

Prototyping Audiovisual Performance Tools: A Hackathon Approach

Nuno N. Correia
Department of Computing
Goldsmiths, University of London
SE14 6NW London UK
n.correia@gold.ac.uk

Atau Tanaka
Department of Computing
Goldsmiths, University of London
SE14 6NW London UK
a.tanaka@gold.ac.uk

ABSTRACT

We present a user-centered approach for prototyping tools for performance with procedural sound and graphics, based on a hackathon. We also present the resulting prototypes. These prototypes respond to a challenge originating from earlier stages of the research: to combine ease-of-use with expressiveness and visibility of interaction in tools for audiovisual performance. We aimed to convert sketches, resulting from an earlier brainstorming session, into functional prototypes in a short period of time. The outcomes include open-source software base released online. The conclusions reflect on the methodology adopted and the effectiveness of the prototypes.

Author Keywords

Audiovisuals, performance tools, user-centered design, hackathon, expressive interaction, human-computer interaction

ACM Classification

H.5.1 [Information Interfaces and Presentation] Multimedia Information Systems—Animations, H.5.2 [Information Interfaces and Presentation] User Interfaces—User-centered design, H.5.5 [Information Interfaces and Presentation] Sound and Music Computing.

1. INTRODUCTION

We are interested in researching the development of tools for audiovisual (AV) performance in order to address the current practices and growing attention given to the field. This growth can be demonstrated by the several new festivals, seminars and publications (for example [4], [7], [10]) focusing in this area in recent years. In this study, taking into account needs identified in a previous stage [2], we aim to develop prototypes of tools for computer-generated audiovisuals, combining expressiveness, ease of use and visibility of interaction to the audience. In this paper, we present a prototyping approach based on a hackathon, and the resulting five prototypes.

2. TOOLS FOR AV PERFORMANCE

There has been an increased interest in different forms of “screen-based performance”, adopting “a long litany of names such as audiovisual performance, real-time video, live cinema, performance cinema, and VJ culture” [11]. Three notable examples of contemporary audiovisual artists using computer-generated graphics and sound are Golan Levin, Toshio Iwai and Thor Magnusson. They are relevant to this study because they are concerned with creating interfaces and instruments for audiovisual expression. Levin developed a suite of works under the name *Audiovisual Environment*

Suite (AVES) and described his approach to audiovisual performance as being based on painterly interfaces [6]. Iwai creates playful pieces, crossing genres between game, installation, performance (with works such as *Elektroplankton* and *Composition on the Table*) and audiovisual instrument (with *Tenori-On*) [9]. Magnusson uses unconventional Graphical User Interfaces (GUIs) and “abstract objects that move, rotate, blink/bang or interact” to represent musical structures [8]. The tools he develops are often made available online.

Most ready-made commercial software tools for live visuals (such as Modul8 or Resolume) focus on video playback and manipulation. Therefore, artists interested in using video for their performances have a choice of using either turnkey (and easier to use) software, or programming languages / environments (with a steeper learning curve, but offering more flexibility). For artists dealing with procedural graphics, however, there is a scarcity of ready-made, easy to use software.

The notion of expressiveness in AV performances deserves further attention. Hook et al. have studied VJ performances from an interaction design perspective [5]. Their key insights are grouped in terms of expressive interaction in VJ performances around three thematic categories: *aspirational*, *live* and *interaction*. From these, we will concentrate on the last one, as it relates the most to tool design. Within *interaction*, Hook et al. identify the following key insights: *constraining interactions* (the importance of constrains and focus); *haptically direct* (using hardware controllers and having a physical connection to the system); *parallel interaction* (simultaneous control of multiple parameters); *immediacy* (immediate response from the software); *manipulable media* (desire for powerful and varied manipulation of media); *reconfigurable interfaces* (the ability to reorganize the controls to fit a particular performance); and *visible interaction* (to make the performer’s interaction visible to the audience). The last three insights in particular relate to our aims of combining expressiveness, ease of use and visibility of interaction. Hook et al. focus on video manipulation, but state the need for tools creating procedural content, namely visual “devices that mimicked audio-synthesizers”. Their study identifies key themes and needs, and points toward future paths, but does not produce new concepts or prototypes. We aim to further explore the paths laid out by their study, and produce prototypes as outcomes of that exploration.

3. HACKATHON ON TOOLS FOR AV

3.1 Hackathons and User-Centered Design

This study follows a User-Centered Design (UCD) approach. UCD is “a broad term to describe design processes in which end-users influence how a design takes shape” [1]. In particular, our approach is based on a hackathon, and the users are artists in addition to being computer programmers.

