Composing with Emergent Instruments in *Habitats*

Michael Lukaszuk
Queen's University
mpl@queensu.ca

Submitted: 2021-10-10

Abstract: Habitats (2021) is a multi-movement computer music composition by Michael Lukaszuk. Each movement consists of a dedicated algorithmic composition program that explores digital lutherie with extended versions of physical models from the Synthesis Toolkit and custom-built instruments whose bodies change over time. This article describes the manner in which resources from the ChucK programming language are used to shape these instruments, and how aspects of the design and composition of *Habitats* are informed by prominent conceptual frameworks used in the cultural studies field. With regards to digital lutherie, this article addresses how I sought to convey a sense of instrumentality using simulated bodies that are projected through use within a particular ChucK program, instead of actual physical instruments. The code and resultant audio for this piece are research-creation efforts that respond to our everchanging definition of composition. In linking my computer music practice and scholarly background in cultural studies, this article also aims to highlight how creative coding represents participation in the formation of culture. *Habitats* can be heard in full using the following link: https://michaellukaszuk.bandcamp.com/album/habitats. Code examples can be found using: https://www.michaellukaszuk.com/research.

1 Introduction

Habitats (2021) is a multi-movement computer music composition. Each movement consists of a dedicated algorithmic composition program written in the ChucK programming language (Wang, 2004). During the realization of the piece, there is no intervention through human performance. What you hear is the spontaneous formation of digital instruments that are built, modified and performed in reaction to the behaviour and properties other sound making objects that are also coming into being. The purpose of this article is to discuss compositional strategies with emergent instruments and how the creative use of constraints within the ChucK language represent a discursive approach to algorithmic composition that syncs with broader ideas about technology in culture. Do these virtual "performances" say anything about the increase of virtuality in our lives? Does the hacking and plundering of the language's unit generators parallel our regular interactions with the technologies that surround us? While there has been a great amount of research into technical aspects of algorithmic and generative music, this article responds to the idea that these genres and the tools that help form them are participants in the formation of broader cultural dynamics. In particular, I will discuss how this code-as-instrument practice in *Habitats* relates to important conceptual bases for the cultural studies field. This piece can be better understood as a research-creation effort when considered alongside concepts such as the rhizome, as presented by Gilles Deleuze and Félix Guattari. In *A Thousand Plateaus*, the rhizome is described "an acentered, non-heirarchical, nonsignifying system without a General and without an organizing memory or central automation" (Deleuze & Guattari, 1980). In this piece, the use of processes that simulate gesture benefit from additional analysis using Michel Foucault's idea of biopower as a conceptual framework. Biopower is characterized by a change in the strategies of social control from exclusionary toward disciplinary methods, the ultimate goal of which is to inculcate in free subjects the propensity for self-government and self- discipline…" (Karakayali & Alpertan, 2021).

Linking these ideas to instrument design and sonic behaviours opens room to consider how in addition to composing using computer-generated sounds, musical programming becomes a form of discursive practice. In this sense, the use of digital instruments in *Habitats* becomes a microcosm for digitality and the cultural significance of technology-driven automation, interactivity and decentralization in art and media production.

2 Models Inside-Out

The desire to emulate and extend familiar instruments and sound-making objects using sound synthesis has been a preoccupation of many electronic and computer musicians. As the capabilities of digital systems for recreating instrumental sounds has improved over time, the idea of the model has become a liminal space in which artists plunder the threshold between authenticity and sonic experimentation. Just consider the progression from the use of additive synthesis for emulation of bell timbres in Jean-Claude Risset's *Mutations* (1969) (Solomos, 2019), to the use of physical modeling synthesis to produce the sounds of an enormous flute in Paul Lansky's *Things She Carried* (1996) (Chafe, 2004). The increased complexity of the components used to create the model or emulation results in a greater number of possibilities for those who wish to at unearth and place the filters, delays and modulated oscillators closer the foreground of the sonic experience produced by these computer-generated instruments.

