AKSON, AN AUDIOVISUAL ENVIRONMENT FOR NETWORKED INTERACTION AND PERFORMANCE

LUÍS ARANDAS
University of Porto
INESC-TEC
luis.arandas@inesctec.pt

JOSÉ ALBERTO GOMES
Universidade Católica Portuguesa,
School of Arts
Research Center of Science and Technology of the Arts
jagomes@ucp.pt

RUI PENHA
School of Music and Performing Arts
CESEM
ruipenha@esmae.ipp.pt

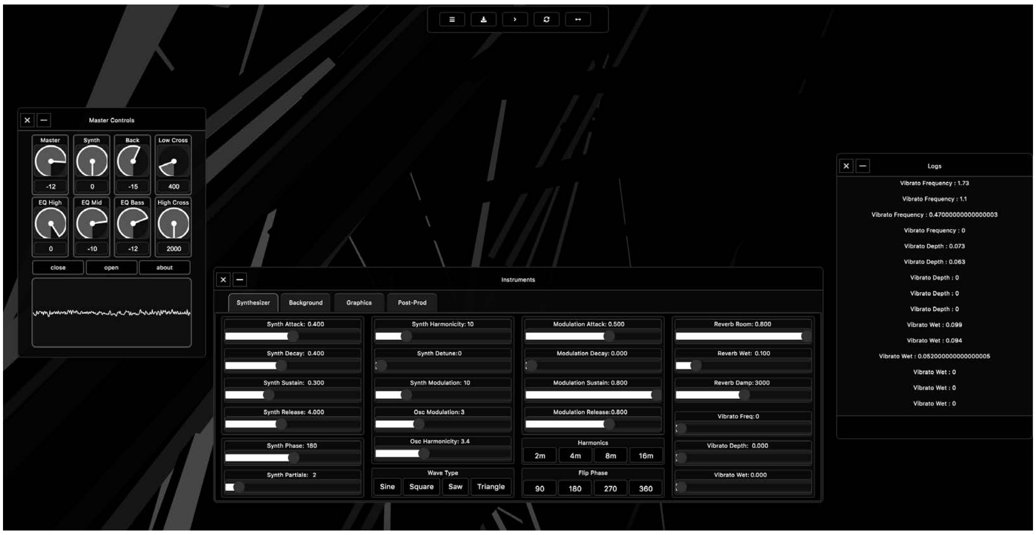
https://doi.org/10.34632/jsta.2021.9448

Journal of Science and Technology of the Arts, vol. 13, n. 2 (2021): pp. 142-158

ABSTRACT

In this article we present Akson, a web-based audiovisual (AV) environment for networked interaction and performance. As a result of research in the context of Braga Media Arts, part of UNESCO's creative cities network (UCCN), is documented why it was developed, its characteristics as a software that promotes networked artistic practice, as well as the tests performed. Taking advantage of the Internet's infrastructure, Akson allows the exploration of hundreds of devices scattered around the world and the study of the interface as an extension of the human gesture.

Keywords: Computer music; Media art; Networked interaction; Audiovisual art; Artistic performance; Networked art; Web technologies;



Master -12
Synth 0
Back -15
Low Cross 400
EQ High 0
EQ Mid -10
EQ Bass -12
High Cross 2000

Synth Attack: 0.400
Synth Decay: 0.400
Synth Sustain: 0.300
Synth Release: 4.000
Synth Phase: 180
Synth Partials: 2
Synth Harmonicity: 10
Synth Detune:0
Synth Modulation: 10
Osc Modulation: 3
Osc Harmonicity: 3.4
Modulation Attack: 0.500
Modulation Decay: 0.000
Modulation Sustain: 0.800
Modulation Release:0.800
Reverb Room: 0.800
Reverb Wet: 0.100
Reverb Damp: 3000
Vibrato Freq: 0
Vibrato Depth: 0.000
Vibrato Wet: 0.000

```
Vibrato Frequency : 1.73
Vibrato Frequency : 1.1
Vibrato Frequency : 0.47000000000000003
Vibrato Frequency : 0
Vibrato Depth : 0.073
Vibrato Depth : 0.063
Vibrato Depth : 0
Vibrato Depth : 0
Vibrato Depth : 0
Vibrato Wet : 0.099
Vibrato Wet : 0.094
Vibrato Wet : 0.052000000000000005
Vibrato Wet : 0
Vibrato Wet : 0
Vibrato Wet : 0
```

Figure 1: Akson AV Environment Graphic User Interface

1. INTRODUCTION

Leveraging the cloud as an ecosystem and medium for collaborative software development is a very common choice today. The easy access to the Internet provided by contemporary digital devices, as well as the various programming interfaces present in browsers, promotes the democratization of collaborative endeavours. In fact, the software is specifically designed with the potential of the Internet *as structure* for connecting people. Given the technical advances in browsers to perform audio and graphics almost near-native speed (see Lyu, 2021) this research sought to design an AV environment – Akson – that would allow an augmented artistic practice distributed around the world. Using contemporary digital devices, provided with screens, as Manovich (2001) said, we can understand and use the screen as a window that shows us another virtual space, that is, a world that is situated in our physical space (Teixeira, 2018).

2. THE INTERFACE AND THE NETWORK

Akson is an AV environment with its own network. Given the nature of the web, and the way software exists on a website, it must be divided between page and server – no matter what features it may offer. A connection structure has been developed from scratch in *node.js* (Mardan, 2018) and hosted in the Heroku infrastructure (Surianarayanan & Chelliah, 2019), a service that has similar properties to Amazon AWS, in order to have a scalable project with a dedicated Unix machine. This structure allows, within Akson, to control not only the environment and its instruments but also how users will interact with each other, opening possibilities of internal communication (see scalability proposal on Arandas et al. 2019). Given the nature of the subject and how one interface can be multiplied, extended and operated by multiple others, it is possible to make the initial statement – that Akson is an environment where users can create sound and graphics as well as create it with each other – see figure 1. The fact that they can create with each other inherits that it must have or exist in a network. Given its centralised nature (see Arkko (2020) for an essay on the topic), everything in the environment – regardless of interface and location – passes through the server. The network boundaries in Akson are specified by the connections made – if it is not connected to the server, it is not part of the virtual network.

On the established server, is where the various *interaction models* are organized – later presented. These *models* define how *users* interact, and users are defined as instances connected to the server, for the same artist can use more than one[1]. It is on the server, where an initial management of the data flow is made. Once a device is successfully connected, and if it can run the project code, the AV system and its controls are then rendered.

