SkipStep: A Multi-Paradigm Touch-screen Instrument

Avneesh Sarwate
Department of Computer Science
Princeton University, Princeton, NJ, USA
avneeshsarwate@gmail.com

Jeff Snyder
Department of Music
Princeton University, Princeton, NJ, USA
jeff@scattershot.org

ABSTRACT

SkipStep is a novel touchscreen application that allows users to write, perform, and improvise music on multiple virtual instruments via MIDI. It is a looping-based instrument that incorporates step sequencer and keyboard inspired interfaces with generative music capabilities. Additionally, SkipStep allows for collaborative performance over Wi-Fi, allowing users to either send musical material between separate SkipStep instances or collaboratively edit a single SkipStep file. This paper will provide the motivation behind SkipStep's design, review similar interfaces, describe SkipStep, present a user evaluation of the interface, and detail future work

1. INTRODUCTION

SkipStep is an attempt at bridging the gap between tools meant for "composing" and tools meant for "performing." What we refer to as composing tools could include digital sheet music based tools, such as Sibelius, digital recording tools, or digital audio workstations (DAWs) that provide sequencing capabilities, such as Pro-Tools. Such tools provide users with a detailed representation of the music being written and give precise control over how the music can be edited, but providing this degree of control and detail often necessitates relatively complex interfaces, which can become unwieldy for rapid input and performance of live music. Additionally, these composing interfaces can take time to get comfortable with both for inexperienced musicians, who may be unfamiliar with the roles of the controllable parameters, and for musicians new to the interface itself.

On the other end of the spectrum are performance-oriented interfaces, which are tools that allow real-time control over music in a live setting. These include DJ and mixing decks and software, such as various products by Traktor and Serrato, as well as sampling and triggering based tools, such as Ableton's Push controller and Native Instruments' Maschine. Performance-oriented tools often provide physical hardware whose actuation triggers some instantaneous change in the music being produced. Of the performance-oriented tools, the DJ and mixing tools are designed for a specific purpose and have almost no affordances for "composition" in a general sense. Ableton Push and Maschine, while very powerful, need to be set up before performance use, and new users could face a steep learning curve when trying to set up the software and hardware.

2. GOALS

The main goal in building SkipStep is to create a fast, easy to use *hybrid* tool by incorporating the most important features of both composing and performing tools by leveraging the touch screen as a way to streamline the interface. The motivation for creating a "hybrid" instrument is two-fold – having a UI that is fast and intuitive enough for live performance may be useful in a "composition" tool, as the tool would not get in the way of entering musical ideas into the software. Having the more detailed graphical display of a "composition" tool would ideally give performers more information about the music they are working with, thus aiding them when performing or improvising live.

Multi-instrument performance by a single user is another goal of Skipstep, and therefore the software aims to give the user the ability to control multiple instrumental voices at once.

SkipStep has generative music capabilities, and another goal of SkipStep is to create an interface that allows for the live use of algorithmic music generation. SkipStep is designed to allow users to seamlessly move between manually and algorithmically creating music in real time, even while performing or improvising.

Conceptually, if we think of a melody as a point in some "musical space", we envision SkipStep as a tool that lets users explore this subspace around this point (which is the space of similar melodies) rapidly and deliberately: users could quickly explore the space of variations around a single melody via the many algorithmic features for quickly modifying a melody. Users could also use SkipStep to quickly create totally new melodies, thus generating a new subspace to explore. With the affordances of live multi-instrument control and live algorithmic music generation, we hope to explore new modes of musical performance and improvisation, both individually and in groups.

3. RELATED WORK

There are several widely available tools that share the features of both "composing tools" and "performing tools." Ableton Push is a powerful hardware device that integrates with the Ableton Live DAW, allowing users to use Push as, among other things, an arrangement tool, step sequencer, and keypad. For use as a keyboard/pad, Push's velocity sensitive pads give it an expressive advantage over most touch screen applications. However, Push's minimal textual display area can make switching between the various modes complicated for a new user. Maschine is a similar integrated hardware/DAW product with much of the same functionality, and its twin screens give it much more display flexibility. Currently, however, it's video screens only function when using the Maschine DAW, limiting its flexibility.