In the second movement of *Habitats*, *Low*, Perry Cook and Gary Scavone's Synthesis Toolkit (STK) (Cook & Scavone, 1995) served as both a set of FM and physical modeling UGens available in the ChucK language, and a jumping off point for digital lutherie. This was primarily achieved by extending the Mandolin and the BandedWG (for modeling various metallophones) classes. Thinking about the use of the mandolin model, an important compositional consideration was whether using a large array of Mandolin UGens would lead to the formation a single meta-instrument, or an ensemble of digitized mandolins. As the standard physical version of this instrument contains 4 courses, each containing 2 or 3 (in the case of the mandriola) strings, the mandolin morphs within a system of constraints that allow the corporeal features of the instrument to move around while still projecting a sense of instrumentality. Computer musician Thor Magnusson writes about how "a relatively high-level system of constraints, encapsulating a defined space for potential expression, whether of compositional or gestural nature" allows digital instruments to engender an identity (Magnusson, 2010). In *Low*, this augmented mandolin instrument is defined and constrained, by the manner in which it is played (establishing physical constraints through use), through the use of effects processing, and by linking parameters (e.g., frequency, gain, and function methods that contribute to a sense of plucking).

The construction of new instruments through use or performance in the digital music domain provides room to consider whether computer-generated musical actions exist as simulations or modifications of the plucking and striking of objects in the physical domain, or as the newly discovered motions of virtual instruments. It is my feeling that in *Habitats*, the idea of instrumentality is asserted, in part, through the use of repeated gestures. One of the uses of this new mandolin-esque instrument, the `MetaMando` class, is to provide a response gesture to various calls performed by other objects that help form this piece. When more than a dozen strings are consistently played as a response through a quick strum or arpeggio and performed with a sense of coherence that implies a single performer's action, the morphing of other components in the instrument such as the perceived body size become more believably "of the instrument." In her PhD dissertation on gesture in electronic arts, Vanessa Chang writes that "…through gesture, contemporary digital art practices borrow and recode forms of presences…to highlight a more diverse array of creative agencies." (Chang, 2017). This `MetaMando` that extends from the standard mandolin UGen relies on an array of 21 instances of the original class from the STK. This aspect of the instrument's design is meaningful in that a degree of variation in arpeggiated and strummed chord-type sounds could be produced using combinations of strings (i.e., array members) that do not appear similar to the 3, or 4-note chord structures that are commonly used in pop or folk mandolin playing. The patching the output of the entire `MetaMando` as a single input for effects processing modules such as bandpass filter banks is used throughout the entire movement as a compositional strategy

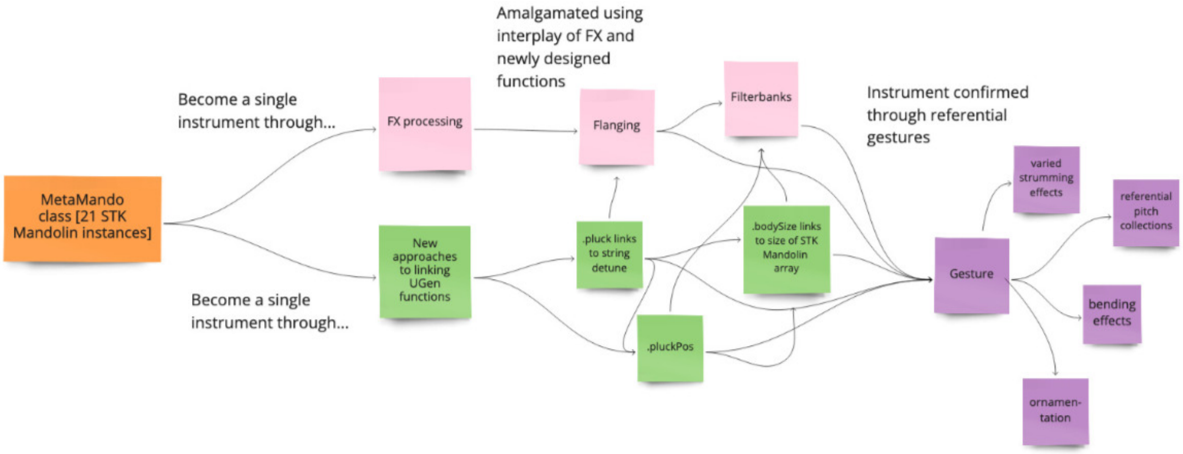
MetaMando class [21 STK Mandolin instances]
Become a single instrument through...
Become a single instrument through...
FX processing
New approaches to linking UGen functions
Amalgamated using interplay of FX and newly designed functions
Flanging
.pluck links to string detune
.pluckPos
Filterbanks
.bodySize links to size of STK Mandolin array
Instrument confirmed through referential gestures
Gesture
varied strumming effects
referential pitch collections
bending effects
ornamen-tation

Figure 1: Concept Map for `MetaMando` class

that encourages the listener to hear this plucked string sound as being of a new instrument and not as a carefully implemented computer-generated mandolin model.

