Pd Poems and Teaching Tools

Jiffer Harriman
ATLAS Research Institute
University of Colorado
1125 18th St. #223
Boulder, Colorado 80309
harrimj@colorado.edu

ABSTRACT

Music offers an intriguing context to engage children in electronics, programming and more. Over the last year we been developing a hardware and software toolkit for music called modular-muse. Here we describe the design and goals for these tools and how they have been used in different settings to introduce children to concepts of interaction design for music and sound design. Two exploratory studies which used modular-muse are described here with different approaches; a two day build your own instrument workshop where participants learned how to use both hardware and software concurrently to control synthesized sounds and trigger solenoids, and a middle school music classroom where the focus was only on programming for sound synthesis using the modular-muse Pd library. During the second study, a project called Pd Poems, a teaching progression emerged we call *Build-Play-Share-Focus* which is also described.

Keywords

NIME, toolkit, modular instrument, laptop orchestra, teaching, learning

ACM Classification

C.0 [Computer Systems Organization] General --- Hardware/Software interfaces K.3.2 [COMPUTERS AND EDUCATION] Computer and Information Science Education --- Computer Science Education

1. INTRODUCTION

Music offers an intriguing context to engage children in electronics, programming and more. Modular-muse is an interaction design toolkit designed to enable novices to engage in *Digital Music Instrument* (DMI) design. The toolkit enables children to get hands-on experience with electronic sensors, interaction design and sound design. The toolkit consists of a modular hardware interface and software library for Pure Data (Pd). Both components of modular-muse (hardware and software) aim to lower the barrier to entry to building DMIs making it accessible and fun for children.

After reviewing related work the paper describes two studies using modular-muse, first a two day workshop where children used the modular-muse toolkit to map sensors to control audio parameters and programmed algorithmic beats from Pd to trigger solenoids. Second is a study based in a middle school music classroom where students focused on programming sound design using the modular-muse library for Pd. Through the experience of a project called *Pd Poems* we developed a teaching progression we call *Build-Play-Share-Focus*.

2. RELATED WORK

Making and tinkering activities [5] have become integral to after school programs and, increasingly, in the classroom to teach kids programming and electronics. It is easy to see why. Imagining, designing and creating a technologically rich artifact engages students in engineering design in meaningful ways. Popular activities in these settings include robotics, video-game design and art and crafts projects embedded with sensors, motors and LEDs [11][7][8][10].

These projects are enabled by tools that scaffold the student work, enabling them to do more than they could otherwise do. For instance, Scratch, a popular children's programming language uses graphical blocks that snap into other only in legal combinations. Scratch features a small set of music functions to capitalize on children's interest in music. Relatively few studies have engaged children with hardware and programming through building DMIs.

Using high level GUI abstractions to engage primary school children in instrument design Trappe [12] identifies "musical playfulness" as a key to success concluding, music controller construction is a context that nurtures self-motivated creation, exploration and play. Sawyer et al [13] describe a workshop in which children use hardware sensors and a graphical programming environment to create musical instruments. A related study by Bukvic et al [3] describes the use of "granularity" as a means to provide multiple points of entry, and in enabling an adaptive tool (Pd-L2Ork, also used by Sawyer) that can match the educational model and skill level of the audience.

Blikstein and Sipitakiat found middle/high school and undergrad students had trouble with differences between analog, digital and PWM pins, the concept of pull-up resistors and the architecture of solderless breadboards [1]. Hardware abstractions have been a fruitful means of engaging children in hardware design by removing the complexities. Examples include the GoGo Board, PicoCricket and LilyPad [1][11][2].

Phidgets and the Grove System represent general purpose interaction design toolkits that can be used for building DMIs [4][9]. While these kits have an extensive offering of modules there are some notable exclusions that I consider important for tinkering with music including membrane sensors and a solenoid drivers[1]. Music toolkits such as the I-CubeX[2] and Livid Instrument Builder[3] are designed to create custom musical controllers. These toolkits focus on input options but their closed nature limit options to create a transition for students to more flexible hardware platforms such as Arduino.

