Modal Kombat: Competition and Choreography in Synesthetic Musical Performance

[The First Ever Instrument-Controlled Video Game Battle]

David Hindman[*]
New York University, ITP[†]
721 Broadway, 4th Floor
New York, New York
david.hindman@aya.yale.edu

ABSTRACT

Public competition has been a part of the human experience for at least all of recorded history. The same is true for music. *Modal Kombat* is a live guitar competition channeled through the video game *Mortal Kombat.* The performance work builds on the idea of *dueling banjos*, one of the few examples of public musical competition. *Modal Kombat* takes this concept one step further by allowing a virtual mythological warrior to embody musical gestures. The result is a musical competition that utilizes advanced technology and popular media to exemplify the human passion for the spectacle sport.

This paper describes the technological and musical challenges that face performers, composers, and programmers when producing performance works that incorporate the control of pre-recorded graphical animations with audio and MIDI. Topics covered will include the musical instrument as an interface, choreography and competition, visual theatrics, performance flow, and compositional structures that allow for asynchronous rhythmic patterns inherent to and resultant from game controller input. In addition to its theoretical inquiry, the paper will discuss the practical artistic and technical needs required to successfully create a public instrument-controlled video game competition/concert, through documentation of *Modal Kombat*, the first ever instrument controlled video game battle.

Keywords

Video Games, Guitar, Interface, Instrument, Performance, Video Game Battle, Music Games, Sonic Games, Controlling Video Games With Sound, Modal Kombat, Mortal Kombat, Competition, Choerography, Human-Computer Interaction.

[*]Composition and Performance of *Modal Kombat* in collaboration with Evan Drummond of the Eastman School of Music

[†]Interactive Telecommunications Program

1. INTRODUCTION

Figure 1: *The Sonictroller (www.sonictroller.com)*, by David Hindman and Spencer Kiser, is a system that allows existing video games to be controlled by traditional, full-range musical instruments. *Modal Kombat (www.modalkombat.com)* builds on this system, extending its application in to the live performance setting as a hybrid musical performance and public video game battle.

Musical video game interfaces have existed for some time, but have not reached their full potential because they are game-specific and lack the versatility and tonal range of traditional musical instruments. With few exceptions, existing music gaming interfaces are nothing but toys and could never be used for music production apart from the single video game they were designed to control.[1] *Real* musical instruments have no doubt been overlooked as interfaces in the gaming industry because the market for such interfaces (skilled musicians) is much smaller than that of unskilled or hopeful instrumentalists. Also, interfacing a real musical instrument with a video game is far more complex than simply engineering a simplified instrument-like interface to be distributed with an accompanying game. This is why the most successful musical interfaces so far have relied on simple musical input, like the rhythmic interface for Nintendo's *Donkey Konga*, and single line melodic recognition

[1]*Karaoke Revolution*, by Harmonix, is an exception, it relies on the human voice for input. The interface for Nintendo's *Donkey Konga* also can be used as an acoustic instrument.

Figure 2: *Modal Kombat* in live performance. Two classical MIDI guitars control video game characters in the popular video game Mortal Kombat. The result is a hybrid musical concert and public video game competition.

or trigger recognition in Harnomix' *Karaoke Revolution* or *Guitar Hero*. As a result, practicing musicians have had to take step backward to participate in the musical gaming process. For a solution to this problem, I have created a system called *The Sonictroller*, a device which allows traditional, full-range musical instruments to control sprites in a wide range of console video games. I was motivated by the following questions: Why not have the option of playing a video game with a real electric guitar or bass? Or piano? Or drums? If anything, this option would engage current instrumentalists, and more importantly, it may create new ones. Furthermore, why limit a musician to a particular video game?

Figure 3: Shown above is the congo drum interface for Nintendo's *Donkey Konga.* It is the closest match to an real musical instrument gaming interface, as the device produces acoustic sound in additon triggering electronic events. The musical input is simple momentary controls in the form of beats, the game accepts no harmonic or melodic input.

Now, with the ability of concert-quality instruments to control video game sprites, we are witnessing the convergence of two previously separate forms of entertainment: the musical concert and the public video game competition. This paper focuses on the new performance genre that *The Sonictroller* has allowed to emerge, and describes the musical and technical considerations required to produce a work in this genre, building from the personal experience of creating *Modal Kombat*.

2. BUTTONS AND HANDLES

The content in this section is largely influenced by William Verplank, through a series of lectures at the Physical Interaction Design Workshop for Music at CCRMA 2004.[2]

Verplank describes how human-computer interfaces can be described as either **buttons** or **handles.** Buttons are momentary controls, switches, a brief change from one state to the next, a plucked string, a drum beat. Handles are always updating, continuously sending information, a long vocal melisma, a bowed violin. Musical sound is nothing but information; it can be listened to by the human ear for the purpose of pleasure, or it can be listened to by computers for the purpose of control. For example, a guitar can not only produce beautiful music for the human audience, it can also be a control panel, an array of 114+ discrete switches. A French Horn could be a mouse or joystick, volume on the x-axis and pitch on the y-axis.