Listing 1: Routing of the mandolin array

```
private class MetaMando extends Chugraph //
{
   PluckInfo pluckUpdate; //extension of Event class, for communication between functions

   Gain collapse;
   Gain mainOut => outlet;
   Mandolin mando[20];
   ResonZ filtBank[8];

   for (0 => int q; q < mando.cap()-1; q++)
   {
      mando[q] => collapse;
      3.0 => mando[q].gain; // compensates for filtering
   }
```

Listing 2: Linking parameters to convey a sense of one single instrument rather than an ensemble. Here, the number of strings played links to the body size of the mandolin model

```
for (0 => int r; r < stringz; r++) // # of strings, i.e. # of members of mando UGen array
    used
{
   if (gainDist == "rand")
      Math.random2f(0.1,0.9) => gainScale;

   else if (gainDist == "uniform")
      1.0 => gainScale;

   else if (gainDist == "melody")
      (1. - (stringz/21.)) + 0.01 => gainScale;

   (1. - (stringz/21.)) + 0.01 => mando[r].bodySize; // scales the bodySize from 0. to 1.0
```

While the use of effects processing such as bandpass filters is more of a superficial approach to projecting a new sense of instrumentality, other aspects of the design of this `MetaMando` component within *Low* go deeper and help project a more spontaneously changing single instrumental body. One example is the link between the number of strings used and the `.bodySize` function in the mandolin UGen.

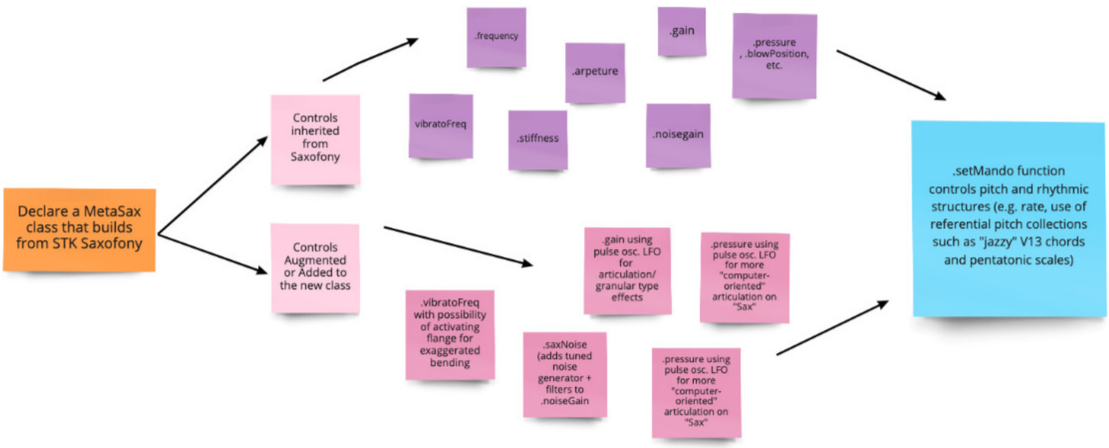
Declare a MetaSax class that builds from STK Saxofony
Controls inherited from Saxofony
Controls Augmented or Added to the new class
.frequency
.arpeture
.gain
.pressure , .blowPosition, etc.
vibratoFreq
.stiffness
.noisegain
.gain using pulse osc. LFO for articulation/ granular type effects
.pressure using pulse osc. LFO for more "computer-oriented" articulation on "Sax"
.vibratoFreq with possibility of activating flange for exaggerated bending
.saxNoise (adds tuned noise generator + filters to .noiseGain
.pressure using pulse osc. LFO for more "computer-oriented" articulation on "Sax"
.setMando function controls pitch and rhythmic structures (e.g. rate, use of referential pitch collections such as "jazzy" V13 chords and pentatonic scales)