The interacting space – i.e., the web page where Akson exists – is developed as a multi-instrument environment. This division has made it possible to establish relationships between the interaction models – that is, between the humans who interact and their roles in performance – and by the various controls it offers it is then possible to explore various visual and sound properties. For a better understanding, we can divide the software presented as: 1) a networked AV environment that allows the use of several instruments; 2) an adaptable connection structure with models of interaction between various instances of Akson in a customised way; 3) a collaborative space in which the gesture of each user can be echoed in various interfaces around the world.

3. GRAPHICAL INTERFACE

The environment consists of an audio system and a graphics system that allows interaction by clicking the screen or the computer mouse. Through interaction with the graphics system the user has both visual and sound reproduction that can later be customized – the graphics system consists of 3D scenes with geometries in space and shaders that can be applied later, the audio system of a polyphonic synthesizer, noise sources and filters. The way to customize these properties is from a graphical user interface (GUI) that is composed of elements such as sliders, buttons, and knobs. This way an interaction approach is made based on touch and not on code writing – see Thompson & Wilson (2020) for recent contributions in *live coding*. Akson was developed using Google Chrome (Gray, 2009) as a target platform and keeping the features described, it has two types of operation – *as desktop* or *as mobile device*. These two terms are used for classification, yet devices such as iPads capable of executing all the necessary controls on a desktop will be assumed to be so – at the beginning of the page the device is scanned to see if it is considered *vintage*, does not have a large enough display or cannot run the necessary libraries – even though it is a *mobile device*. When the interface cannot run the page, there is no GUI and its elements. This is a way to include the ones that are not able to use all the features of the environment and can eventually be used as resonant machines in space (see the work of Clément et al. (2016) in orchestrating the audience during performance). Detailed in the next sections are the various panels that are part of the GUI as well as their functionalities[2].

[1] *User* from a programmatic point of view, is an instance that has established connection to the server. It is always possible for the same person to have several instances. Akson users can be i.e., musicians on stage, as well as members of the audience with multiple mobile phones.

[2] The main repository of Akson can be found at https://github.com/luisArandas/akson.

3.1. MASTER DIALOG

Entitled *Master Dialog* is one of the five panels that make up the GUI – figure 4. Here the user can find general controls of the environment – adopted the name *master* as in *master fader* (see Yamazaki, 2015). The term is inherited as it eventually stands out from other potential controls offering features such as the ability to mute the entire application (Pfeiffer & Green 2015). It consists of eight knobs where: 1) we can control the volume and general lighting of the application; 2) control the volume of the two independent audio generators; 3) control three band-pass filters; 4) apply a high and low crossover – (items 3 and 4 act globally, i.e., in the audio *main out*). It also has three buttons to turn the rendering on and off and open another panel entitled *System Dialog* – figure 4 – that the only function it has is to display text with the specifics of the system – as far as the browser allows us (see Mowery et al. (2011) for device *fingerprinting*). At the bottom of the panel there is an oscilloscope to visualize the audio master signal.

3.2. INSTRUMENTS DIALOG

As mentioned, Akson is designed to be an environment composed of instruments, namely four (see Magnusson 2009 for the *limits* of a musical instrument). Two for the audio system entitled *Lead* (a six-voice polyphonic synthesizer) and *Background* (a filtered noise source), one for graphics entitled *Graphics* (to control various properties of the visual scenes) and a combination of the two entitled *Post-Prod* acting on general AV properties of the environment. The properties of the various instruments that can be modified are present in the *Intruments Dialog* – figure 5 – panel which in turn is divided by tabs. As individual GUI elements in this panel we have 64 sliders and 96 buttons all with numeric display of the modified number percentage and nomenclature.

> *Lead* Tab – Controls such as an amplitude envelope (using ADSR, see Deutsch et al., (1978)), harmonicity, ring modulation, partial harmonics, phase, wave type (sine, square, saw and triangle) and virtual filters such as vibrato and reverb are found.
>
> *Background* Tab – The noise source has an automated filter that can be changed (a filter with a low frequency oscillator (LFO) connected to the cutoff frequency), creating modulation. For both, it is allowed to change properties such as minimum and maximum frequency limit, signal type (white, brown, or pink), gain, octaves, and base frequency, LFO rate and a filter chain composed of phaser and reverb applied to the signal.
>
> *Graphics* Tab – Akson creates, at the moment of loading the page, a 3D field composed of four scenes which in turn are composed of independent geometries. In this panel it is possible to control the

incident light of each one (for the graphic context, artificial lights and camera are necessary), camera aspect, zoom and field of view, and change the dimension of each group of geometries in the three axes independently (scaling methods) (Angel & Shreiner, 2013).

Post-Prod Tab – Both visual and sound, are found methods such as shaders – four pieces of GLSL code (Wolff, 2018), a custom scale/note engine for the *Lead* (ordered synthesizer note groups), and camera offsets (useful for multi-monitor/multi-machine future work), preset buttons and visual meters (frames per second, milliseconds of delay and performance memory). It is possible for the user to click on specific notes (from C2 to B5), clean and create his own relationship scale[3].

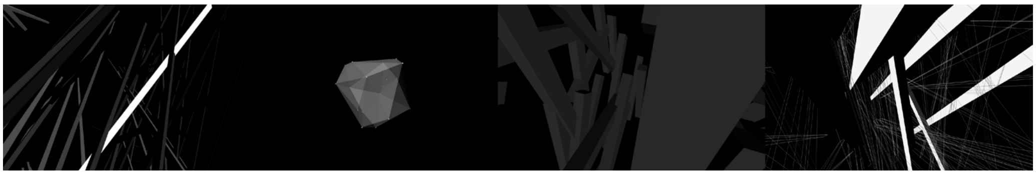


Figure 2: Screenshots of the four default graphical scenes of Akson.

3.3. LOGS DIALOG

The panel entitled *Logs Dialog* is a console that prints text of actions taken towards the environment – figure 4. It prints all methods of the interface done by the user, such as i.e. which note has been played, which slider value is to be modified, the action executed by a button, as well as the actions that another user may have in our environment[4]. It works as a monitoring system for the AV methods.

3.4. TOP BAR

The five mentioned panels have the possibility to be moved in the two-dimensional space so that the user may arrange them to their best taste (for future work, each panel can be used in different tabs of the browser). However there is a third panel considered as a *top bar* with five buttons that can be hidden but cannot be moved. From left to right, these execute: 1) a *modal* window with text about the nature of the project; 2) a dialog entitled *Save Settings*; 3) changes between the 3D scenes; 4) refreshes the web page; 5) a *modal* window with the interaction models.

