

GUA: Development of a Digital Instrument

Nick Hwang
Louisiana State University
Center for Computation and Technology
Laboratory of Creative Arts and Technology
nhwang1@lsu.edu

Jonathan Corey Knoll
Louisiana State University
Center for Computation and Technology
Laboratory of Creative Arts and Technology
jknoll3@lsu.edu

ABSTRACT

Patch-based digital processes are often created and designed for a specific piece, installment, or performance. These processes have limited use and lifespan. GUA was conceived as a process in a musical piece. Constant iterative development of not only process, but also interface design and controller hardware, has evolved GUA into a versatile musical instrument capable of numerous performance settings, most noticeably improvisation. This development has also coincided with a reevaluation of aesthetic goals. This paper outlines the development of GUA from its conception, as musical work written in ChuckK, to its current state as a complete instrument, using Max/MS and iPad. Software, hardware, and aesthetic considerations are discussed, as well as noted setbacks.

Keywords

Max/MSP, ChuckK, Instrument, OSC, TouchOSC, Apple iPad, M-Audio Trigger Finger, Live Manipulation

1. INTRODUCTION

Since the Hub first networked microcomputers in 1984, composers and musicians have been looking at ways to create digital ensembles. Today, laptop ensembles are increasingly becoming important media in the experimental music scene. Entire software programs, complete with digital interfaces and corresponding hardware systems, are increasingly being designed for use in a single piece within these ensembles. Purpose and interface design often blur the line between what is a musical work and what constitutes a digital instrument.

Concerning modern instrument design, composer Dr. Bruce Pennycook stated that “we need to build instruments that are inexpensive, highly portable, simple to use, and that performers will enjoy using... [with] small, self-powered high fidelity loudspeakers that can be put in the milieu of the performers so that from the audience’s viewpoint the sound seems to come from the players.” [1] GUA is an interactive digital instrument closely following Pennycook’s model. It began its life, however, as the core in a networked laptop ensemble piece. In this paper, we track the evolution of GUA from piece to engine to instrument, discussing how advancements in hardware, software, and interface design have contributed to an aesthetic reevaluation of GUA’s role in digital music performance.

2. CONCEPTION OF GUA

GUA was conceived as an experiment in the Laboratory of Creative Arts and Technology (LCAT) at Louisiana State University. The task was to create an original work in the ChuckK music programming language for multiple performers on laptops. Each laptop was to be networked through the use of open sound control (OSC) protocol [2]. Each laptop had a specific function, and a centralized laptop controlled different musical parameters running on the other networked laptops.

2.1 Meaning of GUA

The name GUA was derived from the eight trigrams, *gua*, of the I-Ching. Eight melodies were composed, each based on a *gua*. These melodies were composed with different levels of consonance and dissonance, based on their corresponding level of change (*yin* and *yang*). The melodies were realized in ChuckK and formed the basis of the improvisatory material used by the acoustic musicians.

2.2 Control Stations

It was initially decided that there would be four stations running separate programs. The first station consisted of a live performer on an acoustic instrument (trombone, saxophone, etc.) playing into a microphone. The output of the microphone was processed by a computer and ran through a series of ChuckK unit generators controlled by the performer at Station 4.

These unit generators included live pitch shifting, delay/chorus, gain, and panning controls. The second station also received input from the microphone, which was directed into a recording buffer. The buffer integrated the *LiSa* (LIve SAmpling) object, which allowed for playback looping, variable speed, and bi-directionality of the captured sample. The third station allowed the performer the ability to play back any of the eight *gua* melodies at any time and at various playback speeds. The centralized fourth station controlled both the parameters of Station 1, as well as the gain and panning of Station 2 and Station 3. A separate ChuckK shred also played one of the eight melodies at random once every minute, allowing the performers to keep track of time.

2.3 Interface

Each of the parameters at the various stations was controlled through use of the MAUI interface, a set of GUI buttons, knobs, and sliders incorporated into the ChuckK language.

2.4 Initial Performance

GUA was first performed at Louisiana State University on April 27, 2009 as a part of the High Voltage concert series. Each performer manned all four stations once during performance, rotating to a new station every two minutes. The fourth station independently selected a random playback melody for every minute of performance.

3. Development

The development of GUA emerged from a multi-musician, multifunctioning, collaborative environment to a consolidated, single-musician, multi-tasking instrument. Through iterative revisions¹[3], mostly borne of technical and practical demands, the major changes outline aesthetic shifts seeking to create a robust live-performance instrument.

