Noisock: A musical playful experience

Sergio Manuel Galán Nieto, Matthew Hennessey

Malmö Hogskola- Konst Kultur och Kommunikation (K3), Malmö, Sweden

Intro

The initial development of the *Noisock* concept began with the element of play inside a physical computing project at the Interaction Design Master Course at Malmö University. The task was "Soundify a common household object and control it with an Arduino in one week". From this first basic task to the final *Noisock* concept we followed a design path that included brainstorming, user testing, bodystorming and, of course, fast-prototyping in order to design a physical and playful music experience. During the process we also discovered the importance of Creative Commons content and decided to use *Noisock* as a tool to put Creative Commons musical samples in the hands of non-musicians.

Figure 1 9LyrDip

First prototype: *9LyrDip*

Our first concept was to turn the analog exercise of drumming with utensils on the dining table and its objects, making them digital. How to make multiple sounds out of two utensils and a metallic bowl? The result is shown on [Figure 1] and is slightly different: The player sits with the bowl on his legs. To play a sound he must wear a pair of gloves and touch the bowl with a finger. Eight of the fingers produce a beat while the other two, turn on/off the recording mode or remove the last beat. In this recording mode, each sound that is played is recorded and displayed in a music sequencer on the computer screen.

Bodystorming process

After seeing interest from teachers and classmates in the *9LyrDip* prototype, we decided to keep working and to move forward with the concept. To do this, we try an approach to bodystorming described in [1]. We found that bodystorming is a tool that needs external factors like a comfortable atmosphere and a very good mood. So we went home to play, grabbed some beers, turned up the beats and we began to imagine physically how to create the music with other objects and body movements. We took notes from this process and used them to create the concept of *Noisock*.
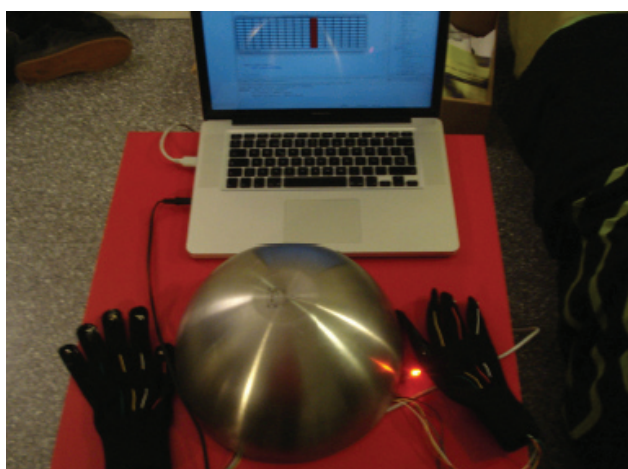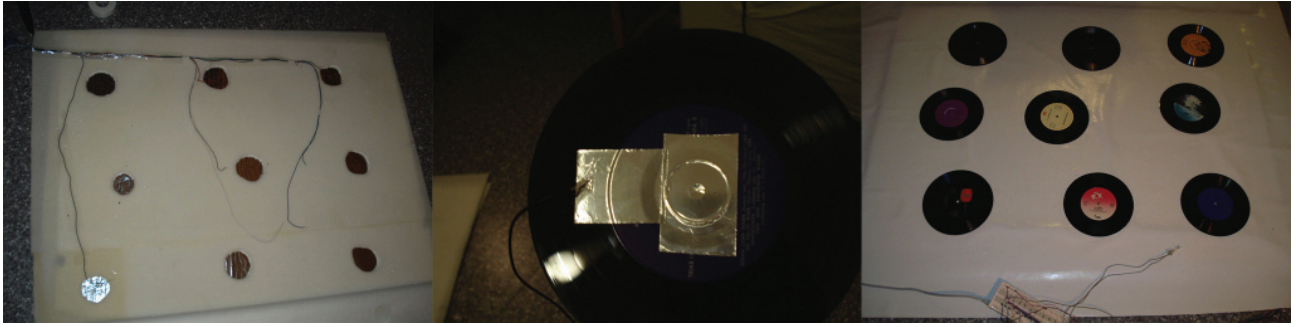
Figure 2 : Building the Noisock *mat*

Noisock : Concept and implementation

To define the concept we thought about:

☞ Scenario: *Noisock* can be used in the same context as games like Guitar Hero or SingStar: It can be played as a "living room-game", but works better in social events with more people around to interact and even play in pairs.

☞ Music games (i.e. Guitar Hero) are not really based on creating music. They focus on the users hand eye coordination. Dance Dance Revolution, is a coordination game, as well. To play with *Noisock* the main sense is hearing, aided by vision.

