Crackle: A dynamic mobile multitouch topology for exploratory sound interaction

Jonathan Reus
STEIM
Achtergracht 19
1017 WL Amsterdam
jon@steim.nl

ABSTRACT

This paper describes the design of Crackle, a interactive sound and touch experience inspired by the CrackleBox. We begin by describing a ruleset for Crackle's interaction derived from the salient interactive qualities of the CrackleBox. An implementation strategy is then described for realizing the ruleset as an application for the iPhone. The paper goes on to consider the potential of using Crackle as an encapsulated interaction paradigm for exploring arbitrary sound spaces, and concludes with lessons learned on designing for multitouch surfaces as expressive input sensors.

Keywords

touchscreen, interface topology, mobile music, interaction paradigm, dynamic mapping, CrackleBox, iPhone

1. INTRODUCTION

Crackle is an interactive sound and touch experience for the iPhone that draws inspiration from the CrackleBox, the iconic "bent" touch synthesizer realized in 1975 by Michel Waisvisz at STEIM [7]. Crackle turns the multitouch surface of the iPhone into a changing sound landscape to be explored and shaped with the fingers.

The Cracklebox is likely the first commercially available portable, self-powered, audio synthesizer [2], and one of the first mobile music synthesizers to use the conductive qualities of the human body as a primary form of control. Modern multitouch smart-phones, heralded by many in the NIME community as a paradigm shift in digital music-making [4], are likewise self-powered, portable, and capable of generating a world of sounds through interaction via touch. It would seem that such a platform would be the ideal digital platform to realize an interactive experience inspired by the classic analog touch synthesizer.

One key challenge in developing Crackle was confronting the role of the human body in the sound-making process. The CrackleBox makes significant assertions about the body as an agent of control, both voluntarily (through applied touch) and involuntarily (by literally positioning the human body as part of the sound generating circuit). The iPhone, as a general purpose hardware platform with myriad levels of software abstraction, does not provide the low-level connection to its capacitive surface necessary to respond to the user's physiology. However, the touch-screen is a uniquely capable sensor for capturing applied touch. In developing his crackle instruments Waisvisz observed that physical effort exerted through human touch has an instantly recognizable way of shaping sound [7]. From this observation comes the core design philosophy of Crackle, to place the human body within an interaction paradigm that exposes the nuances of touch.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

Figure 1: The CrackleBox.

2. THE CRACKLE EXPERIENCE

> "*It could be learned by playing by ear and developing experience and manual/mental skills instead of having to dive into a world of logic, functions, interaction schemes, electronic circuit theory and mathematical synthesis methods. One could play an electronic instrument in direct relation to the immediate musical pleasure of performed sound.*" - M. Waisvisz[7]

In Crackle we sought to re-imagine the expressive, explorative, and surprising qualities of the CrackleBox on the iPhone. In interacting with any functional object there are a set of rules which define the experience empirically. You push *that*, blow into *there*, and then *this* happens. Tanaka proposes that articulation of musical phrases is not typically executed by a single interaction, but rather a set of three interactions that work in conjunction to formulate a musical utterance: 1) Binary (on/off) 2) Basic parametric choices (choice of a string, or key) and 3) Expressive control (continuous control) [6]. This model roughly describes the interaction paradigm of the CrackleBox with the exception of 2). What follows is a description of the key empirical interactive qualities of the CrackleBox that we sought to recreate in Crackle.

2.1 Expressive Control Through Touch

In touching the fingertip-sized conductive leads of the CrackleBox, one is able to create large expressive sound gestures from minute, nuanced finger movements. A small twist of the index finger results in dramatic sweeps, another causes the device to chirp loudly. One could imagine a topological map of the sound world laid out over the surface of the instrument. In some places the curves of hills and valleys are dramatic, allowing wide sound gestures to be created by moving the fingers over a very small space. In other places, the landscape slopes gently, allowing the player to use a similar gesture to control the sound far more precisely. Crackle uses this topological approach to mapping touchscreen coordinates. The precision of the iPhone's touch screen allows for extremely subtle continuous changes in finger coordinates, facilitating the continuous control element described by Tanaka's model.