Save Settings – A small dialog with the ability to record audio and save images. It consists of 5 buttons divided into two boxes called *Save Frame* and *Save Audio*. The first one has 3 buttons, the *New Page* (opens a new tab with a camera capture), *Save to Disk* (downloads a capture from the camera) and *Save 360* (it does a 360º capture from the same camera). The second has two buttons, *Start Audio* (as a record button) and Stop Audio (to end the recording that is running). There is also an audio object to play and download the recorded audio track – see figure 9.

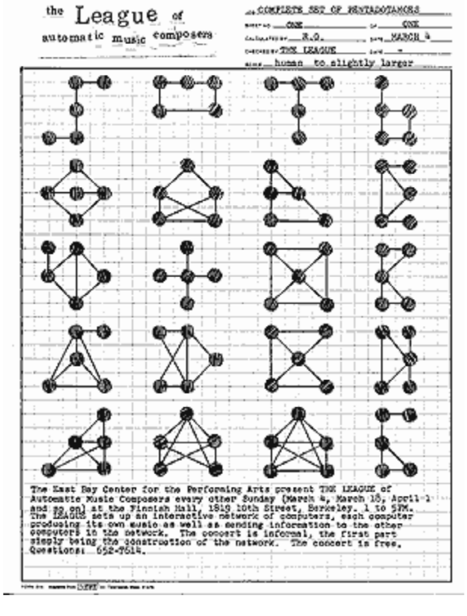
[3] It is also possible to apply presets of scales used around the world (Major; Minor; Melodic; Harmonic; Pentatonic (Major and Minor); Adonai Malakh; Hungarian; Hirajōshi; Ionian; Locrian). The scales are originally all set in C to facilitate a start pattern.

[4] The console also prints actions in the environment taken by other users who share the same space – which will be explained in more detail in the various interaction models.

https://doi.org/10.34632/jsta.2021.9448

4. INTERACTION MODELS

The moment an interface establishes a connection to the server and becomes an Akson instance, it has one of the interaction models. Given the focus of this research on networked artistic practices and given the structure of *server-client* (see recent proposal by Le, et al., 2020) connections, we developed interaction typologies based on Winkler's *Composing Interactive Music* (1998). The book's proposals have all been implemented from scratch and are based on the role performance members play – *Improvisation Model, The Conductor Model and The Chamber Music Model*[5]. The roles are considered from the four instruments described, specifying them as separate[6] (four are proposed using the *String Quartet* as an example). As mentioned in Arandas et al. (2019), the server controls the flow of information, but the interfaces establish the interaction matrices, each user – that is not considered *mobile device* – can instantly change the interaction model if they wish (see Matuszewski et al., 2019). This can be measured by the amount of information that passes through the server regardless of the interface, whether used by the performers or by audience. The connection of devices and people through a network, their application in performance and the fact that we consider information flows contained in communication routes that have barriers can be seen up to *The League of Automatic Composers* and their work in the 1970's (Barbosa, 2006). The aforementioned Winkler (1998) inspirations are transposed to Akson in the form of centralised connection models (Mills, 2019; Baran, 1964). They are entitled *descenter* model, *streamed* model and *allocate* model – figure 9. There is also the possibility to choose the *alone* model which is the absence of interaction (stops all incoming and outgoing connections, ending the collaboration), and the user can always switch dynamically between them.



the League of automatic music composers

COMPLETE SET OF PENTADOTAMORS
ONE ONE
R.G. MARCH 4
THE LEAGUE -
human to slightly larger

The East Bay Center for the Performing Arts present THE LEAGUE of Automatic Music Composers every other Sunday (March 4, March 18, April 1 and so on) at the Finnish Hall, 1819 10th Street, Berkeley. 1 to 5PM. The LEAGUE sets up an interactive network of computers, each computer producing its own music as well as sending information to the other computers in the network. The concert is informal, the first part simply being the construction of the network. The concert is free. Questions: 652-7614.

Figure 3 I Image of a flyer designed by Rich Gold from 1979 showing different network topologies between computers (referenced from Barbosa, 2006).

[5] The *Free Improvisation* model is also considered; however, the act of *improvising* is here regarded as possible in any of the models. The alone model takes this place by the degree of freedom offered.

[6] Given the openness to use the software in customised contexts, the limit of the instrument can be crossed. The various models only offer ways to differentiate them practically.

4.1. FIRST MODEL – *DESCENTER*

The first model of interaction – also the initial when the page is loaded – is entitled *descenter* and is based on *The Improvisation Model*, paradigm of the *Jazz Combo* (Winkler, 1998). It is a model where the user is by default in collaboration receiving and transmitting actions in the *Lead* (as triggers, equal to the click). All other properties of the environment are controlled by the user in question, and when someone enter the server, it is printed in the *Logs Dialog* – the color of the geometries also changes when a click is made, regardless of who did it. This was the first model to be developed and tested. "We are honoured to be in somebody's computer. You are very close to a person when you are on his desktop. I think the computer is a device to get into someone's mind." (Paesmans, 1995, as cited in Jana & Tribe, 2009)

4.2. SECOND MODEL – STREAMED

The *streamed* model allows all instances – which are in this model – to control anything in all others. Any small change in the interface will be sent to all the Akson instances that share this model. This method is based on *The Conductor Model*, paradigm of the *Symphony Orchestra* (Winkler, 1998) and aims to make every instance have the same control over each other. It is designed to be used with one computer and multiple *mobile devices* that do not have access to the controls and are just machines for spatialization. The development of this model derived from feedback obtained after a public performance done with the *descenter* model. All features of the environment can be streamed as a bidirectional event, so Akson can be used as a source of coordination on live performances with large numbers of distributed *mobile devices* or between various instances that can extend the methods to each other. "Concerning live performance, designers must not only consider awareness and communication between the musicians, but also their relationship with the audience." (Fencott, 2012)

4.3. THIRD MODEL – *ALLOCATE*

The *allocate* model allows the user to choose his role for performance. After selecting the 5th button mentioned in the *Top Bar*, the user is presented with a panel called *Allocate System* – figure 9 – where four options with the name of each instrument are present. This method is based on *The Chamber Music Model*, paradigm of the *String Quartet* (Winkler, 1998) and allows the user to take the position of only one instrument and control it, closing the transmission of the various methods other than this instrument. The other participants of Akson that share this model can have the same role or choose other one. This method was born from feedback obtained in the second public experiment and referenced by the laptop performance practice (i.e. the computer network music band *The Hub* (Boutwell, 2009) and Birmingham Ensemble for Electroacoustic Research (Akkermann, 2016).

