Supporting modular performance technique

Hans-Christoph Steiner
IDMI/Polytechnic University
Brooklyn, NY, USA
hans@at.or.at

ABSTRACT

Many novel instruments have been built in recent decades, but they rarely become popular or even get played at any degree of virtuosity. A new model of instrument design is emerging, where creators make instruments tailored to a performer or specific piece, making instruments even more specific. Modular software for synthesis and mapping along with standard human interface devices make this increasingly feasible. In addition, the performer's body of technique could be modular as well, with musicians learning interfaces, mappings, and synthesizers as distinct "modules" of technique. This would allow musicians to play newly created instruments with some degree of existing skill. These modules of technique could exist separately from whole instruments, therefore allowing a shared body of technique to be developed, especially around software synthesizers, popular human interface devices, and the mappings between them. The design of software used for such instrument building can go a long way to furthering this approach. Some specific software design ideas are explored in the context of Pd with the aim of fostering the discussion and development of modular technique.

1. INTRODUCTION

New computer music instruments are rarely played by more than a few people or used for more than a couple very specific styles of music. With most of these new instruments, no one ever achieves any kind of virtuosity with them, making any evaluation of their potential quite difficult [19]. Many new instruments are expensive or difficult to make, further limiting the potential body of users. With few people playing a given instrument, the development and sharing of techniques for playing that instrument is quite limited. The classic goal of an instrument designer is to create an instrument that is so compelling that it becomes widely used. This would allow, among other things, the possibility of building a body of shared technique. Even with increased activity designing instruments of the past century, none of these new instruments are a household name. Indeed, the most successful new interface for musical expression of the past century, the turntable, was not designed as a musical controller at all.

A new model of instrument design is emerging, shifting away from instruments designed for a broad range of musicians, such as the Theremin. Musicians are becoming their own instrument designers, creating instruments designed purely for their own compositional goals. The idea of an instrument as a device for playing a wide range of pieces is also shifting. When teaching instrument design, some go so far as saying, "make a piece, not an instrument or controller."[8] Computer music software is making it relatively easy and affordable for individuals to create their own instrument tailored to their goals, or even to a specific piece or performance. Many performers interested in using new interfaces are employing general software and hardware building blocks that allow the individual musician to create their own instrument. Software such as Ableton Live, Max/MSP, Pd, and Reaktor provide this kind of high-level modular approach to music programming; various USB HIDs (Human Interface Devices) and MIDI controllers can be used with such software.

Traditional instruments have a great advantage over instruments designed for narrow compositional aims. Musicians using standardized instruments can collaboratively develop and share a body of knowledge about how to play that instrument, as well as learn from existing masters. With musical instruments throughout history, learning from and sharing with other musicians are essential means for progressing. Using instruments created using modular software and standard HIDs can potentially allow people to build a shared body of technique without sacrificing the ability to tailor the instrument specifically to specific design goals. The [hid] [1] toolkit and Mapping Library together form a framework for instrument building. A number of design choices where specifically chosen to explore this modular approach to instrument building and musical technique.

2. STANDARDIZED INTERFACES

When talking about interacting with computers, HID has become the standard term for devices that get data from the physical actions of a human. Mice, joysticks, gamepads, keyboards and graphics tablets are all classified as HIDs. Consumer HIDs are attractive for a number of reasons. They have been developed and tested over decades, and are cheap and readily available. While many such devices are not up to the standards needed for musical performance, some HIDs perform quite well. Devices notable for their performance include gaming mice and graphics tablets. Many people already have established a high level of skill with these devices.

[1] a word in square brackets denotes a Pd object

Pd Convention 2.0, August, 2007 Montreal, Quebec, Canada

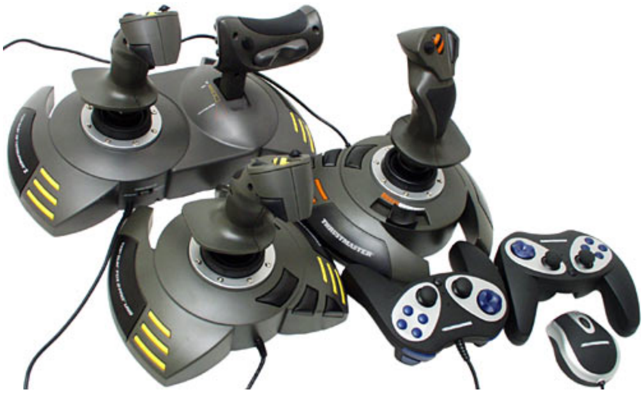

Figure 1: A range of variations in game controllers.

