Sputnik

Project Report HCC Project Seminar 2011

Simon Wallner*

16th 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 Prerequisitions Video CD etc...

user/performer synonymously used

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], *mixiTUI*[Pedersen and Hornbæ k, 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

[1]http://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 Hornbæ k, 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 **??** goes into detail about Sputnik, both on a conceptual and a technical level. Section 10 describes the performed user study and the paper is finally concluded in section 11 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 Hornbæ k, 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 *Wiiolin*, 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 *spatial 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 *contraints* 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 Sputnik

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

5.1 Setup

The system is set up in a room with an overhead mounted video projector. The IR-sensor bar, needed for the Wiimote controller, 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 The Virtual Scene

Figure 2 shows an overview shot of sputnik. The svirtual cene consists of the following parts:

1. The light blue and bent *arc of light* starting in the middle of the lower edge and going *into* the picture. It is directly controlled by the user and is the primary mean of interaction.

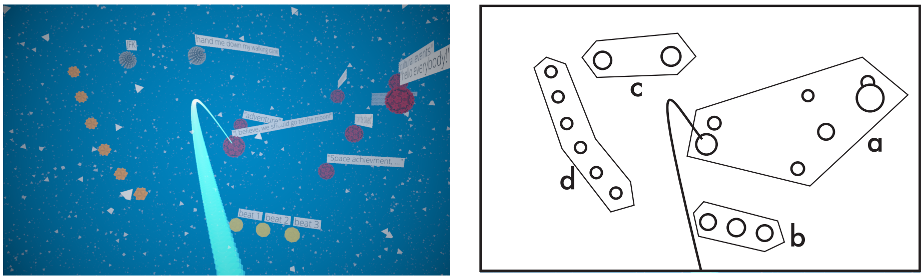

Figure 2: The objects in the scene: (a) Samplers, (b) Players, (c) Tape Machines, (d) Harmonic Harp

2. Objects the user can interact with: sampler (red), player (yellow), tape machine (grey), harmonic harp (orange)
3. Textual labels on the interactive objects. Each label contains a short text or name describing the interactive object. Figure 3 shows a few *samplers* with their labels.
4. Randomly generated star field. It is not interactive and static. It serves an aesthetic purpose as well as providing important reference points for the users orientation.
5. Coloured Fog. The fog provides depth cues for the objects

Figure 3: A group of *samplers* and the close up of a single *sampler* with their attached labels

The visual representation of Sputnik serves a dual purpose. On the one hand it is the sole graphical interface for the user and on the other hand it should be visually pleasing for the audience and allow them to understand the process of music creation in a live setting. Common music software usually focuses on the performer, leaving the visual performance in most cases to a dedicated and specialised VJ. Sputnik tries to bridge this rift by using the visual representation both as an interface to the performer and the audience.

The textual labels in Sputnik are aimed at both the performer and the audience. The text on the labels is hard coded and does not change during run time. It can be used to convey additional information to the performer about a certain object, but it can also allow the audience to gain more information about the piece. The function of the objects can become clear even if the object is not currently in use. This can also introduce an element of anticipation when an object with a certain label is visible but the performer does not yet interact with it.

5.3 Navigation and Camera Controls

Sputnik is controlled from a first person perspective. The user can navigate 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 to the 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 4 illustrates the navigation.

Sputnik's camera uses a fixed *up direction*. The fixed up direction is commonly seen in cinema and video games, even though it is not intrinsic in the space inspired setting. It furthermore should limit the camera's degree of freedom and make it more accessible.

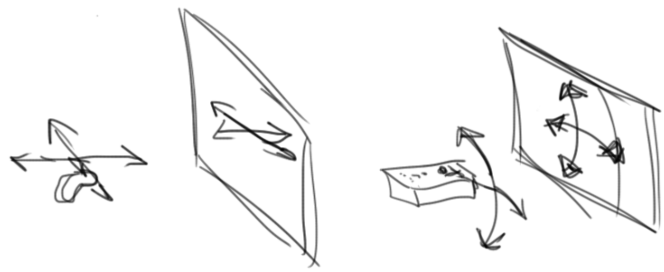

Figure 4: The Nunchuk's analogue stick controls the camera's dolly and track movements, the Wiimote's IR pointer controls the camera's tilt and pan movements.