Figure 2: Organization of method functions for `MetaSax` class

According to treatment of the `.bodySize` method shown in Listing 2, a set number of 21 strings would yield the largest possible body size of this virtual mandolin, whereas the instrument would appear to shrink based on the use of only a small set within the `MetaMando` array. Other attempts at affirming a single `MetaMando` include the linking components from the inherited Mandolin UGen class such as the `.pluck` and `.stringDetune` functions. In *Low*, they are used to allow for intensely exaggerated plucks that are not possible with actual mandolins. In the `MetaMando` class, the `.pluck` function of the Mandolin STK UGen is extended so that it can create exaggerated effects that beyond the sonic results produced 0.0-1.0 range held in the Mandolin UGen. For especially intense detuning effects, timed flanging is made active when the arguments for `.pluck` and `.stringDetune` functions exceed their permitted ranges.

The final movement of *Habitats*, *Bromelaids*, takes a similar approach to the reconfiguration of an STK physical model, in this case using the saxophone Saxofony UGen as the foundation for the design of a broader a code-based `MetaSax` instrument. For example, the use of a pulse wave as a "switch" for parameters beyond the `.gain` method of the Saxofony UGen allows for the sense of an instrumental performance using instantaneous changes in physical parameters such as amplitude (i.e., gain) and breath pressure. This feature becomes part of the software instrument as such effects cannot be updated in a similar fashion with actual saxophones. This `MetaSax` class inherits Saxofony, and at times reconfigures its components by using a similar idea of linking parameters and exaggeration that can be found in the `MetaMando` from the second movement of *Habitats*, *Low*. For example, when values that surpass the normal range for the `.noiseGain` function has been exceeded, the output of the Saxofony UGen within the `MetaSax` class is blended with an instance of the CNoise noise generator UGen with a series of bandpass filters that are tuned to the fundamental frequency and a small number of partials. This reconfiguration presents a version of the model that emits far more incidental noise than would normally be available using an actual saxophone.

Listing 3: The use of an overloaded gain function with added pulse wave. This creates a kind of ornamentation that is relevant to the digital realization of a saxophone

```
//overloaded gain func
fun void gain(float gainy,float switchFreq, int switchRate, string switchy) // switch rate
    is a kind of "grain rate"
{
   Shred square1;

   switchFreq => switchGainLFO.freq;

   switchRate => int rateSq;

   if (switchy == "On")
```

```
      spork ~ square1Gain(rateSq,gainy) @=> square1;

   if (switchy == "Off")
      square1.id() => Machine.remove;

}

fun void square1Gain(int switchRate, float gainy)
{
   float squareLo;

   while (true)
   {
      Math.fabs(switchGainLFO.last()) => squareLo;
      squareLo * (gainy) => this.gain;
      switchRate::ms => now;
   }
}
```

Listing 4: The MetaSax program uses a filter bank to tune noise to the approximate attacks and frequencies of ChucK's Saxofony UGen when the .noiseGain function of the MetaSax class exceeds the normal range of 0.0-1.0

```
else if (noise > 1.0)
{
   spork ~ saxEvent() @=> updateSax;

   rezEnv.keyOn();
   0.5 => this.noiseGain;

   for (0 => int o; o < noiseTones.cap()-1; o++)
   {
      70 => noiseTones[o].Q;

      this.freq() * (o + 1) => noiseTones[o].freq; // overtones approx.

      if (o != 1 && o != 5)
         1.0 - o => noiseTones[o].gain;
      else
         (1.0 - o) * 1.5 => noiseTones[o].gain;

      noiseBump => noiseTones[o] => resonBump;
   }
}
```