¹ “Recently, a shift in musical interface design has been occurring, one in which users create new iterations of an interface, and become the driving force behind development.”

3.1 Software

The principal development in the software underlying GUA constituted a shift from the Chuck programming language to Max/MSP. The chief impetus for this development was visualization (feedback) of the buffers. Without visual cues, it was difficult to pinpoint a starting and ending point to a buffered sound, or even to be sure whether a sound was or was not properly recorded into the buffer. While using the `Waveform~` object in Max/MSP, a graphical waveform is visually written onto the screen as it is recorded into the buffer. This allows for precise selection during playback. Another benefit to Max/MSP is the wider variety of graphic possibilities in the creation of an interface (see 3.3 below). The ability to exponentially map a slider in Max/MSP has also proven useful in the construction of filters and reverb.

Despite the benefits of porting GUA to Max/MSP, there have also been drawbacks. Most notably, certain ready-made unit generators in Chuck have been more difficult to implement in Max/MSP, such as pitch-shift, reverb, and filtering.

The choice of using TouchOSC [4] to create the iPad interface was its similarity to the graphic components of Max/MSP. OSC communication was facilitated into Max/MSP with Rémy Muller's *OSCBonjour* [5] object.

3.2 Hardware

Limitations in mapping physical controls to the user interface necessitated modification of hardware devices. The single-touch control of the MacBook Pro trackpad hamstrung the use of MAUI sliders. The M-Audio Trigger Finger afforded greater control by the manipulation of multiple sliders, buttons, and knobs at once. The Trigger Fingers also allowed gestural nuance that is difficult to control with a single-touch track pad. Likewise, the Apple iPad replaced the Trigger Finger because of mapping deficiencies.

Despite having multiple programmable banks at an instant, GUA demanded a different number and set of mappable control elements. For example, controlling the playback of the buffers was unintuitive on the Trigger Finger. The lack of a method for labeling the controls of the Trigger Finger also made it difficult to learn and use.

By using the iPad with TouchOSC, we create multi-touch virtual interface that is customizable, lightweight, easy-to-use, and that has a one-to-one mapping scheme. While the iPad allows customizable, multi-touch controls, limitations encountered mainly related to touchable real estate on the surface.

3.3 Interface

The original interface of GUA used Chuck's MAUI toolkit. MAUI elements were limited to sliders, buttons, and LEDs. These were all controlled with the laptop trackpad. While these elements worked for many of the modules in GUA (playback volume, pitchshift value, panning), the sampler lacked visual feedback.

Chuck was graphically poor, being limited to its MAUI elements. The major developments of the GUA interface have come after the software was shifted to Max/MSP. The graphical nature of Max/MSP, and its 'Presentation Mode', facilitates creating an effective user interface. By following principles of design (Visibility, Uniform Connectivity, Mental Models, Visual Feedback, and Mapping [6][7]), the goal for the user interface in Max/MSP and TouchOSC has been to create an intuitive connection between the musician and modules being used. The controls follow models of physical device controls, including but not limited to buttons, sliders, knobs, and lights. Each module is grouped into common regions, bounded by

borders and color, in Presentation Mode giving it a compartmentalized feel.

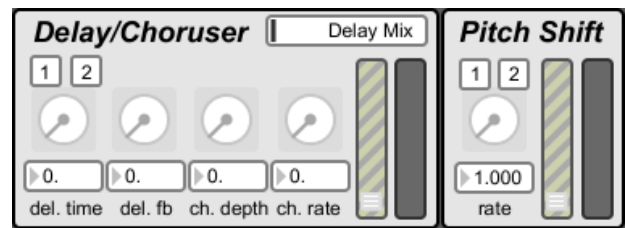


Figure 1. Portion of Interface in Max/MSP

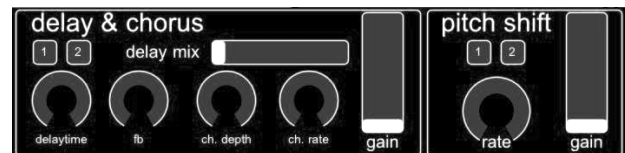


Figure 2. Corresponding Interface Portion on iPad

The graphical, multi-touch nature of the iPad allows the user interface to parallel that on Max/MSP. One slight shortcoming of TouchOSC (which parallels that of Chuck) is the inability to graphically represent recorded audio. While performers may easily step away from the laptop and control every aspect of GUA, they must stay within view of the laptop in order to see the `waveform~` object on the Max/MSP interface.

