

Interaction Techniques for Musical Performance with Tabletop Tangible Interfaces

James Patten
MIT Media Lab
20 Ames St.
Cambridge, Ma 02139
+1 857 928 6844
jpatten@media.mit.edu

Ben Recht
MIT Media Lab
20 Ames St.
Cambridge, Ma 02139
+1 617 253 0620
brecht@media.mit.edu

Hiroshi Ishii
MIT Media Lab
20 Ames St.
Cambridge, Ma 02139
+1 617 253 7514
ishii@media.mit.edu

ABSTRACT

We present a set of interaction techniques for electronic musical performance using a tabletop tangible interface. Our system, the Audiopad, tracks the positions of objects on a tabletop surface and translates their motions into commands for a musical synthesizer. We developed and refined these interaction techniques through an iterative design process, in which new interaction techniques were periodically evaluated through performances and gallery installations. Based on our experience refining the design of this system, we conclude that tabletop interfaces intended for collaborative use should use interaction techniques designed to be legible to onlookers. We also conclude that these interfaces should allow users to spatially reconfigure the objects in the interface in ways that are personally meaningful.

Categories and Subject Descriptors

H.5.2 [User Interfaces]: interaction styles, input devices and strategies J.5: [Arts and Humanities]: performing arts

General Terms

Performance, Design, Human Factors

Keywords

musical performance, tangible interface, interaction techniques

1. INTRODUCTION

While graphical user interfaces (GUIs) have a set of generally accepted interface building blocks, such as buttons, sliders, menus and windows, tangible user interfaces (TUIs) lack an analogous vocabulary. Our development of the Audiopad[10] electronic music performance system provided the opportunity for the iterative design of a set of TUI interaction techniques in the context of a tabletop musical controller. After completing this development, we found that with the addition of a few missing pieces, this set of interaction techniques could be applied to a larger set of applications, including business simulation and cellular telephone tower layout.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACE 06, June 14-16, 2006, Hollywood, California, USA.
Copyright 2006 ACM 1-59593-380-8 /06/0006 ...\$5.00.



Figure 1: The Audiopad electronic music controller.

All of these applications were built using a tabletop object tracking platform, similar to the Sensetable[9]. Beyond real-time tracking of the positions of objects on its surface, our tracking platform supported additional physical interface elements on top of the tracked pucks, such as buttons, dials and interchangeable tokens. While much of the initial interaction techniques (discussed in [10]) we developed for Audiopad leveraged these additional elements, over time we converged on a set of techniques in which each function is controlled either through the movement of one puck, or the relative position of two pucks. Audiopad's design and development process was punctuated by periodic performances and installations, which gave us the opportunity to gauge reactions from performers, audience members and naive users. One of our most important findings during this design process was the importance of the "legibility" of the interaction. Specifically, that onlookers should be able to understand how users were interacting with the system, even if the onlookers were not participating in the interaction themselves.

2. RELATED WORK

In recent years, researchers have developed a variety of tabletop tangible musical controllers. The reacTable [12] uses physical objects on a tabletop projection surface to represent parts of a modular synthesizer. Performers can change the topology of the synthesizer, as well as other synthesis parameters in real-time. Perhaps the most important difference between the reacTable

and the Audiopad is that the reacTable uses modular synthesis, while the Audiopad uses loop-based synthesis. These two synthesis approaches present distinct challenges to the interface designer as well as the performer. Other projects such as the Jam-o-drum [11] incorporate aspects of game play into a musical interaction. Information about other “musical tables” can be found in [6]. In the larger context of tabletop interfaces, our work is inspired by such projects as the Digital Desk[14], graspable interfaces such as Grasp Draw[3], and tabletop tangible systems such as Urp[13]. A list of many other related interactive tabletop systems can be found in [8].

3. MOTIVATION

Audiopad started as an experiment to create a new performance dynamic for electronic music. Often, artists perform their music on stage seated behind one or more laptops. From the audience’s perspective, the performer’s actions are often very similar to what they would be if he or she were reading email. As a result these performances can sometime lack the engagement one finds at a concert where the performers use traditional musical instruments. We believe the crucial difference between these two cases is that the audience can see and appreciate the interaction between a performer and a traditional “analog” musical instrument in a way that is difficult with an on-screen interface. As a result, we aimed to design the interaction techniques within the Audiopad system to allow the audience to begin to see the cause-and-effect relationships between the performer’s actions and the changes in the music.

