

A Design Environment for Musical Instruments

Samuel H. Thibault

November 29, 2000

Contents

1	Introduction	3
2	Educational Frameworks	3
2.1	Papert, Piaget, and Constructionism	4
2.2	Dewey and Experience	4
2.3	Vygotsky and Social Development	4
3	Background Work	5
3.1	Crickets	5
3.2	Cricket Add-ons	5
3.3	The MIDI Bus Device	6
3.4	MicroWorlds and LogoBlocks	6
3.5	MusicLogo	6
3.6	The First LEGO Instruments	7
4	Project Description	7
4.1	The LEGO Instrument Challenge	7
4.2	Hardware Objectives	8
4.2.1	MIDI Bus Device Improvements	8
4.2.2	Pocket PC development	8
4.3	Software Objectives	9
4.3.1	A Melody Editor	9
4.3.2	An Instrument Creator	10
4.3.3	An Expression Generator	11

4.3.4	A Standard Logo Tool	11
4.4	A Tool for Learning	11
4.5	A Cultural Tool	11
4.6	Example Projects	12
5	Projected Timeline	12
6	Possible Extensions	13
6.1	LogoBlocks embedding	13
6.2	Chaotic Music	13
6.3	A LEGO Orchestra Recorder	13

“Better learning will not come from finding better ways for the teacher to instruct, but from giving the learner better opportunities to construct”
-Seymour Papert

1 Introduction

Simple building blocks provide more than just hours of play time for a child – they provide hours of learning too. Through trial and error, a child will learn methods to keep a tall tower of blocks standing while noticing how a poorly stacked pile collapses. Utilizing a more advanced set of building tools will interact with the simple mechanics of gears and pulleys. Electronic components will offer motors and sensors. Each component offers some new idea that the child can learn through their building experience. Each step in expanding this building environment provides better learning opportunities.

One such building environment available at the MIT Media Lab combines LEGO building blocks with electronic and programmable components. These components can be combined in many ways to fashion cars, robots, kinetic sculptures, and other interactive contraptions. There are other things children would like to build like musical instruments. A musical instrument, however, is very difficult to construct with the current building tools available. By expanding the available tools with hardware improvements and new programming environments, musical instruments ranging from traditional to exotic could become simple creations. Moreover, this new capacity would be a useful educational tool to enable children to learn about music and interface design principles.

2 Educational Frameworks

When developing a tool for education, it is important to examine the framework in which the tool will be effective. The environment at one school can differ greatly from the environment at another.

Thus, rather than focusing directly on specific schools, it will be more effective to discuss several powerful models of education.

2.1 Papert, Piaget, and Constructionism

In the constructionist model children use tools to build objects, then reflect on that process to come in contact with new ideas. This model has significant ties to the empirical work done by Jean Piaget (French) many years ago, and ties more closely with the more recent work done by Seymour Papert at the MIT Media Laboratory. The musical instrument construction tool proposed here can be highly effective in a constructionist environment. It allows a domain where learners can explore musical properties, receive guidance as necessary, and use their imagination and ideas to create new instruments. [5, 9, 10, 7]

2.2 Dewey and Experience

John Dewey, who directed a great deal of philosophical thought on education, proposed another related model for education. His ideas centered around the use of experience as a tool for building interest and foundations for additional learning. Dewey discusses tools in a non-specific way, not targeting them directly. Dewey suggests that a tool which provides a valuable and enjoyable educational experience will allow more sophisticated lessons to be learned. By using this musical instrument tool in such an environment, educators can build learners experience with music, design, and engineering to develop a broad foundation for more specific learning in those areas.[4]

2.3 Vygotsky and Social Development

Learning through social interaction was the main idea advocated by Lev Semenovich Vygotsky (Russian). He proposed that during social interaction between a teacher and a learner, the learner

can develop more complex ideas and associate the social interaction with personal development. Vygotsky referred to this concept as the “zone of proximal development”, where the learner is actively involved in the development of a project with the help of a more capable teacher or peer. In this model, the musical instrument tool would need to be combined with highly involved teachers or powerful instructive tools within the created software.[11]

3 Background Work

Many tools have already been developed for this LEGO building environment which serves as the starting point for the musical instrument tool. The foundation is centered around a programmable device nicknamed a “cricket”. Other construction tools include sensors for the crickets, expansion devices that attach to the crickets, and software environments to aid in programming the crickets.