Gamers[2] are quite skilled at using devices such as joysticks, computer keyboards, and mice; graphic designers often are very skilled with drawing tablets such as Wacom devices.

Video games are becoming focus in human-computer interaction research because they "... are one of the most successful application domains in the history of interactive systems" [10]. Gamers spend years perfecting their technique in interactive tasks that require similar levels of control and concentration as making music. Also, gamers place a high value on virtuosity, criticizing games such as Halo 2; it automates aiming, thereby making game play much less reliant on manual skill. A number of novel gaming controllers are available, but the vast majority of gamers use the standard interfaces. Even though there are many variations of these standards, they are all similar enough that technique developed on one can be directly applied when using another. In other words, on the most basic level, a mouse is a mouse, a keyboard is a keyboard. As another avenue for exploration, much of this gamer technique could be applied to playing a HID-based instrument.

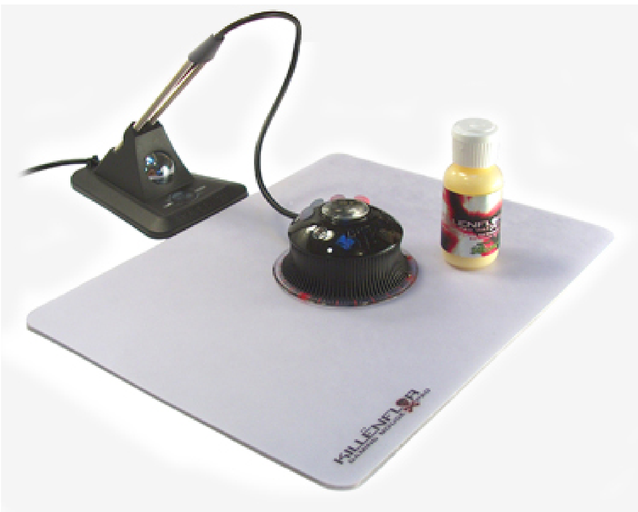

Figure 2: An unorthodox mouse, but still a mouse.

3. MODULAR TECHNIQUE

Gamers and turntablists[3] provide relevant examples of bodies of technique for standardized interfaces. Most gamers specialize on certain physical interfaces and develop that technique. For example, many gamers who play "first-person shooters" use "WASD" keyboard/mouse mapping[4] while many others use a gamepad to control the same game. Video games allow the user to map their interface to the controls, so it is possible to use a range of controllers for a given game, and even heavily customize the mapping of common controllers. Gamers usually stick to a specific controller, even when playing different types of games. Even though some controllers are better suited to specific games, the body of technique developed on that controller outweighs the possible benefit of changing controllers. Similarly, gamers are hesitant to use unorthodox controllers partially because the body of technique would not be available, nor would the community of users be as large. Some gamers even modify their game system in order to use their favorite controllers. For example, a number of people have modified the Xbox so that they could use a keyboard with it, allowing them to use the WASD setup.

We can also look at turntablists. Turntable and DJ mixer interfaces have been long standardized, providing a common platform for turntablists. The body of technique is very established, so even though the turntables leave much to be desired as a musical controller, they are increasingly popular. Turntablists use a variety of techniques for manipulating the turntables and mixer in order to produce the sounds they desire. A canon of standard turntable technique has been established, with many gestures having specific names such as "the chirp" and "the flare" [16]. Using different source records dramatically changes the resulting sound. Scratching with beats versus scratching with tones generates distinct timbres. The turntables are even becoming a general controller with software like Final Scratch or Ms. Pinky, giving the turntablist access to synthesizers of all kinds.

