Developing spaceJam: *The New Sound Spatialization Tool for an Artist and Novice*

By Adriana Madden

Submitted in partial fulfilment of the requirements for the Master of Music in Music Technology in the department of Music and Performing Arts Professions in the Steinhardt School of Culture, Education, and Human Development New York University

Advisor: Dr. R. Luke DuBois
November 25th, 2014
# Table of Contents

Abstract 4  
Acknowledgments 5  

1.0 Introduction 6  
  1.1 Context 6  
  1.2 Motivation 6  
  1.3 Goals 7  

2.0 Background 8  
  2.1 HCI 8  
    2.1.1 HCI and Multimedia 8  
    2.1.2 User Centered Design 9  
  2.2 Artistic Spatialization Practice 10  
    2.2.1 Spatialization and Artistic Authorship 11  
    2.2.2 Overview of Artistic Spatialization Practice 12  
      2.2.2.1 Artist vs. Researcher 12  
      2.2.2.2 Artistic Requirements for Spatialization Tools 13  
  2.3 Technical Spatialization Practice 19  
    2.3.1 Overview of Spatialization Techniques 19  
    2.3.2 Rendering Concepts 22  
      2.3.2.1 Stereo Sound Panning 22  
      2.3.2.2 Ambisonics 23  
      2.3.2.3 VBAP 25  
      2.3.2.4 DBAP 27  
  2.4 Review of Spatialization Products 29  
    2.4.1 Jamoma Modular 30  
    2.4.2 VBAP in Max 32  
    2.4.3 Spatium 34  
      2.4.3.1 Spatium Panning 34  
      2.4.3.2 Spatium Ambi 35
3.0 spaceJam Implementation
   3.1 UX
   3.2 Technical Implementation
      3.2.1 Spatialization Algorithm
      3.2.2 Implementation
   3.3 Other Notable Implementation Decisions
      3.3.1 Non-Modular Interface Design
      3.3.2 Multi-Frame View
      3.3.3 User Defined Levels
      3.3.4 Affordance

4.0 Testing
   4.1 Participants
   4.2 Testing Environment
   4.3 Procedure
      4.3.1 Task 1
      4.3.2 Task 2
   4.4 Results

5.0 Discussion and Future Work
   5.0.1 Learnability
   5.0.2 Performance Effectiveness
   5.0.3 Flexibility
   5.0.4 Error Tolerance and System Integrity
   5.0.5 User Satisfaction

6.0 Conclusion
Abstract

Many spatial sound artists are currently struggling due to a lack of appropriate media authoring products. In active response to this demand, this thesis documents the development of spaceJam: a novel media authoring application dedicated to sound spatialization and optimized for use within the artistic community. A critical design criterion for spaceJam is usability, which has lead to a tool that is accessible to artists, and spatial sound novices alike. A spaceJam prototype was developed and assessed during the preliminary testing stage. This testing conveyed the products potential, and highlighted areas for further development.
Acknowledgments

First, I’d like to thank Luke DuBois for being such a fun, crazy and inspiring advisor. I appreciate all the early morning discussions, late night programming sessions, and necessary chipotle breaks. I couldn’t have done this without your guidance.

To my MTech crew, by virtue of the fact that you’re laughing right now, you know who you are. From New York to Kentucky, Pennsylvania, Vermont, Florida, Colombia, Ecuador, Australia and back – thanks for the memories we’ve made these past few years, you little chestnuts.

And finally to Mum and Paps, I can’t thank you enough for the past two and a half years. Thank you xx
1.0 Introduction

1.1 Context

Music students in western music culture have been encouraged to understand sound(s) within the confines of the fundamental elements of music: rhythm, dynamics, melody, tone-colour, harmony, texture and structure. To consider sound purely within these boundaries ignores many other notable methods to understand and experience sound, including the spatial dimensions of sound.