Nodal is a network-based generative music tool that allows for both composition and real time performance and improvisation [3]. Users set up networks where nodes correspond to certain notes or chords, and the nodes play their pitch content when a network-traversing object called a "voice group" reaches the node. These nodes can be connected such that the network can be traversed either deterministically or randomly. Users can either "compose" using Nodal by creating networks that play without user interaction or manually trigger networks to "play" using MIDI messages, thus allowing them to be used as part of an instrument. While Nodal and SkipStep offer some similar functionality, Nodal dispenses with the representation of music as pitch vs time on an X/Y plane, whereas in SkipStep that idea is a crucial part of the interface.

Notesaaz is the musical performance interface that inspired SkipStep's approach to "musical space" [1]. In Notesaaz, the performer is able to control two different sets of lines moving on a screen, with their intersection determining the sound produced. Performers first use the controller to create a "score" by setting the position of the first set of lines, and then use the controller to freely move the second set of lines over the first, thus "playing the score" by creating new intersections and new sounds. The statically positioned first set of lines allows users to define a musical space, creating a loose limitation on the type of music that can be produced. The movement of the second set of lines allows users to explore that space. In SkipStep, users can similarly define a musical space by creating a melody on the grid and then explore the space of variations on that melody.

4. SOFTWARE COMPONENTS

The iPad interface for SkipStep was built using TouchOSC, an app for the iPad that allows for the building of UIs in a graphical editor by dragging and dropping UI widgets onto a screen. The SkipStep back-end is run on a laptop, and the iPad and laptop communicate via Wi-Fi using the OSC protocol [7]. The back-end was written in Python. ChucK was used to control timing and to send MIDI messages to a DAW, which was used to synthesize the sound. Python and ChucK communicated via OSC, using the pyOSC[1] library [6].

SkipStep has a multi-user mode where users can send melodies to each other. LANdini, a networking utility by Jascha Narveson, was used when multiple SkipStep instances were coordinated in multi-play mode [4]. LANdini simplifies the sending of OSC messages between multiple users on the same Wi-Fi network, and can also provide guaranteed delivery for selected message types.

[1] https://trac.v2.nl/wiki/pyOSC

5. FEATURES

5.1 "Painterly" Note Input

To make SkipStep a tool suitable for both composition and performance, we wanted to create a notation-input interface that was immediate, expressive, and visually informative. We also hoped to design what Golan Levin described as a "painterly interface" [2]. According to Levin, the design of a painterly interface should aim for the following (quoted):

- The system makes possible the creation and performance of dynamic imagery and sound, simultaneously, in real-time.
- The system's results are inexhaustible and extremely variable, yet deeply plastic.
- The System's sonic and visual dimensions are commensurately malleable.
- The system eschews the incorporation, to the greatest extent possible, of the arbitrary conventions and idioms of established visual languages, and instead permits the performer to create or superimpose her own.
- The system's basic principles of operation are easy to deduce, while, at the same time, sophisticated expressions are possible and mastery is elusive.