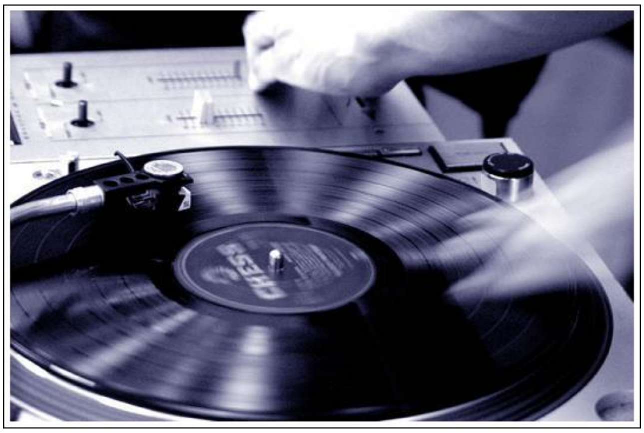

Figure 3: The standard position for the turntablist.

Michel Waisvisz demonstrates the idea of modular interface technique quite well in his performance using *The Hands* [1]. For most of the 20 years that he has been playing his in-

[2] people who play video games seriously, much like a sport

[3] musicians who play the turntable as their instrument

[4] for an example of WASD, see Figure 4

strument *The Hands*, he has been controlling MIDI synthesizers. For his performance at the 2003 NIME conference, he sampled his voice and manipulated it in realtime. His skill with The Hands was evident even though the method of generating sound was quite new to him. There are a number of contemporary musicians who have mastered HIDs. Gerard Van Dongen toured with his force feedback joystick [9]; Loïc Kessous created his instrument using a Wacom tablet and a joystick [15]; Sergi Jordà built the QWERTYCaster using a joystick, keyboard, and trackball [14]; Leon Gruenbaums Samchillian Tip Tip Tip Cheeepeeeee [12] is built upon a standard ergonomic keyboard.

Knowledge of particular synthesis methods is one example of modular technique that has been common is the computer music world for a long time. Many musicians and composers choose a specific synthesizer (software or hardware) and learn the details of the range of sound that it can produce. For example, someone knowledgable about FM synthesizers has a strong idea about what modulating frequencies make vibrato sounds, and, how much to raise that frequency to stop sounding like vibrato. With a phase vocoder, someone with much experience can predict at what point the artifacts will dominate when stretching the sound in time. This knowledge can be applied to any implementation of these synthesis algorithms much like how trumpet mouthpiece technique can be usefully applied to a french horn. It is still useful knowledge regardless of the control interface.

4. EXAMPLES OF SHARING

The sharing of programs, patches[5], and compositions is a long standing tradition in the computer music world. There are many active forums where musicians swap and share software instruments and patches. Mailing lists are a very common medium for this swapping, as well as web site forums such as EM411.com. Many people share their performances patches to exchange ideas and get feedback. Such performance patches usually incorporate the interface, some form of GUI, the synthesis, and the mapping.

Since musical instruments made with HIDs are relatively new and not very widespread, there are only a few examples of HID musicians sharing technique or learning from other HID musicians. In a general way, HID musicians often discuss their controllers after a concert with interested audience members, much like any other electronic musician. A paper by Wessel and Wright[22] gives a more specific example, describing a few techniques they use with the graphics tablet: "Drag and Drop", "Scrubbing", and "Dipping". Some common mapping ideas seem to be developing. For example, Wessel and Wright, and DuBois both break up the area of the tablet into different "regions" of control. Also, DuBois and Kessous both use polar coordinate systems with tablets. These are all interface techniques which can be applied to which ever synthesis method the musician desires.

A body of standard technique has been established for using HIDs for video games, including: "Pawing (verb): the act of lifting a mouse and returning it to the center of your mouse pad. Useful when trying to make a sequence of fast mouse movements" [11]. "WASD" position is a canonical position for playing many games, much like holding a violin properly. Gamers are hesitant to use unorthodox controllers partially because there is less opportunity to learn from others. Watching and imitating is perhaps the oldest form of learning, and it is also a common practice among gamers. At gatherings of gamers, people often watch over the shoulders of others while the play, watching both game play and physical technique. Many video games allow players to remotely view what another player sees, giving them the opportunity for them to learn the game through another person's eyes [10]. These are a common practices: in person, the interface technique can be observed, and via the game software, the game technique can be observed, demonstrating how gamers learn the game separately from from interface.

