

Marije A. J. Baalman

Coding in the context of collaborative, interactive performance

Introduction

Most of my current work is in the context of creating interactive performances or installations in collaboration with several other artists. This work, besides creating the artistic works, involves the development of tools to make the creation of such work easier, so that more time can be spent on experimenting with the technology to try out different artistic concepts, rather than trying to make the technology work. Between 2007 and 2010, I worked on the research-creation project *Sense/Stage* at Concordia University in Montréal, Canada, with Chris Salter and several other collaborators.¹ The research and development in this project was done while simultaneously working on a number of artistic projects, such as the dance performance *Chronotopia* with the Attakkalari Centre for Movement in Bangalore, India, as well as the installation *Just Noticeable Difference*, a collaboration of Chris Salter, Harry Smoak and myself. Since 2010, I have also been involved in a loose collaboration project called *Modality*,² which aims to develop tools for the audio programming language *SuperCollider*,³ to create interactive musical performances. This collaboration brings together several *SuperCollider* developers to combine their efforts to create flexible tools to make music.

In this short text I want to highlight some of the issues that occur in working in the context of collaborative, interactive performance or installations. For readers interested in more specific details of some of this work, particularly around using the programming language *SuperCollider*, I refer to (Baalman, 2010).⁴

Context – collaborative and interactive

Before continuing, I would first like to highlight what I mean with collaborative and interactive. The artistic projects I have been involved in, usually involve several people working within the same space (the performance space, or the location where the installation is placed) with different media, such as sound, video, light, and mechatronics, but also dance, theatre or visual arts. For those media which are controlled from computers, there is usually the need for some way of sharing data between different computers and software environments. For the whole project, everyone involved usually (or should) takes part in discussions as the use of interactive elements not only affects the technical realisation, but also the artistic concept and way of thinking. In my opinion, an interactive dance performance cannot be truly interactive if the introduction of interactive technology does not affect the choreography of the dance, e.g. by introducing elements of improvisation, where the performer has to react on the impulses that are given back to the environment through the technology. The technology in turn reacts on changes in the environment by sensing these changes with sensors, and after some manipulation and processing of the data coming from these sensors, drives or manipulates for example the sound or light in the performance.

With interaction in this context I mean the interactions between machine and humans, machine and environment, and vice versa; these are realised technologically with the aid of sensors and data processing algorithms. This technology aided interaction comes in addition to interactions between performers themselves, or performers and stage set, etc.

My main interest in this field is to find out how such interaction can give us experiences which would not be possible to realise using traditional methods of scripting, composing or choreographing media. The collaborative context is important since an experience

is not tied to one medium alone, and the interplay between different media enriches the experience. Also each medium has its own “sense of time” dependent on our perception of different media and their interplay (crossmodal perception).

The role of code

The role of code in this context is both a functional and expressive one. On one hand the code will enable us to deal with data and create output. Parts of the code will purely aid us in getting data from one place to another, e.g. from the sensing device to software environments and then out to output media again, thus interfacing with the hardware. However part of the code will be the translation of how we want the sensor data to affect the environment. This translation consists of conditioning the data, mapping it from one domain to another, applying different algorithms to work with the dynamics found in the input data, which can then generate or drive dynamics in output media. While there are various algorithms described in the literature, and available in certain software environments, the choice which ones to use (or to develop your own), or how to combine them, is a personal choice dependent on the kind of interactions that you are looking for within the project. As such the code itself becomes an expression of the creative ideas. In the process of creating the work, you are sculpting the code until it generates the kind of behaviour that you think is interesting.

Creative problems and solutions

Coding in a professional performance context has different demands than “product oriented coding”, in the sense that while writing the code, the purpose of the code and its needed functionality is not yet known, but will emerge during the artistic process of discussions, experimentation and rehearsals. This is especially true, when the artistic project involves real-time sensing, where it is not known beforehand what the input data will be, and how it will influence the output media, which are also being shaped in the process of creation.



Chronotopia: Attakkalari Centre for Movement, interactive scenography by Chris Salter and Marije Baalman (Februar 2009). Photo by Anke Burger (rehearsals 2008).

Within the rehearsal process for such projects it is important to have a flexible system which allows for on-the-fly manipulation of audio synthesis processes as well as sensor data mappings. Part of the preparation for the rehearsal process is to create systems that allow for such flexibility, so that many different kinds of interactions can be explored. This is only possible if there is some idea in advance what kind of possibilities there are, i.e. what kind of data is to be expected from the sensors, the type of audio processes that will be used (its compositional structure, as well as its sonic quality), and what kind of interactions the collaborators in the project are interested in. Extensive discussions about this with the other collaborators, as well as short exploratory sessions with the performers, and a basic understanding of some of the movement material of the dancers (so that you can e.g. wear an accelerometer and produce some data yourself while writing and testing code) are essential components in this process. Having some skill at livecoding to quickly develop new interactive processes is also vital for a successful rehearsal process.