We were excited about exploring new TUI interaction techniques in the context of a musical application for two reasons. First, musical performance is a very demanding application from an interface perspective, particularly as far as timing is concerned. The quality of a performance depends in part on the ease of interaction with the interface. Second, musical applications often involve the manipulation of many different parameters, both continuous and discrete, so there were many opportunities to explore interaction techniques for setting these parameters. During the process of its development Audiopad has been used in more than ten public musical performances

and three museum installations. During this process we have observed users with a variety of musical and computer skill levels interacting with it.

4. SYSTEM HARDWARE

Audiopad is based on a tabletop RF tracking system. Each of the tracked objects contains an LC tag that resonates at a unique frequency. The position of the tag on the table is determined by measuring the resonance of the tag with several different antenna elements in the tabletop. The sensing hardware and a video projector are connected to a standard PC running Linux on which the application software runs. The video projector displays the graphical component of the interface on the tabletop, as in figure 1.

5. INTERACTION TECHNIQUES

The techniques described here employ a relatively generic set of tracked objects on the table, or pucks. There are between five and eight pucks that represent audio tracks (or more generally, data). In addition, there is one “selector puck” that is used to change the properties of other objects. This puck always has a star shape as shown at the top-right of figure 2.

These techniques were developed and refined in the context of the Audiopad application, but were later applied to other applications, including video editing, cellphone tower placement, and business supply chain visualization.

5.1 Hierarchical Item Browsing and Selection

In a graphical user interface, pie menus [5] are useful for selecting items from sets of choices. We have explored a variety of related approaches for use in tabletop tangible interfaces for modifying the properties of pucks. One of these approaches is a two-handed, asymmetric approach in which the user’s non-dominant hand holds the puck to be modified, and the dominant hand holds the modifier puck. This approach is based on Guiard’s Kinematic Chain Model [4], which suggests that in asymmetric two-handed tasks, one’s dominant hand acts in the frame of reference provided by the non-dominant hand. For

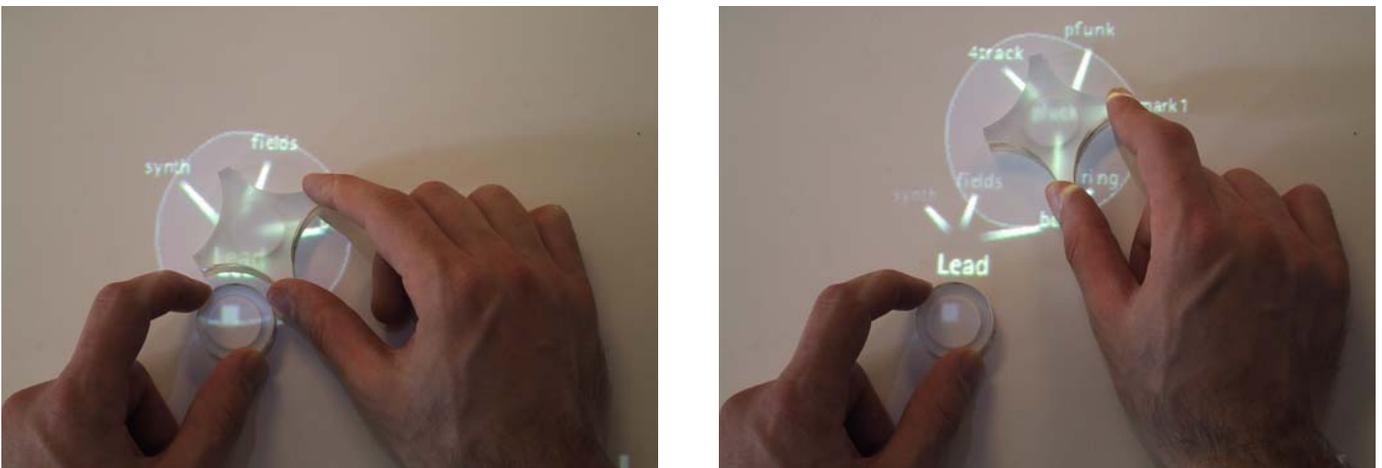


Figure 2: A two handed method for selecting items from a hierarchical menu.

example, when writing with a pen on a piece of paper, right-handed people often orient the paper with their left hand, and this improves their performance in the writing task [4]. Figure 2 shows the two-handed technique in use.