With each performance, needs for practical and musical revisions on the interface, both on Max/MSP and TouchOSC, were discovered. An example includes, number boxes to accompany sliders as added visual feedback, in both Max/MSP and the iPad.

Objects like `meter~`, `waveform~`, and `umenu` and other feedback tools raised the level of interaction between GUA and the musician. The heightened interaction ultimately raised the quality of musicianship in performance.

3.4 Other Developments

A Chorus effect and Feedback were both added to the Delay controls in the initial port. Later, a Reverb with variable 'room size,' 'decay time,' and 'diffusion,' and Filter were added. Several inputs were added (previously limited to one microphone input).

4. AESTHETIC DEVELOPMENTS

The technical and practical adjustments for GUA followed decisions made on musical goals and overall attitude toward the usage of GUA. All decisions culminated in the continued ideal to create an improvisatory instrument.

Three significant decisions for GUA have been consolidation of individual roles into one for a single performer, elimination of pre-recorded media, and streamlining of those previous roles into modularized interfaces affording precise and robust control.

Performance and practice have been the thrust for change and often come from personal opinions from the performers and developers. For example, the shedding of the pre-recorded media was a highly debated idea, but, in the end, liberated the acoustic performers to improvise more freely. For example, reworking the *gua* melodies into a more randomized yet integral role was an option. Instead, the idea of using the modules created for the *piece* GUA to sample and manipulate an improvising acoustic instrument shifted the direction of development. Since then, the developments have followed in the vein of improving on a single improvising engine/instrument capable of multiple sample/recording/live interaction.

The addition of the reverb and filter modules allowed the performers to create environmental effects.

For spatialization, we sought to use localized loudspeakers to project an individualized sound source, eliminating the necessity of panning. The constant consolidation of functions and ease of control helped push GUA to the status of instrument. Audio chain paradigms are constantly discussed: How/Should the elements in GUA follow another? Should the elements be more modularized? Should the performer have more flexibility to customize prior to/during the performance?

5. CURRENT STATE

5.1 Software

The current iteration of GUA has a control interface built in Max/MSP and a condensed control interface on iPad using TouchOSC.

5.2 Hardware

The hardware includes a laptop, an audio interface, and an iPad. The Max/MSP interface can be completely controlled on the laptop's trackpad, but the iPad allows for multi-touch interaction. A MIDI foot pedal was added recently to allow for greater simultaneous control.

5.3 Interface

GUA can be broken into several modules. These modules include delay/chorus, pitch shift, filtering, and reverb.

The Delay/Chorus module has variable delay time, delay feedback, chorus depth, and chorus rate. The Pitch Shift module uses the *pfft~* max object.

Each module has independent and multiple inputs (varying on the number of inputs) and output controls.

The number of samplers has settled at a total of eight, also with variable input (see Figure 3.1). The chaining of these module outputs follows our conceptual model of improvisation using acoustic sound sources.

The Max/MSP and iPad interfaces parallel each other in layout. While the iPad using TouchOSC is a convenient remote multi-touch interface controller, the limited real estate has affected choices in consolidating portions of the user interface.