> “When laptop performance interfaces move beyond simple one-to-one mappings, they present unique ensemble challenges, particularly in terms of the synchronization and sharing of musical material. Specialized improvisation environments for specific performances (e.g. Trueman, 2008) or ensembles (e.g. Rebelo and Renaud 2006) can help groups to negotiate these challenges and structure their collaboration.” (Freeman & Van Troyer, 2011)

5. PUBLIC EXPERIMENTS

This section presents the public experiments done with the Akson environment. The choices made in the development of the software were based on the feedback obtained, promoting a recursive way of doing research between the laboratory and the field (characterised as an inter-disciplinary mixed method research approach conducted ‘in the field’ that serves applied and theory building purposes (Reimann, 2011)). Each experiment informed the next model to be developed until all three were finalised.

5.1. FIRST EXPERIMENT: *CENTRO DE ALTO RENDIMENTO ARTÍSTICO*

The first experiment was held on February 19, 2019, in Centro de Alto Rendimento Artístico (CARA) located in *Real Vinícula*, home of the Matosinhos Jazz Orchestra (OJM), and the equipment used was provided by its Educational Service.

5.1.1. THE PERFORMANCE

The performance was composed of 6 people all with advanced musical knowledge and with experience in artistic performance. Participants knew that it was a public test of experimentation belonging to a research in collaborative AV systems and all had links to the academic environment. There were 9 computers, 9 sound devices, a mixer, a sound system, 3 mobile phones, an electric guitar and the WI-FI connection was relatively quick – in CARA’s network. The computers were of model *iMac* 2017 - 21.5 inch running OSX *High Sierra*, version 10.13.6 with a processor of 2.3GHz Intel Core, RAM memory of 8Gb 2133 MHz DDR5 and an *Intel Iris Plus Graphics* 640 card 1536Mb of 1920x1080. Each sound device was connected to each computer and going to the main mixer – of model Focusrite Scarlett 2i2 (2nd Gen using a sample rate of 96KHz with 24-Bit conversion and 2 pre-amplifiers. The connection was done by TRS terminal to a mixer *Mackie Micro Series 1402-VLZ* of 14 channels and then by XLR to a *Meyer Sound* system. There were two ultra-compact wide coverage loudspeakers of model UPM-1P at each corner of the room that squeezed all these channels to a stereo. Alongside the computers, were 3 mobile phones connected to the system that were not flowing audio to the mixer but served as resonating devices in the room. They were all *Android* devices, two were running *EMUI* 5.0.3 and the other 9.0 *Pie* operating system. One of the

participants that brought his personal guitar of model *Ibanez* RG2570 EX Prestige was playing alongside the other participants and was processing the sound using Bias FX from *Positive Grid* in his own laptop.
There were more devices than participants, thus offering some versatility, and the internet connection showed no problem to solve on the OJM network, the machines were connected, and no latency was noticed. A speed test was carried out before starting to show that ports 8000 and 5000 were open (the ones needed), <4.20Ms latency, <6.26Ms jitter, a download rate of <60.77Mbp/s and a little volatile upload speed of <92.00Mbp/s. The experiment started at 15:00 and took 15 minutes until everyone stopped playing with the system.

5.1.2. THIRD MODEL – *ALLOCATE*

The participants after the involvement demonstrated some similar opinions regarding Akson and its use. There was a clear general desire to increase the relationship between musicians while playing, and a more visible form of contact between them.

> The participants agreed that it would enhance the performance if they had some kind of visual display of what the others were doing in the user interface, while playing.
>
> The participants also felt the need to have a main master control to shut down the system at the same time.
>
> There was a general opinion that there should be “more clarity in sound propagation between machines” when that happens and why.

This data informed the development of the next model and therefore the next experiment. Before the subjects started playing, they knew it was a networked collaborative system, but they actually didn’t really know how to interact with the interface and how to control it. To provide an easy learning curve, was presented an image moments before starting that had some instructions on basic controls such as how to change visual scene and how to trigger the GUI.

5.2. SECOND EXPERIMENT: *GNRATION*

The second experiment was held on April 27, 2019, in gnration gallery located in Braga during its annual *open day*. It was done under the context of Braga Media Arts (BMA) with equipment provided by its Educational Service.

5.2.1 THE INSTALLATION

The second public experiment was presented as an experimentative installation. It allowed the use of Akson by interacting with computers present as well as by establishing a connection using mobile devices. It was presented from 11:00-13:00 and 14:00-17:00 to any type of participant (i.e. regardless of age or previous knowledge). This experience has been much more adhered to than the first, in terms of participants – also thanks to the participatory nature of the installation. It was composed of 5 computers, sound system and video projection. We had the central processing machine which was of model *Mac Pro* (late 2013) with a processor 3.7GHz Quad-Core *Intel Xenon* E5, RAM memory of 12Gb 1866MHz DDR3 ECC, graphics card AMD *FirePro* D3000 2048Mb running OSX *El Capitan* 10.11.6. The projection and sound system were from this machine – the main HDMI connected to an *Optoma* GT1080E, a 1080p resolution projector with 3000 ANSI Lumens. The sound system chain started with an USB 3.0 connection to a *Behringer* U-Phoria UM2, an audio interface of 2 entrances and 2 way outs with 48KHz on 16-Bit. It connected by TRS terminal to an *Alto Professional* 350 Watt MIXPACK Express PA. This sound system came with two 10 inch speakers, a built-in mixer with seven-band graphic EQ and integrated Alesis DSP with 24-bit effects that weren't used. Alongside the main processing machine there were 3 13-inch *MacBook* Air running OSX *Mojave* version 10.14 with a processor 1.8GHz Intel Core i5, RAM memory of 8Gb 1600MHz DDR3, a graphics card Intel HD Graphics 6000 1536MB. And one *MacBook Pro* running OSX *Sierra* version 10.12.6 with a processor 2.2GHz Intel Core i7, RAM memory of 16Gb 1600 MHz DDR3 with a graphics card *Intel Iris Pro* 1536Mb. These served as distributed devices in space and were displayed around a table that was in the room. The video was pointing to a wall without posterior mapping but carefully calibrated and the sound system was equalised in accordance with the room so the other installations wouldn't be disturbed.