According to the online Oxford English Dictionary, hackathon is “an event, typically lasting several days, in which a large number of people meet to engage in collaborative computer programming”. Hackathons share resemblances to workshops – “collaborative design events providing a participatory and equal arena for sharing perspectives, forming visions and creating new solutions” [12] – which have often been used in UCD studies. However, hackathons

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. *NIME '15*, May 31-June 3, 2015, Louisiana State University, Baton Rouge, LA. Copyright remains with the author(s).

are more specific events, and require a technical skillset. Due to this requirement, hackathons are adequate for the creation of working prototypes. Workshops have been used as research methodology for modifying (“hacking”) musical instruments [13], and some collectives regularly organize events dedicated to music or video hacking (for example, the London Video Hackspace¹). Some organizations promote longer term do-it-yourself workshops to develop interactive projects (for example, MediaLab-Prado with the *Interactivos?* programme [3]). But the approach of using a hackathon as a UCD method for prototyping software for audiovisual performance is novel.

3.2 Preparing and Running the Hackathon

In a preparatory phase (presented in [2]) to the research reported here, we conducted interviews with 12 audiovisual performers. We asked the artists about their practice, the tools they use, and their needs as performers. This generated a series of key ideas that informed a brainstorming workshop. The outcomes of the workshop were sketches of tools for audiovisual performance using procedural sound and graphics. These sketches were the input to the hackathon.

A call for the hackathon was distributed among the Goldsmiths and London Video Hackspace communities. Previously to the hackathon, we interviewed four of the more experienced participants to validate our plans for the event, and obtain suggestions. The two-day (8 hours per day) hackathon took place in December 2014, and a pilot was conducted exclusively with Goldsmiths students. 18 participants took part in the hackathon, and five in the pilot (five female and 18 male in total). The participants were divided into six groups, taking into account a previously filled-in questionnaire identifying preferred programming languages and development environments. Groups were created focusing on Processing (three groups), openFrameworks (two) and Cinder (one). Participants with an emphasis on sound (Max/MSP and Pure Data) were distributed through those groups.

The hackathon started with a presentation on the previous stages of the study (interviews and workshop) and results achieved so far. The sketches produced in the workshop were presented in detail, and participants were invited to adopt features of the sketches into their projects. The structure for the workshop was outlined. The first stage was the development of sketches, followed by software programming. A follow-up one-day event was agreed for finishing the projects. One of the groups did not continue to the follow-up event, therefore there were five resulting projects in total.

We used GitHub, a web-based Git repository hosting service, to manage and distribute the code created for each project. We created an “organization” page within GitHub², and each group created a code repository within it. GitHub facilitates the identification of contributions, and modification (“hacking”) of software by others. The groups were encouraged to release their projects as open-source.

4. RESULTS

The resulting five projects from the hackathon are: *ABP*, *drawSynth*, *Esoterion Universe*, *GS.avi* and *Modulant* (Figure 1). They are available for download or for code modification, from the respective GitHub pages. A website was created for documenting the projects.³

ABP is an animation engine and sound visualizer where the user can define color, geometry and animation parameters (position, rotation, size, motion vector) with a GUI. The GUI can be shown to the audience. The visuals are based on a particle system, which is sound-reactive. The visual module is built with Cinder and the sound module is created with Pure Data. The sound consists of two drum synthesizers. Information is communicated between the two programs using the OSC (Open Sound Control) protocol.

drawSynth consists of a GUI to control sound and image. Users can draw vector shapes and select colors. By doing this, they control the FM synthesis engine. The position of the vector points on the screen affects parameters of the synthesis, such as number of carriers, modulators and their frequency. The project is built with openFrameworks for graphics and interaction, and the Maximilian openFrameworks add-on is used for sound.

Esoterion Universe starts with an empty 3D space that can be filled with planet-like audiovisual objects. The objects can be manipulated and can be given different appearances and sounds. The visual component of the objects is audio-reactive. Users can navigate in space and the audiovisual outcome is influenced by that navigation. Generic, media neutral terms such as warmth, sharpness, size and roughness are used to characterize and connect sound and visuals. This semantic approach was chosen instead of a one-to-one parameter mapping. The GUI consists of sliders distributed concentrically, in the shape of a star graph, embedded in the center of the object, and integrating aesthetically with the objects. openFrameworks with OpenGL is used for graphics and interaction. The sound component is a granular synthesizer built with Max/MSP. OSC is used for communication.