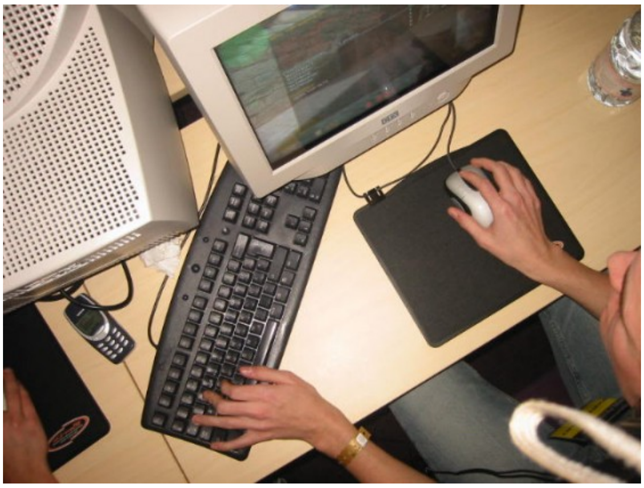

Figure 4: Standard WASD position.

Turntablists also have a tradition of learning new techniques by watching each other perform. Pure imitation, called "biting", is frowned upon, but building upon other people's technique is a strong part of the culture. The body of turntablist technique has become quite codified, with standard named gestures. Standardized DJ equipment enabled the development of turntablist technique, which has become quite well established over the last 3 decades, for example: "[t]he first scratch, normally done in eighth-notes or triplets in time with the music, in called the baby-scratch." [13] There are numerous books, websites, and videos which illustrate the canon; courses in turntablist technique offered, including Berklee's "Turntable Technique" [21]; and even two notation systems. [7].

These two communities provide strong examples of the development of a shared body of technique with new physical interfaces. The gaming community also has some examples of modular technique for controller mapping and game play. Such shared bodies of technique are essential aspects of all well established musical instruments. These examples point to the possibility of similar developments in the realm of new musical instruments. Using standard HIDs as musical controllers enable the building of a body of shared technique in a number of ways. These devices are affordable and easily available, much like turntables. When a device has a broad user base, an instrument based on that device will be much more likely to spread, and communities of users much more likely to form. HID musicians can also build upon the existing body of knowledge that comes from other serious users

[5]a "patch" is a term used by Pd users to mean a program

of these devices, such as gamers or graphics designers.

5. [HID] AND MAPPING LIBRARY

5.1 A Unified Framework

The [hid] toolkit[17] together with the Mapping Library[18] provide a framework for getting gestural data from these devices, controlling haptic feedback, and mapping this data to the desired output, all within Pd. Also, specific attention was given to easing the sharing of instrument patches, generalizing the labels for each element of each device type, using a common button numbering scheme across all devices, and normalizing all data to between 0 and 1. This framework also allows the final instrument software to be made up of distinct modules, so they can be interchanged in ways that would have otherwise required substantial modification of the underlying software.

It was already possible to build instruments with HIDs using a number of software packages such as Pd, Max/MSP, and SuperCollider[3]. Many people have created their own instruments but because of technical difficulties, found it difficult to share with other users. A common problem is that the software is tailored to one specific setup making it quite difficult to get it running in a different context. Another issue is that each HID had a separate software interface. This means that in order to use different HIDs, the user has to learn a software interface for each different HID type. Since the basic principles are the same across the range of HIDs, a common software interface works well.

Two notable frameworks for creating mappings for musical instruments come from IRCAM. "MnM is a set of Max/MSP externals based on FTM providing a unified framework for various techniques of classification, recognition and mapping for motion capture data, sound and music." [5] The ESCHER toolkit for jMax is an earlier effort: "ESCHER is a modular system providing synthesis-independent prototyping of gesturally-controlled instruments by means of parameter interpolation" [20]. These frameworks also aim to modularize instrument design but without specifically addressing measures need to enable the sharing of instrument patches.

5.2 Data Range

A common data format is essential in order to have the input, mapping, and feedback objects interoperate with each other. This frees the instrument designer from having to think about scaling the data with each step. The [hid] toolkit uses the input/output range of 0 to 1 following the computer audio standard is 0 to 1 (amp, pan, etc.), as well other environments like the Gem graphical environment for Pd. Also, 0 to 1 is much easier to convert to any other range. The objects for getting the data from HIDs ("[mouse]", "[joystick]", etc.) also output data in the range of 0 to 1, allowing the same patch to work with different devices of the same type, two joysticks for example, even if they provided a drastically different range of data. Using 0 to 1 for axes makes the data format the same across all of the HID elements as well: axes, buttons, and pseudo-axes all output data in the range 0 to 1. The variation in device data output instead causes a difference in resolution rather than a difference in position.

