DESIGNING EXPRESSIVE MUSICAL INTERFACES FOR TABLETOP SURFACES

Jordan Hochenbaum[1] Owen Vallis[1] Dimitri Diakopoulos[2] Jim Murphy[2] Ajay Kapur[1,2]

New Zealand School of Music[1]
PO Box 2332
Wellington, New Zealand
+64 (4) 463 5369

California Institute of the Arts[2]
24700 McBean Parkway
Valencia, California, 91355
+1 (661) 952 3191

hochenjord@myvuw.ac.nz vallisowen@myvuw.ac.nz ddiakopoulos@alum.calarts.edu
jamesmurphy@alum.calarts.edu akapur@calarts.edu

ABSTRACT

This paper explores the evolution of collaborative, multi-user, musical interfaces developed for the *Bricktable* interactive surface. Two key types of applications are addressed: user interfaces for artistic installation and interfaces for musical performance. In describing our software, we provide insight on the methodologies and practicalities of designing interactive musical systems for tabletop surfaces. Additionally, subtleties of working with custom-designed tabletop hardware are addressed.

Keywords: Bricktable, Multi-touch Interface, Tangible Interface, Generative Music, Music Information Retrieval

1. INTRODUCTION

Interacting with large-scale tangible and multi-touch interfaces such as Bricktable [1] affords users many unique musical experiences. Firstly, Bricktable's vast screen size makes it an excellent candidate for collaborative interaction, such as audio-driven installations or multi-user tabletop performances. Secondly, Bricktable's software-driven architecture allows the creation of many unique musical interfaces tailored specifically for either tangible or touch-based interaction. These new interfaces exploit the interactive intimacies afforded by the 'hands-on' control of tabletop surfaces. Ultimately, we have found that it is possible to create exceptionally engaging and meaningful musical experiences for users that transcend traditional modes of input.

From a software developer's standpoint, increasing platform support for tangible & multi-touch frameworks have further enabled rapid application development. Time previously spent worrying about low-level implementation issues can now be spent on designing novel tabletop interaction. Online communities of DIY builders and hackers (e.g. NUIGroup) represent a large knowledgebase of resources that have made possible the ability for anyone to experiment in tabletop computing. This combination has made the realization of surfaces such as Bricktable increasingly possible for musicians and artists, and the potential for musically expressive tabletop interfaces greater than ever.

Inspired by the influx of musical surfaces emerging in recent years, most notably the seminal work of the reacTable team [2, 3], earlier work such as Toshio Iwai's Composition on the Table [4], the Audiopad [5], and many others including [6, 7], this paper details our journey into the exploration of musical interfaces for tabletop surfaces. To this end, we have documented here some of the considerations and lessons gleaned during the development of five different applications.

NIME2010, 15-18th June 2010, Sydney, Australia

2. THE BRICKTABLE

Bricktable is a 50" diagonal Diffused Illumination (DI) surface, providing multi-touch and tangible object interaction. Bricktable affords users a 360-degree view and has been constructed of lightweight aluminum framing for ease of transportation. Bricktable has undergone three major hardware revisions. Copies of the most recent revision currently reside at the California Institute of the Arts and the New Zealand School of Music.

Bricktable currently uses the open-source vision tracking software Community Core Vision[1] (CCV) for multi-touch finger tracking. In cases where object (fiducial) tracking is required, Bricktable uses the open-source tracker, reacTIVision [8]. More information on the construction, dimensions, and technical details of Bricktable can be found in [1].

3. SOFTWARE APPLICATIONS

In this section, we provide an overview of our five application suite. We later detail much of the *how* in section 4 and *why* in section 5.

