Learning Visual Programming by Creating a Walkable Interactive Installation

Aristotelis Hadjakos
Center of Music and Film Informatics (CeMFI)
University of Music Detmold
University of Applied Sciences OWL
Hornsche Str. 44, 32756 Detmold, Germany
hadjakos@hfm-detmold.de

Heizo Schulze
Center of Music and Film Informatics (CeMFI)
University of Music Detmold
University of Applied Sciences OWL
Liebigstraße 87, 32657 Lemgo, Germany
heizo.schulze@hs-owl.de

André Düchting, Christian Metzger, Marc Ottensmann, Friederike Riechmann, Anna-Maria Schneider, Michael Trappmann
Department of Media Production
University of Applied Sciences OWL

ABSTRACT

In this paper we explore the question of how to teach visual programming languages, such as Max/MSP, Pure Data, vvvv, to experienced media and sound artists. A pedagogical challenge with these students is to keep them motivated after the initial phase of learning the core concepts and creating first simple programs. This is the case because they already have their means for artistic expression and experience a mismatch between their current skill and what they would need to successfully apply visual programming in their art. Our approach consists of letting them develop an interactive installation based on the principles "media first", liveliness, design patterns and technical framework. The core idea is to let the students create and bring their media, in which they are experts creating and use this as the backbone of an interactive installation. Our students developed a walkable interactive installation *bang!*, which serves as a case study for this approach.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation (e.g., HCI)**]: Sound and Music Computing

General Terms

Human Factors

AM15, October 07 - 09, 2015, Thessaloniki, Greece
Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3896-7/15/10 ...$15.00.
http://dx.doi.org/10.1145/2814895.2814914.

Keywords

Visual programming languages, interactive art, Max/MSP

1. INTRODUCTION

Visual programming languages, such as Max/MSP, Pure Data, vvvv, are commonly used by media artists and electronic musicians for creating an element of interactivity in their pieces, performances and installations. One reason for the popularity of visual programming languages is certainly the apparent simplicity of visual programming, which seems to involve nothing more than connecting several objects with each other. However, creating something interesting is oftentimes harder than that first impression might imply. After learning some initial concepts, such as message ordering in Max/MSP and Pure Data, the next important step is to gain basic knowledge about the most important objects and on how to connect them to form commonly used patterns. In our experience (teaching Max/MSP to composers and media producers) a hands-on approach is most effective in that phase. By working on toy problems, the students gain the mentioned familiarity and a sense of achievement. However, the gap between what the students have already learned and what would be necessary to use those techniques in their artistic works is still significant. Therefore, we wanted to create a motivating task for our students, who already have their means of artistic expression in (instrumental) composition or in film making: We wanted to let them create a public walkable interactive installation. This is of course a challenging task for students that are just beginning to learn their first programming language. To make that task feasible, we formulate a teaching pattern consisting of four key ideas:

1. Media first: Let the students create and bring their own media. After all they are experts in making films, animations, sounds or music. Let those media snippets be the backbone of the work and use interactivity to control these elements.