3. MODULAR-MUSE TOOLKIT

In order to create a toolkit for education that would enable a transition to using Arduino and Pd, modular-muse[4] is made to be transparent and to promote hackability. The software library is built with native Pd-Extended objects and can be opened to explore how they were built or make changes. Similarly, the hardware modules use exposed components inviting inspection.

[1] Both kits offer motor drivers but require either I2C or a separate USB connection complicating things for a novice
[2] http://infusionsystems.com/
[3] http://shop.lividinstruments.com/builder-diy/
[4] Please see www.modular-muse.com for more on this project

The software library includes help files for each object and example patches.

3.1 Interface Board

The modular-muse hardware consists of an Arduino based interface board that connects to the I/O modules (figure 1). Modules are connected with 3.5mm stereo jacks that use the three wires of a stereo patch cables to interface to the I/O modules. The three wire cable delivers power, ground and a signal wire for all input and output modules. While this wiring scheme has limitations (e.g. no I2C support) the simplicity is ideal for educational settings. The interface board connects Arduino pins to the cable jacks (Figure 1).

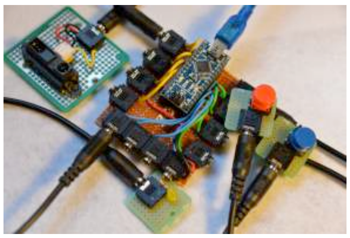

Figure 1: Interface board and with some I/O modules

3.2 Modules

A variety inputs have been created for the toolkit. Digital inputs include buttons, switches and a motion sensor. A converter module enables using standard ¼" switching pedals as digital inputs to the system.
Analog input modules include rotary and slide potentiometers, accelerometers, light sensors, membrane sensors, force sensitive resistors and proximity sensors among others enabling a range from the typical knobs and sliders to the more exotic proximity and light sensors.
Output modules include LEDs and a solenoid driver board. Using a MOSFET driver circuit the solenoid board enables a simple interface for driving solenoids that work well as percussive actuators.

3.3 Software Library

Using Pd-Extended our library of objects was created to simplify interfacing to the hardware and to enable basic sound design. All the objects utilize GUI elements in addition to control inputs making it easy for novices to explore.

3.3.1 Interface Objects

The [mm] object scans and automatically connects to the interface board. Interface objects simplify the process of routing and mapping data. The example in Figure 2 shows an [mm-analog] object with the creation arguments 0, 40 and 800. These values are used to specify the input port number, and the low and high values to map the analog input to respectively. The [mm-digital] object uses a single creation argument to specify the input pin. The GUI elements of the interface objects provide immediate visual feedback on the status of the hardware.

3.3.2 Sound and Control Objects

Audio Generation Objects are for sound synthesis. These include basic waveform generation (see figure 2), FM and pulse width modulation objects as well as a plucked string model. These objects feature similar interfaces making them easy to interchange.
Audio Effects Objects include reverb, delay, overdrive, tremolo and a resonant low pass filter. The effects objects use similar interfaces with a bypass switch and knobs to control parameters with corresponding inlets that align with the GUI interface.

Control Objects allow programmatic control of parameters for automation and gestures. Control objects include envelopes, low frequency oscillator (LFO), and a general purpose sequencer that can be used to trigger events, set note values or control any other number of parameters.

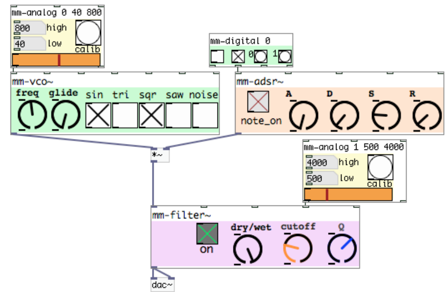

Figure 2: A simple synthesizer built

Because these objects are all built from standard objects available in Pd-Extended they can be explored as examples to understand programming approaches or easily modified to fit different needs.

4. In the Hands of Children

In order to assess the efficacy of the toolkit as well as the context of DMI design for education, modular-muse has been used in preliminary studies in both informal (in a public makerspace) and formal learning environments (as part of a middle school music class).