When a human perceives sound, our biological makeup allows us to decipher the spatial position of the sonic event to an exceptional degree. History may inform us that composers have toyed with our keen localization skills, and the role of space in composition for centuries. In medieval Christian antiphonal music, composers physically spatialized choirs during sacred performances to emphasize a relationship between time, pitch and location. The technological revolution of the 19th century allowed stunned audiences to experience the first recording enhanced with spatial features at Clement Ader’s 1881 Paris Exhibition of Electricity. And in 1958, Edgar Verese amazed the world with the premier of his multimedia performance of *Poème électronique* that incorporated spatial sound technology for virtual reality.

1.2 Motivations

Interest in spatial audio continues to evolve within the research and artistic communities alike, and particularly in the past decade, interest has spiked. Reasons for this may be attributed to ‘increased computing power [that] has also led to the production of workstations that can support data bandwidth required for multiple audio channels. These factors have led to an
interest in sound spatialization techniques, providing enormous potential for accelerated development in the field.\textsuperscript{1} It is due to such technological advancement that spatial audio researches have been able to develop spatialization rendering concepts and tools. This progress can be seen in techniques such as: VBAP, VicMic, Wavefield Synthesis and Ambisonics. While these show tremendous growth in conceptual and algorithmic development, the tools providing accessibility to the techniques are not suitable for conditions outside the research environment, and/or are inappropriate for those without a tertiary education in sound engineering.

Sound spatialization has become an important concept and technique within the artistic community. However, there is no ‘go-to’ commercial product for this artistic practice. This means spatial sound artists are left to exploit inappropriate and/or incomprehensible tools developed by audio researchers. As these tools often alienate many, and artists are forced to use unsuitable, yet familiar methods for sound spatialization. This project hypothesizes that the artistic community will benefit greatly from a sound spatialization tool developed in light of their specific requirements.

1.3 Goals

This paper documents the development of spaceJam: a sound spatialization tool directed at use within the artistic community that allows one to pan multichannel audio within a 3D space. To ensure usability for the target audience, spaceJam will incorporate a novel media-authoring interface, and only incorporate features and requirements that are viable for the artistic community.

\textsuperscript{1} Malham, D, Myatt, A, ‘3-D Sound Spatialization using Ambisonic Techniques’ \textit{Computer Music Journal} Vol. 19, No. 4 (Winter, 1995), pp. 58-70
2.0 Background

spaceJam will be an interactive multimedia application with a keen focus on usability for the target audience. To guarantee the appropriate development considerations are applied, a brief understanding of Human Computer Interaction (HCI) for multimedia software will be discussed here.

2.1.1 HCI and Multimedia

’HCI is a science of design.’

The term multimedia refers to any computer based software or interactive application that combines a complex interaction of text, color, graphical images, animation, audio, and/or full-motion video within a single application (Hall, 1996; McKerlie & Preece, 1993; Northrup, 1995; Tolhurst, 1995). The inherent architecture of interactive multimedia systems depends on a fluid engagement between human users and computer software. This has forced multimedia developers to consider HCI. Carrol (2006) defines the HCI as:

‘...concerned both with understanding how people make use of devices and systems that incorporate computation, and with designing new devices and systems that enhance human performance and experience.’

spaceJam will be heavily concerned with usability, a branch of HCI that is concerned with the development of computer systems that are easy to learn and easy to use (Preece 1994.)

---

International Standards resonate this opinion: ‘Usability: the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use’ (ISO/DIS 9241-11; European Usability Support Centres).

A focus on usability has been enforced, as it is a vital determining factor for the success of any computer system or service (Smith & Mayes 1996). Nils (2011) also recommends that sound spatialization products for artists should focus on ‘low leaning curves’ and ‘good usability’ to ‘lower the entry barriers for artists.’ This opinion is in reaction to the numerous sound spatialization tools that show limitations in regard to user experience (UX,) due to inefficient HCI consideration. Unlike the continued development of rendering concepts conveyed in new tools, interface usability progression is minimal, ‘surprisingly, the interfaces for controlling these systems have remained generally unchanged.’ For those educated within the field of sound technology or alike, this aesthetic subset may merely be superficial. However, for the broader population of potential users, this weak link may be an unnecessary catalyst for user alienation and/or user failure.