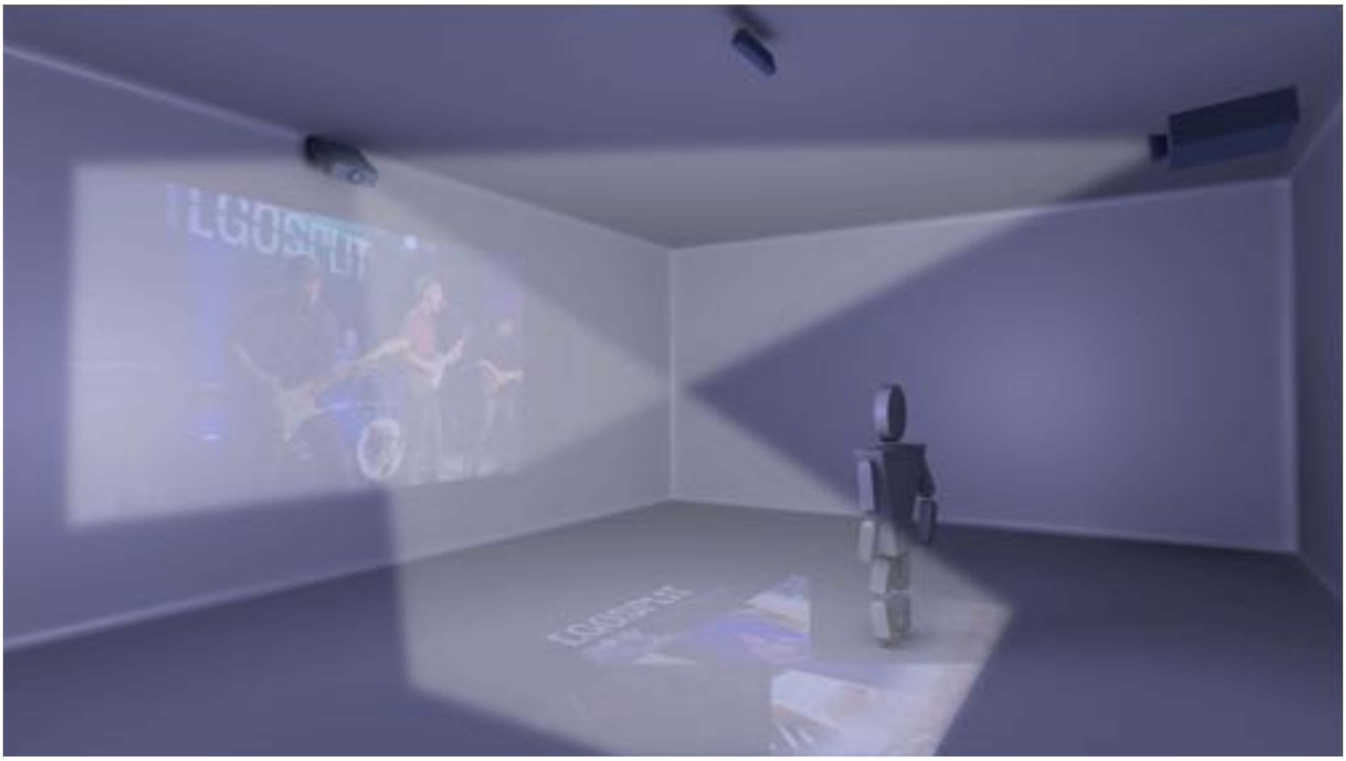

Figure 1: *bang!* is an audiovisual installation with a interactive floor and a immersive video screen.

2. Liveliness: A disadvantage of using such media snippets is that they are static without inherent interactivity. The interaction should be designed for liveliness to avoid that the installation behaves like a rather cumbersome media player. One potential solution is to use effects to change the source material interactively.
3. Design patterns: For the students this is also the first time that they create an interactive installation. Apart from reviewing existing works (see, e.g., [10, p. 120–157]), we found it useful to discuss design patterns for interactive installations.
4. Technical Framework: Provide a framework that solves technical or uninteresting tasks, but let the students control the parts that are central to the design of the interaction.

Interactive installations are discussed in secondary literature [10, p. 120–157] and also in the form of research papers. Some recent systems include an installation in a science history museum that provides sonic interaction with the stones and minerals of the collection [2]. The Cave of Sounds features eight interactive instruments that allow the users to play together as an electronic ensemble [8]. Another installation uses suspended glass sheets with sound transducers on them, forming an oval shape in a room [6]. The vibrations and the projections onto the glass sheets generate an illusion of ephemeral sound, projections, reflections and transparency. The installation Vocal Vibrations lets the user explore her own voice as the vibrations of the voice is reproduced on a special device with tactile actuators, which the user holds in her hand [5].

bang! is a walkable interactive audiovisual installation (see Figure 1). The position of the user, which is tracked with the help of a Kinect depth camera, is used to trigger and control audiovisual content. Two projectors show videos and graphics on a large video screen and on the floor. The installation has three different parts: In one of them, the visitor is suddenly trapped in a Horror House and has to find three keys to get out again. The second part is an interactive music video, where the user can experience the performance of a band from different positions. In the third part, the user can control Alex doing his morning routine.

bang! was developed with Max/MSP using the said teaching pattern with media production students of our university at master level.