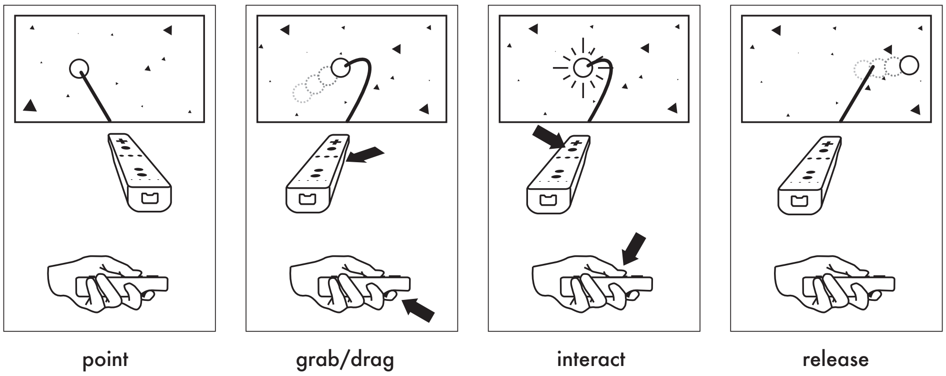

Figure 5: The basic interaction vocabulary of Sputnik

5.4 Interaction

Figure 5 illustrates the basic interaction vocabulary of Sputnik. The user can 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* the user can *point* at objects, *grab* them and also *drag* them around the scene.

Interactive objects behave in a simplified 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 interactive object in the scene can react individually to user interaction. The following classes of objects currently exist in Sputnik:

Sampler (red) The *sampler* object reacts to the press of the *A button* while it is grabbed. While the button is held a preloaded 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) The *player* object reacts to the press of the *A button* while it is grabbed. Pressing the *A button* starts and stops the playback of a preloaded sample. It is not automatically stopped when it is released. Playback is looped and always starts at the beginning.

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

Harmonic Harp (orange) The orange spheres form a kind of *harmonic harp*. Each sphere controls a single sine wave oscillator and the volume is determined by the

the sphere's distance to its origin. Additionally the spheres are pulled towards their respective origins by a constant force that is linear in the distance from the sphere to its origin. The oscillators are tuned to the natural harmonic series which makes them easy to play with and avoids dissonances.

Each interactive object is preloaded with a different sample or sound in an external application. By interacting with the different objects and arranging them in the virtual scene, users can create musical performances that are both musically and visually expressive.

5.5 Simplified Physics

Sputnik uses simplified physics simulations to give the interactive objects their distinct physical properties. An object's properties are described by *mass*, a *movement vector*, as well as a movement *damping constant*. SI units are used unless otherwise noted.

The dragging force of the arc on an object is linear in the distance from the object to the intersection of the unbent arc with the plane that goes through the object, with a normal pointing at the camera. This intersection point also defines the direction of the force vector.

In each frame, the speed of an object is then computed as

$$v_i = v_{i-1} + \frac{F * \Delta t}{m} \tag{1}$$

where v_i is the object's movement vector in frame i, F is the current force, m is the mass and Δt is the frame time. Force is only applied at the center of the objects, and thus never causes a rotation of the object.

Damping of the object's movement is implemented by a simple exponential decay, given as

$$v_i = v_{i-1} * e^{-\lambda * \Delta t} \tag{2}$$

where λ is the damping constant.

5.6 Creating sound

Sputnik's interactive objects individually react to the users input and use the MIDI protocol to communicate with an external application. Sputnik itself does not create any sounds. Sound creation is handled entirely by an external application.

Thanks to the simplicity and the pervasiveness of this protocol, virtually every music software can be used together with Sputnik. In the current set up *pure data (pd)*[4] is used for sound creation. A simple patch is used that builds on *boctok-1*, a small collection of pd patches written by the author prior to this project.

[4] `http://puredata.info/`

Communication is only one way; from Sputnik to the external application. The MIDI channel and controller number every object uses is currently hard coded into Sputnik an can only be changed in the source code.

6 Implementation

Sputnik is implemented in C++ with Mac OS X as its development platform. The project is split into two sub projects: The special purpose Sputnik and the more general purpose *kocmoc-core*. Kocmoc-core provides core services that are then used by sputnik to build the final system. Some code of kocmoc-core existed prior to this project, but most of it was created in the course of the project.

The source code of both Sputnik and kocmoc-core is licensed under the permissive MIT open source license and is publicly available on github[5] [6].

Sputnik uses OpenGL 2.1 with a few extensions to display the virtual scene. The renderer is fairly simple and displays unlit, textured objects with baked ambient occlusion maps. The computation of the homogeneous fog is realised in the vertex and fragment shader and a post processing effect is applied. This effect adds a barrel distortion to compensate the wide field of view and vignetting. A simple form of full screen anti aliasing is also added in this stage.