2.1.2 User-Centered Design

To ensure usability for the target demographic, the user-centered design (UCD) process will be followed during development. UCD places most importance on the end user’s requirements during the development process, as to specifically understand and address their needs (Morariu, 1988; Rubin, 1994).’ UCD advocates three principals:

---

1) Researching and understanding user-end needs for design
2) Designing for usability
3) Iterative testing

These UCD principals will be realized during development of spaceJam in the following ways:
1) Researching and understanding user-end needs for design and implementation
   Sufficient research discovering the needs of the artistic community will be conducted.

2) Designing for usability
   Development and design of spaceJam will directly reflect the findings of principal 1.

3) Iterative testing
   Testing will be conducted; iterations will be time permitting.

---

2.2 Artistic Spatialization Practice

“What is an author?”

Sound spatialization tools are currently available, but many are not suitable for use outside the research environment. This author theorizes that sound spatialization tools for artistic use have not developed because people have failed to assign correct artistic authorship to the spatialization practice.

---

2.2.1 Spatialization and Artistic Authorship

The modern notion of artistic authorship possesses a socially, culturally and politically undetermined, cloudy definition. Ronald Barthes argues that text doesn’t reflect a single ‘message’ of the author, rather ‘text is a tissue of citations, resulting from the thousand sources of culture.’ 7 This invites the concept that artistic authorship is a community-based endeavor, heavily fueled by cultural practice; a collaboratory web. The uncertainty of supreme authorship, and perhaps the substantiation that authorship is a process of collaboration, has only been provoked during the digital age.

‘By examining the role of technology in buttressing the “author-function”, one is immediately reminded of the ways in which music is never its single, autonomous product, but, as Nicholas Cook reminds us, always a “co-product” which requires “mediation” – be it through live performance, media-storage devices or technologies of re-presentation. 8

Sound spatialization holds an increasingly important role within the artistic process of music creation. This is evident in the many different fields the concept has touched: virtual reality, multi-media computing, films, videos, computer games, and more. Even if one is to spatialize sound that is not of their own creation, their premeditated and deliberate placement of the sound within a physical space impacts the audience’s experience of said sound, thus imposing an authorial fingerprint upon the artistic outcome. In light of this, an appropriate sound spatialization tool with a novel media-authoring interface is needed to encourage and aid this artistic practice.

2.2.2 Overview of Artistic Spatialization Practice

Sound spatialization applications currently reflect great progress in algorithmic and sound quality development. However, such tools are often created for research-based conditions, enforcing unnecessary accessibility constraints for the larger community. spaceJam aims to bring this technology outside the lab, and to the artistic community. To help secure usability within this target group, the UCD process has been employed. To meet with UCD principal number 1, an investigation into the needs of spatial sound artists has been performed. Firstly, an insight into why artists and researchers require different appropriations of a sound spatialization tool will be discussed.

2.2.2.1 The Artist vs. the Researcher

“The spatialization equipment and technology have become readily available, but the users haven’t caught up” - Natasha Barrett

The sound artist and the audio researcher may work within the same field, yet they inherently perform different roles within it. The idealized collaboration of the audio researcher and the artists would certainly allow for the development of any successful artistic audio tool. However, according to Blesser and Satler (2008), this ‘is an ideal based on theory, not practice’ resulting from difference in motivation between artists and researchers. ‘Audio engineers, typically working within budget constraints for commercial firms, [and] rarely have a mandate to design an artistically innovative system.’ This difference in motivation is often reflected in interface

---

design and user experience (UX), as researchers continue to develop tools that do not resonate within the artistic community.

‘Limited knowledge often prevents both groups from achieving their personal goals. The quality of the art of virtual space depends on understanding the properties of those tools available. The history of virtual spaces is therefore the story of an evolving relationship between sophisticated audio engineers, creating spatial tools, and impatient artists, incorporating such tools long before they are fully refined.’ (Blesser and Satler, 2008)

Like all technical tools created for artistic purposes, inspiration from both the research and the artistic world will lead to the most fruitful outcome. ‘To benefit from varying viewpoints, individuals involved in artistic practice and those involved in theoretical or applied research need to engage in regular dialogue…we need to understand this lack of coherence between development and creative musical application.’ (Nils, 2011) Therefore, spaceJam will attempt to provide a spatialization tool that is of the same audio quality standard as many research products, however will also provide a UX and incorporate features and requirements that resonate within the artistic community.