2. DESIGN PATTERNS

As the students are just beginning to learn to program, they also have no experience in designing interactive installations. Apart from reviewing existing works, we find it instructive to present and discuss Borcher's design patterns for interactive exhibits [1]. For each pattern, Borchers provides a name, a subjective ranking of its validity, a graphical illustration, a discussion of the problem that is addressed, the suggested solution together with a descriptive illustrations and references to other patterns. Some of the patterns described by Borchers were instructive for the design of *bang!*, in particular:

- Attract-Engage-Deliver [1, p. 105]: This pattern suggests to organize the interaction into three phases. First, the installation has to attract a user. Then the user engages in the interaction. And finally the interaction delivers a message it was intended to convey to the user and concludes. Borchers also advises to plan the interaction time according to the influx of users. We limited each sub-installation to a duration of approximately 90 seconds and designed the overall interaction according to the three phases.

- Attraction Space [1, p. 109]: By looking or sounding interesting, beautiful or innovative, an installation can grab the attention of visitors and motivate them to start interacting with the exhibit. Without a well-designed attraction space users may just move on to other exhibits. However, excessive noise and visual overstimulation may be counterproductive. *bang!*'s home state shows a visually appealing selection screen projected to the video screen and the floor.

- Simple Impression [1, p. 120]: Users typically interact with an exhibit for the first time and only once. Therefore, the exhibit should be easy-to-use and that simplicity should also be also conveyed visually. *bang!* features quite simple interactions: "Alex Got Time" and "Egosplit" use direct mappings from position to videos while the "Horror House" features a gradually more complex interaction with state.

- Invisible Hardware [1, p. 141]: This pattern suggests to hide the computer hardware from the user and show only the devices that the user has to interact with. *bang!*'s hardware was hidden in a corner or the room at the opposite side of the video screen, which also contributes to provide a simple impression (see previous pattern).

- Innovative Appearance [1, p. 137]: Instead of using standard interaction setups (mouse, keyboard, computer screen), exhibits can use non-standard input and output devices to provide an innovative appearance, which may be more interesting for the user. *bang!* uses user tracking and two projections one on the screen and one on the floor.

- Cooperative Experience [1, p. 113]: Since exhibits are commonly visited by groups, at least two persons should be able to use an interactive exhibit at the same time. Bystanders should be able to experience the exhibit passively, which also helps to attract new users. The interaction in *bang!* is designed for only one user at a time. However, the immersive screens allows bystanders to experience the installation passively.

- Easy Handover [1, p. 117]: There should be a simple way to hand over the control from one user to another and still provide a meaningful interaction, even though the new user may not have experienced the entire interaction. In *bang!*, a user can hand over control to another one by leaving the interactive area. *bang!* has very little state making it simple to take over the interaction from another user.

- Immersive Display [1, p. 139]: By using large displays it is easier to immerse the user into the virtuality of the exhibit, which may help to provide an engaging experience. *bang!* even uses two projections, one on a cinema-sized video screen and the second on the floor.

3. MEDIA CONCEPTION

According to our idea "media first" we chose to center the installation around the medium of video. In the following we discuss the media conception for the three sub-installations.

3.1 Alex Got Time

The goal of the installation "Alex Got Time" is to control what happens on the video screen and to interact with the protagonist named Alex. However, once the user starts to think that he can manipulate Alex for all time, Alex gets it his way. He than does completely other things for a short time while the user cannot exert any influence on him. The setting is Alex's bathroom where he stands in front of the mirror doing his morning routine (see Figure 2). Alex is shown from the perspective of the mirror and is combing, blow-drying, shaving, or brushing his teeth. A fish eye perspective effect makes the user believe that Alex is very close. The installation lets the user get in immediate contact with Alex and command him what action to perform. Furthermore, the user can control the execution speed on a five-point scale from very slow to very fast. This simple effect allows for a new and uncommon view on everyday activities, which may provoke the user to think about his own time spent in the bathroom, and also about how the user lets others manipulate him or how he controls the actions of others.

The selection of activities is done by walking into one of the four areas marked on the floor projection (see the floor plan in Figure 3). When the user enters the area marked with the comb, Alex starts combing. One step further and Alex is blow-drying. Further to the front, Alex shaves and, even further Alex brushes his teeth. Towards the left, the actions become slower and slower, while they get faster and faster towards the right.