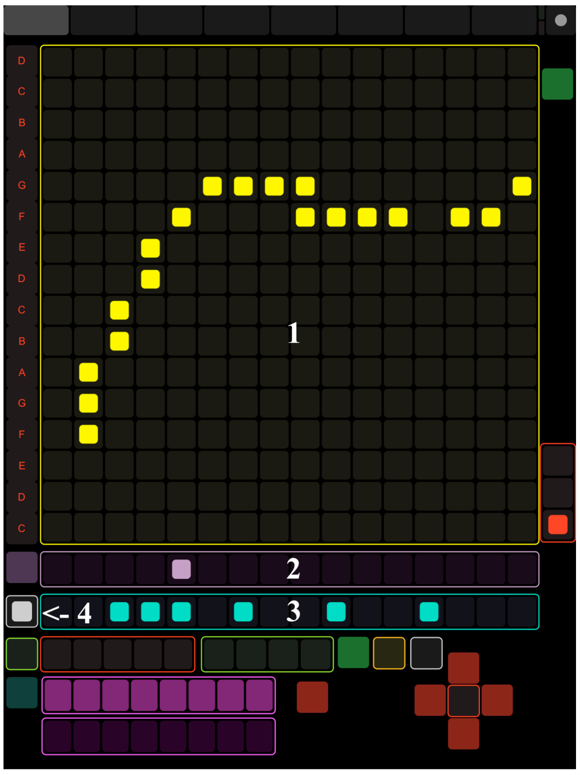

Figure 1. Painterly interface controls

We found the step sequencer interface, particularly one as implemented on a touchscreen, to meet these goals. A step-sequencer incorporates the 2D pitch vs time representation of music in a minimal, intuitive fashion. What is particularly attractive about touchscreen step sequencers as opposed to physical hardware ones is that users can create melodic contours by simply swiping across the screen. This allows for a very fluid, intuitive input of musical material, especially for users with no musical experience. For greater expressive control, users can change the scale played by the step sequencer as well as transposing the entire sequencer.

We wanted to include equally intuitive, visual controls for varying playback of the melodic content of the grid, providing ways to explore the musical space defined by the grid without explicitly altering it (and thus changing the musical space). The step jumping and subset looping controls serve this purpose. The purple dot (Figure 1, element 2) is always underneath the column that is being played by the step sequencer. If the user presses under some other column, the step sequencer will play that column next and continue playing forward from that column. This allows users to disrupt the linearity of the loop they have created and is similar to the way a turntablist can reset the flow of a record by physically spinning it to different positions.

The subset looping feature allows users to select a subset of the columns (Figure 1, element 3) and when subset looping is turned on (Figure 1, element 4), the step sequencer only loops (still in order) over those selected columns. The columns can be added or removed while subset looping is on, allowing for a very dynamic, interactive looping experience. Together, step jumping and subset looping allow users to engage more deeply with a "static" loop/grid.

5.2 Interactive Algorithms

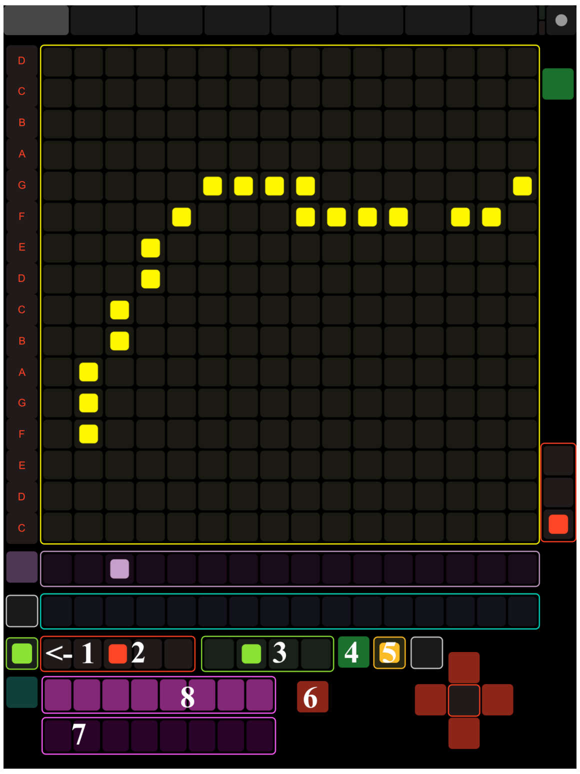

Figure 2. Algorithmic interface controls

SkipStep currently implements a simple (but soon to expand) set of algorithms that can modify whatever music is displayed on the main grid. We focused on interactivity in designing both the algorithms and the interface for their use. All of the algorithms take 2 parameters: the grid to be modified and the intensity of the modification (an integer from 1-5). The algorithm to be used can be selected via element 3 in Figure 2, and the intensity set via element 2 in Figure 2. The algorithms can be applied manually, via element 4 in Figure 2, or can be set to "automatic", so that they are applied every time a loop starts, via element 1 in Figure 2. There is also an "undo" button (Figure 2, element 6) that restores the grid to the state before an algorithm is applied. There as also an auto-undo mode (toggled via Figure 2, element 5): when this is turned on, a "snapshot" of the current active grid is taken. Any modifications to the active grid, (whether manual or algorithmic) are played once and then reverted backed to the snapshot.

Users can save up to 8 different grid patterns, and can instantly recall them to loop. To save the grid currently on the screen, users simply click on an unlighted bank (Figure 2, element 7), which then lights up, indicating that a grid is saved in it. To load a bank and loop it, users can press the button (Figure 2, element 8) above the lighted bank they wish to load. These interface features were chosen to encourage iterative modification of music, allowing for the seamless interaction of algorithmic and manual modification.