4.1 Make Your Own Instrument Workshop

In August of 2014 modular-muse was used in a two day workshop at the ideaLab, a makerspace for teens in the Denver Public Library. The space provides free access to tools and workshops for youth in an urban area. The lab is a space for kids to get access to tools like a 3D printers and Arduinos and to learn how to program and use digital design tools.
The workshop was held over two days for 2 hours each day. We had five participants ages 10-13. None of the participants had any previous electronics or programming experience. One ideaLab employee (with music experience but no programming or electronics experience) was on hand for help.
The workshop was structured with a focus on hands on building. Topics were briefly introduced followed by hands on exercises where I asked students to build along with me. Each student had a laptop with modular-muse and Pd installed. The hardware toolkit was handed out after the students gained some familiarity working with Pd and some of the library elements.

4.1.1 Workshop Topics and Goals

The goals of this initial workshop were to get kids comfortable with the workflow of working with the toolkit and key concepts of Pd as well as providing an opportunity to assess and refine the toolkit.
The workshop was set up to get students creating sound as quickly as possible while enabling customization and exploration. Using the modular-muse library everyone was playing with sound synthesis and audio effects in minutes. The similar interface of sound generation and effects objects made it easy for participants to swap out different sound generators and effects during their explorations.

4.1.2 Introducing Hardware

After getting comfortable with some basics of creating and controlling sound using the modular-muse objects we introduced the hardware. Each participant was given an interface board, patch cables and a series of inputs and output modules.

Verplank's concept of *buttons* and *handles* [14] was used to introduce interaction design concepts and to explain the difference between digital and analog inputs. Focusing on common NIME challenges of *expressiveness* and *mapping* [6].

Figure 3 – (left) Exploring sensors at the ideaLab; (right) Julio with an early iteration of his "air guitar"

4.1.3 Workshop Results

Participants were all successful in constructing a variety of interfaces. The following vignette illuminates the type of engagement and use of the toolkit common in the workshop:

> *Julio, a twelve year old was riffing happily on an interface he had just built. He was adding effects and exploring different means of controlling them, using a knob to control pitch and a distance sensor to control a tremolo effect. He liked the tremolo interaction but identified a limitation of the knob as a pitch controller, noting that he couldn't control both pitch and tremolo at the same time, as his other hand was occupied with a button to trigger the sound. His novel solution was to swap out the knob for a ribbon* sensor *that detects the position of a touch along a narrow ribbon, which he could operate with the same hand as the button. He dubbed his instrument an "air guitar" because the linear touch sensor interaction reminded him of sliding a finger on a string and the distance sensor let him control the sound by waving his hand in the air.*

This exemplifies the potential of using DMIs to engage children in a rich set of interdisciplinary activities. The behavior of the different sensors was only briefly introduced yet through exploration Julio (figure 3) taught himself how they *felt* for different controls. He's also exploring the ergonomics of the interaction, looking for ways that will allow him to control and perform with multiple parameters.

4.2 Manhattan Middle School - Music Focus

Designing DMIs involves both developing an interface and mapping the interface to control different sound parameters. There is a lot for a novice to think about. In the middle school class, as a way to break the task down we decided to initially focus only on the sound design for a project before introducing any hardware. This allowed us to focus on the teaching process and led to the development our learning progression, *Build-Play-Share-Focus*, highlighted below.

The class, Music Focus, is an elective and all the students are also enrolled in either band, orchestra or choir. The class has 19 seventh and eighth grade students. Prior to the introduction of Pd and the modular-muse library they had discussed the concept of timbre, some basic music theory and "rules of melody", which they were using to compose their own pieces.

With the students in groups of 3 or 4 we built some basic patches together (*Build*), starting with the sound generation and the effects objects. After explaining how to use some control objects and showing them how to find the help files, the groups were given free reign to explore (*Play*). The results were chaotic and charming as the kids tried different things out and explored the boundaries. At the end of the class period the groups reported out on their results. They described their sounds: "*psycho-ambulance*", "*electric-turtle*", "*we made water*". Groups were asked to present the techniques they used and ask questions (*Share*).

This exploration is seen as extremely valuable time for the students to get comfortable with the tools and learning some of the possibilities. It allowed them to explore the boundaries they found interesting while the sharing session allowed them to learn from each other different techniques they hadn't considered or settings they didn't have time to explore.

