

Sputnik

Project Report HCC Project Seminar 2011

Simon Wallner*

9th December 2011

This paper evaluates *Sputnik* a 3D environment with which the user can freely interact through an elastic *arc of light/fishing rod* metaphor, to explore, create and interact with virtual *sound objects*. These sound objects are placed in the scene and react to the user's input by sending MIDI commands to an external audio program thus creating or manipulating the sound.

1 Reading Guide

Appendices Code Prerequisites Video CD etc...

2 Introduction

Computer music is around us for some time now and through the use of the computer musicians have sheer endless possibilities of musical expression. With this plethora of possibilities comes the need for constraints and control to harness this expressive potential. Over the recent years many standard and non-standard interface have been developed, ranging from the ordinary button-fader-nob MIDI interface to more elaborate interfaces and systems like the *reactable*[Jordà et al., 2007], *mix-iTUI*[Pedersen and Hornbæk, 2009] or commercial solutions like the *Novation Launchpad*¹ or *Native Instruments Maschine*² to name but a few.

With the advent of motion based controllers in consumer entertainment systems, marked by the release of the *Wii*³ console in late 2006, motion controllers became

*me@simonwallner.at

¹http://www.novationmusic.com/products/midi_controllers/launchpad

²<http://www.native-instruments.com/#/en/products/producer/maschine/>

³<http://de.wikipedia.org/wiki/Wii>

widely and cheaply available. This and their interface capabilities make them the ideal tools to explore the realm of *new interfaces for musical expression*.

A common problem of computer music interfaces is that often the process of sound creation is not readily comprehensible. Seeing a performer on stage behind their laptop twisting knobs and adjusting faders might be ambiguous to an uninformed observer. It can be hard to relate the artist's action to the resulting sounds. This can hinder the experience and might go as far as to the point where the audience suspects that an artist just pressed play, as interviews conducted by [Pedersen and Hornbæk, 2009] show.

This paper introduces *Sputnik*, a system that uses a *Wiimote* controller to interact with a dynamic 3D scene. In the scene, a variety of sound creating objects are placed that send MIDI signals to an external audio program upon the user's interaction.

Users can freely navigate the 3D scene and interact with it through an elastic *arc of light/fishing rod* metaphor. It seems as if the *arc of light* was coming out of the *Wiimote* and reaches into the scene, acting as an extension of the user's body into the virtual space. With this bodily extension users can *grab* and *drag* objects around the 3D scene.

In this paper I evaluate the qualities of the *arc or light* metaphor and how the design decisions/constraints of the system influence its expressive potential both visually and musically. This evaluation is grounded in a user study of XXX users.

Based on these findings and the theoretical framework of [Ullmer and Ishii, 2000] the similarities and differences between *Sputnik* and tangible user interfaces are discussed.

3 Paper Outline

The following section gives an overview over related work in the field of *New Interfaces for Musical Expression* and tangible user interfaces. Section 5 goes into detail about *Sputnik*, both on a conceptual and a technical level. Section 6 describes the performed user study and the paper is finally concluded in section 7 where the findings are discussed.

4 Related Work

Practical Work Only a few projects exist that go into a similar direction as *Sputnik*. The *Virtual Xylophone*[Mäki-Patola et al., 2005] is a virtual reality system in which the user can place xylophone bars of different pitch in the scene and then struck them with a virtual mallet. By translating the configuration and mapping of the real instrument into the VR environment, new modes of play emerge. [Zappi et al., 2010] created a virtual controller for *Ableton Live* that allows users to create simple proxy objects in a VR environment, bind them to certain controls and use them effectively as

virtual sliders. [Rodet et al., 2005] created a virtual environment for an exhibition setting. Users interact with the system via a 6-DOF motion tracker with tactile feedback. However, the user’s actions in the system are highly constrained.

More projects can be found in the realm of Tangible User Interfaces. With *mixiTUI* [Pedersen and Hornbæk, 2009] created a table top tangible interface for a sequencer that aimed not only to be functional but also to visually enrich the artists performance. Interviews with musicians and an extensive user study have been performed. [Jordà et al., 2007] created the famous *reacTable*, also a table top tangible interface that allows the creation and manipulation of music by composing various objects on its surface.