The preliminary tests that were done in the used WI-FI network were very satisfactory given the volatile nature of the network in a public gallery. The speed test showed a latency of <11Ms, a jitter of <3Ms, a download speed of <58.51Mbp/s and an upload speed of <56.95Mbp/s. It also showed that the port 8000 was opened and 5000 closed.

5.2.2 THE PARTICIPANTS

In the interaction model used in this experiment – *streamed* – everyone that connected to the system this day was by default in this model. Allowing the users to build an experience using Akson as a *single* distributed interface and given the fact that all actions echoed throughout all connected devices, participants got a clear notion of what other people were doing. There was no clear opinion that all participants shared, however, some behaviours were consistent throughout the day.

> Most people used the equipment provided by the installation and not their own mobile devices to interact with the system throughout the experience.

Apart from artists that demonstrated knowledge about digital audio processing, asking questions about the system and entering into discussion, the participants who spent most time interacting were children and young people up to 30 years old.

Overall, while interacting with Akson, people were more focused on the 3D scenes until they discovered the capabilities of the GUI.

Even if the system had a dedicated screen to show attendees an image with key controls to interact with the keyboard, help was always given to anyone entering the installation space. Most people also interacted verbally when they came across the setup and some participants also used their devices.

5.3 EXPERIMENT IN OPENFIELD

The third experiment was done on May 6, 2019, in Openfield Creativelab, located in the center of Porto, home of a media art collective. It was performed with equipment provided by the Atelier.

5.3.1 THE PERFORMANCE

The third public experiment was conducted between the main author of Akson and a member of the collective, as a performance. The nature of this experiment resides on the networked laptop performance (Hugill, 2005; Schooler & Touch, 1993) experimental tradition, done without any previous rehearsal and exploring the *allocate* model (two performers, one for the *Lead* and the other to the *Background*). The system was composed of two computers, a sound system and a projector. This experiment took 10 minutes and started at 10:15 in the morning with 6 researchers as public. One computer was a *MacBook Pro* (late 2013) with a processor of 2.7GHz Intel Core i5, 13.3-inch, RAM memory of 8Gb 1867MHz DDR3 and a graphics card *Intel Iris Graphics* 6100/1536Mb running OSX *Mojave* 10.14. The second computer was of model *MacBook Pro* (late 2015) with a processor of 2.2GHz Intel Core i7, RAM memory of 16GB 1600MHz DDR3 and a graphics card *Intel Iris Pro* 1536Mb running OSX *Sierra* 10.12.6. The first computer was connected to a Scarlett 2i4 (2nd Gen), the second computer was connected to a *Roland Quad-Capture*, and these two interfaces connected to the main mixer of model Alto L-12, a twelve channel legacy mixer with integrated DSP of 24-bit that connected to 2 *Tannoy Reveal* 502 speakers. The second computer connected to a projector EPSON EB-S05 of 1080p with 3100 ANSI Lumens by HDMI.

Both the equipment configuration and the performance interaction definition were resolved before it happened. It was defined that the two performers would have two distinct functions, the *Lead* and the *Background*. The computer connected to the projection could also control the visuals. The tests were done in the WI-FI network one hour before the performance and were satisfactory enough so we didn’t have to setup our own network for speed improvement. We ran speed tests that demonstrated a latency of <13Ms and <4Ms of jitter. The download speed was of <45.00Mbp/s and upload speed of <53.00Mbp/s. We also had the port 5000 opened to serve Akson.

5.3.2 THE PARTICIPANTS

The participants in this experiment were involved with the performing arts and with the academia as researchers. They did not know the method of interaction of the two artists, nor was it explained how the performance would take place. They were only told that if they wanted to, they could connect to the system through the official link – before starting the performance an image with information was projected on the wall with the address. There was a clear agreement on the various topics mentioned.

> Participants liked to know more about Akson and how it works as a system, not just the content it created.
>
> They wanted to know exactly what the performers were doing, separately.
>
> They agreed that, given the aesthetics of the sound and the graphics load, the performance should take more than 10 minutes.

6. DISCUSSION

This article tries to report and discuss the technical and conceptual foundations of an AV environment done in the context of Braga Media Arts. Using Engelbart's (1962) vision towards technology, considering an *increased human intellect* and Stelarc's (Atzori & Woolford, 1995) extended-body analogy, Akson allows the study of the augmentation of the human gesture by the network of visual and sound interfaces. It focuses on the exact moment where the action of a human is amplified by various devices around the world and transmitted to the other humans present (see Colomina, 2001) and an essay on image spatialization using multiple screens for big audiences). An aprioristic desire is applied to merge the visual and sound modality as well as to use the interface to mediate the interaction (as proposed by the early work of Greenberg (1997). An AV relationship is found, even though not compromised as *language* i.e. Kandinsky (Ione and Tyler 2003) or Quayola (Panourgia et al. 2018). Previous research on network performance is used, as well as the types of interactions that humans create. There are three *types* of pre-established interactions, but an openness is maintained to proposals for new paradigms, such as automatic reorganisation and custom programming. The devices are extended to the audience as well as to the gallery space. The connection maintained with the network and with the interface eventually brings to light new forms of relationships between humans. As a result of academic research, Akson allows to investigate areas such as: digital networks and the transmission of AV content, relations, and interactions in the performative space, to develop new AV languages, to spread a certain performance around the world, to promote inclusion in the performative space, to explore levels of freedom (relating to the type of gesture in question), as well as to compose audiovisually.

REFERENCES

Arandas, L., Gomes, J. & Penha, R. (2019). Towards Large Scale Artistic Practice with Web Technologies. *Proceedings of the Web Audio Conference.*

Atzori, P., & Woolford, K. (1995). Extended-Body: Interview with Stelarc. *CTheory*. https://journals.uvic.ca/index.php/ctheory/article/view/14658/5526

Akkermann, M. (2016). Performing Computer Network Music. Well-known challenges and new possibilities. *Proceedings of the International Computer Music Conference*, 395-401.

Angel, E., & Shreiner, D. (2014). *Interactive Computer Graphics with* WebGL. Addison-Wesley Professional.

Arkko, J. (2020). The influence of internet architecture on centralised versus distributed internet services. *Journal of Cyber Policy, 5*(1), 30-45. https://doi.org/10.1080/23738871.2020.1740753

Baran, P. (1964). *On Distributed Communications Networks. IEEE Transactions on Communications, 12*(1), 1–9. https://doi.org/10.1109/tcom.1964.1088883

Barbosa, Á. (2006). *Computer-supported Cooperative Work for Music Application.* Universitat Pompeu Fabra.