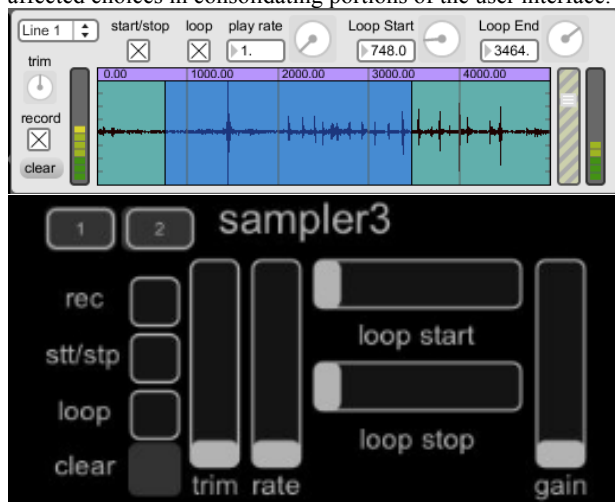


Figure 3.1 A sampler module in Max/MSP and TouchOSC

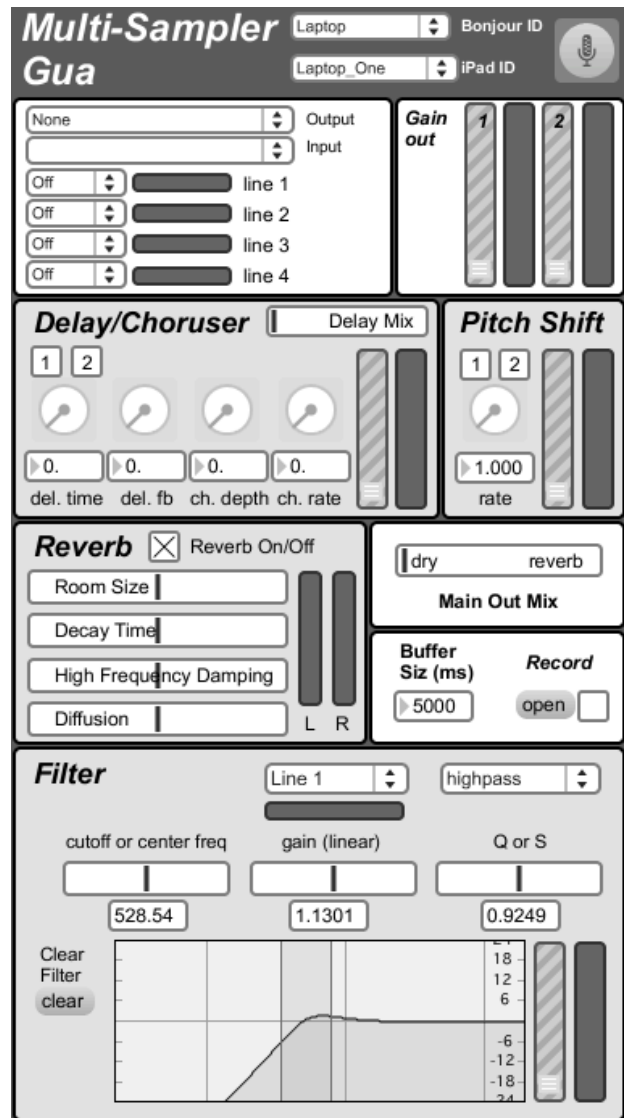


Figure 4. Interface of Modules (without Samplers) in Max/MSP

6. THE VERSATILITY OF GUA

6.1 The Expansion of Modules

With the variety of modules and independent inputs, GUA can be played in many ways, however the flexible live manipulation modules help it to excel at improvisation. In our performance history, the usage of modules varies often with the setting, sound sources, and other musicians, which speaks to the flexibility of the instrument.

6.2 Developing a Performance Practice

Having multiple sampling modules and capability to perform 'basic tape techniques',² GUA can be seen to have its roots from the ideals of musique concrète, MicroSound, and augmented instruments.

The ability to precisely control several parameters at an instant helps GUA to be an easily learned and mastered instrument capable of both subtlety and intensity. While a degree of expressiveness may be quickly achieved, a level of virtuosity can only come from experience and practice.

² Variable speed, looping, section splicing, and reverb.

7. CONCLUSION

The current build of GUA is highly portable, runs with Max Runtime on any platform, and needs only a wireless connection to an iPad. The iPad as an external interface allows a high level of interactivity, while allowing the performers to free themselves from the laptop screen. With its interface, GUA is easily playable, but also commands a high level of experience to perform well due to its large amount of modules.

With its high level of expressivity, as well as a plethora of modules, GUA has become a highly versatile instrument, capable of the subtle nuance and freedom to allow a personal connection with its performer. The evolution of GUA was not a preconceived progression, but an iterative developmental process. Discussion, experimentation, and performance, as well as changes and advancements in technology, have all propelled modification of GUA.

In the future, development plans include extensions to the external interface such as foot pedals, which will allow an even greater freedom from the laptop, and will free the performer from dependency to the iPad interface. Also, customizable chaining of modules will allow each performer to personalize GUA for his or her needs, allowing for a truly interactive environment.

8. REFERENCES

- [1] Chadabe, Joel. *Electric Sound*. Prentice-Hall, Upper Saddle River, NJ, 1997.
- [2] OpenSoundControl, <http://opensoundcontrol.org/>
- [3] Vallis, Hochenbaum, Kapur, "A Shift Towards Iterative and Open-Source Design for Musical Interfaces" Nime 2010 Conference.
- [4] TouchOSC, <http://hexler.net/>
- [5] R my Muller's OSCBonjour Object:
<http://recherche.ircam.fr/equipes/temps-reel/movement/muller/index.php?entry=entry060616-173626>
- [6] Norman, Donald. *The Design of Everyday Things*. Basic Books, New York, 2002.
- [7] Lidwell, William, Katrina Holden, Jill Butler and Kimberly Elam. *Universal Principles of Design*. Rockport, Beverly, Mass., 201.