The icons and markings on the floor plan are suggestive of the various activities and velocities. However, there are also hidden video clips, which are not indicated. They are activated whenever a user stays on the same spot for more than eight seconds, Then Alex reacts to the passing of time by evading the manipulation and doing something unexpected, e.g., he drinks a beer because he got bored. After a short amount of time, Alex is available for the user again. If the user lets Alex comb his hair with the slowest speed, Alex dozes off after a short amount of time. If the user stays in the middle, Alex will eventually grab his smartphone and

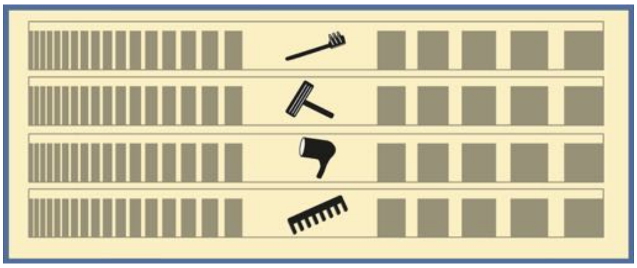

Figure 3: Alex Got Time: the floor plan with four areas corresponding to the four action types. Playback speed is controlled by moving left (slower) and right (faster)

Figure 2: Alex is doing his morning routine

busy himself with it. Or if the user lets Alex brush his teeth at the highest speed, Alex will eventually just use his electric tooth brush.

Liveliness: To create liveliness “Alex Got Time” uses two mechanisms. For one, it modifies the playback rate of the videos based on the left-right position of the user. Second, Alex Got Time defines 20 (4 morning routines × 5 discrete video speeds) separate areas, making it a very “busy” installation. To underline that business each change is underlined by a sound sample.

3.2 Egosplit

In this part, the spectator has the opportunity to participate in a concert of the alternative rock band Egosplit. The video on the main screen shows the band’s performance from one of six different directions, depending on the position of the user in the tracking area (see Figure 4). When the user moves to the right, the camera position switches to the right and vice versa. This provides the impression that the user can walk around the band during a performance. To create this effect, the cameras were positioned symmetrically in a semicircle with an uniform distance. A floor plan aids the user (see Figure 5). It provides an indication of which perspectives can be triggered and the five areas indicate which band member is at the center of attention.

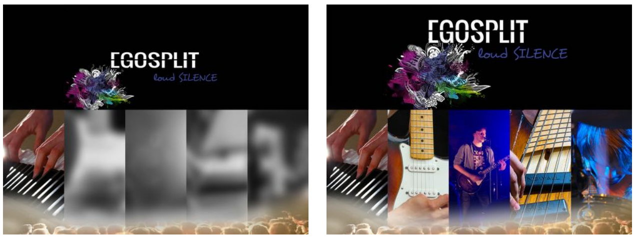

Figure 5: Egosplit: The interactive floor highlights the currently active field (left). There five areas (right) provide an indication of which band member is at the center of attention.

The band’s sound is interactively altered depending on the position of the user. The drums were located on the right side so that the drumming sound is emphasized when the user moves to the right while the other instruments and the singer are attenuated. Similarly the guitar is emphasized when the user moves towards it. Furthest to the front, the user enters the backstage area (see Figure 4, bottom) and hears a muddy sound as one would expect when standing behind the loudspeakers. The audio interactivity was realized by using a multi-track studio recording that the band provided. The left-right position of the user is used to control the sound level of each track continuously. The video was displayed by sending OSC messages to the video player described in Section 4.2. To make sure that audio and video stay in sync, the audio progress is used as a reference. Whenever, the video switches the video player is informed of the current progress and plays the video starting from that