GS.avi is a gestural instrument that generates continuous spatial visualizations and music from the input of a performer. The features extracted from a performer’s gesture, using the GVF software⁴, defines the color, position, form and orientation of a 3-dimensional Delaunay mesh – its composite triangles, vertices, edges and walk. The music, composed using granular synthesis, is generated from features extracted from the mesh – its colors, strokes, position, orientation and patterns. The project was created using Processing and Max/MSP. OSC is used to communicate between the two.

Modulant allows for the creation of images and their sonification. The present implementation is built upon image-importing and freehand-drawing modules that may be used to create arbitrary visual scenes, with more constrained functional and typographical modules in development. The audio engine is inspired by a 1940’s synthesizer, the ANS, which scans across images. In this scanning, one axis is time and the other axis is frequency. *Modulant* thus becomes a graphical space to be explored sonically and vice-versa. The project is built with Processing for graphics and interaction, and Ruby with Pure Data for sound.

A public performance took place in February 2015⁵, and an audience evaluation was conducted. An additional evaluation of the projects by AV performers has also taken place meanwhile. We are now in the process of analyzing the results from these evaluations.

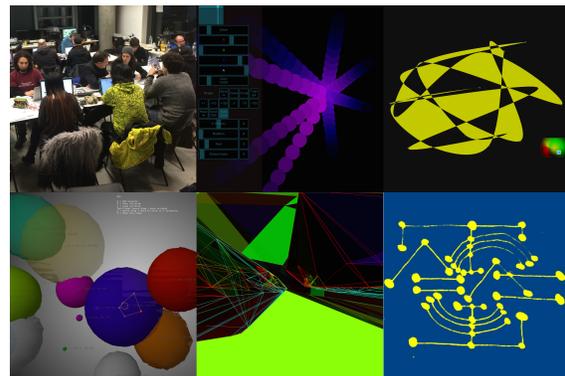


Fig. 1 Left to right, top row: hackathon, *ABP*, *drawSynth*; bottom row: *Esoterion Universe*; *GS.avi*; *Modulant*.

¹ <http://www.videohackspace.com>

² <https://github.com/AVUIs>

³ <http://avuis.goldsmithsdigital.com/gen-av-feb-2015/>

⁴ <http://eavi.goldsmithsdigital.com/resources/gesture-variation-follower-gvf/>

⁵ Video from the performance: <https://vimeo.com/124065089>

5. DISCUSSION

5.1 Prototypes

Looking at the five projects through the lens of expressive interaction of Hook et al., the projects diverge in their degree of conformity with the design insights, particularly in the thematic category *interaction*, and the related key themes identified for this study of *manipulable media, reconfigurable interfaces* and *visible interaction*.

Two of the projects offer immediate *manipulable media* capabilities in the visual domain – *drawSynth* and *Modulant* – since they are drawing applications. They allow for varied graphical manipulation: vector-based, in the case of *drawSynth*, and bitmap-based, in the case of *Modulant*. Of the two, *Modulant* is the most versatile, since it provides a larger diversity of drawing tools, and it allows for the saving and loading of bitmap images. *drawSynth* is the most fluid, since it produces immediate sonic results upon drawing. However, both offer very simplistic drawing capabilities at this stage. The graphics of the remaining projects are not as manipulable.

Sonically, the projects that offer greater manipulation are the ones based on granular synthesis – *Esoterion Universe* and *GS.avi* – as they allow for the loading of different sounds, leading to more varied results. *Esoterion Universe* offers more sonic manipulation capabilities, based on its semantic approach, although they do not change the sound characteristics drastically. The sonic manipulation capabilities of the different projects are not very powerful, leading to results that are not substantially varied.

None of the projects allow, on the surface, for truly *reconfigurable interfaces*. However, the fact that the projects are open-source allows for reconfiguration. The reconfiguration of interaction is particularly easy in the case of the projects that use the OSC protocol for data communication: *ABP*, *Esoterion Universe* and *GS.avi*. The OSC messages can be easily rerouted to other parameters, or entirely different sound generation sources.

Regarding *visible interaction*, most of the projects (*ABP*, *drawSynth*, *Esoterion Universe* and *Modulant*) display the media manipulation act to the audience. The screen seen by the performer is also intended to be the screen shown to the audience. *ABP* and *Modulant* rely on more traditional GUIs – buttons and faders in the former, and a “drawing tools palette” in the latter. In *drawSynth*, the drawing tool is fixed – a line-drawing tool, complemented by a color picker. In *Esoterion Universe*, a radial GUI with faders is embedded in the “planets”. In all four, the cursor is showcased to the audience, revealing the choices made. In *ABP*, both GUI and cursor can be hidden. *GS.avi* is the exception – it was designed for gestural interaction, preferably with a tablet. The interaction with the project is not made visible to the audience. All five projects rely in keyboard shortcuts, in alternative to, or in addition to, the pointer- or touchscreen-based interaction.