After the release of the Wii in late 2006, the Wiimote motion controller received some attention in and outside the field of musical interfaces: [Kiefer et al., 2008] assess the general qualities of the Wiimote as a musical controller and [Miller, 2010] uses the Wiimote and sensor bar to create the *Wiolin*, a virtual violin that mimics the real instrument and can be played either in an upright position like a cello or horizontally like a violin. It senses the button presses and tracks the movement of the *bow*, i.e. the sensor bar to create the sounds.

Not a Wiimote but still impressive, [Miyama, 2010] uses an low resolution distance sensor array to control the many parameters of a synthesizer. A small gui application is merely used for monitoring the system’s state, and sound creation is done in pd.

Theoretical Work The field of *Tangible User Interfaces (TUI)* provides part of the theoretical background for this work. Work of [Fitzmaurice et al., 1995] and then later [Ishii and Ullmer, 1997] introduced this term and the wider concept. [Shaer, 2009] Gives a very good overview over this field as well as the history of TUI studies. [Ullmer and Ishii, 2000] introduced *MCRpd*, a formal model for describing and analysing TUIs that will be used in section ??.

[Sharlin et al., 2004] introduced *spacial TUIs* that focuses on *I/O unification* by tightly coupling the action and perception spaces and embodying a clear state representation across all sensory modalities.

Entering the musical realm [Fels and Lyons, 2011] give a good overview and general introduction into the field of *NIMEs (New Interfaces for Musical Expression)*. [Cook, 2001] shares 13 general principles for designing computer music controllers that resulted from his long lasting experience in this field. [Dobrian and Koppelman, 2006] asks the question of virtuosity and expression by pointing out the elephant in the room, e.g. the lack thereof and also the lack of a comparable standard repertoire.

In contrary to that [Gurevich and Treviño, 2007] question the hegemonic *composer–interpret–listener* relation in favour of a more holistic *ecological* view of musical expression. Later work by [Gurevich et al., 2010] evaluated a highly constrained, prototypical one-button instrument that spurred a wide variety of play styles in test users.

Closing the loop to design and HCI, [Magnusson, 2010] gives a good overview over

the field of *affordance* and elaborates on *constraints* from different viewing angles and how they impact and support creativity. Finally, [Wanderley and Orio, 2002] goes into depth over evaluating input devices for musical expression in the context of HCI.

5 Results

7-10 pages

Description of the Implementation according to the learning goals:

5.1 Sputnik Overview

Sputnik is a *New Interface for Musical Expression* that combines 3D graphics with the capabilities of the Wiimote controller. The user is presented with colourful 3D scene that contains various simple objects. The user can freely navigate this scene and interact with these object to create sounds.

The user can interact with the system via a Wiimote and Nunchuck controller. The Nunchuck controls the users movement in the 3D space and the Wiimote is used to control the camera and to interact with the objects in the scene.

5.1.1 Setup

The system is set up in a room with an overhead mounted video beamer. The sensor bar can either be placed on the upper or lower edge of the projected image. It consists of two IR emitters that allows the IR camera in the Wiimote to track its relative orientation in space. A Wiimote and Nunchuck controller are used and only a single person at a time can use the system. Figure 1 illustrates the setup.

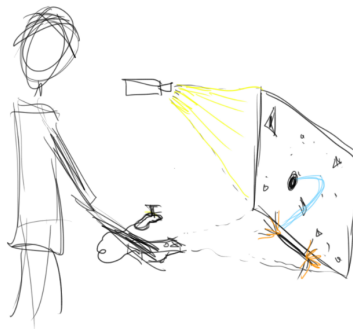


Figure 1: Standard set up of Sputnik

5.2 Navigation and Camera Controls

Sputnik's virtual scene uses a fixed up direction and the user can move through the scene by pushing the Nunchuck's analogue stick in the respective direction. Pushing the stick forward moves the camera into the scene, pushing it left moves the camera to the left and vice versa.

Tilting and panning is controlled by pointing the Wiimote to the top/bottom/left/right of the screen. The farther it is pointed away from the neutral center position the faster the camera movement is. Figure 3 illustrates the navigation.

