SurfaceMusic: Mapping Virtual Touch-based Instruments to Physical Models

Lawrence Fyfe, Sean Lynch, Carmen Hull, Sheelagh Carpendale
University of Calgary
2500 University Dr. NW
Calgary, AB T2N 1N4
Canada
ljfyfe@ucalgary.ca, sglynch@ucalgary.ca, cphull@shaw.ca, sheelagh@ucalgary.ca

ABSTRACT

In this paper we discuss SurfaceMusic, a tabletop music system in which touch gestures are mapped to physical models of instruments. With physical models, parametric control over the sound allows for a more natural interaction between gesture and sound. We discuss the design and implementation of a simple gestural interface for interacting with virtual instruments and a messaging system that conveys gesture data to the audio system.

Keywords

Tabletop, multi-touch, gesture, physical model, Open Sound Control.

1. INTRODUCTION

In creating musical applications for interactive multi-touch tabletops, one of the most important considerations is the mapping of gestures on the tabletop to control of sound. The tabletop becomes the gestural interface part of the four components of computer-based musical instruments described in Wessel et al [11]. One approach to gestural control is to manipulate physical objects that are recognized by the tabletop. The AudioPad [9] uses round pucks that are identified by radio-frequency tags. Players manipulate the pucks to control the volume and effects for sets of sample loops. A similar approach to gestural control, used by the reacTable [7], involves the use of fiducial markers to control sound synthesis objects like oscillators, LFOs and sequencers.

Another approach to gestural control is to allow players to touch the interface directly. Compostion on the Table by Iwai [5] is an early tabletop interface that allows players to interact directly with the tabletop to control sound. The Jam-O-Drum from Blaine and Perkis [1] uses velocity sensitive MIDI drum pads in combination with visual feedback on a tabletop. Another touch interface, by Davidson and Han [3], uses interface elements like sliders, knobs and keys and implements them on a large multi-touch surface to control sound synthesis.

SurfaceMusic is an interactive multi-touch tabletop instrument that uses gestural interaction with virtual instruments to control sound parameters in physical models. Figure 1 shows players interacting with SurfaceMusic. The goal of SurfaceMusic is to create a natural mapping between gesture and sound control while retaining the configurability of a tabletop interface. Other work has mapped gesture to physical model including Jones and Schloss [6] where a force sensor maps to a 2D waveguide mesh. Dimitrov et al [4] created mappings between gesture and physical models for the reacTable. While these interfaces are interesting, for SurfaceMusic, a more transparent [12] mapping between gesture and sound control is achieved by allowing a direct touch interaction with the tabletop that in turn effects the sound.

Figure 1: Playing the virtual instruments.

The tabletop environment for SurfaceMusic is the Microsoft Surface where all visuals and interaction were done. The audio engine for SurfaceMusic is provided via the ChucK [10] programming language. The physical models used in ChucK were originally developed for the Synthesis Toolkit (STK) [2]. Sounds were developed in ChucK rather than in the STK itself for rapid development and because ChucK offers built-in support for communication using Open Sound Control [13]. Gestural parameters are OSC addressable as per Wright et al [8].

2. VIRTUAL INSTRUMENTS

The virutal insturments in SurfaceMusic have abstract cicular shapes that are not meant to represent actual instruments directly. A circular shape was chosen to allow players to rotate the instruments in a variety of orientations depending on their location around the table. The internal, playable parts of each instrument are meant to suggest interactions with actual instruments. For example, a real string instrument can be strummed. In SurfaceMusic, a string is strummed using a swiping gesture across the virtual strings that is similar to a strum on a real string instrument. Virtual touch-based instruments cannot offer

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

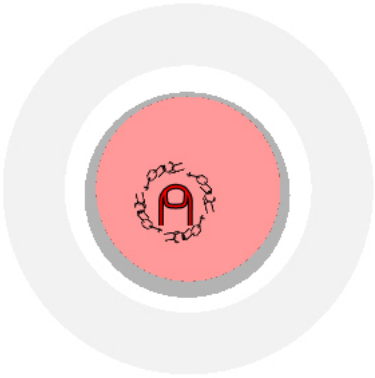

Figure 2: Playing the percussion instrument.