☞ *Noisock* should create a full physical engagement between sound and body movements.

In *Noisock* we replaced the bowl for a big mat with different sounds at each touch point. Stepping on a different touch point plays that specific sound. To control the sounds we wanted to use arm movements. In the body storming session we came up with these gestures/controls.

☞ Record: Hi five with another player or clap the hands.

☞ Volume: Arm movement up and down

☞ Undo last action in the sequencer: Single arm movement from left to right.

☞ Change the sound set: Both arms moving at the same time from right to left.

☞ To implement this concept we created a cheap electronic mat with 9 contacts covered by a vinyl LP [Figure 2] based on a instructable [3]. When stepping on the LP the sample is played in the same way that it was played with the *9LyrDip*. Using a Nintendo Wiimote to incorporate some of the distinct gestures and replace others with

button clicks helped us to avoid the complex problem of capturing gestures. For instance, recording is not a clapping gesture but an action controlled by the A button [Figure 3].

☞ The hardware used is an Arduino [6] board while the software was Processing [7] at the beginning, and OpenFrameworks [8] currently. To read the information from the Wiimote we use Darwiinosc. [9] In different incremental iterations we added more functions to the software, now *Noisock* has the following features:

☞ Rhythm tune: If the player is not in sync with the rhythm, *Noisock* automatically adjusts the sample that is being played.
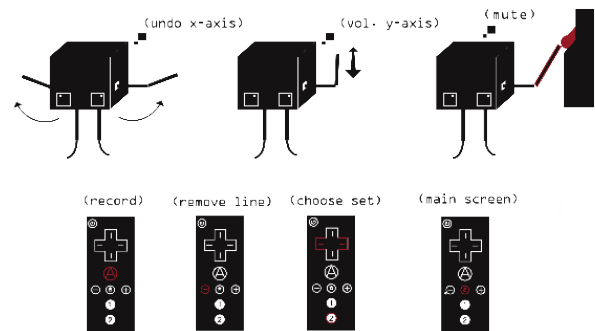

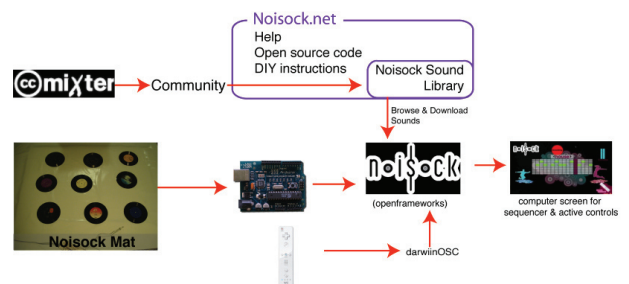
Figure 3 : Some of the Noisock *controls*

Figure 4 : Elements in the Noisock *system*

24

☞ Graphical interface based on a step sequencer: A 16 × 9 cell grid where rows represent each sound and columns represent the time that the sound will be played. A line iterate through the columns to show the current loop position. When the recording mode is activated every time that a new sound is added the cell representing the sound in a particular moment changes colour. This is a very extended interface in the music software and we found that is also easy to learn for non musicians.

☞ Song-like output: Enables inexperienced players, after recording a few sounds in the sequencer, to obtain a pleasant sounding loop.

☞ Easy to start: it doesn't have the expressive power of a musical instrument but it is easier to learn.

☞ Playing not gaming: *Noisock* isn't governed by goals or rules. The user value is about play and enjoyment. [10]

☞ Different sound sets at the same time with mute functions: Increases user's ability to mix sound sets and create a kind of DJ experience.

☞ Download new sound sets from the Internet.

From technical constraints to new ideas: Examples during the whole process

During this project we learned the importance of tinkering and how learning with technical constraints can move forward the concept. Here we present two examples. The first one, is related with *9LyrDip*. As explained before, we wanted to make different sounds hitting different parts of a bowl, but finding no solution in a few hours of research lead us to move on. Leaving behind the idea of "soldering the different sounds" to the bowl, to keeping the bowl as a ground and "soldering the sounds to the fingertips of gloves". We found that, because of wearing the gloves, the immersion was higher and the link between body movement and sound was stronger.

The second example happened when developing *Noisock*. Our first thought was to deconstruct famous songs and spread their samples across the mat. The goal of the game would be: "Piece together the simples and try to build the song as close as possible to its original form."

Fortunately, this is not an easy task. CD tracks are flattened compositions of all the instruments together making it impossible to extract a single instrument. When we attempt to build a song cutting it into pieces the result was bad and the process painful. Needing studio tracks (one instrument per track) we turned to the Internet, finding that there are some artists that not only provide the final song, but also the studio tracks and they use a site called ccmixers.org, a community remix site filled with amateur and professional samples for users.

