<br>28 audio-effect |
| <b>Smart-device support</b> | mirroring even on browsers through Mira | mirroring                         | mirroring                         |
| <b>Price</b>                | free                                    | from 79 €                         | from 299 €                        |

**Table 1.** Comparison of some significant features amongst *MaxPiccolo*, *Ableton* and *Bitwig*.

## 6. CONCLUSION AND FUTURE WORK

The strength of *MaxPiccolo* resides in its limitations. It must be clear that the aim of the prototype is not to create a free digital audio work station that, even if targeted to young students, could be seen as an alternative to its commercial competitors, but rather to study the best way to actualise an effective electroacoustic music curriculum that approximates authentic music cultures within primary and secondary school.

From this point of view the planned development of an effective set of educational scenarios becomes crucial, to be actualised within primary and secondary schools, consisting of detailed pedagogical objectives, strategies and activities, the outcomes of which can be scientifically verified.

The efforts in the development of the prototype goes in the direction of allowing connections with pieces of hardware. This is the sense of the ongoing implementation of MIDI and OSC messages in order to enable the support for the widest range of controllers and midi-instruments. A further way to increase the pedagogical potential of the software is the implementation of the possibility of mirroring it easily it on browsers through Mira<sup>20</sup>, in order to be able to use the prototype on tablets.

## 7. REFERENCES

1. L. Hayes, “Sound, Electronics and Music: an evaluation of early embodied education”, in Proc. International Computer Music Conference (ICMC2016), Utrecht, 2016, pp.388-393.
2. A. Anatrini, “The state of the art on the educational software tools for electroacoustic composition”, in Proc. Int. Conf. Sound and Music Computing (SMC2016), Hamburg, 2016, pp. 15-20. Available at <http://smcnetwork.org/resources/smc2016>.
3. R. Penha, P.M. Rodrigues, F. Gouyon, L.G. Martins, C. Guedes, A. Barbosa, “Studio Report: Digitópia at Casa da Música”. Available at <http://quod.lib.umich.edu/i/icmc/bbp2372.2008.103/1>.
4. D. J. Elliott, “Music matters: A new philosophy of music education”, Oxford University Press, 1995.
5. J. Martin, “Electroacoustic music in middle and secondary education: Some concerns regarding curriculum development”, in Proc. Int. Conf. Teaching electroacoustic music: Tools, Analysis, Composition (EMS2010), Shanghai, 2010, pp.1-12.
6. S. Luca, “Esplorazione e creazione musicale con le nuove tecnologie” in *Musica Domani* n. 147, 2008, pp. 12-19.

<sup>17</sup> I decided to compare the prototype with commercial pieces of software because as far as I know the ones specifically targeted to children entail a different conceptual approach.

<sup>18</sup> <http://www.ableton.com>

<sup>19</sup> <http://www.bitwig.com/en/home.html>

<sup>20</sup> Mira is an app that automatically connects to Max and mirrors your interface, <http://cycling74.com/products/mira/>

7. M. Prensky, "Digital Game-Based Learning", New York, McGraw-Hill, 2001.
8. H. Gaunt, H. Westerlund, "Collaborative Learning in Higher Music Education", London, Routledge, 2013.
9. L. Hanna, K. Riden, M. Czerwinski, K.J. Alexander, "The role of usability Research in Designing Children's Computer Products", in A. Druin (ed.) "The design of children's technology", San Francisco, Morgan Kaufmann, 1999, pp. 4-26.
10. S. Chiasson, C. Gutwin, "Design Principles for Children's Software", in Technical Report HCI-TR-05-02, Computer Science Department, University of Saskatchewan, 2005. Available at [http://hci.usask.ca/publications/2005/HCI\\_TR\\_2005\\_02\\_Design.pdf](http://hci.usask.ca/publications/2005/HCI_TR_2005_02_Design.pdf)
11. J. Piaget, "Dal bambino all'adolescente. La costruzione del pensiero", Firenze, La Nuova Italia, 1969.
12. A. Druin, B. Bederson, J.P. Hourcade, L. Sherman, G. Reville, M. Platner, S. Weng, "Designing a Digital Library for Young Children", in Proc. Int. Conf. ACM JCDL 2001, Roanoke, 2001, pp. 398-405.
13. M.P. Johnson, A. Wilson, B. Blumberg, C. Kline, A. Bobick, "A Sympathetic Interfaces: Using a Plush Toy to Direct Synthetic Characters", in Proc. Int. Conf. ACH CHI 1999, Pittsburgh, 1999, pp. 152-158.
14. N. Said, "An Engaging Multimedia Design Model", in Proc. Int. Conf. ACM IDC 2004, Baltimore, 2004, pp. 169-172.