5.2 Hackathon Approach

The hackathon approach was successful in converting sketches into functional prototypes in a short period of time. Adding an extra day to the planned two-day hackathon proved to be decisive for the completion of projects. Therefore, one important lesson learnt was to have a flexible timeline for the hackathon. However, an extension of the hackathon has the risk of leading to a higher number of dropouts – some of the participants did not continue to the extra day.

GitHub proved to be an essential tool for groups to collaborate and share code, not only during the hackathon but also for collaborating remotely outside of the event. Additionally, it provided an important platform for sharing the projects and their code. One of the groups used GitHub for communication and knowledge sharing as well.

Organizing the teams around the same technology was important for the success of the hackathon. Running a questionnaire before the workshop for identification of preferred technologies saved valuable time during the event. However, an AV project does not have to be built using a single technology. In particular, the sonic and visual

components can be built with different technologies, and communication between those can be facilitated by OSC. Building the project with a single technology does have an advantage – ease of distributing and setting up the software.

6. CONCLUSIONS

Some of the prototypes (notably *Esoterion Universe* and *Modulant*) present potential for addressing the needs of audiovisual performers. As prototypes, all of the projects have room for improvement. But even at this stage, they contain strengths that can be of use and inspiration to other projects. Additionally, the code from these prototypes can be reused, given their release as open-source. The GitHub platform, where the projects are hosted, facilitates this reuse. Most of the projects adopt the OSC protocol, allowing for easy remapping of audio and visual parameters. Hence, these hackathon outcomes are also *hackable instruments* that allow for “discovery of novel working configurations”, as defined in [13].

With the present study, we introduced a novel approach for the user-centered prototyping of tools for audiovisual performance, based on a hackathon. This approach was successful in converting sketches into prototypes in a short period of time. It was also successful in providing different creative perspectives to an initial challenge. We believe that hackathons offer potential to be used for UCD studies in collaborative prototyping for creative fields beyond audiovisual performance. We are planning further developments taking into account the forthcoming evaluation results.

7. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement n° 627922. We would like to thank the hackathon participants, Alessandro Altavilla for the photo and video documentation, and Peter Mackenzie for technical assistance.

8. REFERENCES

- [1] Abras, C., Maloney-Krichmar, D., and Preece, J. User-Centered Design. In W. Bainbridge, ed., *Encyclopedia of Human-Computer Interaction*. Sage Publications, Thousand Oaks, 2004.
- [2] Correia, N.N. and Tanaka, A. User-Centered Design of a Tool for Interactive Computer-Generated Audiovisuals. *Proc. 2nd ICLI*, 2014.
- [3] Corsín Jiménez, A. and Estalella, A. The hospitable prototype. *Prototyping cultures*, 2010.
- [4] Faulkner/D-Fuse, M. *VJ: Audio-Visual Art and VJ Culture*. Laurence King, London, 2006.
- [5] Hook, J., Green, D., McCarthy, J., Taylor, S., Wright, P., and Olivier, P. A VJ centered exploration of expressive interaction. *Proc. CHI '11*, ACM, 2011.
- [6] Levin, G. A Personal Chronology of Audiovisual Systems Research - Golan Levin and Collaborators. *Proceedings of NIME '05*, (2005).
- [7] Lund, C. and Lund, H. *Audio Visual: on Visual and Related Media*. Arnoldsche, Stuttgart, 2009.
- [8] Magnusson, T. Screen-Based Musical Instruments as Semiotic Machines. *Proc. of NIME '06*, 2006.
- [9] Nagle, P. Yamaha Tenori-On Review. *Sound On Sound*, February 2008.
- [10] Rainer, C., Rolling, S., Daniels, D., and Ammer, M., eds. *See This Sound*. Walther König, Köln, 2010.
- [11] Salter, C. *Entangled: Technology and the Transformation of Performance*. MIT Press, Massachusetts, 2010.
- [12] Soini, K. and Pirinen, A. Workshops - Collaborative Arena for Generative Research. *Proc. DPPI '05*, 2005.
- [13] Zappi, V. and McPherson, A. Design and Use of a Hackable Digital Instrument. *Proc. 2nd ICLI*, 2014.