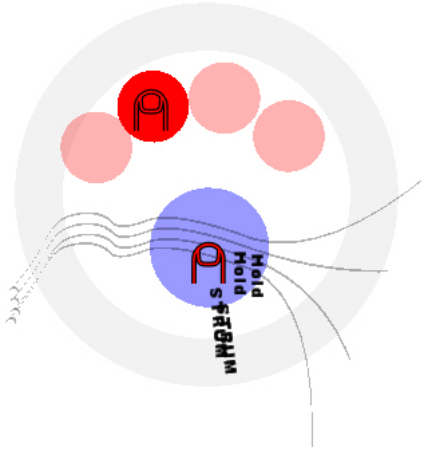

Figure 3: Playing the string instrument.

the affordances of real instruments but interactions with them can be made simpler by mimicking the gestures used in playing real instruments.

2.1 Adding and Removing Instruments

In order for people to play the instruments in SurfaceMusic, they must first grab them from the instrument container, a circle in the center of the screen that contains a copy of each virtual instrument. This can be seen in Figure 1. The instrument container serves to both create and destroy instruments. When an instrument is grabbed from the container and pulled out, a new copy of the instrument is created that can be played. Multiple virtual instruments of the same type can be pulled onto the surface and played simultaneously. To remove an instrument from the surface, a player moves the instruments back into the container and it disappears.

2.2 Rotating, Translating and Scaling

All instruments have a circular grey border to help players perform actions on the instruments without making sound. When any instrument's border is grabbed with one finger, and dragged, the instrument can be rotated 360 degrees or tranlated to any location on the surface. In addition, by grabbing the border with two fingers, instruments can be scaled to larger or smallers sizes, allowing players to scale the instrument to a size that suits them. Rotation and translation have no effect on the sound of the instruments while scaling changes the volume of the instrument.

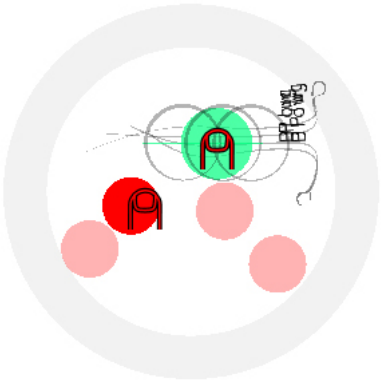

Figure 4: Playing the wind instrument.

2.3 Instrument Types

The three instrument types created for this application are a percussion instrument, a string instrument and a wind instrument. The physical models used are the BandedWG model for percussion, the Mandolin model for the string and the Clarinet model for the wind. Each model has specific parameters that are controlled via touch events and gestures on the tabletop. These physical models were chosen because of their associations with real instruments and because of their sound control parameters.

2.3.1 Percussion

The percussion instrument shown in Figure 2 is the simplest of all of our instruments, being a circle that players simply touch to interact with. A touch triggers a call to the BandedWG pluck function.

2.3.2 String

The string has four circles for determining the note played and strings for strumming with a swipe gesture. Players touch the circles to active a note which can be held as long as a finger is placed within its boundaries.

The action is shown with two fingers in Figure 3 where note circles become red when touched and text appears and moves away from the strings when strummed. Strums trigger the noteOn function in the Mandolin model. In addition, the angle of the strum gesture maps to pluck position and the speed of the strum maps to attack velocity in the physical model. Those parameters where used to give a natural feel to the strum gesture.

2.3.3 Wind

The wind instrument is similar to the string instrument in that it contains four note circles. However, it differs in the interaction used to generate sound. In order to generate sound, a player touches the green circle and moves it to the right along the grey path shown in Figure 4. The distance that the green circle is moved controls the intensity of the sound via the reed stiffness parameter of the Clarinet model. When a player releases the green circle, it moves back into its original position. This gesture gives the player the feeling of putting more or less energy into the sound depending on how far the circle is moved and how long it is held. Note that the circles are used as note controls in the same way as with the string instrument.

2.4 Text Visualization

Text is used to help visualize actions performed with the

virtual instruments. For the drum, when it is hit, the word "Touch" appears and then flies off the screen. For the string instrument, "Strum" appears whenever it is strummed and for the wind instrument "Blow" appears when the green circle is moved.

2.5 Playing Notes

Both the string and the wind instruments have circles that can be touched in different configurations to produce notes in ChucK (see messaging below). Notes for the virtual instruments work by touching from zero to of all four of the circles. Each circle, from left to right (see Figures 3 and 4), represents 1, 2, 4 and 8. As each hole is selected, the number is added to the total giving 16 possible tones from only four holes. That number, from 0-15, is then sent to ChucK and added to 60 to create a MIDI note range from 60-75.