Show control, show time and rehearsals

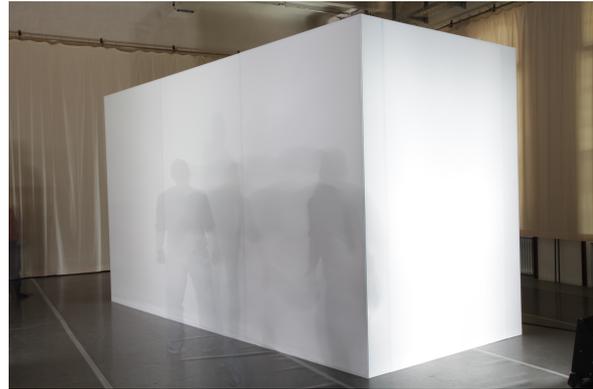
For the eventual showtime in the theater or at an exhibition, it is important to have a robust “show control” system⁵ from which the show can be run, while at the same time being flexible to adapt to differences in set-

up (e.g. audio balance/mix), based on the venue in which the performance takes place. Ideally, you should be able to adapt “cues” during the show, should there be the need. Backup solutions, in case sensing infrastructure breaks down, can also be useful (even just as a reassurance).

In the case of installations, the code (and the machine it runs on) may need to be prepared to be started and stopped by gallery personnel who have no knowledge at all about coding, and in some cases even of computer environments. In the ideal case the machine running the code can boot up and start the code automatically, so the computer only needs to be turned on.

The design of a show control system for interactive, collaborative work is far from straightforward. While traditional show control systems assume a linearly progressing timeline, for interactive works timelines may not be linear, but instead depend on improvisational elements or dynamics within the sensor data. Cues (a specific event or start of an event) can be set in time (absolute with regard to the start time of the work, or relative to the start time of a section, or another cue), or be manually triggered (e.g. by the artist controlling the computer), or be triggered based upon certain conditions that are met by the data resulting from sensors (or otherwise). Additional complexity comes in for synchronising the different media and the software environments driving them; will one software environment be the master of the show control and others follow, or is there a degree of independence? Where is the logic for the behaviour over time placed? How do you deal with these issues in the creation process? How will a change of the structure in one software environment affect the other media? How do you rehearse with such a system? The algorithms for processing the sensor data may need a specific history of the data, as it has evolved over time in the previous part of the piece. How do you deal with stopping and starting processes?

In my recent works I have experimented with different approaches to deal with this problem, and these approaches are now slowly merging together. The development of the tool goes hand in hand with the creation of the work, that is to say, using the tool.



Exegesis – *Just Noticeable Difference*, Chris Salter in collaboration with Marije Baalman and Harry Smoak. Pakt Zollverein, ISEA 2010, Essen. Photo by Thomas Spier.

Adhoc solutions, reusable tools and maintenance

While working on artistic projects there is always a trade-off between developing “general-purpose” tools that are robust and flexible in use, and quickly putting something together, that is usable and reliable enough for the project at hand, but may not translate well to other projects. However, going into the next project, some of the ad-hoc tools may be translated to general-purpose tools for future projects. But how do you then maintain this? If you perform or show the old work again, do you spend the time to update it to use the (possibly improved) general-purpose tool, or use it in its old state? What if you want to make changes to it, and you know that it will be easier with the more general tools? In how far are tools personal to the artist or the work, or can other artists use them? Can you give the tools into the hands of a technician to go on tour with the piece, being able to troubleshoot problems that may occur in different locations?

Conclusion

I have tried to address some of the issues that come up in the context of coding for collaborative, interactive performance and installation projects. Quite a few of the issues are topics for further discussions and elaborations, and I've only given brief personal viewpoints on these issues here. As my work continues in developing and using software infrastructures for interactive

projects I will be addressing these issues, and hopefully come up with solutions that work not only for one specific project, but that can be used as tools to create multiple projects, and be used by artists other than me. The "Sense/Stage"-infrastructure already addresses part of these issues and I hope that other artists will enjoy using this to create their own works.

Endnotes

1. See <http://sensestage.hexagramca> for an overview of the outcome of this research project, as well as a full list of collaborators in the project. Accessed on April 14, 2011.
2. See <http://modality.bek.no>. Accessed on April 14, 2011.
3. See <http://supercollider.sourceforge.net>. Accessed on April 14, 2011.
4. Marije A.J. Baalman. *5 years of using SuperCollider in real-time interactive performances and installations – retrospective analysis of Schwelle, Chronotopia and Semblance*. In *Linux Audio Conference (LAC) 2010*, Hogeschool voor de Kunsten, Utrecht, The Netherlands, May 1-4, 2010.
5. In theater/performance the collective control for all events supporting the performer's action on stage (i.e. sonic, light, mechatronics, video) happening on stage is usually referred to as "show control".

Abstract

The context of collaborative, interactive performance and installation is a challenging environment for developing technology that will enable creating such artistic works. Code becomes a medium in which artists express themselves and the challenges of writing code in a context where problems are created as they are being solved demand flexibility and realtime control over the code that defines the interaction. In particular, show control systems that can deal with interactive and improvisational elements are a topic for further investigation. This article reflects on various issues that come up in this context.

Author

Marije Baalman studied various things at various universities in different countries and at different continents. In her artistic work she is interested in the real-time components of the work, in that nothing is pre-composed as such, but rather the (mostly, but not exclusively) sonic output depends on realtime interactions, be it of the performer, or of the audience. Thus composition becomes more the composing of behaviours and interaction modalities, creating processes, rather than fixed sound tracks. This is expressed with tools such as physical computing (performance interfaces and/or installations), livecoding (both as a skill, as well as a performance interface), digital and analog sound processing, and improvisation.

She has collaborated with various people, amongst which Alberto de Campo, Chris Salter, Michael Schumacher, Attakkalari Dance Company and Workspace Unlimited.

<http://www.nescivi.nl>.

Title

Marije Baalman, *Coding in the context of collaborative, interactive performance*, in: *kunsttexte.de*, Nr. 2, 2011 (4 pages), www.kunsttexte.de.