3 Becoming

These adaptive interpretations of STK models used in *Low* and *Bromelaids* share similarities with more ubiquitous technologies that we encounter in ordinary life. They highlight the manner in which creative coding does not only involve the construction of new sound making devices but expresses a form of what cultural theorist Stuart Hall describes as "conjunctional knowledge." For Hall, this refers to knowledge that is relevant to immediate cultural/political circumstances. This includes research-creation efforts capable of acknowledging that the structures that form culture are instruments of power (Hall, 2007). Media artist and scholar Olga Goriunova supports the cultural expressivity of code art by arguing "…that software is a culture in its own right. And that code, as a language system, reflects at least two cultures, that is software and that of the coder's context…" (Diamond, 2013). The rhizome can act as a useful conceptual lens for understanding the cultural significance of digital technologies in computer music (e.g., programming languages) and beyond. In *A Thousand Plateaus*, Deleuze and Guattari's discussion of rhizome emphasizes the principle of asignifying rupture, in which the state of being acentered allows some of the lines that form a structure to break without impeding its function. In his book *Understanding Digital Culture*, sociologist Vincent Miller provides useful links between digital technologies and the rhizome. Miller focuses on the internet and how it encounters older media and devices such as news and telephones through websites and software applications, suggesting that the system grows through transformation and not reproduction (Miller, 2011). This is kind of "becoming" is paralleled in Habitats with the idea of forming single meta-instruments that grow out of inherited classes from the STK that add new capabilities by augmenting and connecting existing function methods. The `MetaSax` serves as a good model. It grows out of the STK Saxofony UGen in ChucK and possesses controls that are transformations of the UGen's original `.freq`, `.noiseGain`, etc. methods, but overloaded—i.e., the augmented versions of the same parameter are available, using the same name, and accessed when new behaviours (e.g., exaggerated instrumental noise) are used.

Another parallel between *Habitats* and this rhizomatic sense of becoming is the way that these new instruments behave within the programs that are run to form each piece. The main generative strategy for this piece was for each movement's program to act as an ecosystem where actors (i.e., digital instruments played by the computer) depend on one another, but in a fragmented manner that avoids any kind of central automation. For example, the final movement, *Bromelaids*, a pitch tracker can be activated to catch and "drag" around parameters (e.g., `.gain`, `.freq`, or even instrument specific functions) belonging to instruments in the ensemble. A lack of apparent hierarchy is maintained by the use of random number generators that trigger the function based on the possibility of matching another randomly chosen number.

Listing 5: Declaring an instance of Blip, a custom instrument that extends simple oscillator UGens to give a sense of discrete control over a number of overtones

```
///////////////////////////////////////////////////routing/signal chain////////////
Blip blip1 => Chorus chor => NRev rev => Gain mainOut => dac;
// Blip class here is an emulation of Blip Ugen from SuperCollider
// (impulse generator with a discrete control over # of harmonics)
Saxofony saxo[4];
```

Listing 6: An example of linked parameters in which the system permits interaction through an instrument's ability to "catch" or "drag" other players until another function triggers a release

```
fun void blipPlay() {
   while (true)
   {
      float blipPitch;
      if (pitch.get() == 0)
         55 => blipPitch;
      else
         pitch.get() => blipPitch; // when this function has been made active
                               // pitch detector "drags" the Blip class' fundamental Hz around
                               // until another function triggers the release
      1.0 => blip1.blipFades;
      blip1.blipOn(blipPitch, Math.random2(5,17), "harmonic", 0.5,0);
      10::second => now;
   }
}
```

Vincent Miller's discussion of digitality in culture also points to how digital technologies such as internet exist in a state of tension between the rhizomatic and the arboreal in the way commercial (e.g. advertising) and political pressures (e.g., censorship) can begin to form roots and hierarchies (Miller, 2011). Similar dynamics occur in electronic/computer music composition where despite workarounds, the use of beat-oriented time displays and the divisions of notes in a MIDI editor using a 12-note octave represent a kind of cultural residue that roots the creative process. The manner in which digital instrument builders are especially drawn to the use of free and open-source software (FOSS) is reflective of a resistance to the roots formed by tools that result from biases that often stem from the commodification of electronic/computer music production. In *Habitats*, the reliance on a particular set of FOSS technologies for the realization of new works reflects a point of view that digital instruments, and the concept of code-as-instrument in particular, are more than informational technologies but instead part of a desire to express cultural positionalities. The manner in which this approach leans into the rhizomatic is a response to the inherited Eurocentric model for composition that prioritizes fixed, rendered and fully realized works that tend to permeate formal music education settings. The relationship between FOSS, code and self-expression can be traced back decades ago to the controversy surrounding AT & T and the privatization of the UNIX operating system (now used as the basis for Apple's Mac OS). The way in which programmers rallied against AT&T demonstrates a feeling of identity in the development of technologies (Kelly, 2012). The manner in which musical programming languages such as ChucK allow users to continuously customize and update the bodies of their instruments offers a similar space to discuss identity that cannot be accessed so easily in the digital audio workstation.