Boutwell, B. (2009). The League of Automatic Music Composers, 1978–1983. With John Bischoff, Jim Horton, Tim Perkis, David Behrman, Paul DeMarinis, and Rich Gold. New World Records 80671-2, 2007. *Journal of the Society for American Music, 3*(2), 263-264. https://doi.org/10.1017/S1752196309091147

Colomina, B. (2001). Enclosed by images: the Eameses' multimedia architecture. *Grey Room*, 6-29.

Clément, A., Ribeiro, F., Rodrigues, R., & Penha, R. (2016). Bridging the gap between performers and the audience using networked smartphones: the a. bel system. *Proceedings of the International Conference on Live Interfaces.*

Deutsch, R, and Deutsch, L. J. (1978). ADSR envelope generator. U.S. Patent No. 4,079,650. 21 Mar.

Engelbart, D. C. (1962). *Augmenting human intellect: A conceptual framework.* Stanford Research Institute.

Fencott, R. (2012). *Computer Musicking: Designing for Collaborative Digital Musical Interaction* (Doctoral dissertation, Queen Mary University of London).

Freeman, J., & Troyer, A. V. (2011). Collaborative Textual Improvisation in a Laptop Ensemble. *Computer Music Journal, 35*(2), 8–21.

Gray, J. (2009). Google chrome: The making of a cross-platform browser. *Linux Journal, 2009*(185), 1.

Greenberg, S. (1997). Collaborative interfaces for the web. *Human factors and web development, 18*, 241-254.

Hugill, A. (2005) Internet music: An introduction. *Contemporary Music Review Journal, 24*(6), 429-437. https://doi.org/10.1080/07494460500296094

Ione, A., & Tyler, C. (2003). Was Kandinsky a synesthete?. *Journal of the History of the Neurosciences, 12*(2), 223-226.

Jana, R., & Tribe, M. (2009). *New media art*. Taschen America, LLC.

Le, C. D., Calcagno, M., Hamaker, J., Chambers, R., Plumpe, M., & Wilson, T. (2020). *U.S. Patent No. 10,749,989.* Washington, DC: U.S. Patent and Trademark Office.

Lyu, S. (2021). High-Performance Web Frontend Using WebAssembly. *Practical Rust Web Projects* (pp. 193-249). Apress.

Magnusson, T. (2009). Of epistemic tools: Musical instruments as cognitive extensions. *Organised Sound, 14*(2), 168-176. https://doi.org/10.1017/S1355771809000272

Manovich, L. (2001). *The Language of New Media*. The MIT Press.

Mardan, A. (2018). *Deploying Node.js* Apps. Practical Node.js (pp. 365-388). Apress.

Matuszewski, B., Schnell, N., & Bevilacqua, F. (2019). Interaction Topologies in Mobile-Based Situated Networked Music Systems. *Wireless Communications and Mobile Computing, 2019*, 1–9. https://doi.org/10.1155/2019/9142490

Mills, R. (2019). Towards a Theory of Tele-Improvisatory Collaboration. *Tele-Improvisation: Intercultural Interaction in the Online Global Music Jam Session* (pp. 119-141). Springer, Cham.

Mowery, K., Bogenreif, D., Yilek, S., & Shacham, H. (2011). Fingerprinting information in JavaScript implementations. *Proceedings of W2SP*, *2*(11).

Panourgia, E. I., Wheelaghan, F., & Yang, X. (2018). Digital interactions. *Airea: Arts and Interdisciplinary Research*, (1), 3-10. https://doi.org/10.2218/airea.2732

Pfeiffer, S., & Green, T. (2015). Manipulating Audio Through the Web Audio API. In *Beginning HTML5 Media* (pp. 227-276). Apress.

Rai, R. (2013). Socket. *IO Real-time Web Application Development*. Packt Publishing Ltd.

Reimann, P. (2011). Design-based research. *Methodological choice and design* (pp. 37-50). Springer.

Renaud, A., & Rebelo, P. (2006). Network performance: Strategies and applications. *NIME Conference Paper.*

Schooler, E. M., & Touch, J. (1993). Distributed music: a foray into networked performance. *International Network Music Festival, Santa Monica,* CA.

Surianarayanan, C., & Chelliah, P. R. (2019). Fundamentals of Cloud Computing. *Essentials of Cloud Computing* (pp. 33-67). Springer.

Teixeira, I. (2018). *O Espaço Aumentado na Experiência Artística: Ecrãs, Sistemas e Contextos Físicos*.[PhD Dissertation]

Wilson, E., & Thompson, A. (2020). Live coding the code: an environment for 'meta' live code performance. *International Conference on Live Coding*.

Winkler, T. (1998). Composing interactive music. *Techniques and ideas using Max.*

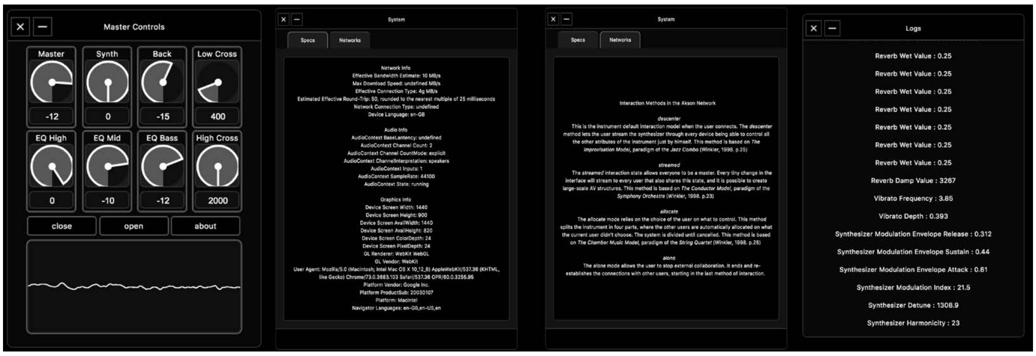
Wolff, D. (2018). *OpenGL 4 Shading Language Cookbook: Build high-quality, real-time 3D graphics with OpenGL 4.6, GLSL 4.6 and C++* 17. Packt Publishing Ltd.

Yamazaki, T. Electronic device with a master fader and a plurality of slave faders. U.S. Patent No. 9,215,021. 15 Dec. 2015.