In Audiopad this approach is used to select a musical sample from a set of samples. The samples are arranged into various groups, and those groups may be collected into larger groups and so on. When the user places the modifier puck close to an area marked with a small '+', known as a hotspot, near the puck to be modified, the first level of choices spring out of the modifier puck. When the user moves the modifier puck over one of these items, any of its child items spring out, and so on, as shown in figure 2. A terminal node in this tree contains a colored square. Selecting one of these nodes by placing the modifier puck on top of it indicates the selection process is finished, and the tree disappears. In the case of Audiopad, these terminal nodes represent the actual musical samples, and selecting them causes a new sample to start playing. If the user wishes to cancel the selection of a new item from the tree, he or she can move the modifier puck away from the tree and the tree will disappear after a couple of seconds.

As this technique depends only on the relative positions of the modifier puck and the object it is modifying, one can also select items from the tree using one hand. This hand can move either puck alone to select items. In informal demonstrations or museum installation settings users almost always select items using one hand on the modifier puck, while in performance contexts, performers typically use both hands, though sometimes use only one hand when the other is occupied with another task. This difference may be due to the stricter timing requirements in the performance context, as well as the performers being more familiar with the interface.

One problem with the first version of this interaction technique was that the selection process gave no feedback about recently selected items. In the context of Audiopad, similar sounding samples are located near each other in the selection tree. During a performance one often wants to focus on a certain group of samples for awhile, and then move to another group. Without feedback from the interface about which items had been used recently, performers wasted time repeatedly searching for certain samples within the tree. To address this issue, we changed the interaction such that the location of the most recently selected item is displayed when the tree is first activated. While this greatly reduces the time spent searching for an item, it is still difficult to switch quickly between items that are located several levels deep in the tree because the user must repeatedly move the modifier puck between the hotspot and the item to be selected. For cases in which quick selection among a few items is needed, we developed a separate technique called floating menus which is discussed in the next section. A condition in which two handed interaction becomes important is when the tree extends off of the table while selecting an item several levels deep. In these cases, the user can simply move the base of the menu using the non-dominant hand to bring the entire tree onto the table. Two user interfaces of note which have employed asymmetric two handed interaction are Toolglass[1] and GraspDraw[3]. With the GUI-based Toolglass, one hand

controls the mouse cursor, while the other hand positions a set of tools in the workspace. The GraspDraw system uses two 6 degree-of-freedom trackers as a method of physically interacting with a drawing application. In his thesis, Fitzmaurice states that he originally focused on using asymmetric gestures to create objects such as circles. [3] He notes that the hands obscured portions of the circles, and thus any benefit achieved through the asymmetric use of hands was overcome by not being able to see the results of the interaction [3]. We did not observe users having problems with occlusion of graphics during the selection of items using the tree, probably due to two differences between these applications and GraspDraw. First, GraspDraw, running on the ActiveDesk [3], uses rear-projection. The Audiopad relies on projection from above. If the user places his or her hand on the table on top of some graphical information, the information shows up on top of the hand, though it will be somewhat distorted. Second, while navigating the tree the most important graphical elements, the children of the current node in the tree are displayed in front of the modifier puck where they will not be occluded by the puck or the user's hand. To further avoid occlusion, each series of choices in the tree is displayed within 120 degrees of arc in front of the modifier, rather than completely surrounding it.