The *Assimp*[7] and *devIL*[8] libraries are used to load assets. Font rendering is implemented using the *freetype*[9] library.

RtMidi[10] is used to send MIDI messages to external applications and *WiiC*[11] was used to interface with the Wiimote and Nunchuck controller.

Figure 6 illustrates the data flow in the main run-loop. First, the devices are polled, which causes input callbacks to be fired. The components are subsequently updated and finally the scene is sent to the renderer.

7 Wiimote Input

The Wiimote controller was chosen for its pointing functionality as well as the motion sensing capabilities, even though they haven't been used in the final project. The accompanying Nunchuck controller was also used to allow the standard analogue stick driven navigation.

[5] `https://github.com/SimonWallner/sputnik`
[6] `https://github.com/SimonWallner/kocmoc-core`
[7] `http://assimp.sourceforge.net/`
[8] `http://openil.sourceforge.net/`
[9] `http://www.freetype.org/`
[10] `http://www.music.mcgill.ca/ gary/rtmidi/`
[11] `http://wiic.sourceforge.net/`

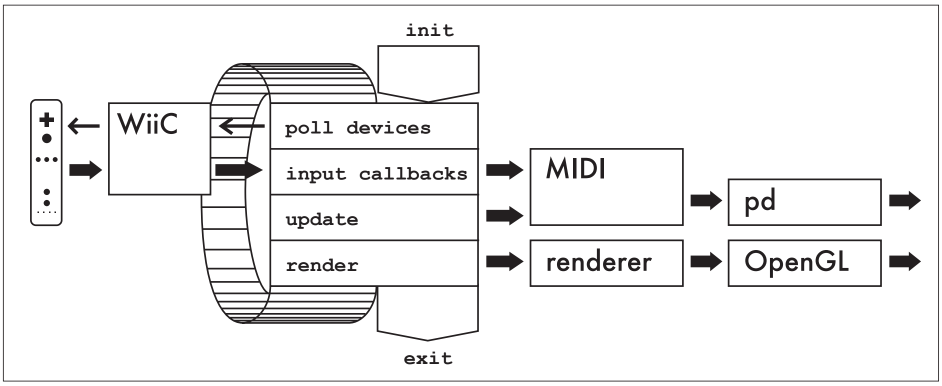

Figure 6: The data flow in the run-loop.

The Wiimote has an built in low resolution IR camera and its feed is directly processed on chip. The feed itself is not accessible but data of up to 4 tracked IR points can be read. the Wiimote's update rate is reported to be $100Hz$ which gives an lower bound for the worst case lag of $10ms$. The used Wiimote library takes care of most of the processing and conveniently returns the computed pointer location in relative screen coordinates.

A drastic constraint of the Wiimote's IR pointer however is the relatively narrow field of view. It is very easy to leave the small sensing area. During the development it turned out that using the Wiimote for the tilt/pan movements of the camera can alleviate this problem a little. Users seem to intuitively try to compensate the camera's rotation, thus maintaining a focus on the neutral area in the center of the screen. Without it users easily lost the focus an had troubles finding *back onto the screen*.

The used Wiimote library supports more than one Wiimote and the input system in Sputnik would also support multiple input devices at the same time. The option to *dual wield* two Wiimotes and to interact with two arcs of light simultaneously was given up in favour of the more standard analogue stick track/dolly controls, that allow the users to easily navigate the scene.

8 The Arc Of Light

The arc of light forms the foundation of Sputnik and is its core contribution. It acts as a bodily extension of the users body into the virtual scene. Through the arc the user can use the interaction vocabulary described above. It has the following behaviour.

If no object is grabbed, the arc follows the user's input directly. This input is not filtered and is used directly *as is*. This introduces a slight jitter but on the other hand does not take away from the system's responsiveness.

Low pass filtering however is desired in most scenarios and is indirectly implemented via the physical properties of the interactive objects. The higher the weight of the object for instance, the stronger the filtering is. while grabbing an object the arc bends accordingly to the the physical properties of the object and the dragging force of the user. It bends like a fishing rod, a natural metaphor that seems to readily understandable to the users.

One of the core advantages of the arc is that even very strong filtering can be achieved without making the system feel laggy and slow. It happens in a way that is transparent and understandable for the user, and in a informal user study the testers reported that the system is very responsive.

9 Mapping the System

What further distinguishes Sputnik from other projects or interfaces, and a commonality it shares with some tangible interfaces, is the spatial component. Users can move the interactive objects in 3D space in a meaningful way.