How else does *Habitats* connect to the rhizome, and what is to be gained from this connection? The idea of asygnifying rupture, can be linked to more ubiquitous forms of technology such as mobile phone apps that depend on participation such as Facebook and similar social media services, where the removal of a branch or node (e.g., a participant or group) does not compromise the system itself. This feature can be found in other forms of FOSS-dependant computer music. David Ogborn, who directs the Cybernetic Orchestra at McMaster University, has written on the diverse routes in which code travels, and the possibility for scalability beyond ensemble membership as characteristics that allow for non-hierarchical sharing and participation (Ogborn, 2014). The Birmingham Ensemble for Electroacoustic Research (BEER) also performs using an approach that moves beyond the notion of the and corresponding to a single electronic instrumentalist in their use networked technologies (Wilson, 2014). This characteristic of the rhizome is embraced in *Habitats* by funneling sound output into looping and stretching effects processing where both recordings from previous instances of the instrument earlier and the piece in combination of with live input produce the overall output being sent to ChucK's audio output. This is one of the main features of the third piece from the set, *Hydrilla*. Using an instances of the Spectacle UGen from ChucK, an FFT delay that produces spectral time stretching, and the LiSa live sampling utility, the program for *Hydrilla* can asynchronously recall earlier sonic gestures from instruments that may be silent or interfered with due to the kind of "dragging" functionality discussed with regards to Listing 7. As shown in the listing below, the `chirpPluck` (a high-pitched tremolo type gesture) is called within a particular moment, it is also being sent for later use by Spectacle and LiSa for spectral delay and looping. Viewing *Habitats* in relation to the rhizome highlights the possibility for future versions of the work that plunder the generative processes in different ways and even encourage interactivity with human actors. An important part of the future work for this piece will be an exploration of digital lutherie and rhizomatic transformation that gives space for simulated and actual/physical instruments. As *Habitats* is simultaneously a piece and a group of programs, there is room for it to grow even further from the arborescent to the rhizomatic through augmentation, extension and inheritance of digital and physical instruments.

Listing 7: Storing output for later use by a spectral delay as "asignifying rupture"

```
fun void chirpPluck (float stretch)
{
   stretch => pluck.stretch; // spectral delay times for when audio is later recalled
   [1.,1.,1.,1.] @=> float noTrans[];
   [3.5,2.0,1.25,0.33,1.5] @=> float transIt[];
   0 => int h;
   float panAxis1;
   10::second => pluckEnv.duration;
   float next;
   0 => int q;
```

```
while (true)
{
   Math.random2f(-0.6,0.5) => panAxis1;
   pluckEnv.keyOn(1);
   while (true)
   {
```

4 Biopower

Michel Foucault's concept of biopower is used to describe a form of control in society that is generative, continuously organized and "…internalized by subjects rather than exercised through acts or threats of violence." (Taylor, 2011) While Foucault's own discussion of power dealt with large social issues such as discipline in penal systems (e.g., his 1977 book *Discipline and Punish*), some scholars have adapted the idea to look at aspects of digital culture. In the digital domain, the idea of computer hacking supports Foucault's idea of biopolitical systems requiring "continuous regulatory and corrective mechanisms." For Matthew Kelly, who writes on digital literacies and new media, the emergence of error and bugs (as produced by hacking) is a necessary component of more elaborate and efficient forms of progress (Kelly, 2012). In "If…Then: Algorithmic Power and Politics," Tania Bucher responds to biopower with the idea of "productive power" as emphasizing the "…significant role users have in reconfiguring the algorithmic spaces that they themselves inhabit." (Bucher, 2018) Bucher highlights how algorithms present in prevalent technologies such as Facebook's new feed are shaped once become aware of and encounter them. The musical works of creative coders resonates with the idea of algorithmic power when we think of hacking as reconfiguration. Similar algorithmic music-creation practices that depend on spontaneous changes in lutherie such as live coding show a precedent for growth through error. In a TEDx presentation, live coding artist Alex McLean tells us that "live coders are much more interested in causing problems than solving them." (McClean, 2017) As a kind of virtual performance within a computer-generated sonic environment, the emergent ensemble that is played through the software (i.e., ChucK code) for *Habitats* resonates with the idea of discipline through continuous self-correction and a kind of looped, self-imposed hacking as the instrument's bodily features are continuously reformed during the process of playing (i.e., running an instance of) the piece. It is through the interplay of hacking (e.g., augmenting and extending inherited parameters of an STK physical model) and the assertion of this new identity for the object that a convincing realization of the piece is made.