3.1 Crickets

Each cricket is a small computer only slightly larger than, and powered by, a 9-volt battery. A single cricket is capable of powering two LEGO motors and monitoring two sensors. Crickets can also communicate with other crickets or a computer interface using infrared light. [8] A dialect of the Logo programming language is used to program the crickets. The language includes procedure calls, simple control structures, and standard numeric operations. Logo also has functions for motors, sensors, timers, and playing tones.

3.2 Cricket Add-ons

There have been several expansions to the basic cricket as well. The selection of sensors, once including only light and touch sensors, now provides sensors to detect temperature, reflectance, acceleration, and more. In addition, expansion devices are available which can be plugged into the

crickets bus. One such device provides more inputs for sensors, allowing four additional sensors beyond the original two.

3.3 The MIDI Bus Device

One recently created bus device that is especially useful for a cricket based musical instrument is the Musical Instrument Digital Interface (MIDI) bus device. With a set of primitive commands the cricket can play musical notes using standard MIDI voices and channels, just like a musical synthesizer. MIDI commands are more complicated than other basic Logo commands, however, which makes MIDI programs much more difficult to write.

3.4 MicroWorlds and LogoBlocks

Some software tools have been written to aid in Logo programming. The MicroWorlds environment offers a command center interface for running code one instruction at a time. LogoBlocks is a visual programming environment in which Logo commands and control structures are represented by graphical blocks. A sequence of instructions is created by “snapping” the blocks together. [2]

3.5 MusicLogo

One particular musical project developed by Jeanne Bamberger utilized a similar block style interface like LogoBlocks to teach music. The software tool, called MusicLogo, allowed learners to combine short musical phrase blocks to create larger musical phrases. In this process, students developed intuitions about mathematical relations to music.[1]

3.6 The First LEGO Instruments

The idea of constructing a musical instrument from cricket components is not new. In 1996, an undergraduate researcher (UROP) constructed a simple musical instrument. [6] The instrument used a simplified set of MIDI commands. Due to the interface, the LEGO instrument was also immobile.

4 Project Description

The goal of this project is to improve existing tools so that the creation of a musical instrument from LEGO and cricket components is easy. Furthermore, the instruments should be capable of performing at the speeds of a standard musical instrument. When a note on a piano is pressed, the is heard immediately; LEGO instruments should perform just as well. Towards the end of this project, the new instrument creation tools will be presented to middle-school children for additional feedback and revisions.

4.1 The LEGO Instrument Challenge

While all the basic building blocks are in place to create a musical instrument, they are still exceedingly difficult to build. The MIDI commands available to Logo are very complicated. Every note must be turned on and off individually, specifically noting one of the sixteen available MIDI channels. Special modifications to notes such as accents also require special commands. Even worse, the graphical environments like LogoBlocks do not have any standard blocks to program MIDI commands. Since even a simple musical device requires many lines of code, writing the Logo code by hand is difficult and error-prone.

The performance of the hardware is an additional problem. The current cricket devices are too

slow to handle all the processing of a musical instrument. With even a small number of sensors, the cricket will be slow in playing notes or completely miss a pressed key. Because music is so inherently connected to timing, faster hardware performance is also a must.

4.2 Hardware Objectives

One significant step in creating the musical instrument environment is improving the current hardware. The current cricket does not have the processing power to handle the timing and computation intensive demands of a large musical environment. Therefore, there will be two separate platforms for using the instrument tool. The original cricket with improved bus devices for simple instruments, and a version of the software compatible to work with the Pocket PC.