2.2.2.2 Artistic Requirements for Spatialization Tools

As spaceJam intends to provide a successful UX, and incorporate appropriate features and requirements, an understanding of the artist’s needs and concerns is necessary. Such needs are directly addressed by Peter Nils (2011) in his article ‘Current Technologies and Compositional Practices for Spatialization: A Qualitative and Quantitative Analysis.’ This research provides
pivotal insight into the current challenges faced by spatial sound artists’, and documents their needs for the artistic process. A brief review of his most fruitful findings will be presented here.

The research conducted was limited to those with active artistic involvement with spatial sound. The survey was performed over a two-week period via the web and fifty-two surveys were completed internationally. On average respondents had composition experience of 20 years, 14 years of which was computer-aided, and 10 years of which involved spatialization.

<table>
<thead>
<tr>
<th>Table 1. Why Composers Use Spatial Aspects in Their Music</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category</strong></td>
</tr>
<tr>
<td>To enhance the listening experience</td>
</tr>
<tr>
<td>As a paradigm for artistic expression</td>
</tr>
<tr>
<td>To organize and structure sounds</td>
</tr>
<tr>
<td>To experiment with technology and spatial effects</td>
</tr>
<tr>
<td>Perceptual motivation</td>
</tr>
<tr>
<td>Segregation and blending of sounds</td>
</tr>
<tr>
<td>To add motion and dynamism</td>
</tr>
<tr>
<td>To make sounds more natural and vivid</td>
</tr>
</tbody>
</table>

Fifty responses in open-comment form. Respondents chose multiple categories.

Figure 1, Nils (2011), ‘Why Composers Use Spatial Aspects in Their Music’

From figure 1, one may gather that most artists are using spatial sound techniques ‘to enhance the listening experience.’ While this answer is somewhat vague, it infers a general view that adding spatial dimensional variants to a sonic display will enhance the artistic outcome and listener experience. This response forwards the importance of multi-dimensionality.

requirements, a tool that allows for 3D and 2D speaker arrays. The second and third most popular responses ‘as a paradigm for artistic expression’ and ‘to organize and structure sounds,’ reaffirm this multi dimensionality requirement.

Figure 2 reveals that spatial audio artists are predominately spatializing prepared electronics and live electronics. From this finding one may deduce one important factor, that users are moving from traditional instrumentation choices to modern, digital instrumentation. The term ‘electronics’ of course refers to any electronic instrumentation choice, such as laptops and synthesizers. However, as the past two decades have brought an abundance of composition and mixing based DAWs to suit almost any user, this author is making the assumption that most of these spatial composers are using computer software produced electronics. Therefore, creating the need that spaceJam be digitally accessible; software not hardware.

Figure 2 – Nils (2011), Orchestration and Musical Context Chocies

Figure 2 also indicates most artists utilize spatialization techniques and tools in the concert and installation performance contexts. This identifies the importance of spaceJam to contain

---

features that aid performance, such as gesture control tracking, and real-time movement capabilities. Context based requirements and issues are further explored in figure 3. This figure reveals the 38% of spatial sound artist have issues with the ‘technical limitations of the venue.’

The algorithms controlling many sound spatialization tools call upon strict speaker arrays that often are inaccessible to the average museum, art space, or performance venue. Or, these performance venues are unable to customize their established speaker configurations for the artist, as reflected in the third largest context issue: ‘non-ideal loudspeaker and audience location.’ This issue is echoed by Malham et al. (1995) ‘the exploration of sound spatialization is a preoccupation of many composers and performers of electroacoustic music. Two-channel stereo techniques are widely used in the genre, but more sophisticated forms are often restricted to those with access to significant technical resources.’