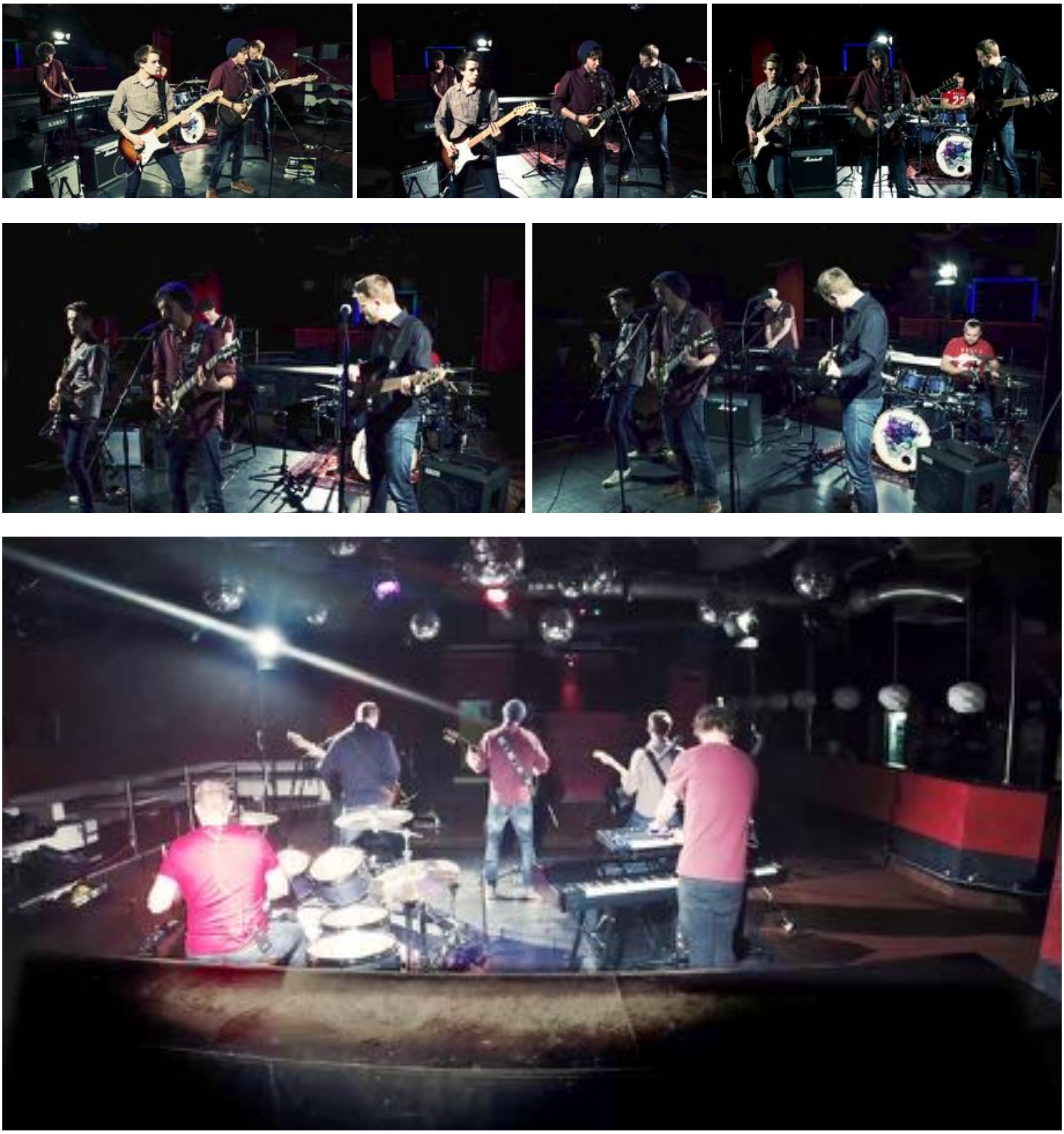

Figure 4: The installation Egosplit lets the user experience the band from different camera angles and emphasize specific instruments in the audio track.

particular position.

Liveliness: To provide greater liveliness Egosplit uses the auditory modality. When moving inside the tracking area, the users position continuously emphasizes and deemphasizes the different audio tracks, making the installation musically interesting.

3.3 Horror House

The protagonist wakes up. Instinctively he touches an area in the back of the head that hurts. He opens his eyes slowly and looks around. There is very little light, but he can make out high walls, old furniture, withered plants. "Where am I?," he says to himself when he begins to feel the need to cough. The air is full of whirled up dust; the floor he is lying on is cold. Slowly, he recognizes that he is in a very old house. His memory is blurred and he has no idea how he came to this house. Fear is spreading. The protagonist notices a double-winged door behind him in an area that appears to be an entry portal. He pushes down the door handle, he rattles at the door, but nothing happens. Panic is spreading. He rattles more violently and is frightened when he hears a loud growl from the other side of the door. He takes a few steps back. He still hears howling as if a sharp wind is blowing around the house. However, there is another sound, very silent, hardly hearable, like a whisper: "Find the three keys."

The projection on the floor shows the layout of the horror house (see Figure 6). If the user enters one of the areas marked with a red smiley, a spooky yet funny video is played on the main screen. In three of these locations the user will get a key of which he is informed in the video. If the user goes to the exit area indicated by the lock symbol on the floor plan, the doors will remain locked unless the user has already acquired all three keys. If the user has all three keys, the doors open only to find a zombie that attacks the protagonist. An overview of the sequence of events is sketched in Figure 7.