4.2.1 MIDI Bus Device Improvements

Faster and more powerful bus devices could enable simple instruments to perform effectively with the basic cricket. Currently each instruction is a very simple primitive. For example each note must be turned on with one command and subsequently turned off by a second command. The total number of commands the cricket itself needs to run could be reduced by changing the MIDI device to accept higher level commands. By sending a command that turned a note on for a particular amount of time, the cricket would only need to send one instruction. Meanwhile, the MIDI device could handle both the note-on and note-off commands to complete the operation.

4.2.2 Pocket PC development

The Pocket PC provides a very powerful hardware device, nearly the power of a laptop, in a device that is no larger than the user's hand. By writing software for this device, the Pocket PC itself could be used as the cricket, providing immense processing power and other media functions

such as playback on its internal speaker. Moreover, the development environment for the musical instrument tool can also be run on the Pocket PC. Some hardware will need to be created to adapt the sensor ports to the inputs of the Pocket PC. However, this should not be too difficult given the standard port interfaces provided by the device.

4.3 Software Objectives

Although the above work does need to be made to the hardware, the major work will involve creating a new environment for programming the musical instruments in a simple and straightforward manner. The software will provide tools for creating songs for the cricket to play, selecting from the full range of MIDI sounds and commands, assigning sensors with specific musical actions, and allowing for all other standard cricket commands for motors, infrared, etc.

4.3.1 A Melody Editor

One basic element of the software environment will be a melody editor. This tool will provide a graphical interface where music will be displayed in a standard notational format. Each instrument selected by the user will appear on its own staff. For each measure, the user will be able to select time signatures and key signatures. For inserting notes, the user can select from a palette of the types of notes and simply click on the staff to add the note. Just as easily notes could be deleted or moved. In addition, any variety of accent or dynamic marking will be available from the palette. For very exact note attacks the user can modify every aspect of the MIDI command for the notes. The program will also provide a playback feature so the user can listen to the score. This interface should make editing a melody straightforward and easy to accomplish.

4.3.2 An Instrument Creator

The main component of the software environment will be an instrument creation tool. Within this environment the user can begin by describing what devices are connected to their cricket. For example, the user could specify that an additional sensor input is attached. Next the user would declare each of the sensors that is attached the cricket. The sensors could be any combination of the available sensors for the cricket. Next, actions can be assigned to each of the sensors. As a simple example, the user could specify that whenever a touch sensor is pressed, the instrument should play the note middle-C. The number of possible actions is much broader than just playing a note when a sensor is pressed. For a ranged sensor, the user will be able to specify a specific sub-range (or several) to connect with an action. Moreover, the user can select the action to trigger inside the range, outside the range, or on a transition between the ranges. In all, there are six specific actions which can then be performed:

1. Play a note: play one or more notes on a specific MIDI voice.
2. Play a melody: play a melody previously created in the melody editor.
3. Play a selected root tone: the user will choose a single pitch that is the “root” tone of the instrument. That note is played during this action.
4. Modify the root tone: temporarily raise or lower the currently played root tone.
5. Sustain current notes: sustain notes like the sustain pedal on a piano.
6. Run other cricket code: perform any other cricket actions, like turning on motors.

Once all the actions have been assigned, the instrument creator will automatically generate all the code necessary to operate the instrument. This step hides all the hard-to-use MIDI commands from the user.

4.3.3 An Expression Generator

The last major tool contained in the software will be a tool for designing musical expressions. The basic interface appears to mechanical in many ways. If using a touch sensor, sounds are simply on or off. The expression tool will allow a series of minor variations in sensor readings to map to more delicate and human musical gestures. These tools will allow the instruments to have more of the feel and capabilities of traditional instruments.

4.3.4 A Standard Logo Tool

As evident by the sixth possible action in the instrument creator, the software interface will be capable of much more than just programming a musical instrument. The interface will allow any more “normal” uses for the cricket devices, like cars or robots, to be programmed as well. Apart from the cricket code assigned for a sensor action, there will also be a global cricket code region where larger procedures used throughout the cricket program may be written.