5.2 Floating Menus

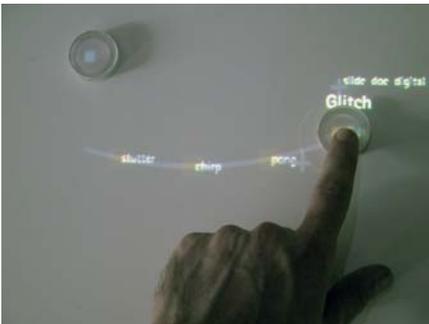
As discussed above, users of Audiopad in performance found it tedious to repeatedly select samples from several levels deep within the sample tree. To address this issue, we developed a floating menu that can follow objects around as they move on the table. The menu is shown in figure 3. To select an item from the menu, one simply moves the object on top of the desired selection. In the context of Audiopad, these menu items represent audio samples that are related to the sample currently being played. As the user moves the object around the table, the menu follows it, so that the user can easily select something from the menu with a quick gesture.

The important design issue in this interaction is when the menu should move, and when it should be stationary. If the menu moves too much, it can be difficult to select something from it, while if it moves too little, it will usually be far from the object it corresponds to. To determine when the menu should move and when it should be still, we define an area surrounding the icons called the selection area, as shown in figure 4. When the puck is inside of this area, the menu stays still to make selection easier. If the puck moves outside of this area for more than 3 seconds, the menu recenters around the puck, such that the currently selected choice from the menu is underneath the puck. When the puck moves, the menu lags behind it slightly. This gives the user freedom of movement in case he or she would like to move a puck to a specific area on the table without accidentally selecting an item from the menu.

In the original version of this technique, the menu would move toward the puck whenever the puck left the selection area surrounding the icons. This approach sometimes caused problems, because a user would accidentally move the puck outside of this area while trying to select a menu item. The user's motion would cause the desired menu item to move, making it difficult to select. We experimented with increasing the size of the selec-



The floating menu presents a list of choices to the user.



The user selects one by moving the puck along the arc.



The user can move the puck anywhere else once the desired item is selected.



After a brief time delay, the floating menu moves back under the puck.

Figure 3: Floating menus in the Audiopad application. One selects an item from the menu by placing the puck on top of it. When the user moves the puck away from the menu, the menu follows it.

tion area to make menu selection easier, but this caused the menus not to follow the pucks when users thought they should because the puck was still inside of the selection zone. The time-based approach works well because users can stray outside of the selection zone when moving the puck toward an item in the menu without having the menu move in response. This time-based tolerance means that the selection zone around the icons can be small, ensuring that the menu will follow the puck as the user moves the puck around on the table.

Another issue with the design of this technique was how the menu should recenter around the puck. In the initial design, the menu recentered by moving toward the puck until the puck was once again in the selection zone. This approach occasionally led to items in the menu being inadvertently selected after the menu had recentered itself several times. Recentering the menu by moving the currently selected menu item underneath the puck resolves this problem.

5.3 Changing Continuous Parameters

Many applications involve the manipulation of continuous parameters. For example in Audiopad, each audio track has a volume parameter. One early approach to this problem was to rotate pucks on the table to change their volume. Graphical feedback, in the form of an arrow and a bar graph were displayed beside the puck to indicate the current setting, as seen in figure 5. We avoided using a physical dial as was used with the Sensetable system [9] because this approach must deal with what Buxton calls the 'nulling problem' [2]: a condition resulting when the physical state of a dial and its computational state are inconsistent. There were several problems with this approach to parameter control. First was a tradeoff between precision and speed when adjusting a parameter. The software could be configured such that several revolutions of the puck were needed to fully traverse the range of possible parameters. In this case it was possible to set the puck to a value with several digits of precision, but it took a lot of rotating to reach a desired value. Alternatively, with the entire parameter space accessible in one revolution of the puck (or less), parameter changes could be made quickly but it was difficult to make them precisely. Another issue with this approach was that it was difficult to change multiple parameters at the same time. Any more than two parameters was essentially impossible with two hands.

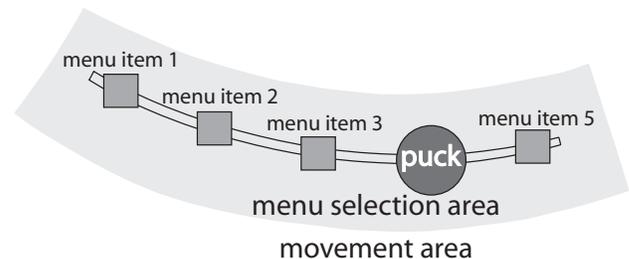


Figure 4: The selection area around a floating menu. When the puck is in this area, the menu will not move.