4.2.1 Pd Poems

In the following class session we wanted to give them a task to ground their sound explorations (*Focus*). This is what came to be known as *Pd Poems*. Each group was given a poem and asked to design at least three sounds to accompany a reading of the poem. The poems chosen were rich with language relating to sound, describing rain, steam engines and rocks falling. When working towards a specific sound goal the students worked in a much different manner. They were recalling techniques they had explored when in "free-play" and applying them to the new task. There was a different energy in the room as teams focused and tried to analyze what changes they should make to create their sound effects. In two classes of work the students became comfortable with the basic workflow in Pd using the modular-muse library while designing sounds to accompany the poems. The success of this activity resulted in the students sharing them at the school's Winter Arts Festival.

4.2.2 Pd Poems Results

Over the course of four class periods the students formed new groups and chose new poems and programmed new sounds. At this point I noticed several students had trouble with the use of the envelope objects, wanting to send audio signals through them rather than multiply them by other signals. The groups used a variety of creative approaches to design their sounds occasionally using objects well outside of the original intention. For example, one group simulated the sound of rain drops by using the sequencer, which outputs control numbers to directly drive the audio output, resulting in a series of impulses, creating opportunities to discuss new concepts. Seeing a need for more expressive control a new object that captures gestures by drawing shapes in an array was introduced which quickly became popular and was used by all but one of the groups.

At the performance one group made use of key strokes to trigger sound gestures, this led to other groups wanting to know how to use this "trick". Exemplifying the benefits of sharing and teaching each other. The loosely structured assignment and performance for family and peers parallels the motivated learning by exhibitions rather than competitions as discussed by Rusk et al [11] in their robotics workshops.

5. DISCUSSION

In both the workshop and in the music class, children were able to quickly grasp the workflow of building patches using the modular-muse library in Pd to create a variety of sounds. By contrast, during a Pd and Arduino class taught at Sparkfun Electronics, much more overhead time is spent on both learning Pd's idioms and on building circuits before the fun begins. While additional research is needed, in these preliminary studies modular-muse has lowered the barrier to entry and provided breadth to *keep young students engaged*.

During the ideaLAB workshop hardware was introduced as early as possible. For the four hour workshop this gave participants access to easily constructed, expressive interfaces as they explored sound programming.

In the middle school music classroom students built capacity for sound design refining their intuition about synthesis and effects parameters.

The experience with Pd Poems has led to the development of a high level teaching framework which will be used to develop future workshops and learning materials. The progression, *Build-Play-Share-Focus*, describes a four step approach to teaching (figure 4). First a new topic, idea or technique is introduced by demonstration and building together. Next students are given time to play and explore the idea. After sharing ideas a task is given to focus the explorations and refine their approach as was done with the *Pd Poems* project.

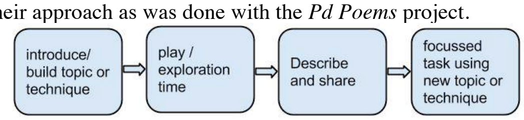

Figure 4 - Introduce, Play Focus teaching progression

5.1 Challenges

On several occasions children struggled with the concept of multiplying an audio signal to control the volume. Explanations describing sine waves and amplitude initially drew stares of confusion (as 7th and 8th grade students they have little experience with trigonometry). However, we decided to keep the existing design in tact because it enforces a common idiom and a basic concept in Pd and other audio programming languages, *making explicit the manipulation of an audio signal by math*. In this case it was resolved by an adjustment to the pedagogical approach. Utilizing the visual nature of Pd to plot an audio signal in real time as it was multiplied by a decreasing number they were able to connect the aural experience with the concept of controlling amplitude .

5.2 Outside Education

The flexible nature of this toolkit makes it an intriguing paradigm for artists and musicians as well. Because modular-muse is reconfigurable it becomes several instruments in one. In 2014 I used it for performances with a laptop orchestra in which a number of interfaces were used (and built) throughout the performance.

5.3 Future Work

These preliminary studies have helped us refine our tools and teaching approaches. Workshops which utilize pre/post tests to assess learning gains will provide insight into the learning potential for using our toolkit approach and DMI design as a learning context.