The spatial arrangement of objects can be reconfigured at run time. This allows the performer to create new configuration during the performance.

For the objects that react to their position of movement the spatiality is mapped directly to the sound creating application. The *tape machine* for instance maps its movement speed to the playback speed of the sample and the *harmonic harp* maps the distance of its spheres to their respective origins to the volume of each oscillator. These are mappings that cannot be achieved by a conventional hardware interface. The informal user study also hinted, that these mappings where the most interesting and novel ones.

10 Evaluation

10.1 Method

Sputnik was evaluated in a small and informal user study with only three participants. The testers where asked to use Sputnik and a small questionnaire and interview where held afterwards. Sputnik was tested with only one person at a time.

Due to the very small number of participants this study can only be seen as a precursor to fully fledged user study. A relatively wide range of questions and topics was used in the evaluation to gain a broad overview about how Sputnik is used, what general problems arise, and how suitable it is as an interface for musical expression.

The whole test was recorded on video and the interview afterwards was audio recorded for later analysis.

10.2 Participants

Only 3 participants (all male) took part in the test, with an average age of 26 (SD = 5.2). All had a musical background having played classical instruments but not on a professional level. Two had additional experience with computer and sample based music.

None of the participants had used a Wiimote prior to the study, all where right handed and colour normal.

Two participants have seen short explanatory demo videos of Sputnik that where used in order to find participants.

10.3 Set Up

The test was conducted in a darkened room with the participant standing about 4 meters away from the overhead projected screen with a diagonal of about 2 meters. The screen had an aspect ratio of about 16/9 and the screen's center was about 1.5m above ground.

The Wiimote sensor bar was positioned a few centimetres below the lower edge of the picture. A small stereo speaker set was used that was located on the floor right behind the participant.

10.4 Procedure

At the beginning of the test, participants where asked for basic statistical data. The testers where informed about the test plan and the controls where briefly explained.

The first task was to try out the navigation and familiarise with the controls. This included basic navigation and orientation as well as grabbing and interacting with objects.

In the second stage, participants where asked to describe for every class of interactive objects, what they do and how they react to user interaction.

The third stage consisted of a brief impromptu musical performance. Time was given to plan and structure it in advance and participants where asked to start when they felt ready. The length of the performance was not specified, only a vague rule of *not too short, not too long* was given.

The test was concluded with a brief ordering task. The participants where presented with four objects and where asked to bring them in order according to their weight. Those objects did not create sounds.

After the practical test, a one-page questionnaire consisting of 4 5-point Likert-items and 7 questions inspired by QUIS[Chin et al., 1988]. A short interview guided by a set of pre-written questions was conducted afterwards.

The questionnaire and interview form can be found in the appendix. **?? ??**

10.5 Design

The answers to the Likert items are discussed individually and mean and standard deviation values written as mean/SD are given for the QUIS section. Since linearity of the scale in the QUIS items cannot be assumed, calculating mean and standard deviation is not correct. Also the very low number of samples speaks clearly against it, however it is a convenient way to look at the data.

10.6 Results

Overall reactions to Sputnik were mildly positive which is hinted in the results of the QUIS inspired questionnaire seen in figure 7. Due to the small number of participants no direct and strong conclusions can be drawn from this evaluation and it should rather be taken as a hint on what Sputnik does good, and where there is more room for improvement.

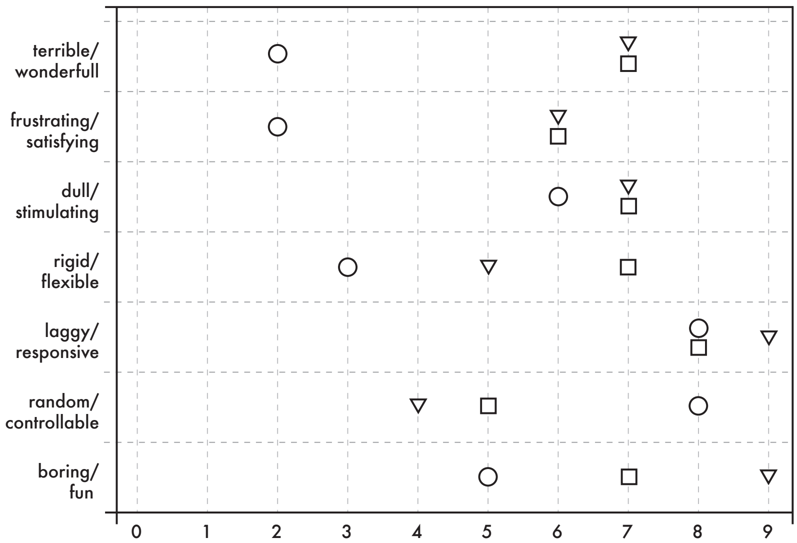

Figure 7: QUIS results...