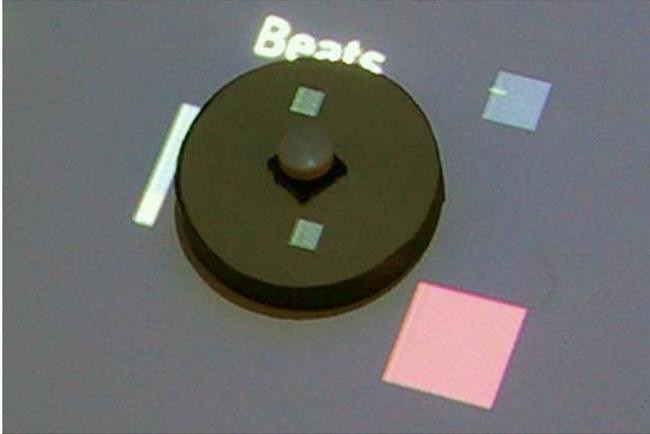


Figure 5: Our first approach to controlling the volume of a track in Audiopad. The puck is rotated to change the volume, just like a volume knob. The volume is displayed in a small bar graph to the left of the puck.

A more subtle issue was that when a user would rotate a puck, their hand often obscured it from the view of others. This made it difficult for others to observe the manipulation being performed and understand its effect in the context of the application. In the context of Audiopad, this was a concern because we wanted the audience of a musical performance to see the causal relationships between the performers actions and the music they were hearing. We believe this difficulty in seeing causal relationships could be of concern in face-to-face collaborative applications as well, where the TUI becomes a shared medium for expressing ideas.

Based on these observations we developed a technique that allows one to manipulate multiple parameters simultaneously with coarse motor movements. The value of the parameter is determined by the distance between the puck and another master puck. In the Audiopad application, this technique is used to control the volume of all of the tracks. The distance between each track and a special puck, called the microphone,

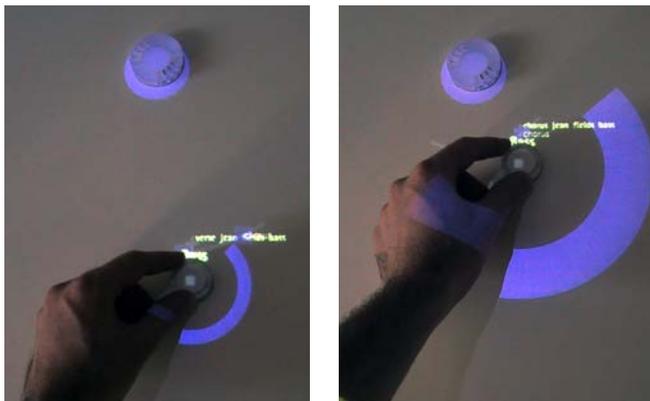


Figure 6: A later approach to changing volume: the distance between the microphone (top puck) and an audio track (bottom puck) determines the current volume of that track. The size of the colored arc in the photos represents the current volume of the track it surrounds.

mines all of the volumes: tracks that are closer to the microphone are louder than those that are farther away. To change the volume of a particular track, one simply moves it closer or farther away from the microphone as shown in figure 8. One can grab several tracks with each hand and move them simultaneously, or move the microphone itself to change the volume of all tracks together. If the user wants to change the volume of most tracks while leaving a few of the volumes constant, he or she can move the microphone with one hand, while moving the other tracks with the other hand, so as to maintain a constant distance between them and the microphone. (figure 7)

One detail important for making this technique work well is the function mapping distance to the parameter being controlled. After some experimentation in the context of Audiopad we arrived at a transfer function shown in figure 8. Within the range of 8 cm. of the microphone, the volume is at its maximum level. From this point the volume decreases linearly until a distance of 27 cm, where the volume reaches zero. This mapping means that there are always some areas of the table where the move-