3.1 Roots

Roots is a dynamic, visually responsive musical interface, allowing users to sequence music along a continuum from generative to through-composed means. When a user presses on the table's surface, a vine-like structure branches out and generatively maneuvers around the surface– actively triggering sounds or loops associated with invisible zones on the screen. After touching the initial starting location of the generative root, the user can influence the generative system through a series of knobs located on the bottom of the screen. These parameters

[1] http://ccv.nuigroup.com/

include things such as speed at which the vine travels, 'wiggleness', and 'branchiness', effecting how linearly or chaotically the vine maneuvers in direction. With a few simple touches, a single user, or multiple people can very quickly create dense and lush generatively evolving soundscapes and re-compositions of the musical material.

In addition to releasing vines *into the wild*, users can further influence the environment by placing tangible "force field" objects on the table. Rotating the force-field fiducials will increase / decrease the vines attraction to the objects. Thus the users begin influencing the environment, and working within a semi-generative musical system. Lastly, if the user presses and drags their fingers over the surface, the sounds will be triggered in a 1-to-1 relationship with their movement, providing full compositional control to the user.

Figure 1 – *Roots* Generative Music Environment

3.2 Spaces

Spaces is a hyper-minimal musical interface for the composition of ambient music. Up to four people can either perform the interface as a live instrument, or experience the interface in an installation-style setting. Grown out of a desire to empower users with an intuitive, visually engaging interface, *Spaces*' simplistic expressivity is realized through the use of a basic drag gesture for control.

Breaking away from the fret, key and marker/position based visual feedback systems found in traditional musical instruments, *Spaces* instead chooses to use color to convey *feeling.* A user's touch visually morphs each lane from cool to warm, altering parameters in an ambient music system implemented in Reaktor.

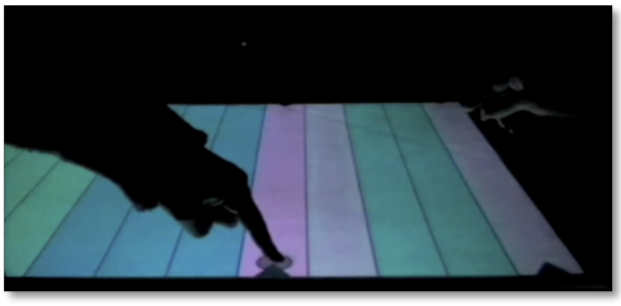

Figure 2 – Spaces Musical Interface

3.3 AhText!

AhText! is an installation that was developed for *AH!,* an interactive multimedia "opera-no-opera"; created and directed by David Rosenboom and co-creator Martine Bellen. *AH!* is based on 13 interconnected stories whose relationships to one another are exposed by a mandala story wheel (see Figure 3). The *AhText!* installation allows participants to explore and re-contextualize these 13 connected stories by navigating the mandala, and then re-organizing the poetic text associated with each active story. This remixing of textual content is reinforced through music generated by the spatial manipulation of short excerpts of text.

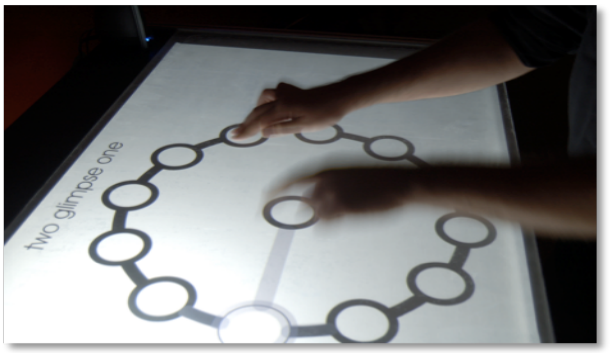

Figure 3 – Main menu from the AhText! installation

3.4 Maps

Maps [9] utilizes a Kohonen self organizing map (SOM) as a graphical front-end to display a library of music. By analyzing a set of pre-computed audio features for each song in the library, a sorted SOM groups similar music and provides basic genre classification. Each node on the organized map represents a song that is surrounded by musically similar neighbors. While *Maps* allows for automatic playlist generation, the primary focus of *Maps* was to demonstrate a model of interaction for browsing and sorting large collections of music not found in traditional applications for music playback.