5.3 Mapping

The Mapping Library provides a number of mapping objects for commonly used operations. All expect an input range of 0 to 1, and their output is scaled to 0 to 1. This allows them to interoperate with all of the toolkit objects, but could make them less useful for general purpose applications. Objects such as [curve] and [curve_log] allow the designer to curve HID data, [iir] and [median_n] are useful for smoothing out noise in data streams. [one2two] and [one2three] allow from one-to-many mappings, and [polar] and [spiral] generate polar coordinates, with [spiral] also tracking revolutions. Other more general objects such as [autoscale], [buttongate] and [keygate] can be used in many contexts.

5.4 Accessing HIDs

Another frustration is how to specify the device to use. Usually to open a device, the number of that device is specified. That number can vary widely depending on the setup. If the name is used instead, then that patch will be tied to that specific type of device. The [hid] toolkit provides objects for device types to solve this problem: [mouse], [joystick], [tablet], [gamepad], etc. Specifying the number ([mouse 0], [joystick 1], etc). chooses only from that device type. When no device number is specified (i.e. [mouse]), the object outputs data from all attached devices of that type. So if a patch was built for a mouse on a desktop with only one mouse attached, and then that patch was opened on a laptop with a built-in trackpad, [mouse 0] would be the mouse on the desktop and the trackpad on the laptop. If the patch specified no device number, then all attached mice would work with the patch. These objects generalize access to devices of a given type so that patches can easily be used with different versions of the same controller, e.g. different models of joystick.

5.5 Event Scheme

The event scheme for the [hid] toolkit is derived from the USB HID [2] and USB PID[4] event scheme and is used on all platforms. The Linux input.h scheme was also examined in detail to provide context. The Linux scheme has some aspects of it that are too specific, making it hard to abstract, i.e. button names for each device type, rather than just button numbers. Numbered button labels allow buttons on one device to work in patches written for other devices. Also, the events are named descriptively rather than specific to a given device. While USB HID specifies a number of different kinds of absolute X axes for different "usage pages" like "Generic Desktop", "Simulation", "VR", "Sports", and "Game", hidio uses the same event names for all related movements: "x", "y", "z", "rx", "ry", "rz", [6] "slider", "dial", "wheel", and "hatswitch". For outputting data to control LEDs and force feedback, [hid] sticks quite close the the USB specifications.

A joystick will have absolute X and Y axes at the bare minimum. A gamepad will also have absolute X and Y axes. Both of these axes are description using "absolute x" and "absolute y". This allows a patch built for a joystick to be controlled by a joystick without modification. Since the events are labeled in a manor that describes the physical action of the human with the device. For the gamepad and the mouse, "absolute x" means that the human is moving a physical element towards and away from herself, while

[6] "r" stands for rotation here

"absolute y" means that the human is moving that same physical element from left to right.

6. EVALUATION

The [hid] toolkit and the Mapping Library have been available in some form for well over a year now, so there is some experience to measure the success of the above ideas. There has not yet been formal testing of these ideas, therefore so far, the evaluation is mostly from the direct experience of the author and informal discussions with users, both in person and on the Pd list. Since the introduction of the [hid] toolkit, there seems to be more instrument design activity involving USB HIDs, such as August Black 's El Lechero [6], and users have given some reports of relative ease of use with the [hid] object. Much has been learned about how to handle USB devices in a simple and generalized way, and this knowledge is being applied to the next generation of the [hid] toolkit, called [hidio].

7. FUTURE WORK

Now that a substantial portion of this framework is implemented, the next steps will be to get feedback from people using this framework to build instruments. As the software is more clearly defined, formal usability testing will be performed to see where these ideas have worked, and what is still lacking. In addition to HIDs, this framework can be applied to a broad range of devices. In particular, microcontrollers could be programmed to communicate using the HID protocol, then sensors could be set up as generic axes, or to emulate a mouse (relative axes) or a joystick (absolute axes). Work is already underway to figure out how best to support various microcontroller platforms in this context. MIDI-based controllers would also fit well into this framework. Objects for specific devices could be written as a interface to MIDI control surfaces like the Peavey PC1600, making them output data in the standard 0-1 range and use the standard event names.