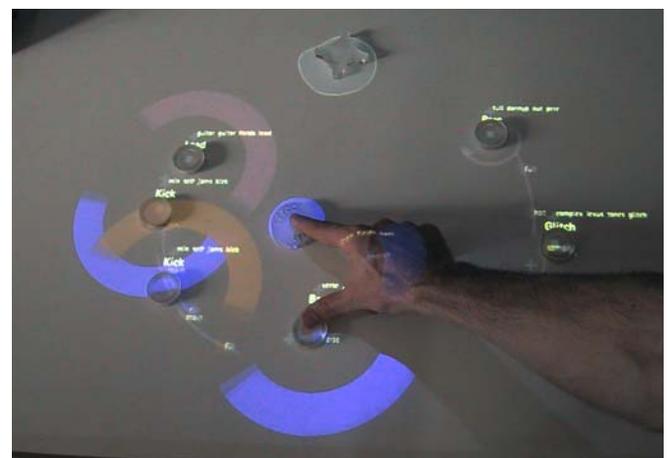
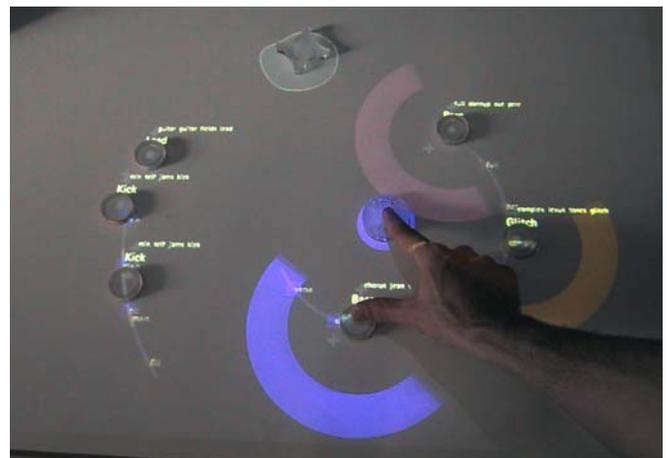


Figure 7: Using the microphone to adjust the volume of many tracks at one time, as one might do when transitioning between songs. The user is moving one audio track with his thumb to keep its volume constant while he adjusts the volume of the other tracks. The blue circle underneath the user's index finger is the microphone.



Figure 8: The volume parameter as a function of distance from the microphone puck.

ment of a puck has no effect on its volume. For parameters that are changed infrequently, one might want to use a mapping in which the active area was smaller. This would give the user maximum flexibility in how objects in the rest of the space were organized.

5.4 Setting Two-dimensional Parameters

While the technique above works well for controlling a one-dimensional parameter such as volume, there is no clear way to apply it to a two-dimensional continuous parameter. In the context of Audiopad, we explored two techniques for modifying two-dimensional continuous parameters. Users employed these techniques to change digital effect parameters on a track-by-track basis, for example the high frequency and low frequency cutoff of an audio filter.

The first technique was the use of effect “zones” on the table where the two-dimensional motion of the puck controlled the two parameters. In this case, the absolute position of a puck in the effect zone determined the value of the parameter. Figure 9 shows a picture of this technique. With this approach, the direct mapping of a particular point on the table to a particular setting of effects parameters reduced flexibility in terms of where pucks could be on the table. This rigidity made it difficult for users to arrange objects in other ways, for example to line tracks up in a row according to the order in which they were to be played. Second, this interaction technique did not give feedback about how the parameters were changed over time. If a musician wanted to gradually change a parameter a certain amount, the interface made it difficult to know when that change was complete. To address these issues we explored a technique for making relative adjustments to two-dimensional parameters. The user places the modifier puck on a hotspot toward the bottom of the puck. Then, the two-dimensional

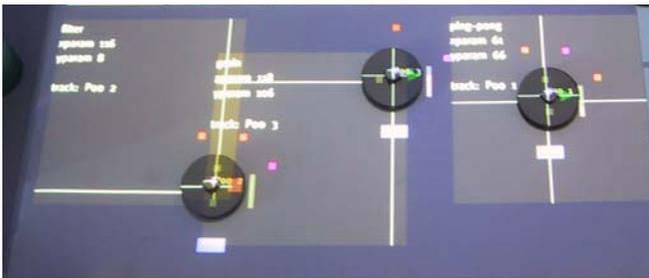


Figure 9: An early method of controlling audio effect parameters in Audiopad using “effect zones” where effect settings corresponded to absolute positions on the table.