Controlling video games sprites with musical gestures relies heavily on the concept of the musical instrument as valid human-computer Interface. When controlling the visual gestures of video game characters, I have found the knowledge of fundamentals of human-computer interaction to be invaluable in understanding how the musical gesture maps to the virtual physical gesture. At the heart of this lies the two main components of human-computer interaction: momentary and continuos controllers.

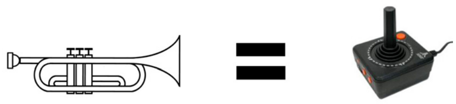

Figure 4: Wind Instruments, like the trumpet, are capable of acting as *handles*, continuous control devices because of their ability to output a constant stream of information.

Figure 5: Plucked or percussive instruments, like the MIDI piano or guitar shown above, are capable of acting as *buttons*, momentary control devices, because of their ability to trigger instantaneous, short-lived data. Note: Many instruments, such as wind or bowed instruments, can act as buttons or handles.

3. SHALL WE DANCE

The original goal behind the creation of the *The Sonictroller* was to create situations in which musical performers could compete through the control of video game characters. The most surprising by-product of this invention was the

[2]CCRMA: Center for Computer Research in Music and Acoustics. Physical Interaction Design workshop for music, July 2004.

discovery that the graphical animations of virtual characters could be re-purposed to make them dance, rather than fight, advance, shoot, conquer, race, etc. While the pallet of moves in this case is limited by the programmers intentions, there is no reason why a game character's moves can not be recycled specifically for the purposes of testing small or large-scale dance choreography. In this specific case, the single notes cause both characters to duck, so along with the audible staggering of pitches we see a visual staggering of characters kneeling and standing. The end result looks very much like the characters are dancing along *with* rather than *because of* the music.

3.1 Virtual Choreography

Movement I of *Modal Kombat*, entitled *Rain Dance*, makes use of sonically-driven virtual choreography through controlling the character named *Rain*. The peice begins with two guitars alternating single, staggered pitches, shown in Figure 6 and Figure 7. Each of these single notes triggers a single mommentary action of a corresponding video game character.

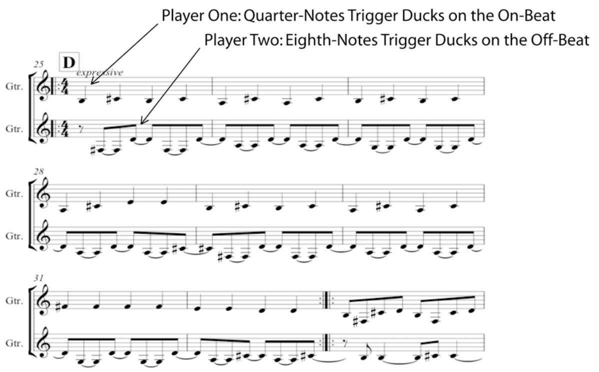

Figure 6: Above, the score for Movement I of *Modal Kombat* is shown. To start, staggered single notes between the two guitars trigger a choreographed sequence. Each character ducks on their respective beat, resulting in a dance-like effect.

4. COMPOSITION

While an unexpected outcome of controlling video games with instruments has been the effectiveness of the choreographed virtual dance, the main goal for this research was motivated by the idea of musical competition reflected in a visual feedback scenario. The challenge of creating a competition-based musical performance is mainly devising a compositional structure that allows for asynchronous rhythmic patterns inherent to and resultant from game controller input.

4.1 Harmonic Structure in the Musically Controlled Video Game Competition

Once the initial mechanism for mapping musical gestures to game characters is in place, a greater challenge arises. The challenge is creating listenable music out of random streams of input. Since video game competition is by nature unplanned, the music in *Modal Kombat* is largely *harmonic contextual*, a series of tone areas that work well as a foundation for improvisation. Once a harmonic ground is established, parts of it can be looped- recorded live, and then improvised over. The improvisation of guitar solos over a harmonic ground drives the competition aspect of the performance. This way, the performance can be unplanned and different each time, two fundamental components of the video game competition. While random cacophony is nice at times, the challenge of this work is to create listenable music while still allowing the inevitable madness native to a public video game battle. The next section describes various mechanisms for establishing harmonic foundations on which to base improvised game play.

Figure 7: Shown above is the visual accompaniment to measures 1-4 of the score shown in Figure 6. As we can see, each character ducks when a single note is struck, resulting in a choreographed section resembling a tribal dance. This can viewed online at http://www.modalkombat.com/trailer.mov

4.2 Technical Mechanisms for Maintaining Harmonic Continuity

There exist a number of different mechanisms for looping harmonic passages for the purposes of contextual improvisation. Hardware devices such as the Lexicon Jamman or Gibson Echoplex are popular in the looping community. *Modal Kombat* employs a custom performance patch written in Max/Msp that records live passages and then loops them under improvised solos. The patch, entitled *LiveLooper* is controlled with a MIDI footpedal and allows up to seven live overdubbs. Built-in to the patch include digital delay and separate solo and looping volume controls.