4.4 A Tool for Learning

Above all, the LEGO instrument creation environment is a tool for children to learn about design and have fun. To ensure that this project reaches that goal, the final stage will give classrooms at some selected schools the chance to utilize the new tools. By allowing children to interact with the instrument creator, we will gain valuable information about the effectiveness of the tools and ways to improve their performance and functionality.

4.5 A Cultural Tool

The instrument development tool is also a way of providing cultural development. Several students with their own newly designed instruments can create their own orchestra. In this environment

they can explore musical performance and search for the musical sound they desire. This goal has been a common one for instrument builders, who struggled for centuries to create the sound of the violin we hear today.[3]

4.6 Example Projects

There are several instruments which would be nice to build as examples of what can be constructed with this tool. The first is a basic piano. We have built a simple version of this from legos which we would like to enhance. Other instruments to cover the range of traditional instruments include a string instrument, a wind instrument, and a drum set. In addition, the construction of several non-traditional or abstract instruments is desired.

5 Projected Timeline

Below is a basic list of what is expected to be completed throughout the remainder of the project:

- January: Completion of PocketPC environment.
- February: Development of example instruments, PocketPC debug. Contact schools to arrange sessions for children to interact with tools.
- March: Travel to schools for feedback from both teachers and students. Heavy work on thesis draft.
- April: Implement improvements based on feedback. Complete development of thesis draft.
- May: Completion of final thesis draft and approval.

6 Possible Extensions

If the project moves quickly enough it may be possible to incorporate some additional elements within the software environment. These additional features, however, will not be a mandatory condition on the completion of this thesis work.

6.1 LogoBlocks embedding

Currently, other Logo programming environments such as LogoBlocks do not provide an interface for utilizing the new MIDI bus device. It would be nice to embed features of the new instrument creation environment into LogoBlocks so that it can be used for creating music.

The instrument creator and LogoBlocks would be a great pair of tools since each one approaches cricket devices in a different way. The LogoBlocks environment controls crickets in a much more sequential manner. One command performed, then another. The instrument creator, on the other hand, allows things to occur in a very random manner. Actions will be programmed as responses to the various sensors. Both of these approaches are useful.

6.2 Chaotic Music

Other musical teaching tools could also be very useful. One possibility is a mapping tool for chaotic music.

6.3 A LEGO Orchestra Recorder

Another possibility is a tool to “record” several people playing their LEGO instruments. This collaborative tool could allow the recording to be generated into a new MIDI file, or combined with other media in exciting new ways.

References

- [1] Jeanne Bamberger. Musiclogo: A laboratory for developing intuitive music skills. *MIT Department of Music*, 1989.
- [2] Andy Begel. Logoblocks a graphical programming language for interacting with the world. *MIT Media Laboratory*, 1996.
- [3] Ernest E. Boesch. The sound of the violin. *Mind, Culture, and Activity*, 1997.
- [4] John Dewey. *Experience and Education*. Simon and Schuster, New York, NY, USA, 1997.
- [5] Aaron Falbel. Constructionism: Tools to build (and think) with. *LEGO DACTA*, 1995.
- [6] Carol E. Foltz. Learning through design of programmable musical instruments. *MIT Media Laboratory*, 1996.
- [7] Howard E. Gruber and J. Jaques Voneche eds. *The Essential Piaget*. Jason Aronson, Inc., Northvale, NJ, USA, 1995.
- [8] Fred Martin, Bakhtiar Mikhak, and Brian Silverman. Metacricquet: A designer's kit for making computational devices. *IBM System Journal*, 2000.
- [9] Seymour Papert. *Mindstorms*. Basic Books, Inc., New York, NY, USA, 1980.
- [10] Seymour Papert. What's the big idea? steps toward a pedagogy of idea power. *IBM Systems Journal*, 2000.
- [11] Jame V. Wertsch. *Vygotsky and the Social Formation of Mind*. Harvard University Press, Cambridge, MA, USA, 1985.