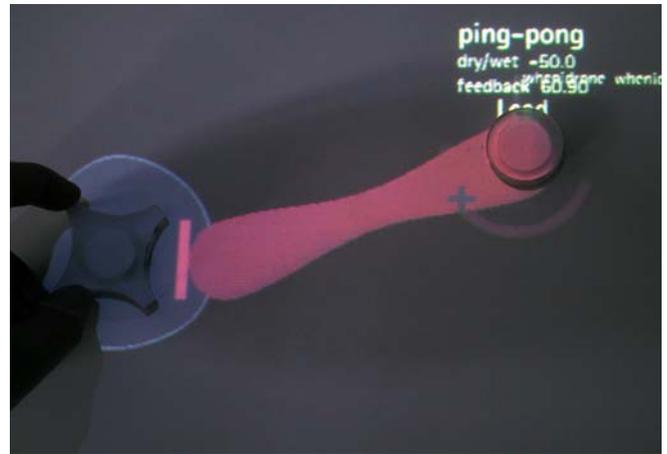
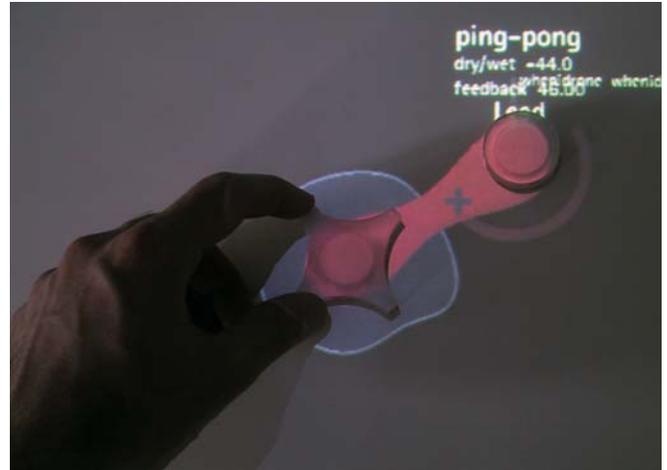


Figure 10: Using the modifier puck to change the effect parameters of an audio track. Here the effect setting is determined by the relative position of the two pucks. In the right picture, the user has exceeded the bounds of the parameter, so the red colored area stops following the modifier puck.

motions of the modifier puck control the two-dimensional parameter setting. Graphical feedback shows how the setting has changed since the modification started, as well as the current absolute setting of the parameter, as shown in figure 10. One can move either puck to change the parameter. What matters is their relative position. If the parameter has a bounded range of possible input values, the graphical feedback indicates this as shown in bottom picture of figure 10. The colored area stops following the modifier puck, and remains at the edge of the area of valid input. A red line indicates that the modifier puck has moved past the limit of the parameter setting. Once the user has set the parameter to the desired value, he or she can lift the modifier puck off of the table to deactivate the parameter modification mode.

6. CONCLUSIONS

The lessons learned while creating and testing applications in musical performance and business simulation suggest two design principles for use with tabletop tangible interfaces.

6.1 Make Interactions Legible for Observers

An important issue to consider in the design of these systems is the legibility of the interaction from the perspective of an observer. We first observed the importance of this principle when testing the Audiopad in a performance situation. The initial iteration of the system used the rotation of objects on the table to control the volumes of individual tracks. One of the limitations of this approach was that observers could not easily tell that a performer was rotating an object on the table because the performer's hand usually obscured the object. One of the reasons that linear movements of the objects on the table worked better for changing parameters was that audience members could see them more easily. They could observe the correlation in time between certain motions on the table, and corresponding changes in the sound produced by Audiopad, and thus begin to understand what the performers were doing.