In *Habitats*, the connection to biopower is present in the kind of shifts in bodily discipline that emerge from the behaviour of the digital ensemble that is active in each work's ChucK program, and its potential to respond to traditional notions of virtuosity. In *Sonic Writing*, computer musician Thor Magnusson describes how virtuosity in digital instruments has shifted from confronting the edges of the constraints found in a performer's body to plundering the constraints of the instrument (Magnusson, 2019). This kind of virtuosity –based on fluid interactions with code-based instrumental "bodies," can be found in *Tardigrade*, the fourth movement of *Habitats*. Its program is focused on the use of a custom-build "micro-panning" class that produces audio artifacts resulting from changes in panning that are instantaneous or microsonic (approximately less than 100ms) in duration. As with granular synthesis, the shape of the grain envelope that affects the resultant timbre of grains joined into clouds and streams is used to alter timbre of panning artifacts by using a rapid tremolo with different oscillator shapes (i.e., the jagged edge of a sawtooth wave produces different artifacts than that of a smoother sine curve.) Virtuosity in this sense is the program pushing itself to explore thresholds where it is capable of producing increasingly audible and sonically varied artifacts that stem from the kind of malfunction that would typically be avoided. As with the rhizome, the relationship between this piece and biopower benefits from the piece's generative nature and the possibility of revisions that allow for human interaction with the unfolding musical processes–drawing on Bucher's notion of reconfiguring algorithmic spaces.

Listing 8: Interaction with sampling rate (48,000 Hz) settings affects envelope durations

```
while (true)
{
   if (sampCount == 48000 * 10)
      0 => sampCount;
   if (sampCount < 48000 * 5)
   {
      0 => w.gain => vc[0].gain => vc[1].gain; // creates down env
      Math.fabs(lfo1.last()) * 0.2 => s.gain;}
      else
      {
         0 => s.gain;
         Math.fabs(lfo1.last()) * 0.2 => w.gain => vc[0].gain => vc[1].gain;
      }
      1::samp => now;
      sampCount++;
   }
```

Rituals, the first movement of *Habitats*, times certain parameters based on samples and not fixed values such as seconds or minutes. Based on the sampling rate that is specified to ChucK, and even possibly one's desire to venture beyond the 44.1kHz or 48kHz standard into the extremes of high or low fidelity, the program interprets this change and stretches or truncates timed events such as how long it takes for a "beeping" sound to switch between using a sine tone and the STK model of a singing voice. *Rituals* also draws on biopower in the use of a musical motive that is emitted approximately every 45 seconds to 1 minute that consists of a STK BandedWG UGen using the Tibetan prayer bowl as the instrument preset for the model. This is a "resetting gesture" that can reset various arguments being passed to UGen functions and can "unstick" parameters that are previously linked. The consideration of algorithmic music behaviours as being connected to biopower syncs with Foucault's interpretation of power dynamics being process-dependant, instead of inherently placed within people or spaces. The rhythms and orderings of components represents an ongoing negotiation of force (Carmi, 2020).

5 Conclusion

Media scholar Friedrich Kittler's essay *Gramophone* argues that technology is made viable by its demand or use within culture and not by presence or availability of a resource or required skills (Kittler, 2007). This is complicated in digital music making that takes place in an algorithmic or generative piece such as *Habitats* in that instruments and performance practices can emerge spontaneously through extending, hacking and using repeated gestures within a set of constraints that define instrumentality. Kittler's position, in which development and demand exist as two separate stages, is incompatible with this kind of sound work in that it diminishes the cultural expressions that emerge from the ambiguous identity that digital, and especially code-based instruments assume. Computer-generated mandolins and saxophones with exaggerated and changing bodies may provide access to cultural expressions that allow users to contemplate digitality in culture in a way that is not available using the actual versions of these instruments. It is important to consider how process-based sound works that emerge from creative coding belong to larger cultural dynamics that are fueled by the ubiquity of computing in our everyday lives. Engaging with significant sources from the cultural studies field has encouraged by my own journey towards the use of custom tools that allow me to engage with algorithmic/generative music beyond just a set of interesting sounding processes or technical experiments. Exploration of concepts such as the rhizome and biopower affirm that this kind of creative coding does not constitute an experiment in a vacuum. By considering computer music composition alongside other features of digital cultures (e.g., mobile networks), we can understand how the building and playing of digital musical instruments become acts of cultural participation—that I am saying something about behaviours and habits as I experience them in my own world. Therefore, the algorithmic strumming of my `MetaMando`, for example, is based on my interaction with other simulated forms of creativity. In the words of

Nicholas Negroponte, founder of the MIT Media Lab, “Computing is not about computers anymore. It is about living.” (Negroponte, 1996). It is my hope that *Habitats* provides a framework for similar research-creation efforts, and leads to the production of future work dedicated to the study of digital instruments in practice.