Thanks to the technical constraints we moved forward and created a much more powerful concept: Not only an average game, but a play experience that includes the ideological implications of using Creative Commons content.

Creative Commons and ccmixers.org

In the current copyright war, the debate lingers on, pay for content/don't pay for content, piracy... Beyond this, the free culture movement wants to warranty rights to creators and users. As Lessing [2] explains:

We live in a "cut and paste" culture enabled by technology [...]Using the Internet and its archives, musicians are able to string together mixes of sound never before imagined; filmmakers are able to build movies out of clips on computers around the world. [...] All of these creations are technically illegal. Even if the creators wanted to be "legal," the cost of complying with the law is impossibly high. Therefore, for the law-abiding sorts, a wealth of creativity is never made. And for that part that is made, if it doesn't follow the clearance rules, it doesn't get released.

With no change to the copyright laws, Creative Commons offers the only solution to creators, to allow other creators to take their work and "sample it" to build something new. ccmixers.org is a Creative Commons community focused on music. It's possible to find different kinds of musical content there:

☞ Professional artists distribute their songs' samples and studio tracks- the source code of the music- to allow others to remix them.

☞ Amateur musicians release their sounds and samples to allow other musicians to build their songs.

In the free software community the technical material, source code, is available. This is a useful tool for developers that could be perceived as useless

25

for users, usually only concerned about money and functionality. It's the same with cultural products: end-users are rarely concerned with the value of freedom to artist material. That's why *Noisock* wants to contribute to spread the importance of free content to users. *Noisock* takes sounds from ccmixers.org to build the sound sets. It takes the free musical content that is mainly used by professional/amateur musicians and puts this content into a mainstream-like electronic game.

The future of *Noisock*

We tested *Noisock* in one of its potential scenarios: A dinner party. We observed that all who played with it, found it funny, even with the current sound sets and limitations. To move forward with *Noisock* we thought ways to spread it. *Noisock*

is hardware+software so in order to distribute, releasing the software on the Internet is not enough.

In recent years a digital "Do It Yourself" (DIY) culture has been growing on the net. Webs like Make Magazine [4] and Instructables [3] have big communities that create new content daily for tinkerers [5]. We don't have the infrastructure to create *Noisock* mats and send them everywhere, thus we are working on a community web to allow people to build their own version. On *www.noisock.net* we want to provide tutorials and tools for:

☞ Users: Those who want to build the mat and run the software.

☞ Content creators: Those who want to create sound sets. The better the sound sets, the better the *Noisock* experience.

☞ Developers: Those who want to download the source code to make their own versions
With these elements we want to build a community where all contribute to design the game that, on the other hand, still needs to be developed in order to be a finished product. Two main redesigns are needed:

☞ The sequencer is useful to build the songs but in the tests we did, we found that players spend too much time looking at the screen and the physical engagement that we want to build gets broken.

☞ Software and hardware are not eye-catching at all. To achieve a successful product the aesthetic experience should be complete.

Conclusions

When we started the project, the brainstorming session didn't work, but anyway we took the less bad idea for us, began to create something, and step by step, because of technical constraints or just because new ideas came up while we worked, we moved far beyond the first concept. Here we realize that creativity doesn't usally come from post-it collections, but it emerges while you work.

Now *Noisock* is a working prototype that we have tested and we realize that it can have a path beyond the course assignment. We're now working on new concepts to make it better, well-known and available to anyone.

Acknowledgments

26

References

1. BUXTON, W. (2007). *Sketching user experiences: getting the design right and the right design.* Amsterdam. Elsevier/Morgan Kaufmann.

2. LESSING L. (2004). *Free culture.* http://www.free-culture.cc/

3. *Instructables.* http://www.instructables.com/id/Interactive-Twister-Game/step1/Materials/ (Accessed in Jan. 2010)

4. *Make Magazine.* http://makezine.com (Accessed in Jan. 2010)

5. LAHART, J. (2009) *Tinkering Makes Comeback Amid Crisis.* http://online.wsj.com/article_email/SB125798004542744219-lMyQjAxMDI5NTM3MDkzODAwWj.html (Accessed in Jan. 2010)

6. *Arduino.* http//Arduino.cc (Accessed in Jan. 2010)

7. *Processing.* http//processing.org (Accessed in Jan. 2010)

8. *OpenFrameworks.* http://www.openframeworks.cc/ (Accessed in Jan. 2010)

9. *Darwiinosc* http://code.google.com/p/darwiinosc/ (Accessed in Jan. 2010)

10. ZIMMERMAN E. (2002) *Gaming Literacy.*