Online documentation will be designed to provide an outline for future workshops and to support self guided learning consisting of step by step tutorials and example projects. We will also create interface objects for toolkits and shields such as Grove, the PicoBoard[5] and others which are widely available to enable a motivated individual at a remote makerspace to use their existing hardware to begin building DMIs.

As of this writing the Music Focus students are using the next iteration of the toolkit to design and build their own tangible interfaces for controlling sound and music. A workshop using the *Build-Play-Share-Focus* progression is being developed for a five-day workshop this August where children ages 11-13 children will build DMIs.

6. SUMMARY

The preliminary studies presented here demonstrate two approaches to engaging children in DMI design through a workshop setting and as a several week unit in a middle school music class. By scaffolding the activity with the modular-muse toolkit the participants gained hands-on experience with sound design, programming as well as interaction design. Further studies and alternative hardware designs are underway. This work also yielded the teaching progression we refer to as *Build-Play-Share-Focus* which will be used for future workshop development.

7. ACKNOWLEDGMENTS

This project is supported by the ATLAS Institute and NSF #0841423. Thanks to Michael Theodore for sharing his endless ideas, to Leah Beauchley for pushing me to uncover this idea and to CCRMA for my first taste of interaction design for music.

8. REFERENCES

[1] Blikstein, P., & Sipitakiat, A. (2011). "QWERTY and the art of designing microcontrollers for children." Proceedings of the 10th International Conference on Interaction Design and Children - IDC '11, 234–237.

[2] Buechley, L., & Eisenberg, M. (2008). The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. *The SIGCHI Conference on Human Factors in Computing*, 423–432.

[3] Bukvic, I., Baum, L, Layman, B., & Woodard, K. (2011). "Granular Learning Objects for Instrument Design and Collaborative Performance in K-12 Education", 12–14

[4] Greenberg, Saul, and Chester Fitchett. "Phidgets: easy development of physical interfaces through physical widgets." *Proceedings of the 14th annual ACM symposium on User interface software and technology*. ACM, 2001.

[5] Honey, M. and Kanter, D.E. (eds.) Design | Make | Play: Growing the Next Generation of STEM Innovators, Routledge 2013.

[6] Jordà, S., & Mealla, S. (2014). A Methodological Framework for Teaching, Evaluating and Informing NIME Design with a Focus on Expressiveness and Mapping. Proceedings of the 15th Conference on New Interfaces for Musical Expression, 233–238.

[7] Kafai, Yasmin B. Minds in play: Computer game design as a context for children's learning. Routledge, 1995.

[8] Kafai, Y. B., & Peppler, K. a. (2011). Youth, Technology, and DIY: Developing Participatory Competencies in Creative Media Production. *Review of Research in Education*, *35*(1), 89–119.

[9] Overholt, D. (2012). Musical Interaction Design with the CUI32Stem: Wireless Options and the GROVE system for prototyping new interfaces. Proceedings of 12th Conference on New Interfaces for Musical Expression.

[10] Resnick, M., & Maloney, J. (2009). Scratch: programming for all. *Communications of the ACM*. Retrieved from http://dl.acm.org/citation.cfm?id=1592779

[11] Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M. (2007). "New Pathways into Robotics: Strategies for Broadening Participation." Journal of Science Education and Technology, 17(1), 59–69. doi:10.1007/s10956-007-9082-2

[12] Trappe, C. (2012). "Making Sound Synthesis Accessible to Children." Proceedings of the 2012 conference on New Interfaces for Musical Expression (NIME).

[13] Sawyer, B., Forsyth, J., O'Connor, T., Bortz, B., Finn, T., Baum, L., Bukvic, I., Knapp, B., Webster, D. (2013). Form, Function and Performances in a Musical Instrument MAKErs Camp, Proceedings of the 44th ACM technical symposium on Computer science education. 669–674.

[14] Verplank, W. 2009. Interaction Design Sketch Book. Unpublished paper. Available on-line at www.billverplank.com/IxDSketchBook.pdf Accessed 12 Jan 2015

[5] http://www.picocricket.com/picoboard.html