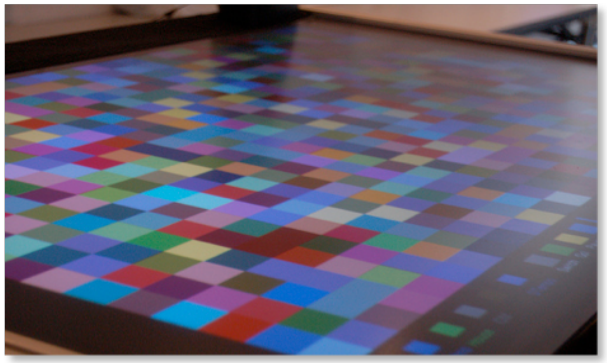

Figure 4 – *Maps* Music Browser

3.5 Argos

Argos [10] is a graphical user-interface builder aimed at extending the range of musical performance available to multi-touch devices. The application was developed with both large tabletop surfaces as well as smaller multi-touch devices (such as the recently released Apple iPad and many forthcoming tablet devices) in mind. The multi-platform interface builder is used to create networkable OSC & MIDI interfaces complete with common UI widgets such as buttons, sliders and knobs.

4. IMPLEMENTATION

This section illuminates some common software and programming considerations encountered during the development of our applications.

4.1 Communication

In exploring the creation of interfaces for musical expression, a common question reappears in nearly every application: where should the audio be generated? Since frameworks and toolkits for user interfaces do not always have the strongest support for audio, we often develop the musical system separately and rely on inter-application communication protocols like OSC [11] and MIDI. A full-circle implementation of Brickable's inter-application communication is as follows: a vision tracker (e.g. CCV or ReacTIVision) sends out TUIO [12] messages to a visual interface, containing data for the location and velocity of touches. The target interface then interprets the input into a musically-useful form and forwards the input via OSC or MIDI to the audio-generating application.

4.2 Languages and Frameworks

Software for tabletop surfaces can be easily developed in a plethora of programming languages, and across a variety of platforms. The TUIO protocol is fully cross-platform with a library for most major languages (Java, C, C++, C#, Python, etc.). Because of the amount of language options available, it can be hard to decide where to start. For our applications, we have exclusively used Java via Processing and C++ via openFrameworks. We find these languages preferable due to the extensive libraries available for advanced rendering, gesture recognition, and multi-touch interaction. Libraries such as MT4J and ofxMultiTouch help filter out the complexities of handling TUIO data, and greatly accelerate the process of application development.

The interfaces for Roots and Spaces were developed using the Processing programming environment, and currently communicate (via OSC) with special proprietary instruments created using Native Instruments' Reaktor; similarly, the *AhText!* interface was created in Java, and uses ChucK [13] to drive the audio component of the installation. The Maps application is unique in that it is written completely in Java, and given that it requires no real-time synthesis, it uses the Minim audio Java library for basic audio playback. Lastly, Argos was developed using the cross-platform C++ framework, *openFrameworks*, and communicates with either OSC or MIDI capable hosts.

5. DISCUSSION

In this section, we talk about our experiences in developing musical interfaces across three primary areas: Hardware Considerations, Performance Considerations, and Installation Considerations.

5.1 Hardware Considerations

This section describes various hardware limitations that have presented us with the greatest challenges in developing for the Bricktable.

5.1.1 Temporal Resolution

As discussed in section 2, Bricktable is a DI based surface with a maximum framerate of 30fps. In our experiments, and as noted by other researchers [6], this temporal sampling rate does not provide sufficient speed for rapid trigger-based (percussive) events. This constraint has led us to consider some alternative ideas regarding interaction design.