Figures 2 and 3 reveal that flexibility and usability of hardware and software within concert and installation settings is paramount. The sound spatialization tool should have the ability to morph between contexts without hassle.

---

14 Ibid
Figure 4 shows an interesting dichotomy between knowledge and use that is currently experienced by artists. 65% - 88% of respondents are aware of products created for sounds patializing (DBAP, VBPA, Ambisonics, Wavefield Synthesis) yet, only 10 – 20% use these actively. This discrepancy reveals that artists are feeling alienated from the tools that have been designed for their specific needs. Artists are forced to use unsuitable, limiting technology, as projected in figure 4: 80% of artists use panning automation in DAWs, and 60% using pan pots on mixing consoles.

These two methods are not appropriate, and have not been optimized for multi-dimensional spatialization purposes. Panning with an audio sequencer refers to panning automation applied by a user to painstakingly small audio tracks on DAWs. This method will often call for
additional mixing between channels, and is not real-time. Furthermore, the interface metaphor for panning is a right / left panning knob, which does not conceptually invite multidimensional movement. A similar experience is found using pan-pots in mixing-consoles that require using unwieldy mixing board knobs. Although this method may be real-time, multiple knobs must be motioned simultaneously, meaning this method prone to accident.

Perhaps one of the most important questions on the survey is an enquiry into why artists work with their chosen spatialization equipment versus using other tools. Figure 5 reveals the most common answers revolve around usability and accessibility. These may be practically reflected as flexible hardware requirements, intuitive interfaces, and the inclusion of appropriate features. This idea is further supported by the fact that ‘half of the respondents use fewer features than their spatialization tools offer.’ This is most likely due to the inclusion of features that are too advanced, too time consuming to learn, or unrelated to artistic endeavours. Nils argues correctly that developers have to ‘acknowledge the higher technical complexity of multi-loudspeaker setups.’\[16\] Features should be filtered to fit user requirements, which will additionally reduce GUI clutter.

One may infer from Nils’ (2011) study that while sound spatialization tools with sufficient features are readily available and known of, they pose practical and technical obstacles to many. From this research, five requirements for an artistically driven sound spatialization tool have been found:

1. The product must allow for multidimensional sound spatialization
2. It must be accessible from a computer
3. It should include features that are appropriate for live performance and installation contexts
4. Hardware requirements should be flexible
5. It must have an intuitive interface

2.3 Technical Spatialization Practice

Sound spatialization tools allow one to perceptually move a sound source within a predefined speaker array. By taking advantage of our natural binaural localization cues, spatialization techniques trick the brain into perceiving virtual sound source movement. Binaural localization is caused by the separation of the ears, and allows humans to psychophysically evaluate sonic signals presented to each ear, and determines the position of the auditory events (Blauert 1997.)

2.3.1 Overview of Spatialization Techniques:

There are four techniques typically used in spatialization algorithms: inter-channel time delays, inter-channel level differences, sound field reproduction, and binaural reproduction. Inter-channel time delays imitate the natural binaural localization cue: inter-aural time difference
(ITD.) By delaying selected channels of a sound source, it mimics the time difference between a source hitting the ipsilateral ear (the ear that receives the sound wave first), and then the contralateral ear (the ear that receives the sound wave second). The placement of the sound source in the virtual scene depends on the delay length. Inter-channel level delays mimic the natural human localization cue: interaural intensity differences (IID). IID corresponds to the reduced intensity level of the sonic event between ears, due to reflection and absorption of the sound source by the body. When used in spatialization, this level difference defines the placement of the source. Sound field reproduction synthesizes an acoustical wave field with a large number of loudspeakers. Psychoacoustic effects from the sound event are ideally minor, as it is assumed that listeners respond to the synthesized sound field in the same way as the real acoustic event. Lastly, binaural spatialization techniques attempt to simulate a virtual source through the dynamic filtering of a sound source with the Head Related Transfer Function (HRTF.) This technique necessitates headphone reproduction.