6 References

Bucher, T. (2018). *If. . . Then: Algorithmic Power and Politics.* Oxford University Press. https://doi.org/10.1093/oso/9780190493028.001.0001

Carmi, E. and Jones, S. (2020). *Media Distortions.* Peter Lang Inc. https://doi.org/10.3726/b15334

Chafe, Chris. (2004). Case Studies of Physical Models in Music Composition. *Proceedings of the 18th International Congress on Acoustics.* https://ccrma.stanford.edu/~cc/pub/pdf/phyModMusComp.pdf

Chang, Vanessa. (2017). *Tracing Electronic Gesture: A Poetics of Mediated Movement.* (Doctoral Dissertation, Stanford University, Stanford, United States). Retrieved directly from author.

Cook, P. R., and Scavone, G. (1995). The Synthesis Toolkit in C++ [Computer software]. https://ccrma.stanford.edu/software/stk/

Deleuze, G. & Guattari, F. (1980). *A Thousand Plateaus: Capitalism and Schizophrenia.* Athlone Press.

Diamond, S. (2013). Reframing the Cathedral: Opening the Sources of Technologies and Cultural Assumptions. In Arthur and Marilousie Kroker (Eds.), *Critical Digital Studies Reader.* University of Toronto Press.

Hall, S. (2007). Cultural Studies and its Theoretical Legacies. In Simon During (Ed.), *The Cultural Studies Reader* (pp. 97-113). Routledge.

Karakayali, N. & Alpertan, B. (2021). Mood playlists, biopower, and the “functional turn” in online media: What happens when a pre-digital social control technology is transferred to the internet? *The Information Society*, 37(1), 20-34. https://doi.org/10.1080/01972243.2020.1826616

Kelly, M. (2012). All Bugs Are Shallow: Digital Biopower, Hacker Resistance, and Technological Error in Open-Source Software. *CTheory.* https://journals.uvic.ca/index.php/ctheory/article/view/14987

Kittler, F. (2007). Gramophone. In Simon During (Ed.), *The Cultural Studies Reader* (pp. 298-304). Routledge.

Magnusson, T. (2010). Designing Constraints: Composing and Performing with Digital Musical Systems. *Computer Music Journal*, 34(4), 62-73. https://doi.org/10.1162/COMJ_a_00026

Magnusson, T. (2019). *Sonic Writing: Technologies of Material, Symbolic, and Signal Inscriptions.* Bloomsbury Press. https://doi.org/10.5040/9781501313899

McLean, Alex (2017, June 2). *Algorithmic Dance Culture* [Video]. YouTube. https://www.youtube.com/watch?v=nAGjTYa95HM

Miller, V. (2011). *Understanding Digital Culture.* Sage Publications.

Negroponte, N. (1996). *Being Digital.* Vintage.

Ogborn, D. (2014). Live Coding in a Scalable, Participatory Laptop Orchestra. *Computer Music Journal*, 38(1), 17-30. https://doi.org/10.1162/COMJ_a_00217

Solomos, M. (2019). *From Music to Sound: The Emergence of Sound in 20th- and 21st-Century Music.* Routledge. https://doi.org/10.4324/9780429201110

Taylor, C. (2011). Biopower. In Diana Taylor (Ed.), *Michel Foucault Key Concepts.* Acumen.

Wang, G. (2020). *ChucK : Strongly-timed, Concurrent, and On-the-fly Music Programming Language* (Version 1.4.0.0) [Programming language]. https://chuck.cs.princeton.edu/release/

Wilson, S. (2014). Free as in BEER: Some Explorations into Structured Improvisation Using Networked Live-Coding Systems. *Computer Music Journal*, 38(1), 54-64. https://doi.org/10.1162/COMJ_a_00229