ACKOWLEDGEMENTS

The research leading to these results was financially supported by the Portuguese Foundation for Science and Technology (FCT), through the individual research grant 2020.07619.BD and by the project “Experimentation in music in Portuguese culture: History, contexts and practices in the 20th and 21st centuries”, co-financed by the European Union, through the Operational Programme Competitiveness and Internationalization, in its ERDF component, and by national funds, through the Portuguese Foundation for Science and Technology.

FIGURES



Master: -12
Synth: 0
Back: -15
Low Cross: 400
EQ High: 0
EQ Mid: -10
EQ Bass: -12
High Cross: 2000

Network Info
Effective Bandwidth Estimate: 10 MB/s
Max Download Speed: undefined MB/s
Effective Connection Type: 4g MB/s
Estimated Effective Round-Trip: 50, rounded to the nearest multiple of 25 milliseconds
Network Connection Type: undefined
Device Language: en-GB

Audio Info
AudioContext BaseLantency: undefined
AudioContext Channel Count: 2
AudioContext Channel CountMode: explicit
AudioContext ChannelInterpretation: speakers
AudioContext Inputs: 1
AudioContext SampleRate: 44100
AudioContext State: running

Graphics Info
Device Screen Width: 1440
Device Screen Height: 900
Device Screen AvailWidth: 1440
Device Screen AvailHeight: 820
Device Screen ColorDepth: 24
Device Screen PixelDepth: 24
GL Renderer: WebKit WebGL
GL Vendor: WebKit
User Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 OPR/60.0.3255.95
Platform Vendor: Google Inc.
Platform ProductSub: 20030107
Platform: MacIntel
Navigator Languages: en-GB,en-US,en

Interaction Methods in the Akson Network

descenter

This is the instrument default interaction model when the user connects. The *descenter* method lets the user stream the synthesizer through every device being able to control all the other atributes of the instrument just by himself. This method is based on *The Improvisation Model*, paradigm of the *Jazz Combo* (Winkler, 1998. p.25)

streamed

The *streamed* interaction state allows everyone to be a master. Every tiny change in the interface will stream to every user that also shares this state, and it is possible to create large-scale AV structures. This method is based on *The Conductor Model*, paradigm of the *Symphony Orchestra* (Winkler, 1998. p.23)

allocate

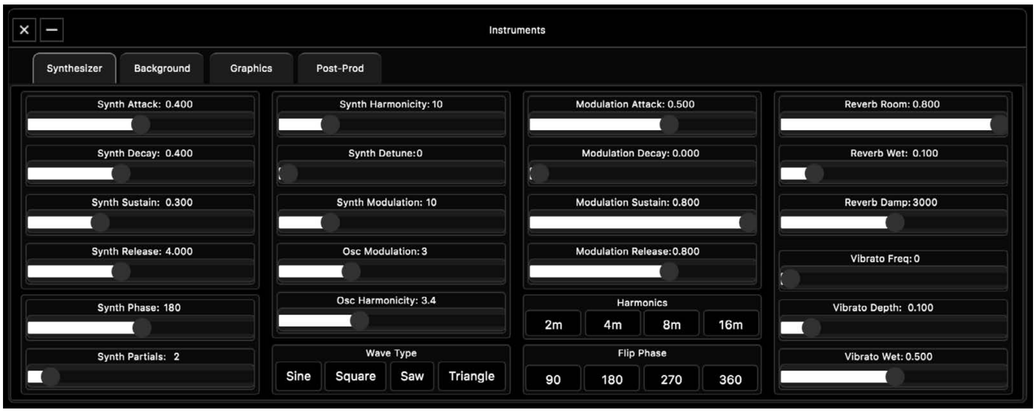
The *allocate* mode relies on the choice of the user on what to control. This method splits the instrument in four parts, where the other users are automatically allocated on what the current user didn't choose. The system is divided until cancelled. This method is based on *The Chamber Music Model*, paradigm of the *String Quartet* (Winkler, 1998. p.25)

alone

The *alone* mode allows the user to stop external collaboration. It ends and re-establishes the connections with other users, starting in the last method of interaction.

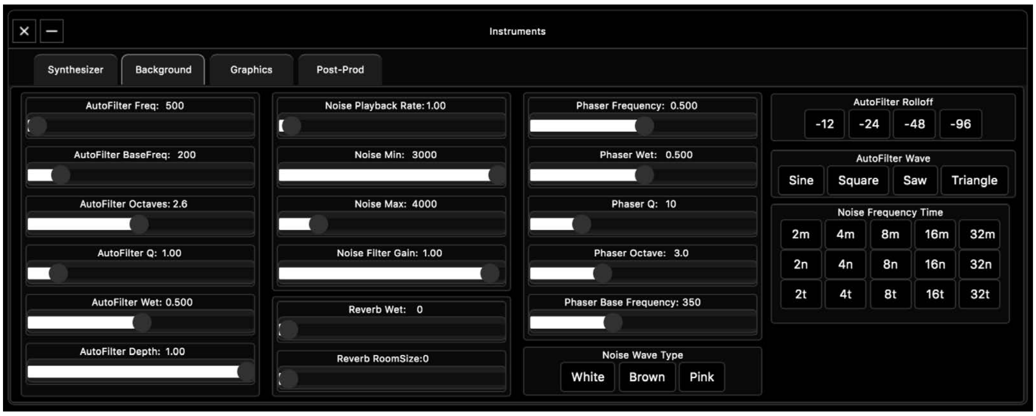
```
Reverb Wet Value : 0.25
Reverb Wet Value : 0.25
Reverb Wet Value : 0.25
Reverb Wet Value : 0.25
Reverb Wet Value : 0.25
Reverb Wet Value : 0.25
Reverb Wet Value : 0.25
Reverb Damp Value : 3267
Vibrato Frequency : 3.85
Vibrato Depth : 0.393
Synthesizer Modulation Envelope Release : 0.312
Synthesizer Modulation Envelope Sustain : 0.44
Synthesizer Modulation Envelope Attack : 0.61
Synthesizer Modulation Index : 21.5
Synthesizer Detune : 1308.9
Synthesizer Harmonicity : 23
```

Figure 4: GUI Elements of Akson: 1) Master Dialog; 2) System Dialog; 3) Logs Dialog – from left to right.