Spatialization algorithms will be either channel based or sound object based, and layout dependent or layout independent. Channel based refers to an algorithm that requires an absolute channel number to pan sound. Sound objects, on the other hand, evoke another layer of abstraction where sound objects are situated within a space, with their own set of properties, namely audio and spatial position. Layout dependant and layout independent simply refer to the algorithms ability to provide a complete representation of the spatialized sound if the speaker setup is altered.
Figure 6 reveals the numerous spatialization tools, and groups them into their respective technique classifications. As one can see, inter-channel level differences for sound spatialization is the most developed area.

Figure 6, Nils (2011) Sound Spatialization Tools and Their Technique Classifications ¹⁷

2.3.2 Rendering Concepts

As one may gather from figure 6, there are many spatial sound rendering concepts. However, this section will provide a review of rendering concepts that adhere most to the artistic requirement: hardware flexibility, as found in section 2.2. The following spatialization rendering concepts will be discussed: stereo panning, Ambisonics, VBAP, and DBAP.

2.3.2.1 Stereo Sound Panning

Stereo sound panning is a truly tried and tested method for altering the virtual position of a source between two speakers. It will be discussed here as this method provides to be the basis of many other spatialization algorithms. We can see in figure 7 that stereophony requires two speakers and a receiver forming an equilateral triangle (Rumsey 2001). During sound rendering, a listener will be receiving two signals at each ear. If these two signals are exactly the same, the virtual placement of the source will be perceptually placed in the middle of the two speakers. According to Blauert (1996), if the two signals differ only in one of the following - amplitude, time, or phase - they will continue to be considered as one coherent signal, but their virtual placement will be altered.

Figure 7
'Optimum arrangement of two loudspeakers and listener for stereo listening Stereo Panning' Lossius (2007)
If the signals differ in amplitude, this inherently becomes the method commonly referred to as Equal Intensity Panning (or -3dB panpot law.) This is a widely popularized 2D panning method that allows a source to graduate within the stereo image through inter-channel amplitude differences. The relative amplitude coefficients of each speaker can be found via the angle between the listener to each speaker.

\[ v_L = v_s \cdot \cos \theta, \quad v_R = v_s \cdot \sin \theta, \]

Figure 8, Lossius (2007) \( v_s = \) mono sources amplitude, \( v_l = \) amplitude of left speaker, \( v_r = \) amplitude of right speaker

The source has a constant intensity \( (I) \) of 1, found by the sum of the squared amplitudes for each speaker (figure 9). The intensity remains constant, because intensity is perceptually related to the distance of the source to the listener (Lossius 2007.) To control the distance accurately, some psycho-acoustical phenomena should be taken into account, as well as some other sound elements such as reflections and reverberation (Pulkki 1997)

\[ I = v_L^2 + v_R^2 = 1 \]

Figure 9 Lossius (2007), Constant Intensity \( I = \) Intensity

The concept behind stereophonic panning can be extended to the surround sound formats 5.1, 10.2, 22.2 etc. In such cases, the virtual placement line is continued above and around the listener, creating more possible virtual positions.

2.3.2.2 Ambisonics

Ambisonic theory is a spatial sound concept that is based on capturing a sound field with regard to a central reference point, and decomposing the sound field into spherical harmonics for
sound rendering. Spherical harmonics, a mathematical approach to define functions on the surface of the spherical sound field, determine the spatial resolution of an Ambisonic system. The Ambisonic order number dictates the quantity of harmonics into which a sound field is decomposed. The higher the Ambisonic order, the greater the harmonic amount, and the greater the resolution.