The *Spaces* interface has been used to drive mostly ambient and slowly moving texture pieces, implying slower and fluid like intervallic changes. Additionally, because pitch events happen along an unmarked linear continuum, pitch change locations along an instrument's pitch slider (although equally distributed) are less obvious. In practice, we have noted that this has made user-invoked pitch changes less rapid, and created a curiosity quotient noticeable in new users. By exploring continuous controls rather than trigger-based actions in *Spaces*, we have maintained an engaging experience for users while diminishing the reliance on temporal resolution.

As the cost of cameras drop, and driver support increases among open-source vision trackers, higher framerates are becoming an affordable option. We hope to upgrade the Bricktable system with one of these cameras in the near future.

5.1.2 Touch & Fiducial Tracking Precision

Our vision tracking system operates with a resolution of 320x240, which if improved, would make object recognition much more robust. Additionally, being able to recognize finger blobs in finer resolution would enable more accurate blob-positioning, and the ability to have increased subtleties in the musical control. On larger surfaces, enlarging fiducials and designing UI elements can offset the effects of less than ideal camera precision. We have found that making hit-test zones extend slightly beyond the visible regions of a widget greatly reduces the user frustration of a poorly calibrated table.

5.2 Installation Considerations

5.2.1 Ease of Use

One of the main considerations when designing musical interfaces for public installation settings is finding a balance between expressive interaction and interface familiarity. Typical interaction design theory states that users feel most comfortable interacting with interfaces that express some degree of familiarity to them. To address this in *AhText!*, the main user menu was designed to emulate a mandala design which already appeared throughout promotional and informational materials of the opera. Additionally, similar interactive interfaces were embedded into an online web application beforehand, so that users could acquaint themselves with the interface before ever experiencing the AhText! installation.

5.2.2 Multi-User Interaction

Another interesting discovery while designing the *AhText!* UI was the issue of multiple people browsing a singular interface at the same time. While parts of the installation interaction yielded collaboratively explored, but independently controlled user actions (e.g. reconfiguring the text), selecting a story to browse is an example where only one user's action could be 'active' at a given time. To enable multiple people to browse the story-selecting mandala menu simultaneously, a two-stage latch system was created in which users could click the nodes on the mandala to preview the story name, which after a few seconds would bring up the words "touch." If another node was selected, the process would start over again, letting the users collectively decide which story to jump into. This proved to be an easily learnable solution that not only solved the issue of multiple people browsing a singular menu interface, but also promoted dialogue between users.

5.2.3 Mixed-Input Interaction

Real-world interactions and experiences are often heightened as a result of their multimodal nature; the fact that you can see, smell, touch, and taste food all combine to create the sensation a person feels while eating. A similar effect can be achieved by harnessing the strengths of both multi-touch and tangible interaction within a singular interface. With *Roots*, we were able to achieve a heightened experience for users without diminishing the learning curve by mixing tangible input with multi-touch input. We attribute the "heightened interactions"

achieved when pairing tangible interaction with multi-touch interaction to two factors:

- Tangible interaction is *tactile*—compensating for the loss of haptic feedback when only using multi-touch interactions.
- (Multi)Touch and click input is intuitively similar to the point and click input we are accustomed to with the mouse. Because of this, users tend to experience real-world objects interacting with virtual (on-screen) elements with a heightened sense of novel play with the system.

That being said, while combining both means of input has proven to be beneficial for some of our interfaces such as Roots, it is not a solution for everything. For some of our less sophisticated interfaces such as *Spaces*, and *Maps*, minimal design with simple visual cues has proven to be more than effective enough.

5.3 Performance Considerations

5.3.1 Visibility and Interaction

Bricktable's large display, which lends itself so well to many users clustered completely around the table during installation settings, can present an interesting set of visibility problems in a more traditional performance environment. One of the major problems involves the audience's line of sight. We have found it beneficial to the audience's experience to perform leaving the front of the surface unobstructed by performers. However, even while this helps to ensure a direct line of sight for audience members, the viewing angle of the screen can often be less than ideal. A duplicate projection of the tabletops screen behind the performers, and/or use of an aerial camera setup above the performers is a great way to remedy this common problem.