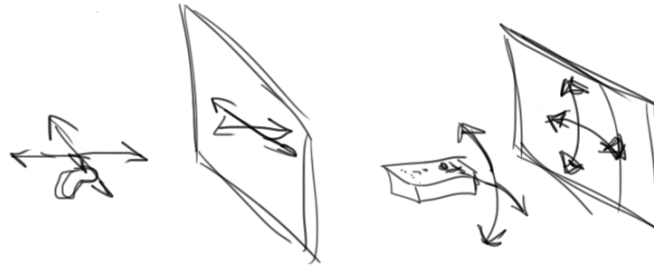


Figure 2: The Nunchuck's analogue stick controls the camera's dolly and track movements, the Wiimote's IR pointer controls the camera's tilt and pan movements.

5.2.1 Interaction

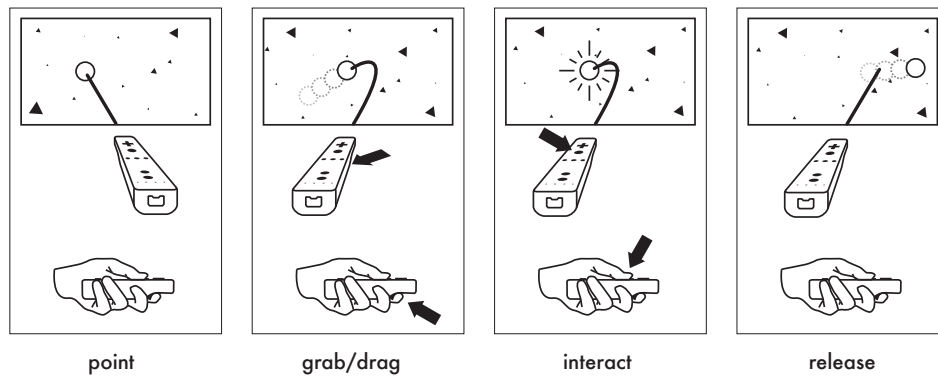


Figure 3: The basic interaction vocabulary of Sputnik

Figure 2 illustrates the basic interaction vocabulary of Sputnik. The user can in-

interact with the scene via an *arc of light metaphor*. It seems as if the arc of light was coming out of the Wiimote and reaches into the scene, acting as a bodily extension of the user into the virtual space. Through this *arc* objects in the scene can be *pointed* at, *grabbed* and *dragged* around as well as *released*.

Objects behave in a simplified yet physically plausible way, each featuring distinct weight and friction. Dragging objects causes the arc to bend like a fishing rod, reflecting the physical properties of the object.

Each object in the scene can react individually to user interaction. The following classes of objects exist in Sputnik:

Sampler (red sphere) The *sampler* object reacts to the press of the *A Button* while it is grabbed. While the button is held the sample is played in a loop and stops immediately when the button or the object is released. Playback always starts at the beginning of the sample.

Player (yellow sphere) The *player* object reacts to the press of the *A Button* while it is grabbed. Pressing the *A button* starts and stops the player. It is not automatically stopped when it is released. Playback is looped and always starts at the beginning.

Tape Machine (grey sphere) Modelled after an old tape machine and inspired by *musique concrète* the *tape machine* object controls the speed of the play back by the object's movement speed in the virtual space. The faster it moves the faster the playback. Playback is looped.

Harmonic Harp (orange spheres) The orange spheres form a kind kind *harmonic harp*. Each sphere controls a single sine wave oscillator and the volume is determined by the the spheres distance to is origin. Additionally a force exists that drags the spheres towards their respective origins. The oscillators are tuned to the natural harmonic series.

By interacting with the different objects and arranging the in the virtual scene users can create musical performances that are both musically and visually expressive.

5.3 Implementation

Sputnik is built on C++ and is developed on Mac Os X. It is split up in two projects with *kocmoc* building the (more general purpose) core foundation for Sputnik. Small parts of *kocmoc* already existed before this project, but most of it was written in this project.

5.3.1 Component Based

At its core *kocmoc* uses a simple component based architecture to manage *objects*. An object is simply a collection of components and components can directly communicate

with each other and are configured at runtime. One of the benefits of component based systems is that new objects can be easily be created by reusing existing component. Extending existing components is often only a matter of adding an existing component to it. It is a relative flexible approach that allows quick changes and keeps the class hierarchy flat.