The algorithms implemented in SkipStep are intended to be transformative algorithms, which modify an existing grid, rather than generative algorithms, which could populate an empty grid. Since we did not want to arbitrarily limit the type of grids that are output by the algorithms, the algorithms focus on geometric approaches to creating variations rather than relying on music theory heuristics, considering only the shapes of the grids rather than the notes they encode. All of the algorithms are probabilistic, and the parameters were chosen so that they provided a high amount of variation, but did not produce overly chaotic output. The algorithms were designed to allow for the rapid development of complex patterns from a simple melodic contour. The algorithms are:

5.2.1 Semi-Structured Random Variation

Let k be the intensity value. Let the "neighborhood" of a grid element e be all other elements that are within 5 vertical steps and 3 horizontal steps of e. Let v be a random variable with outcomes p, q, r, and s. In this random variable, p is the event that e is turned off, q is the event that e is moved to another location in its neighborhood, r is the event that an off element in e's neighborhood is turned on, and s is the event that no modification is made. In event q, the location to which the new element that is added or the old element is moved is distributed uniformly over e's neighborhood. All 4 probabilities are parameterized by k. For each "on" element in a grid, a random trial of v is run to see which outcome of $\{p, q, r, s\}$ occurs. See Figure 3 for an example.

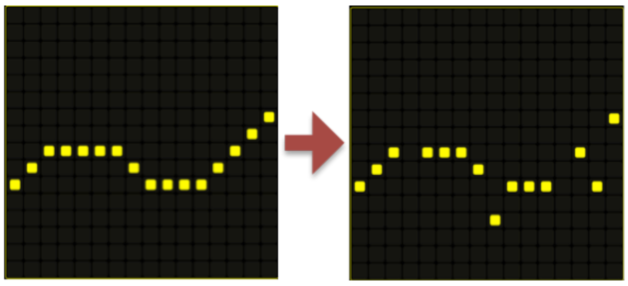

Figure 3. Semi-structured random variation applied to a melodic contour

5.2.2 Random Melodic Filtering

Let *k* be the intensity value. For each column in the grid, *k* of the "on" elements will be selected at random, and all others will be turned to "off". If there are less than *k* elements that are "on" in the column, none will be turned "off". See Figure 4 for an example.

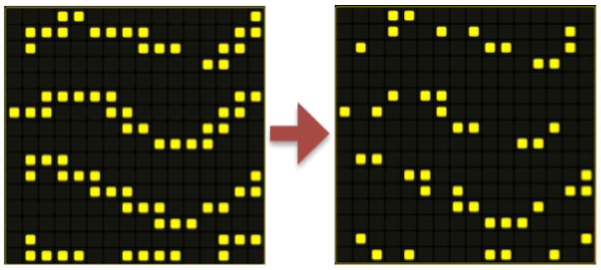

Figure 4. Random melodic filtering (with intensity value 3) applied to a harmony

5.2.3 Random Rhythmic Filtering

Let *k* be the intensity level. With equal likelihood, this algorithm takes selects either 2*k* or 2*k*-1 columns from the grid uniformly at random and clears their contents. This algorithm is intended to take a melody with no rhythmic breaks and create random breaks to facilitate the exploration of interesting rhythms. See Figure 5 for an example.

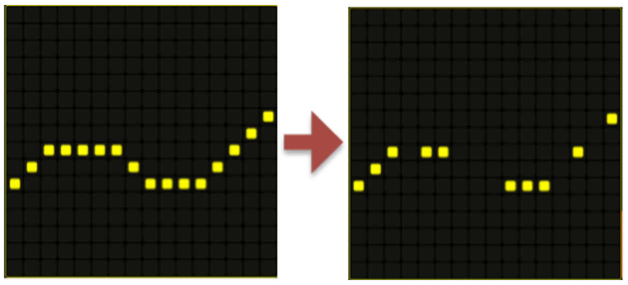

Figure 5. Random rhythmic filtering applied to a melodic contour

5.2.4 Noisy Harmony Generation

Let *k* be the intensity level. Given a grid *g*, two variants (g^1 and g^2) are created by using a modified version of the algorithm 5.2.1 (a neighborhood of *k*/2+1 vertical radius and 0 horizontal radius is used). Then two integers (*a* and *b*) are selected from the non-zero in the set [-*k*, *k*]. g^1 is shifted vertically by *a*, and g^2 shifted vertically by *b*. Finally, *g*, g^1, and g^2 are all superimposed. See Figure 6 for an example.