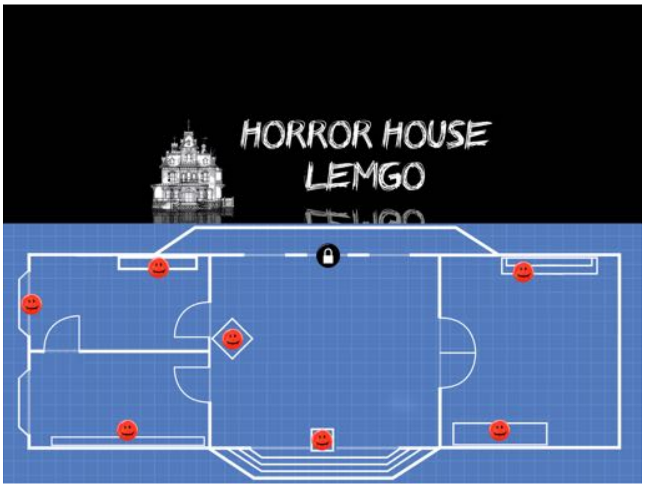

Figure 6: Horror House: floor plan

Liveliness: The Horror House is essentially a small computer game, creating a fun experience by a playful interaction. The liveliness originates from the game experience rather than from audiovisual interactivity.

4. TECHNICAL FRAMEWORK

This section describes the framework, which was provided to the students since it deals with technical aspects that were not central for the interaction design.

4.1 User Tracking

We used a Kinect camera mounted at the ceiling to track the users and employ a tracking method similar to the methods described in [3, 4]: The Kinect camera provides a grayscale image where pixel brightness depends on the distance from the camera. The Kinect films the user in the tracking area from above. Areas further away from the camera, in this case the floor, are darker and areas close to the camera, in this case the user's body, are brighter. Since the distance between the camera and the floor is fixed, thresholding can be employed to distinguish between floor and user pixels. The binary image that results from this operation is further processed. Blob detection is applied to the binary image and the largest blob is identified. This removes spurious false detections from the binary image. If the blob is sufficiently large, the user is detected as the center of that blob.

4.2 Video Playback

The installation totals 35 videos that are oftentimes switched in rapid succession based on the position of the user. In a preliminary implementation, we tried to use the video playback facilities provided by Max/MSP. However, the continuous and repeated switching lead to severe stability problems, making it necessary to restart the installation every couple of minutes. To evade this problem, we programmed a native video player for Mac OS X and provided that to the students. The video player is controlled with OSC messages from Max. Three messages are used to trigger and control video playback:

```
/show electric_brush.mp4
/showFrom band_camera3.mov 3.14
/rate 1.3
```

The "show" message starts the video at the beginning, while the "showFrom" message also specifies the time in seconds where the video should start. Finally, the "rate" message changes the playback rate of the video. Whenever the playback reaches the end, the video player sends an "endnotify" message, informing the client of that event:

```
/endnotify electric_brush.mp4
```

The application can then react appropriately. The Horror House, e.g., uses this message to switch back to the home screen once after a video has been completed.

4.3 Floor Projection

The projection on the floor was realized with a beamer that was mounted over the video screen area (see Figures 1 and 9). By placing it there, we ensured that the shadow of the user's body is not in sight when the user faces the video screen. The projector is tilted sufficiently towards the ground in order to avoid the user to be blinded by the projector beam. Instead of using a projector with keystone correction, we used an own implementation in Max/MSP to make sure that the projected image is not distorted: Using OpenGL, the original image is tilted around the horizontal axis (see Figure 8). Once projected on the floor, the image is again undistorted. In comparison to using the keystone