5.3.2 OpenGL

OpenGL is used for rendering the scene. Sputnik uses OpenGL 2.1 with a few common extensions. The renderer is rather simple and only supports the few features that are required by Sputnik. It has not been tuned to production standards but should be reasonably fast for the task at hand.

5.3.3 MIDI Output

The RtMidi⁴ was used to interface with the MIDI protocol⁵. MIDI was used because of its simplicity and wide adoption in the field of computer music. By using this ubiquitous protocol Sputnik can potentially communicate with virtually every music program.

A disadvantage of of MIDI is the rather low resolution discrete value space. Control values are in the discrete range of $[0, 127]$. Depending on the application this can be a severe limitation, especially if one wants to map for example $[0, \infty)$ to a midi value. Another problem is the limited bandwidth of MIDI.

OSC⁶ is the natural successor of MIDI and alleviates most of MIDI's problems. On the contrary it is more complex and support in application is not as widespread.

5.3.4 Wiimote Input

Wiimote input is handled by the WiiC⁷ library. It allows to access all parts of the wiimote but unfortunately it is not a stable production grade library which can be troublesome at times.

5.4 The Virtual Scene

Sputnik uses a relative simple 3D virtual environment. It contains the interactive objects as well as a randomly generated *star field* and a slight coloured fog that makes objects disappear smoothly in the distance. This fog however serves as an important distance cue to the human perception along with the relative size of objects.

⁴<http://www.music.mcgill.ca/~gary/rtmidi/>

⁵<http://en.wikipedia.org/wiki/MIDI>

⁶http://en.wikipedia.org/wiki/Open_Sound_Control

⁷<http://wiic.sourceforge.net/>

The user sees the scene from the first person perspective and the virtual camera has a relatively wide field of view. To compensate the strong stretching of geometry in the corners of the perspective projection a barrel distortion is introduced as a post processing effect. This makes the image a bit smoother and also adds full frame anti aliasing in this step.

Adhering to the convention found in most video games and also films I decided to use a fixed up axis. Even though the scene is set in *space* the camera cannot be completely freely manoeuvred and *up* always points in the positive *Y* direction. Limiting the camera's movement a little should make it easier for users and fits the *common* camera model.

5.5 Wiimote input and control

The user interacts with Sputnik through a Wiimote and a Nunchuck controller that were originally created for the Wii video game console. The Wiimote is quite similar in shape to a standard TV remote, has a few buttons and a built in IR camera that allows it to function as a pointing device by pointing it at a *sensor bar* composed of IR-LED lights. The Nunchuck controller only consists of an analogue stick and two buttons.

[Wiimote + Nunchuck Schematic here!]

Camera tilt and pan is controlled by the Wiimote pointer. Pointing at the center of the screen results in no camera motion. Pointing to the left or the right causes the camera to pan in the respective direction and pointing up and down causes the camera to tilt. Movements are smooth and the amount of movement is relative to the distance to the center position. Additionally an continuous strictly monotonic increasing function is used as an acceleration function.

A severe drawback of the Wiimote's pointing capabilities is the relatively small field of view of the IR camera. It is not adequate to capture bigger motions. It is best controlled by using mostly the wrist and very little arm and body movement. This contradicts the initial goal of enabling Sputnik to capture bigger movements.

The Wiimote processes the IR camera feed directly on chip and provides the locations of up to 4 tracking points as output. Several sources on the internet report the update rate of the Wiimote at 100Hz

5.6 Creating sound

Introduction to pd, why it was chosen, What it does.

5.7 Mapping the whole system

overview over the whole system and design decisions behind it. why it is what it is.
Positioning objects in the scene – spatial arrangement.

6 Evaluation

3-5 pages

Describe the evaluation according to the research questions. Describe the process and the observed results.

6.1 What to evaluate

6.2 Assumptions, expected outcome

6.3 describe the evaluation

6.4 give the results

7 Discussion

3-5 pages

Discuss the results from the evaluation and answer the research questions.

1. How can the arc of light/fishing rod metaphor be used for intuitive interaction.
How does lag impact the system?
2. What meaningful mappings can be derived from the interaction with and the visualisation of the virtual scene.