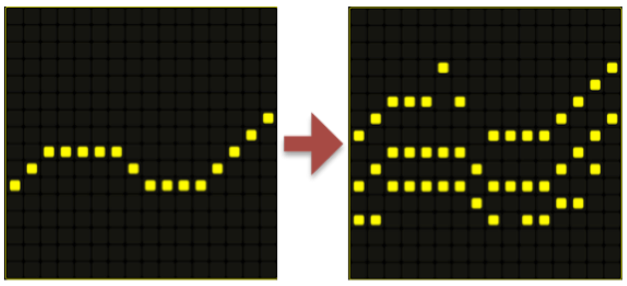

Figure 6. A harmony algorithmically generated from a melodic contour.

5.3 Piano Mode

SkipStep also allows users to "play" the step sequencer as if it were a piano. When piano mode is turned on (via element 2 in Figure 7) the step jumping and subset looping controls are replaced by the piano keys (Figure 7, element 1). Pressing a key underneath a column plays the chord in that column. Multiple columns can be played at once.

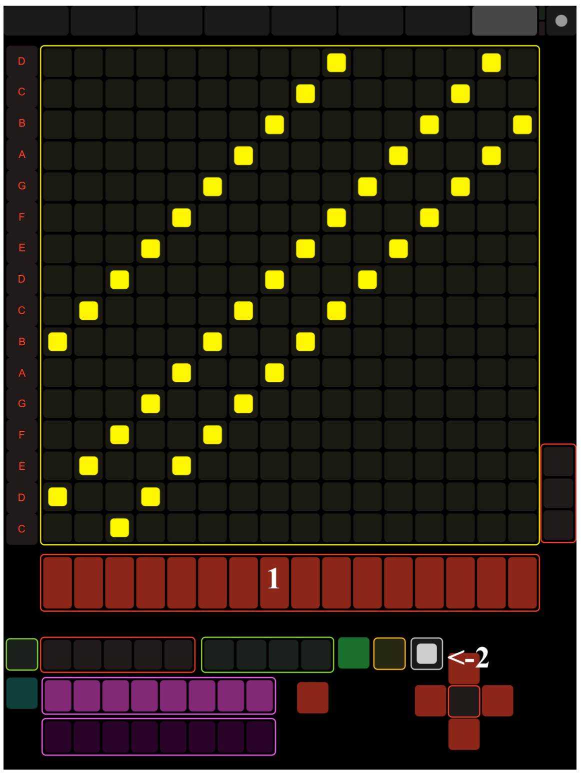

Figure 7. Piano Mode interface

All of the same features that work with the normal step sequencer still work with piano mode, allowing for a dynamic, variable keyboard. In auto-undo mode, the original chord is restored after the varied chord is released. Piano mode was designed to give users a more directly instrument-like way to play music. When the grid is a single ascending diagonal, it is analogous to a keyboard, (albeit, one adhering to a certain scale). In general, playing grids with piano mode allows the user to explore the musical space of the grid totally free of time constraints. It adds a continuous time dimension to the otherwise quantized time of the step sequencer.