3D sound fields are captured using a sound field microphone with four capsules configured in a regular tetrahedron. The signals recorded are the A-format signals, which contain directional information of the sound field, but must be first converted to B-format to correspond to the first-order spherical harmonics and define the sound field. A-to-B format conversion involves a simple set of equations for combining the A-format signals into each of the B-format signals. B-format signals describe the omnidirectional harmonic of the sound field W, and the three bidirectional harmonics corresponding to the X, Y, and Z axes, respectively. Non-coincidence correction filtering is then applied to the B-format signal to obtain a flat frequency response for the sound field by correcting for the spectral coloration caused by phasing due to the non-coincidence of the soundfield microphone capsules. Once the final B-format signals have been attained, the captured sound field can be accurately reproduced at a central listening position via any number or configuration of loudspeakers through the implementation of an Ambisonic decoder. Ambisonics B-Format signals may be decoded to any regular speaker layout. Traditional layouts include: square, hexagon, cube, sphere. Irregular speaker configurations have been recently suggested, however are not currently viable.
2.3.2.3 VBAP

Vector Based Amplitude Panning (VBAP) is a spatialization rendering concept developed in 1997 by Ville Pulkki as a means for virtual source positioning in a pre-defined 2D or 3D speaker array. Previous to Pulkki’s efforts, stereo panning and Ambisonics were the most widely used sound spatializing methods. In terms of 2D stereo arrays, Pullki extended equal intensity panning methods for 3D arrays, and claimed that ‘a natural improvement [of Ambisonics] would be a virtual sound source positioning system that would be independent of the loudspeaker arrangement and could produce virtual sound sources with maximum accuracy using the current loudspeaker configuration.’

An undefined number of input sources may be placed within the virtual space, and possess a static or dynamic nature. VBAP does not require a set speaker amount, and speakers may be placed in 2D or 3D arrays. However, to render this technique successfully, speakers should be placed equidistant from a listener, leading to a spherical formation in 3D arrays. As the name suggests, this panning technique is based on vectors within an array.

\[ p = g_1l_1 + g_2l_2 \]

Figure 10, Pulkki (1997) Virtual source Vectors are a linear combination of speaker vectors, 2D array

VBAP virtually positions sources upon the active arc according to predefined vector bases that are described by the length of the vector between the listener and speaker (seen as \( l_i \) in figure 10 and 11.) ‘\( p \)’ conveys the virtual sources vector, which is a linear combination of the speaker vectors.

---

Pulkki, V. (1997) Virtual Sound Source Positioning Using Vector Base Amplitude Panning, Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, HUT, Finland page 1
When using more than two speakers within the VBAP system, speakers are grouped into pairs with each speaker able to join more than a single pair. As speakers are added, the active arc is extended ‘the sound field that can be produced with VBAP is a union of the active arcs of the available loudspeaker bases.’\textsuperscript{19} The position of the source in the arc controls the gain factor of each speaker. As the source moves away from a speaker, the gain factor graduates towards zero before the ‘change-over’ point, which performs a cross-fade between speaker gain factors.

To broaden the dimensionality of the VBAP model, additional speakers may be placed above or below the original 2D model, but must continue to maintain equidistance from the listener. 3D arrays form a spherical shape around the listener. From the listener’s central perspective, 3D configurations will result in a triangular speaker formation for which the virtual source may be placed within (figure 12). Panning the sound within active triangles is also based on relative amplitude panning. The virtual source placement (‘$p$’, in figure 12) is a linear combination of all three speakers within the active triangle.

\textsuperscript{19} Pulkki, Ville (1997) Virtual Sound Source Positioning Using Vector Base Amplitude Panning, Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, HUT, Finland page 2
2.3.2.4 DBAP

Lossius et al. (2009) created Distance Based Amplitude Panning (DBAP) as a panning based sound spatialization method for relaying sound over multichannel systems in arbitrary 2D or 3D arrays. Engineered through a creative lens, Lossius crafted the first sound spatialization tool that held no assumptions regarding listener position and speaker layout. A non-existent ‘sweet spot’ or speaker position requirement allows for a highly flexible system when considering the range of possible users, possible locations, equipment choices, and hardware requirements.

When considering the DBAP algorithm, the speakers and virtual source placement can be expressed via the Cartesian co-ordinate system. DBAP then assigns gain factors based on the distance between the source(s) and the speakers. The distance between each speaker and a source can be found with the following equation:
\[ d_i = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2 + (z_i - z_s)^2}. \]

Figure 13, Lossis (2009): ‘finding the distance between speaker and source’ \( d_i \) = distance between speaker and virtual source, \((x_s, y_s)\) = virtual source coordinates, \((x_i, y_i)\) = speaker positions \(i\), \(r\) = spatial blur.