8 Future Work

9 Conclusion

0.5 pages

10 Acknowledgements

thank Esben, thank the participants,

References

- [Cook, 2001] Cook, P. (2001). Principles for designing computer music controllers. In *Proceedings of the 2001 conference on New interfaces for musical expression*, NIME '01, pages 1–4, Singapore, Singapore. National University of Singapore.
- [Dobrian and Koppelman, 2006] Dobrian, C. and Koppelman, D. (2006). The E in NIME: musical expression with new computer interfaces. In *NIME*, pages 277–282.
- [Fels and Lyons, 2011] Fels, S. and Lyons, M. (2011). Siggraph 2011 Course Notes Advances in New Interfaces for Musical Expression. *Notes*.
- [Fitzmaurice et al., 1995] Fitzmaurice, G. W., Ishii, H., and Buxton, W. A. S. (1995). Bricks. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95*, pages 442–449, New York, New York, USA. ACM Press.
- [Gurevich et al., 2010] Gurevich, M., Stapleton, P., and Marquez-Borbon, A. (2010). Style and Constraint in Electronic Musical Instruments. In *NIME*, number Nime, pages 106–111.
- [Gurevich and Treviño, 2007] Gurevich, M. and Treviño, J. (2007). Expression and its discontents. In *Proceedings of the 7th international conference on New interfaces for musical expression - NIME '07*, page 106, New York, New York, USA. ACM Press.
- [Ishii and Ullmer, 1997] Ishii, H. and Ullmer, B. (1997). *Tangible bits*. ACM Press, New York, New York, USA.
- [Jordà et al., 2007] Jordà, S., Geiger, G., Alonso, M., and Kaltenbrunner, M. (2007). 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 - TEI '07*, page 139, New York, New York, USA. ACM Press.
- [Kiefer et al., 2008] Kiefer, C., Collins, N., and Fitzpatrick, G. (2008). Evaluating the wiimote as a musical controller. *Proceedings of the International Computer Music Conference*, pages 17–17.
- [Magnusson, 2010] Magnusson, T. (2010). Designing Constraints: Composing and Performing with Digital Musical Systems. *Computer Music Journal*, 34(4):62–73.
- [Mäki-Patola et al., 2005] Mäki-Patola, T., Laitinen, J., Kanerva, A., and Takala, T. (2005). Experiments with virtual reality instruments. *Virtual Reality*, pages 11–16.
- [Miller, 2010] Miller, J. (2010). Wiilin: a virtual instrument using the Wii remote.

- In *NIME*, number June, page 497ff.
- [Miyama, 2010] Miyama, C. (2010). Peacock : A Non-haptic 3D Performance Interface. In *NIME*, number Nime, pages 380–382.
- [Pedersen and Hornbæk, 2009] Pedersen, E. W. and Hornbæk, K. (2009). mixiTUI. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction - TEI '09*, page 223, New York, New York, USA. ACM Press.
- [Rodet et al., 2005] Rodet, X., Lambert, J.-P., Cahen, R., Gaudy, T., Guedy, F., Gosselin, F., and Mobuchon, P. (2005). Study of haptic and visual interaction for sound and music control in the phase project. *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05)*, pages 109–114.
- [Shaer, 2009] Shaer, O. (2009). Tangible User Interfaces: Past, Present, and Future Directions. *Foundations and Trends® in Human-Computer Interaction*, 3(1-2):1–137.
- [Sharlin et al., 2004] Sharlin, E., Watson, B., Kitamura, Y., Kishino, F., and Itoh, Y. (2004). On tangible user interfaces, humans and spatiality. *Personal and Ubiquitous Computing*, 8(5):338–346.
- [Ullmer and Ishii, 2000] Ullmer, B. and Ishii, H. (2000). Emerging frameworks for tangible user interfaces. *IBM Systems Journal*, 39(3):915–931.
- [Wanderley and Orio, 2002] Wanderley, M. M. and Orio, N. (2002). Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI. *Computer Music Journal*, 26(3):62–76.
- [Zappi et al., 2010] Zappi, V., Italiano, I., Brogni, A., and Caldwell, D. (2010). OSC Virtual Controller. In *NIME*, number Nime, pages 297–302.