5.3.2 UI Orientation

Another interesting performance consideration deals with the orientation of UI elements. In performance contexts, we have found it best to design interfaces that are non-orientation specific (e.g. the layout and implied relationships in Spaces do not change depending on the performers' or audience members' location). With more sophisticated interface's however, this is simply not possible. For example, not orienting the UI elements in front of the performer significantly reduces its usability of Argos. We are currently experimenting with free form positioning of UI elements via on-the-fly rotation gestures, to support performers using Argos from any angle around Bricktable.

6. Conclusion

In a domain dominated by traditional commercial hardware and software, tabletop applications via DIY hardware present an emerging platform for new musical expression. By exposing some of our challenges and considerations, we hope to increase the accessibility and approachability of designing new interfaces by others in the field.

7. ACKNOWLEDGEMENTS

We would like to thank David Rosenboom for his artistic insight on the AhText! application and his continuing support for the multi-touch research at the California Institute of the Arts; Seth Sandler for his insight on Argos; Mehmet Akten for his collaboration on Roots; and the New Zealand School of Music for their support of the Bricktable.

8. REFERENCES

[1] J. Hochenbaum and O. Vallis, "Bricktable: A Musical Tangible Multi-Touch Interface," in *Proceedings of Berlin Open Converence 09'*, Berlin, Germany, 2009.

[2] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner, "The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces," in *Proceedings of the 1st international conference on Tangible and embedded interaction* Baton Rouge, Louisiana: ACM, 2007.

[3] S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bencina, "The reacTable," in *Proceedings of the International Computer Music Conference* Barcelona, Spain, 2005.

[4] T. Iwai, "Composition on the table," in *ACM SIGGRAPH 99 Electronic art and animation catalog* Los Angeles, California, United States: ACM, 1999.

[5] J. Patten, B. Recht, and H. Ishii, "Audiopad: a tag-based interface for musical performance," in *Proceedings of the 2002 conference on New interfaces for musical expression* Dublin, Ireland: National University of Singapore, 2002.

[6] P. L. Davidson and J. Y. Han, "Synthesis and control on large scale multi-touch sensing displays," in *Proceedings of the 2006 conference on New interfaces for musical expression* Paris, France: IRCAM; Centre Pompidou, 2006.

[7] J. A. Paradiso, K.-y. Hsiao, and A. Benbasat, "Tangible music interfaces using passive magnetic tags," in *Proceedings of the 2001 conference on New interfaces for musical expression* Seattle, Washington: National University of Singapore, 2001.

[8] M. Kaltenbrunner and R. Bencina, "reacTIVision: a computer-vision framework for table-based tangible interaction," in *Proceedings of the 1st international conference on Tangible and embedded interaction* Baton Rouge, Louisiana: ACM, 2007.

[9] D. Diakopoulos, O. Vallis, J. Hochenbaum, J. Murphy, and A. Kapur, "21st Century Electronica: MIR Techniques for Classification and Performance," in *Proceedings of the 2009 International Society on Music Information Retrieval Conference*, Kobe, Japan, 2009.

[10] D. Diakopoulos and A. Kapur, "Argos: An Opensource Application for Building Multi-Touch Musical Interfaces," in *Proceedings of International Computer Music Conference (ICMC) 2010*, New York, 2010.

[11] M. Wright, A. Freed, and A. Momeni, "OpenSound Control: state of the art 2003," in *Proceedings of the 2003 conference on New interfaces for musical expression* Montreal, Quebec, Canada: National University of Singapore, 2003.

[12] M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza, "TUIO - A Protocol for Table Based Tangible User Interfaces," in *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation* 2005.

[13] G. Wang, "The ChucK Audio Programming Language: A Strongly-timed and On-the-fly Environ/mentality." vol. PhD: Princeton University, 2008.