2.6 OSC Messaging

Touch events trigger OSC messages that are sent to ChucK to trigger control changes in the physical models. Each virtual instrument maps to a shred (thread) in ChucK. New virtual instruments create new shreds and the removal of an insturments removes shreds. With this kinds of shred management, SurfaceMusic is completely polyphonic with the number of instruments limited only by the tabletop's visual surface area.

OSC messages sent from SurfaceMusic to ChucK look like:

```
/instrument, i i i i f f f
```

This is the only message sent regardless of the instrument or its specific parameters. This simplifies the message receiving and parsing process in ChucK. The message parameters from left to right are:

1. id (integer)

   This is the ID of the instrument. It is shared between SurfaceMusic and ChucK so that specific instruments can be distinguished from one another. For example, two winds can be played simultaneously with different notes.

2. action (integer)

   Action values are 0 for creating a new instrument/shred, 1 for removing an instrument/shred, 2 for starting to play an instrument and 3 for stopping an instrument that is currently playing.

3. instrument type (integer)

   Types values are 0 for percussion/BandedWG, 1 for string/Mandolin and 2 for wind/Clarinet.

4. note (integer)

   This integer value maps to a MIDI note value in ChucK.

5. speed (float)

   The speed of a swipe gesture. Currently implemented only for the string instrument.

6. angle (float)

   The angle of attack for the string instrument.

7. size (float)

   Size is the current size of any instrument on the table. It maps to gain with larger sizes having higher gains and vice versa.

3. SUMMARY

SurfaceMusic is an interaction system that allows for the gestural manipulation of multiple virtual instruments on a multi-touch tabletop. The virtual instruments of SurfaceMusic use simple gestures that map to parameters in physical models. The physical models used represent percussion, string and wind instruments. A communication system, developed using Open Sound Control, allows the tabletop to send gestural information to the audio engine for controlling the parameters of the physical models.

4. ACKNOWLEDGEMENTS

We would like to thank the Natural Science and Engineering Research Council of Canada, the Alberta Informatics Circle of Research Excellence, SMART Technologies, Alberta Ingenuity, and the Canadian Foundation for Innovation for research support. We would also like to thank the members of the Interactions Lab at the University of Calgary for feedback and support during the development of this project.

5. REFERENCES

[1] T. Blaine and T. Perkis. The jam-o-drum interactive music system: a study in interaction design. In *DIS '00: Proceedings of the 3rd conference on Designing interactive systems*, pages 165–173, New York, NY, USA, 2000. ACM.

[2] P. R. Cook and G. Scavone. The synthesis toolkit (stk). pages 164–166. International Computer Music Association, 1999.

[3] P. L. Davidson and J. Y. Han. Synthesis and control on large scale multi-touch sensing displays. In *NIME '06: Proceedings of the 2006 conference on New interfaces for musical expression*, pages 216–219, Paris, France, France, 2006. IRCAM — Centre Pompidou.

[4] S. Dimitrov, M. Alonso, and S. Serafin. Developing block-movement, physical-model based objects for the reactable. TeX Users Group, 2008.

[5] T. Iwai. Composition on the table. In *SIGGRAPH '99: ACM SIGGRAPH 99 Electronic art and animation catalog*, page 10, New York, NY, USA, 1999. ACM.

[6] R. Jones and A. Schloss. Controlling a physical model with a 2d force matrix. In *NIME '07: Proceedings of the 7th international conference on New interfaces for musical expression*, pages 27–30, New York, NY, USA, 2007. ACM.

[7] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146, New York, NY, USA, 2007. ACM.

[8] C. W. Osc and M. W. A. Freed. Managing complexity with explicit mapping of gestures to sound. In *in International Computer Music Conference, (Habana*, pages 314–317, 2001.

[9] J. Patten, B. Recht, and H. Ishii. Audiopad: a tag-based interface for musical performance. In *NIME '02: Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–6, Singapore, Singapore, 2002. National University of Singapore.

[10] G. Wang and P. Cook. Chuck: a programming language for on-the-fly, real-time audio synthesis and multimedia. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 812–815, New York, NY, USA, 2004. ACM.

[11] D. Wessel, M. Wright, and J. Schott. Intimate musical control of computers with a variety of controllers and gesture mapping metaphors. In *NIME '02: Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–3, Singapore, Singapore, 2002. National University of Singapore.

[12] M. Wright. Problems and prospects for intimate and satisfying sensor-based control of computer sound, 2002.

[13] M. Wright, A. Freed, and A. Momeni. Opensound control: state of the art 2003. In *NIME '03: Proceedings of the 2003 conference on New interfaces for musical expression*, pages 153–160, Singapore, Singapore, 2003. National University of Singapore.