The idea of legibility of interaction from the perspective of an observer is relevant for systems involving collocated collaboration as well. For example, in case where multiple users are interacting with a simulation, such as a business supply chain or computer network simulation, this work suggests that observers would more quickly understand the causal relationships present in the simulation if rotating gestures to change simulation parameters were replaced with linear movements of pucks on the table.

6.2 Relative Versus Absolute Mappings

Two possibilities for setting continuous parameters in a tabletop tangible interface are to use a relative mapping based on the positions of other pucks, or an absolute mapping based on a puck's position on the table itself. In application domains such as urban planning [13], an obvious mapping exists between the positions of buildings on the table and hypothetical buildings in the real world. In these types of applications, a direct spatial mapping of physical objects in the interface to a hypothetical urban site makes it easy for a user to understand and participate in the interaction. However, many applications have no such obvious direct spatial mapping. Using spatial mappings based on objects' positions relative to each other seems to work better in these cases. For example, the technique shown in figure 10 worked better than that shown in figure 9. Research by Kirsh [7] illustrates a variety of ways that people can use physical objects to offload computation from their brains to their environments. The use of absolute mappings in a tabletop TUI can prevent the user from moving the pucks on the table to employ these types of techniques. Relative mappings can also better afford multiuser collaboration, as users standing around the tabletop interaction surface can define their own reference frame within the context of their body by orienting their pucks appropriately. For many applications, it seems better to leave some degrees of freedom open to interpretation by the user.

7. ACKNOWLEDGEMENTS

The authors would like to thank Joe Paradiso, John Maeda, Tod Machover, Alex Gelman, Gerfried Stocker and all of the DJs who tried Audiopad for valuable feedback and support.

8. REFERENCES

1. Bier, E., Stone, M., Pier, K., Buxton, W., DeRose, T., "Toolglass and Magic Lenses: A See-Trough Interface," Proceedings of ACM SIGGRAPH 1993, p. 73-80.
2. Buxton, W. (1986). "There's More to Interaction than Meets the Eye: Some Issues in Manual Input." in User Centered System Design., pp. 319 - 337., 1986.
3. Fitzmaurice, G., Graspable User Interfaces. Ph.D. Thesis, University of Toronto, 1996.
4. Guiard, Y., "Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model," J. Motor Behavior, 19 (4), 1987, pp. 486-517.
5. Hopkins, D., "Directional Selection is Easy as Pie Menu!", in ;login: The Usenix Association Newsletter, Vol. 12(5), 1987.
6. Kaltenbrunner, M., "reactTable* Related", <http://www.iaa.upf.es/mtg/reactTable/?related> Referenced March 14, 2006.
7. Kirsh, D., The intelligent use of space, Journal of Artificial Intelligence, 73(1-2), 31-68, 1995.
8. Nova, N. "A List of Interactive Tables" <http://tecfa.unige.ch/perso/staf/nova/blog/2005/01/10/space-and-place-a-list-fo-interactive-tables/> Referenced March 14, 2006.
9. Patten, J., Ishii, H., Hines, J., Pangaro, G., Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces, in Proceedings of ACM CHI '01, ACM Press, pp.253-260, 2001.
10. Patten, J., Recht, B., Ishii, H., Audiopad: A Tag-based Interface for Musical Performance, in Proceedings of Conference on New Interface for Musical Expression (NIME '02), 2002.
11. T. Blaine, T. Perkis, "Jam-O-Drum, A Study in Interaction Design," Proceedings of the ACM DIS 2000 Conference, ACM Press, NY, August 2000.
12. The reactTable* Jordà, S. & Kaltenbrunner, M. & Geiger, G. & Bencina, R. Proceedings of the International Computer Music Conference (ICMC2005), Barcelona (Spain)
13. Underkoffler, J., and Ishii, H., Urp: A Luminous-Tangible Workbench for Urban Planning and Design, in Proceedings of Conference on Human Factors in Computing Systems CHI '99), ACM Press, pp. 386-393, 1999.
14. Wellner, P., Mackay, W., and Gold, R. Computer Augmented Environments: Back to the Real World. Commun. ACM, Vol. 36, No. 7, July 1993.