8. ACKNOWLEDGMENTS

Cyrille Henry contributed in a number of key ways. He was involved in the formations of the original ideas regarding mapping, and helped me focus my ideas through discussions about mapping, and wrote a wealth of objects for the Mapping Library. Thanks Derek Nelsen and Anton Marini for insight into the gaming world, and Takuro Mizuta Lippit for insight into turntablism. Thanks to Tom Igoe for helping me clarify my ideas and for discussions of how this framework could be applied to microcontrollers and sensors.

This paper is dedicated to the memory of Gerard van Dongen, who did some of the key early work on using HIDs with Pd.

9. REFERENCES

[1] The hands. http://www.crackle.org/TheHands.htm.

[2] HID Information. http://www.usb.org/developers/hidpage/.

[3] SuperCollider. http://audiosynth.com.

[4] USB PID. http://www.usb.org/developers/devclass_docs/pid1_01.pdf.

[5] F. Bevilacqua, R. Müller, and N. Schnell. MnM: a Max/MSP mapping toolbox. In *Proc. of the Conference on New Interfaces for Musical Expression (NIME05)*, Vancouver, Canada, 2005.

[6] A. Black. El Lechero. http://aug.ment.org/lechero/.

[7] J. Carluccio. Turntablist transcription method, 2004. http://www.ttmethod.com.

[8] P. Cook. Principles for designing computer music controllers. In *New Interfaces for Musical Expression Workshop, CHI2001*, 2001. http://citeseer.ist.psu.edu/cook00principles.html.

[9] G. V. Dongen. http://www.xs4all.nl/~gml/.

[10] J. Dyck, D. Pinelle, B. Brown, and C. Gutwin. Learning from Games: HCI Design Innovations in Entertainment Software. In *Proceedings of Graphics Interface 2003*, 2003. http://citeseer.ist.psu.edu/dyck03learning.html.

[11] ExtremeTech. Review: RTR-720 Whoop-Ass Mouse. http://www.extremetech.com/article2/0,1558,1203359,00.asp.

[12] L. Gruenbaum. Samchillian Tip Tip Tip Cheeepeeeee. http://samchillian.com.

[13] K. F. Hansen. Turntable music. In L. Jonsson, K. Oversand, and M. Breivik, editors, *Musikklidenskapelig Årbok*, pages 145–160. Trondheim: Dept of music, Norwegian University of Science and Technology, 2000. http://www.speech.kth.se/prod/publications/files/796.pdf.

[14] S. Jordà. New musical interfaces and new music-making paradigms. In *New Interfaces for Musical Expression Workshop, CHI2001*, 2001. http://www.iua.upf.es/~sergi/articles/NIME2001.pdf.

[15] L. Kessous. Bimanuality in alternate musical instruments. In *Proceedings, Conference on New Interfaces for Musical Expression (NIME-03)*, 2003. http://citeseer.org/kessous03bimanuality.html.

[16] S. Smith. Compositional strategies of the hip-hop turntablist. *Organised Sound*, 5(2), 2000.

[17] H.-C. Steiner. [hid] toolkit: a unified framework for instrument design. In *Proc. of the 2005 conference on New Interfaces for Musical Expression (NIME05)*, Vancouver, BC, Canada, 2005.

[18] H.-C. Steiner. Towards a catalog and software library of mapping methods. In *NIME '06: Proc. of the 2006 conference on New interfaces for musical expression*, pages 106–109, Paris, France, 2006. IRCAM Centre Pompidou.

[19] M. Wanderley and N. Orio. Evaluation of input devices for musical expression: Borrowing tools from hci. *Computer Music Journal*, 26(3), 2002.

[20] M. Wanderley, N. Schnell, and J. B. Rovan. ESCHER-modeling and performing composed instruments in real-time. *IEEE Systems, Man, and Cybernetics*, 1998. http://intl.ieeexplore.ieee.org/xpl/abs_free.jsp?arNumber=727836.

[21] S. Webber. Turntable technique, a course at berklee school of music, 2004. http://www.berklee.edu/opi/2004/0217.html.

[22] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):1122, 2002.