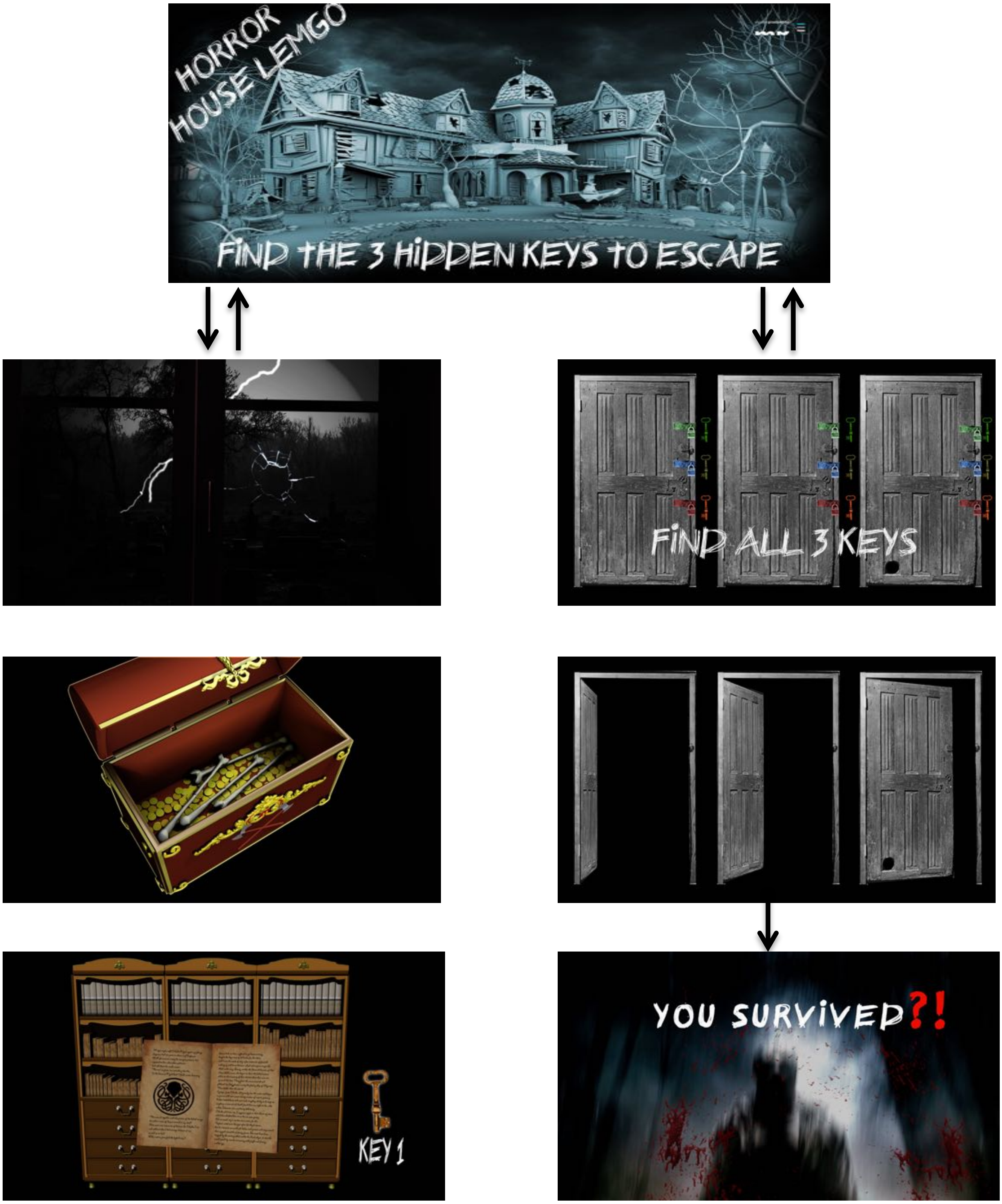

Figure 7: The Horror House, screenshots from selected videos: Starting from the home screen, videos are triggered when the user enters one of the regions marked in the floor plan. If the user enters a red smiley region, he may or may not get a key (left column). To exit the Horror House, the user needs all three keys and move to the exit region (right column): Either the user is still missing one or more keys and cannot open the door or he opens the door only to meet a hostile zombie.

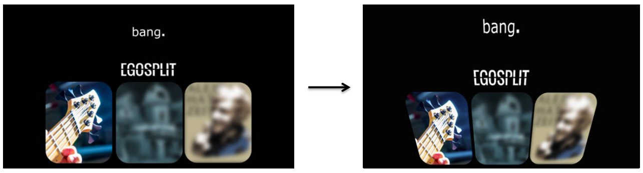

Figure 8: Floor projection: The original image (left) is tilted around the horizontal axis (right) so that it looks undistorted when projected to the floor.

correction of the projector, we gain an additional trapezoidal projection space above the rectangular part. We used that part to display logos and texts on a black background thus hiding the trapezoidal shape of that part.