5.4 Collaboration

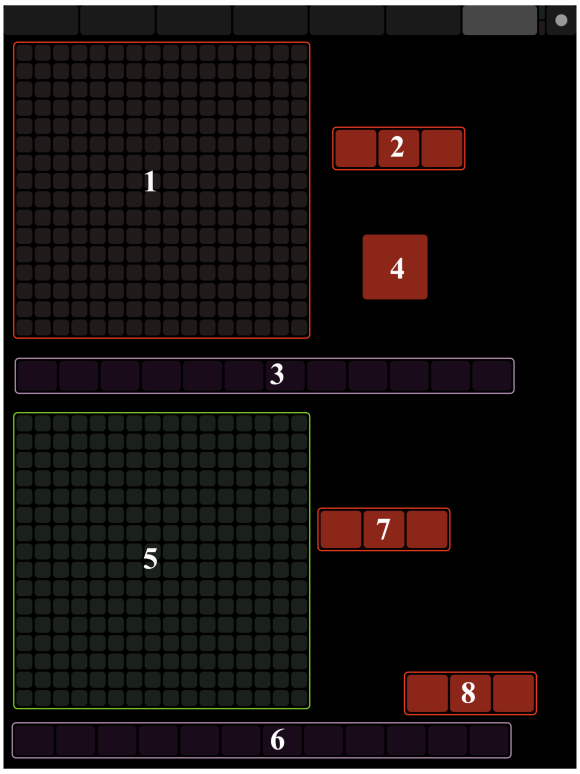

Figure 8. Collaboration interface

When multiple instances of SkipStep are running on the same local area network (LAN), the users can sync their metronomes to step at the same rate. Users on the same LAN can also send grids and scales to each other via the send/receive interface. Users first load a grid and scale into the pre-send editor via element 2 in Figure 8. Pressing the first button loads the grid and scale currently being looped in the first instrument, and so on. In the pre-editor (Figure 8, elements 1 and 3), users can modify the grid and scale before sending it without affecting the instrument it was taken from. By tapping element 4 in Figure 8, users can broadcast this grid to all other users on the LAN. When a grid and scale are received, they are displayed on elements 5 and 6 in Figure 8, where they can be edited again. To send the received grid to loop on the i^{th} instrument, users can press the i^{th} button on element 7 in Figure 8. Pressing the i^{th} button on element 8 in Figure 8 changes the scale of the i^{th} instrument to the received scale.

Additionally, multiple users can collaboratively edit a single SkipStep instance, in much the same way that multiple users can edit a single document on Google Docs. In this case, multiple iPads connect to a single instance of SkipStep, and the control events from one iPad are bounced to update the UIs of all others. All users can still access all of the different SkipStep features.

6. INTERACTIVE CAPABILITIES

6.1 Offline and Online Decision Making

An important benefit of the step sequencer interface (and interactive looping systems in general) is that it allows for a blend of online and offline decision making/activity. When improvising with an acoustic instrument, for example, most of the decisions made are online, or in "real-time" - the choice of what note to play at the next beat must be made before that beat arrives, and a delay that causes one to play unintentionally "late" is an error. In composing, the decisions being made are generally offline; there is no sense of "lateness" in deciding what note to enter into a score, and there is unlimited time to consider the entire piece of music composed so far.

SkipStep allows for both immediate actuation, which allows for keyboard-like performance, and looping, which allows music to play when the user is not interacting with the device, allowing them to make offline decisions without halting the musical output. Also, the automated algorithmic variation can prevent the repetition of static loops while offline decisions are being made. Offline decision-making can allow users to "plan ahead" while improvising, which could be especially useful for novice improvisers. If SkipStep is being used with multiple users, the looping functionality allows performers to stop "playing" and communicate directly with each other (verbally or textually), which could greatly change the dynamic of group improvisations. If SkipStep is being used in conjunction with other physical devices, users could manipulate other devices while SkipStep loops with automatic variation in the background.

6.2 Multi-Instrument Control

SkipStep's looping functionality allows users to control multiple instruments at once. While Ableton and other DAWs provide ways to start and stop pre-composed loops across different instruments in real time, they do not provide the ability to edit the loops themselves in real-time. SkipStep allows for both the editing and launching of loops in real time across multiple instruments. In addition, SkipStep allows for multiple modes of musical modification to occur simultaneously: users can apply automatic algorithmic variation on some instruments (using different algorithms on different instruments), while using the piano mode on yet another instrument.

6.3 Improvising Without Musical Knowledge

SkipStep allows users with little to no musical knowledge to engage in expressive improvisation. The step sequencer interface allows users to easily see the melodic contours they are creating, letting them edit music based on what it "looks like" rather than what it sounds like. Also, the sequencer allows users with no instrument or music-software experience to quickly create and hear melodies in real-time. The algorithms provided can help novice users quickly create variations on input, and the real-time save/load and undo features can help new users quickly and intuitively engage in theme-and-variation approaches to improvisation. In particular, editing melodies with auto-undo on is a simple, immediate and highly visual way to explore theme and variation.