2.2 Exploration of New Spaces

To play the CrackleBox is to be in a constant state of discovery. As finger placement changes, so does the sound circuitry and thus the sound world of the instrument. It can even happen that you make a large gesture and get no sound at all. In Crackle, a digital system where such behavioral anomalies must be formalized, this exploratory paradigm is given as a conditional rule: when the user changes their arrangement of fingers beyond a certain threshold the topology of the interface must also change, presenting a new set of sound possibilities to the user (Fig. 3). Through play the user gains an intuitive sense of what movements are needed to produce what kinds of utterances. The changing landscape becomes a learned part of the playing process. Wessel proposes a formal control scheme for instrumental interaction in which he describes such exploratory learning using the metaphor of babbling [8]; the voice utterances which play a critical role in the development of speech in infants. Through its constantly changing interface topology, Crackle keeps the user in a state of babble. One is constantly relearning the interaction through exploration and intuition as the sound space changes and reveals itself.

2.3 Binary Articulation

The CrackleBox defies Tanaka's tripartite classification system in that there is no basic parametric control beyond the binary active and inactive. When nothing is touched, the box is silent. One articulates different notes by touching and releasing the playing surface. It would be reasonable, then, to consider an alternative classification system for certain instrument categories; one that collapses the binary and parametric choices into a single control type. Essl comes to a similar conclusion when analyzing the touch screen as a generic input modality. Observing that the multitouch screen, sans visual metaphors which segment the interface in non-tactile ways, offers two, not three, key interaction types: 1) two-dimensional local and moving coordinate sensing (continuous) and 2) explicit support for timed tapping on the screen (binary) [3]. With timed binary articulation such an important element of multitouch interaction, care was taken in Crackle to limit topological changes during such gestures for the sake of rhythmic reproducibility.

2.4 Open Interface Metaphor

The interface metaphor can be defined as a narrative framework in which to place the possibilities within the system into a context that is logical for the user [9]. Crackle, as a rule, favors a simple and suggestive interface metaphor over the musically denotative metaphors used in iPhone music applications such as Pocket Guitar (Bonnet Inc) and Pianist (MooCowMusic). This design choice echoes the progeny of the CrackleBox, which began with Waisvisz's desire to escape the connotations of religious music and western tonality inherent in keyboards used to control early analog synthesizers [7]. In Crackle, the iPhone's multitouch surface is visually segmented into six rectangular areas which give a hint where to begin, but do not enforce a particular interaction beyond presenting the touch surface as a playable object, encouraging the user to jump in, touch, and explore.

3. IMPLEMENTATION

The following is a discussion of how the interaction paradigm described in the previous section are implemented in Crackle. The interface is implemented conceptually through a combination of surface segmentation, mapping generation, dynamic remapping, and a pseudo-chaotic sound model. The combination of the individual segment mapping topologies with the unmapped interstitial space between segments creates a single complex surface topology for exploring the possibility space of the sound engine.

3.1 Touchscreen Interface Segmentation

At the bottom of it all is the iPhone's multi-touch sensor, whose Cartesian coordinate space is segmented into six rectangular sub-spaces. This initial segmentation is communicated on-screen as six rectangular touch zones; a simple visual queue that suggests where to begin touching. The user interacts with the interface by placing up to five fingers on the touch zones and by moving fingers within and between the zones. Aside from a small settings dialog button in the upper left corner of the screen, the entire screen surface is playable. More detailed information about the underlying topology could have been communicated through visualization techniques such as color-coded topological heat maps. However, it was decided not to implement such feedback in order to encourage this topological information to be discovered through touch and listening rather than visually.

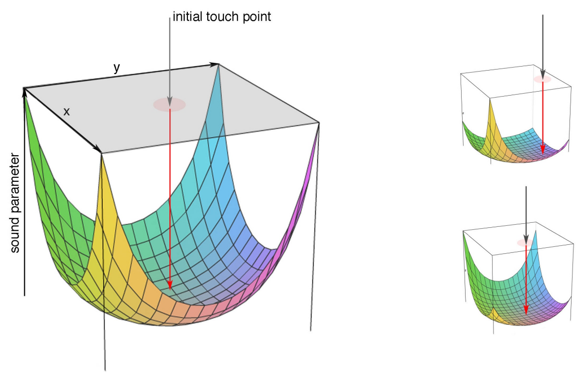

Figure 2: Graphs of conversion functions from segmentation coordinates to single control parameters. The graphs form a basin throughout much of the coordinate space and grow exponentially towards the edges.