5. VISUAL THEATRICS: USING OPEN GL TO REMIX LIGHT AND ORIENTATION

Once a harmonic context is established for improvisation and a seamless program is implemented, we can add visual theatrics to the sonically-controlled video game battle. Cycling '74's Jitter library for Max/Msp provides an excellent way of adding extra visual dimensions to existing game graphics. Open GL gives us three-dimensional world in which we can render the two-dimensional graphics of older console video games. Using the OpenGL environment in Jitter allows built-in lighting and camera effects at minimal cost to processing speed. The following paragraphs discuss various OpenGL-based theatrical lighting and camera effects that can be controlled with sound, to enhance the visual experience of existing video games.

Figure 8: OpenGL allows 3-dimensional theatrical lighting effects to enhance the 2-dimensional space of the game Mortal Kombat. The lighting features in OpenGL allow for different instruments to generate different sonically mapped color explosions. In this figure, the volume from Guitar 1 generates red explosions of light.

5.1 Lights

Jitter's OpenGL functionality includes built-in lighting parameters for RGB color including light position, diffusion, color, and intensity. Movement II of *Modal Kombat* is based on sonically controlled theatrical lighting and camera effects made possible by Jitter's OpenGL. One obvious but effective mapping of audio in this movement is sonic amplitude to lighting intensity. As Figure 8 illustrates, it is possible to focus light on certain characters to create the illusion of the sprites performing under theatrical spotlights on a stage. Mapping the sonic amplitude to lighting intensity allows for effective explosions of focused light, that follow the many discrete levels of audio volume.

Equally interesting lighting effects include specific color mappings. In *Modal Kombat*, volume of each guitar generates certain colors: Guitar One generates red light while Guitar Two makes blue light. This is a simple but meaningful use of color to distinguish one competitor from the other, in the same way that teams in sporting events must where noticeably different colored uniforms.

5.2 Camera

Lighting is one feature of OpenGL that maps well from sound. Another component that we can use for theatrical effects is camera position. Some effects include setting the camera to zoom in on key harmonic figures or creating automated zooms to emphasize particular musical moments. Figure 9 shows a portion of an automated zoom placed at the end of Movement II of *Modal Kombat.* Automated zooms are useful for creating effects concerning the orientation of the two-dimensional screen. For example, figure 9 shows a the OpenGL window in the max patch for *Modal Kombat*, with the video plane tilted and camera at an altered z- axis, yielding the impression of the screen flying through the air. Note that the game charactrers are still fully functional- we can interface sound to all elements of screen orientation as well as to the actual content dislayed in the original game.

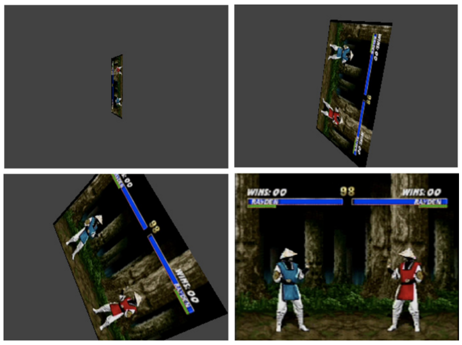

Figure 9: Camera positions can also be automated, to create the effect like the one shown above. The 2-dimensional plane flies through the 3-dimensional space, an effect inspired by the glass prison in the movie Superman 2.

6. CONCLUSIONS

The production of *Modal Kombat* has required an understanding of the musical instrument as a control device, a mechanism for this interface, thought-out compositional structures, well-planned performance transitions and remixed visual presentations. Certainly a video-game based musical production need not demonstrate all of the above qualities, but the success of *Modal Kombat* lies not simply in the ability to control video games with musical instruments but in the ability to use this technology to create large, seamless, well-planned performances. *Modal Kombat* has given careful consideration not only to just the technology that forms the foundation for its existence but to the musical language and presentation of this new genre of multimedia performance. This paper, in addition to serving as written documentation of the existing work *Modal Kombat* should also provide an overall outline of artistic and technical considerations for anyone choosing produce a work in this genre. Game Over.

7. ACKNOWLEDGMENTS

We would like to thank the following: Tom Igoe, Spencer Kiser, Chris Carrington, Jeff Gray, Mary and Jeff Hindman.

8. REFERENCES

[1] *Physical Computing*, by Dan O'Sullivan and Tom Igoe
[2] CCRMA Physical Interaction Design Workshop, Summer 2004, Lectures by William Verplank
[3] Midway Games, Inc
[4] Nintendo Co., Ltd.
[5] Harmonix Games
[6] Cycling '74, http://www.cycling74.com/