SkipStep could also be used to construct "novice-friendly" musical spaces to explore. Musical novices could be provided scale presets that contain few dissonant intervals (ex. Pentatonic scales) to improvise in, allowing them to experiment freely without creating dissonant chords. Furthermore, the dissonance free scales could allow novice users to play freely with the keyboard without worrying about playing "wrong" notes. Preset grids of chords could also be provided to allow users to experiment with chord progressions rather than just melodies.

6.4 Interactive Algorithm Usage

SkipStep allows users to engage with musical algorithms in a truly interactive way. As mentioned before, the interface is constructed to allow for seamless transition between algorithmic and manual variation. The undo features in particular allow for a very iterative and powerful workflow, enabling users to immediately revert any changes that they don't like. When combined with grid saving, the algorithms allow users to quickly move through musical space, instantly saving those points they wish to return to. This allows a historied, branching exploration rather than a linear path through variations. According to Shneiderman, enabling rich history keeping and collaboration are two key features of successful "creativity support tools" [5].

7. EVALUATION

Informal user evaluations were conducted near the end of development of the current (at the time of writing) version of SkipStep. Nine users, ranging from musical novices to professional musicians, tested the interface and provided feedback. Overall, all of the users were able to learn how to use all of the features within 30 minutes and indicated that it was generally easy to use. Their feedback prompted the addition of toggled labels on all of the interface controls, and the redesign of the piano mode controls to be more central to the interface. A more rigorous user test is planned for the future.

SkipStep is currently being used for a piece performed by the Princeton Laptop Orchestra. In this networked-performance piece, seven performers are all running their own SkipStep instance, with all of the metronomes synchronized. A "conductor" holds up signs that give qualitative descriptions of melodies, (such as longer, higher, sparser, etc), and can direct either individuals or groups of performers to start or stop playing melodies, or modify them according to his signs. The performers are free to interpret the descriptions as they wish when modifying their melodies.

8. FUTURE WORK

Immediate plans for SkipStep involve taking the collaborative editing feature, which currently only works on a LAN, and implementing it to work over a wide area LAN, and implementing it to work over a wide area network (WAN), thus allowing collaborative editing by any two SkipStep users connected to the internet, regardless of their location. Internet latency has historically been a problem for long-distance, real-time performance, but SkipStep's quantized stepping provides a mechanism that could be very robust to network delays. The time between steps of the sequencer would act as a buffer against late packets.

Longer-term plans for SkipStep involve porting it to iOS, where low-level touch data could be used to implement vibrato for the piano mode feature.

9. CONCLUSION

SkipStep is an in-progress project that aims to allow both musical novices and experienced musicians to engage in expressive musical performance. We hope to use its combination of algorithmic capabilities, collaborative performance, and multi-instrument control to create new modes of musical expression, and believe it has the potential for, in Levin's words, "sophisticated expression" yet "elusive mastery."

10. REFERENCES

[1] Dezfouli, E., van der Heide, E. "Notesaaz: a new controller and performance idiom," in *Proc. Int. Conf. New Interfaces for Musical Expression*. Daejeon, South Korea, 2013.

[2] Levin, Golan. "Painterly Interfaces for Audiovisual Perfomance," M.S. Thesis, Program in Media Arts and Sciences, School of Architecture, Massachusetts Inst. of Technology, Cambridge MA, 2000.

[3] McCormack, J., McIlwain, P., Lane, A. & Dorin, A. "Generative Composition with Nodal," *Workshop on Music and Artificial Life* (part of ECAL 2007), E.R. Miranda (ed.), Lisbon, Portugal, 2007.

[4] Narveson, J., Trueman, D. "LANdini: a networking utility for wireless LAN-based laptop ensembles," in *Proc. SMC Sound, Music and Computing Conference*. Stockholm, 2013.

[5] Shneiderman, B. Creativity support tools: Accelerating discovery and innovation. Communications of the ACM 50(12): 20–32. 2007.

[6] Wang, Ge. "The ChucK audio programming language: A strongly-timed and on-the-fly environ/mentality," PhD dissertation, Dept. Comp. Sci., Princeton Univ., Princeton, NJ 2008.

[7] Wright, M. 1997. "Open Sound Control-A New Protocol for Communicating with Sound Synthesizers," *Proceedings of the 1997 International Computer Music Conference,* Thessaloniki, Greece, 1997.