5. CONCLUSIONS

bang! was finally presented as part of the open house day of our university. The installation was visited and tried out by many users and the overall reception of the installation was good. One minor and rare usability issue was discovered during that day. In the home screen (see Figure 9), which allows the user to select one of the three sub-installations, a countdown triggers if the user remains on one of the three areas for some time. The countdown was only displayed on the video screen and not on the floor and some users did not notice if they were looking at the floor projection currently. We learned, that in such setups with multiple projectors crucial information needs to be displayed everywhere.

Overall our pedagogical concept worked out. By using the particular expertise of the students, who are experts in producing film and media, it was possible to create an interesting walkable installation. We find it promising to apply this strategy based on “media first”, liveliness, design patterns and technological framework for teaching other visual (and non-visual) programming languages to media artists and sound artists.

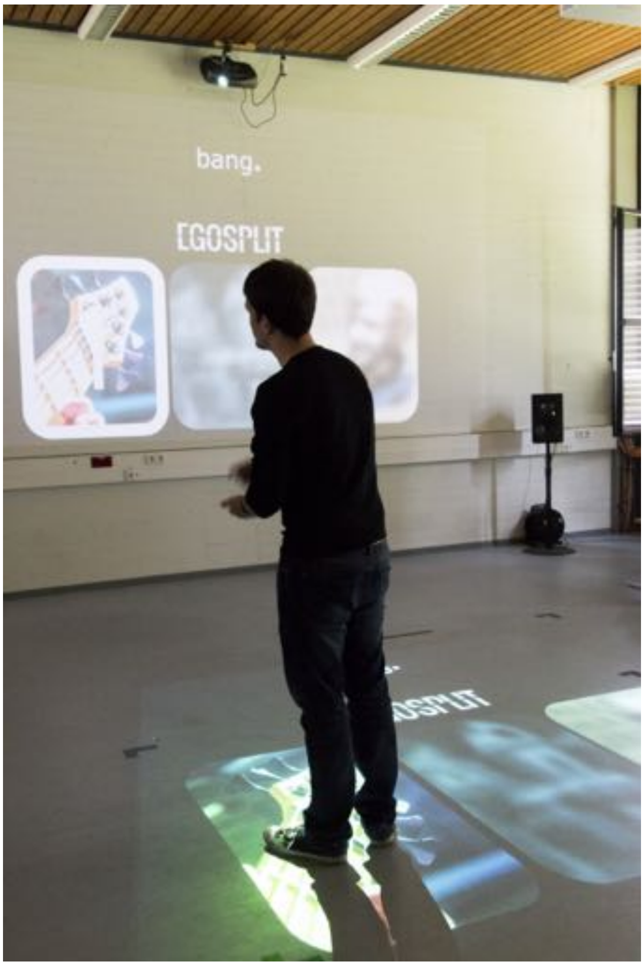

Figure 9: *bang!* at the open house day

6. REFERENCES

[1] J. Borchers. *A Pattern Approach to Interaction Design*. Wiley, 2001.

[2] J. Bowers and T. Shaw. Reappropriating museum collections: Performing geology specimens and meterology data as new instruments for musical expression. In *NIME*, 2014.

[3] A. Hadjakos. Pianist motion capture with the kinect depth camera. In *Sound and Music Computing*, 2012.

[4] A. Hadjakos, T. Grosshauser, and W. Goebl. Motion analysis of music ensembles with the Kinect. In *NIME 2013*, 2013.

[5] C. J. Holbrow, E. Jessop, and R. Kleinberger. Vocal vibrations: A multisensory experience of the voice. In *NIME*, 2014.

[6] O. Lähdeoja. Structure-borne sound and aurally active spaces. In *NIME*, 2014.

[7] B. Laugwitz, T. Held, and M. Schrepp. Construction and evaluation of a user experience questionnaire. In *USAB 2008 - 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society*, 2008.

[8] T. Murray-Browne, D. Aversano, S. Garcia, W. Hobbes, D. Lopez, T. Sendon, P. Tigas, and K. Ziemianin. The cave of sounds: An interactive installation exploring how we create music together. In *NIME*, 2014.

[9] C. Poepel, J. Feitsch, M. Strobel, and C. Geiger. Design and evaluation of a gesture controlled singing voice installation. In *NIME*, 2015.

[10] R. Simanowski. *Digital Art and Meaning: Reading Kinetic Poetry, Text Machines, Mapping Art, and Interactive Installations*. University of Minnesota Press, 2011.