3.2 Segment Mappings

Once segmented, the coordinate space of each segmentation is assigned a dynamically generated mapping to the sound engine's parameters. The mapping algorithms fall into four categories, based upon the target control parameters of the sound engine: period, modulation, period and modulation, or dead-zone. When a mapping is assigned to a segmentation, unique conversion functions are generated based upon initial touch position and pseudo-random variation. The conversion functions reduce the two-dimensional parameter space of the segment to a single normalized sound control parameter. In the case where a segmentation simultaneously controls modulation and pitch, two independent conversion functions are generated. The dead-zone mapping is a mapping which controls nothing. The purpose of this mapping is to create an additional element of unpredictability and anticlimax during play. The analog Cracklebox also has this surprising quirk, but it bears repeating that with digital instruments surprises must often be formalized.

A 3D graph of the family of conversion functions illustrates the topological features of these mappings (Fig. 2). The graphs show a parabolic scoop whose shape flattens into a basin around the minimum point. The coordinate space of the flattened basin area is mapped to the less chaotic sonic qualities of the sound model, making these sounds easy to find and precisely controllable. As the finger approaches the far edges of the basin, the conversion function grows exponentially large, evoking sonically different and often chaotic results from the DSP algorithm. This mapping approach comes from personal observation of traditional instruments, where expected, "pleasant" tones are easy to find and control relative to more eccentric musical utterances. For example, a saxophone has a wide range of possible sounds, from the recognizable timbre of its stable tones to more abrasive honks and squeaks. The sax is built in a way, through the arrangement of finger keys and form factor, to deftly and easily command stable tones. While to master the more cacophonous, yet extremely expressive, part of the saxophone's sound world such as chirps, growls, portamento and quartertones, a performer must explore the eccentric edges of the instrument's interface. This was an apt approach for mapping Crackle's surface coordinates. The end result being that within segmentations Crackle gives the user a wide range of sonic expression but prioritizes the available sound world through ease of discovery and control.

3.3 Interstitial Space and Overall Topology

The interstitial space between touch zones is unmapped. However, a sample-and-hold strategy is employed on sound source parameters as a user moves her fingers smoothly from one touch zone to another. The slow de-zipping qualities of the DSP engine smooth any sudden jumps in parameter values as a new touch zone is entered from the unmapped space between the touch zones, making the user feel as if they are "jumping up to", or "falling in to", the scoop of the new area. Creating such discontinuities in the interface allows for complex sonic gestures which would not be possible using strictly continuous mapping.

3.4 Dynamic Mappings

In order to create the desired sensation of exploring an ever-changing sound space, the mappings of the six segmentations are dynamically remapped each time a finger is either added or removed from the touch surface. (Fig. 3)

During initial development, two algorithmic approaches were taken to generate the overall layout of six mappings: an evolutionary model, and a pseudo-random model. The evolutionary model generated a new mapping for each touch zone as needed, choosing the target control parameters and conversion function based upon the current mappings of the other zones. So long as a segmentation was still being touched, it's conversion functions were retained. When a segmentation was no longer touched its conversion functions would be removed from the configuration until it was touched again, at which point new functions were generated. Target parameters for newly touched segmentations were generated using an algorithm which first would analyze the existing configuration to ensure a functioning instrument that can produce sound (i.e. containing at least one pitch control mapping).

In contrast, the pseudo-random model generated a completely new mapping configuration for all six segmentations each time a significant touch configuration change occurred. The distribution of control parameters and conversion functions in the pseudo-random model was generated probabilistically, with additional checks to ensure a set of mappings that would create a functioning instrument.

The pseudo-random model proved to be the more successful of the two, based upon comparisons with the interaction experience of the CrackleBox. This is the algorithm used in the final implementation.

In addition, when a user moves fingers from one configuration to a new one and then back again (tapping the screen), the mappings of the first configuration are remembered and restored, providing the desired element of reproducibility.

3.5 Sound Model

The sound engine uses a system of squared sinusoids with a de-zipping algorithm applied to smooth changes in frequency and amplitude. This model was chosen because of its sonic similarity to the basic tone of the CrackleBox, and for the harmonics and pseudo-chaotic behavior found when given extreme parameter values as input, making this sound engine an especially apt choice for a wide range of sonic variety. The de-zipping algorithm is also deliberately slowed to create sounds similar to the characteristic pitch sweeps of the CrackleBox.