Unfortunately, the results of the Likert items where inconclusive: The first question, "Sputnik is easy to use" resulted in the answer triple (strongly disagree, neutral, agree) and all other items resulted in (disagree, neutral, agree).

10.6.1 Controls and Navigation

Even though none of the participants had prior experience with the Wiimote, all managed to use Sputnik effectively during the test. After only a few minutes the participants where able to navigate the scene and interact with objects.

Problems however were the narrow field of view of the Wiimote resulting in uncontrollable camera movement until the Wiimote is brought back into the range of the sensor bar. During the test this happened about 3-5 times per participant.

Another problem was missing precision from the pointer. All participants had at least once problems to accurately grab an object. This could in parts be attributed to the slight jitter of the pointer.

The camera was often held static and the participants where then using the scene from the current position until the position became *unusable* and the camera was moved. Phases of camera movement and interaction with the pointer where often clearly separated and camera movement was often rather erratic.

–¿ confirm with Video!!!

10.6.2 Visual

All participants where missing feedback of the interactive objects regarding their musical state, e.g. a way to see if a player object was playing or not. One participants mentioned explicitly that the visual representation has no meaning stating that: "[...] objects and their forms, and so on..., didn't make any sense". Answering another question he also mentioned that it would be more interesting if also the star field was interactive and react to the music performed .

During the improvisational performance the participants seemed to ignore the textual labels of the objects. This could be in connection to the camera navigation, since the camera was often in a position where the labels where just not readable.

10.6.3 Interaction

All participants could deduce the functions of the sampler and player objects, and descriptions of the tape machine and harmonic harp were not as precise but captured their general mode of operation.

Participants seemed to have no problems using the arc and all participants responded positively when being asked about immersion into the scene. One participant even said at the end that: "I liked the link from me to the objects,.. when you hold the objects. It made sense".

A point that was noticed and mentioned by all participants was the arbitrary distance limit of the arc. Objects could only be grabbed if they where outside a certain range. Another thing mentioned by all participants was the urge to pull objects closer to the camera, a feature that does not exist in Sputnik. Right now, objects can only be dragged around the camera, not closer to it.

When exploring the system, all participants actively moved around the objects, even those that do not react to their position or movement speed. This behaviour continued even after the point where they should have realised that it has no effect on the object. It seems that dragging is a natural affordance of the system. Two of the 3 participants also tried to waggle the Wiimote, a gesture which is not directly used in Sputnik.

In regards to the samplers all three participants indicated the need for a way to quickly and directly access some of the objects to either have certain sounds available all the time or to quickly change between sounds.

10.6.4 Improvisational Performance

All participants used a similar pattern for their improvisation, alternating between the different groups of objects. The participant without prior experience with computer and sample based music had trouble to put together a coherent performance and seemingly disliked the task

The two users with experience in computer and sampler base music, arranged the objects before the performance and seemed to *plan ahead*. A procedure that is probably similar to how they would approach such a task with their usual set up and tools.

Likeability of objects seems to correlate with the amount of interactivity they provide and how it is used in the system. The tape machine and harmonic harp got generally better comments than the sampler and player object. Two participants said, that they disliked the player object for being uninteresting and lacking interaction.

10.6.5 Object Sorting

Two participants sorted the objects correctly, one brought the lightest two in the wrong order. All participants used a continuous grab to assess the weight of the objects. During the development another user tried to *throw* the objects to observe their weight, but this behaviour was not observed in this study.

10.6.6 Misc.

All participants expressed the need to exchange the sounds and fit Sputnik to their needs in order to be useful in a live setting.

One participant reported fatigue in his shoulders, ans aspect of Sputnik that was not accounted for in the study, but is probably an important factor in real world usage scenarios

11 Discussion

3-5 pages

Discuss the results form 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.

12 Future Work

13 Conclusion

0.5 pages

14 Acknowledgements

thank Esben, thank the participants,

References

[Chin et al., 1988] Chin, J. P., Diehl, V. A., and Norman, L. K. (1988). *Development of an instrument measuring user satisfaction of the human-computer interface.* ACM Press, New York, New York, USA.

[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). Wiiolin: 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 Hornbæ k, 2009] Pedersen, E. W. and Hornbæ k, 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.