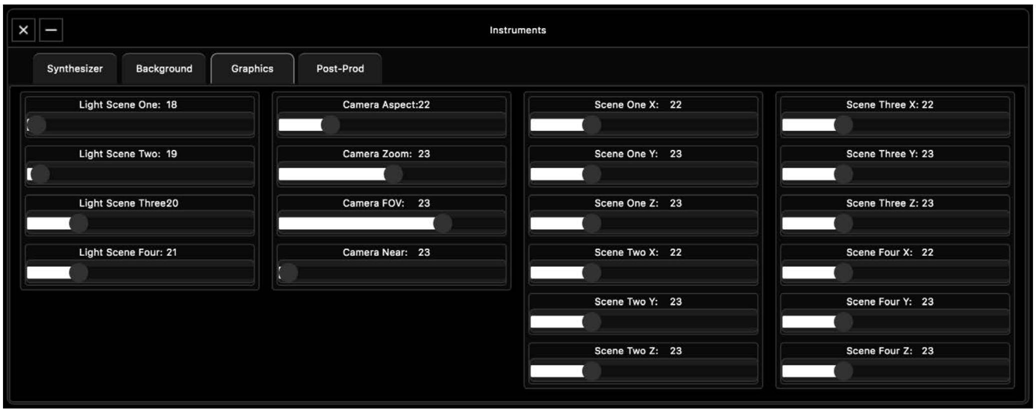
Synth Attack: 0.400
Synth Decay: 0.400
Synth Sustain: 0.300
Synth Release: 4.000
Synth Phase: 180
Synth Partials: 2
Synth Harmonicity: 10
Synth Detune:0
Synth Modulation: 10
Osc Modulation: 3
Osc Harmonicity: 3.4
Modulation Attack: 0.500
Modulation Decay: 0.000
Modulation Sustain: 0.800
Modulation Release: 0.800
Reverb Room: 0.800
Reverb Wet: 0.100
Reverb Damp: 3000
Vibrato Freq: 0
Vibrato Depth: 0.100
Vibrato Wet: 0.500

Figure 5: GUI Elements of Akson: 4) Instruments Dialog – Lead tab.



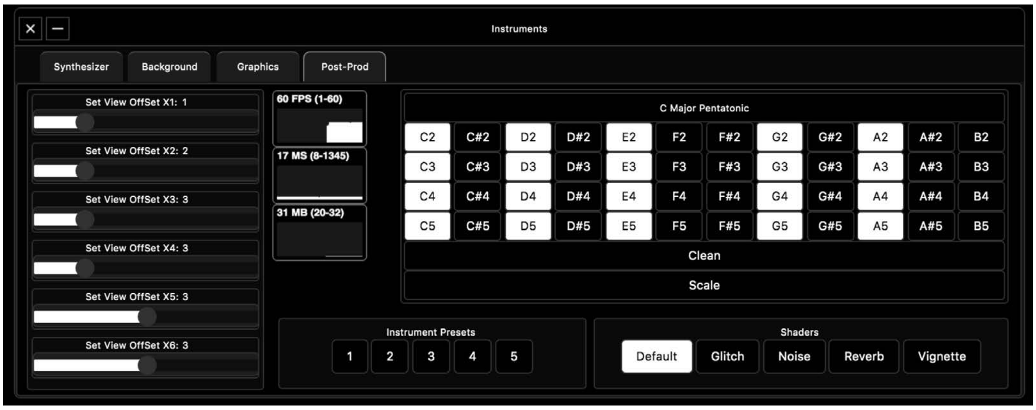
AutoFilter Freq: 500
AutoFilter BaseFreq: 200
AutoFilter Octaves: 2.6
AutoFilter Q: 1.00
AutoFilter Wet: 0.500
AutoFilter Depth: 1.00
Noise Playback Rate: 1.00
Noise Min: 3000
Noise Max: 4000
Noise Filter Gain: 1.00
Reverb Wet: 0
Reverb RoomSize:0
Phaser Frequency: 0.500
Phaser Wet: 0.500
Phaser Q: 10
Phaser Octave: 3.0
Phaser Base Frequency: 350

Figure 6: GUI Elements of Akson: 5) Instruments Dialog – Background tab.



Light Scene One: 18
Light Scene Two: 19
Light Scene Three20
Light Scene Four: 21
Camera Aspect:22
Camera Zoom: 23
Camera FOV: 23
Camera Near: 23
Scene One X: 22
Scene One Y: 23
Scene One Z: 23
Scene Two X: 22
Scene Two Y: 23
Scene Two Z: 23
Scene Three X: 22
Scene Three Y: 23
Scene Three Z: 23
Scene Four X: 22
Scene Four Y: 23
Scene Four Z: 23

Figure 7: GUI Elements of Akson: 6) Instruments Dialog – Graphics tab.



Set View OffSet X1: 1
Set View OffSet X2: 2
Set View OffSet X3: 3
Set View OffSet X4: 3
Set View OffSet X5: 3
Set View OffSet X6: 3
60 FPS (1-60)
17 MS (8-1345)
31 MB (20-32)
C Major Pentatonic

Figure 8: GUI Elements of Akson: 7) Instruments Dialog – Post-Prod tab.

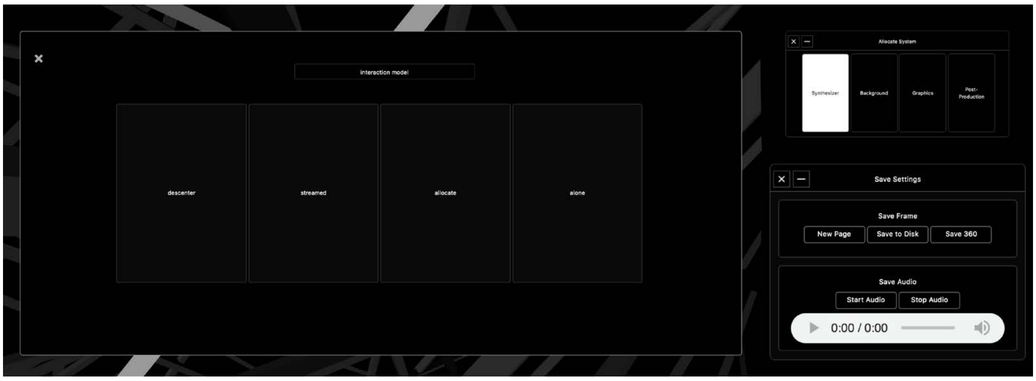


Figure 9: GUI Elements of Akson: 8) Interaction modal; 9) Allocate System; 10) Save Settings dialog.

Article received on 29/07/2020 and accepted on 22/07/2021.