4. FUTURE WORK

In future work, we would like to apply the Crackle interaction paradigm to alternative sound models. We envision using Crackle as an encapsulated, general purpose interface for controlling any parameter space. Additional functionality could be implemented in the application to send normalized parameter values out via Open Sound Control to manipulate synthesis parameters of arbitrary sound models. A key question is discerning where the interaction experience ends and the sound model begins, and ultimately whether they can be divided at all. If a dividing line can be found, this implies that the application could be used as a general purpose controller for navigating sample material, granular clouds, or even lighting and visuals in a very characteristic way. Experiments and analysis would be necessary to determine if such a re-application would retain the same interaction experience and topological feel. Mapping has been covered extensively in the NIME literature, but the proposition of utilizing a mobile device as a black-box mapping and interaction system has yet to be fully explored. Such an encapsulated interface provides an attractive alternative to the limitations of simple one-to-one mappings without falling victim to the curse of programmability that hinders artistic mastery [1]. It is interesting to note that this approach flies in the face of the trends observed in many modern musical interfaces, such as the Monome and Syderphonics Manta, which promote an extremely open, programmable, architecture. At the very least, an encapsulated approach has potential as a method of developing widgets which could provide a canon of new multi-dimensional control metaphors for music on touchscreen-enabled computing devices.

5. DISCUSSION

Figure 3: Touchscreen segmentation and different finger configurations which would cause a remapping of the segments.

In designing Crackle, care was taken to preserve the role of the body's perceptual knowledge in interacting with the touchscreen. On-screen queues were provided to hint at the interaction metaphor, but not explicitly reveal it. Initial beta tests have suggested that soon after engaging with the application, visual feedback becomes a secondary modality to sound and touch. This attention switching, from visual to sonic and haptic, suggests that the visual modality at some point becomes less important to the interactive experience. Recent studies in cognitive science show that attention switching between sensory modalities such as vision and audition comes at a behavioral cost. Experimental subjects have exhibited slower perceptual judgements when constantly switching attention between modalities when compared to not switching [5]. This would imply that, as our beta testers began to actively inhibit one of the three sensory modalities (visual), they learned to minimize cross-modal attention switching and achieved a more immediate connection to sound production. Whether this suggests that the visual modality is in competition with touch and audition during musical performance is an interesting question warranting further experiments and empirical observations.

6. CONCLUDING REMARKS

Through developing Crackle we have learned a great deal about designing sonic interactivity for touchscreen computing devices. It is clear that creating an interesting underlying topology plays a key role in the success of using a touchscreen as a generic input modality for musical expression. The approach of mapping eccentricities of the sound model to less centralized (or extended) parts of the interface worked well, and we believe this is an approach that deserves further exploration in the design of new instruments for musical expression. It is also clear that limiting the visual metaphors of a touch screen to those which are not musically denotative encourages intuitive learning of an interface through babble, which can result in new and novel sound utterances. In treating the mobile touchscreen as a general purpose input modality, it appears that the key to a successful control scheme is a focus on topology, especially compound topologies. The mixing of continuous controls with discontinuities reveal new and novel ways of exploring a sound possibility space.

7. ACKNOWLEDGMENTS

The author would like to thank Frank Baldé for his assistance during the initial stages of development, Daniël Schorno and Dick Rijken for their numerous insights on instrument building and interaction design, and STEIM for supporting this work.

8. REFERENCES

[1] P. Cook. Re-Designing Principles for Computer Music Controllers: A Case Study of SqueezeVox Maggie. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME). Pittsburgh, Pennsylvania: Carnegie Mellon University*, pages 218–221, 2009.

[2] E. Dykstra-Erickson and J. Arnowitz. Michel Waisvisz: the man and the hands. *Interactions*, 12(5):63–67, 2005.

[3] G. Essl and M. Rohs. Interactivity for mobile music-making. *Organised Sound*, 14(02):197–207, 2009.

[4] N. Kirisits, F. Behrendt, L. Gaye, and A. Tanaka. *Creative Interactions - The Mobile Music Workshops 2004-2008*. University of Applied Arts Vienna, Vienna, Austria, 2008.

[5] S. Lukas, A. Philipp, and I. Koch. Switching attention between modalities: further evidence for visual dominance. *Psychological research*, 74(3):255–267, 2010.

[6] A. Tanaka. Mapping Out Instruments, Affordances, and Mobiles. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, 2010.

[7] M. Waisvisz. Crackle history. `http://www.crackle.org/CrackleBox.htm`, March 2004.

[8] D. Wessel. An enactive approach to computer music performance. *Actes des recontres musicales pluridisciplinaires, Lyon, Grame*, 2006.

[9] C. Zwick, B. Schmitz, and K. Kühl. *Designing for Small Screens*. AVA, 2005.