For the true reproduction of DBAP, the algorithm constantly assumes two factors: 1) the source retains constant intensity at all times. This assumed property was inherited from constant intensity stereo panning to multiple channels, and is upheld as to not falsely impose distance perception. The second assumption is that all speakers are active at all times, and the relevant amplitude of the speakers is found with the following equation (figure 14):

\[ v_i = \frac{k}{d_i^a} \]

Figure 14: Lossis (2009) \( v_i \) = amplitude of the \(i\)th speaker, \(k\) = is a coefficient depending on the position of the source and all speakers, \(a\) = coefficient for inverse square law roll off

The \(k\) coefficient is added to mitigate extensive source directionality. \(k\) is a value found from the distance between the source and all the speakers, and is then applied to the gain factor of each individual speaker to ensure a sound source is spread appropriately amongst all the speakers. Spatial blurring is then introduced to account for the issues that arise when the virtual source is placed and rendered solely from one speaker; where ‘\(d\)’ = 0. Conceptually, the spatial blurring co-efficient may be thought of like dithering. There is a relative amount of noise added to each speaker which makes its harder for the virtual source to gravitate towards just one speaker. The more noise added makes this harder for the source to gravitate towards just one speaker. Ideally, this co-efficient is dynamic with speaker array size, so one may impose their
own judgment on speaker directionality. Lossius adjusts the algorithm for spatial blurring factor in figure 15 – $r$:

$$r_s \geq 0$$

$$d_i = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2 + r_s^2}$$

Figure 15, Lossius, (2009) Finding the distance between speaker and source and adding spatial blurring factor - $r$

The convex hull is a supplementary algorithm for sources that are virtually placed outside the speaker array. The distance between the source and the closest speaker output is then calculated as the speaker value. The final stage of DBAP calculation involves speaker weights. $w$ is another constant applied to the gain factor of each speaker, that is intended to provide individual speaker persuasion.

2.4 Review of Spatialization Products

Studies suggest that artists are experiencing alienation from many spatialization products currently available. To assess this claim, this section will evaluate spatialization tools in light of the five primary requirements found in section 2.2.2.2 of this paper:

1. The product must allow for multidimensional sound spatialization
2. It must be accessible from a computer
3. It should include features that are appropriate for live performance and installation contexts
4. Hardware requirements should be flexible
5. It must have an intuitive interface
2.4.1 Jamoma Modular

Jamoma, the brainchild of Trond Lossius inventor of DBAP, is a series of tools and frameworks dedicated to the creation of digital art. This analysis will focus on Jamoma Modular, a particular toolset that has functionality within Max, the chosen programming environment for spaceJam. Jamoma Modular is essentially a library of module patches dedicated to data, audio, and video processing, of which one may build a customized tool from. Modules include OSC communication with external software. The Jamoma module for spatializing (not including Ambisonic modules) is the Max patch named: jmod.sur.dbap~.maxpat. When connected to other appropriate modules, the Max jmod.sur.dbap~.maxpat patch implements the DBAP algorithm for spatialization in 2D and 3D spaces. In regards to the five artistic requirements noted above, when this module is connected to other modules correctly, it boasts four of the five requirements, but lacks interface intuitively. Figure 16 is an example of the Max help patch named: jmod.sur.dbap~.maxhelp. This help patch is a ‘ready-made’ patch for sound spatialization using the DBAP algorithm.
In this author’s opinion, the UX of Jamoma Modular modules are not suitable for users without programming skills, or users without a sound engineering background. This is first felt by the modular architecture. This architecture is an acceptable solution for those who wish to create highly personalized, particular applications. However, in other cases, modular architecture can be burdensome and unwieldy. Asking one to build an application is a demanding task, as it requires extensive knowledge of the module library and module functionality. This only heightens the software’s learning curve. Furthermore, Max programming idiosyncrasies must be ascertained. These include the patching chord metaphor, and the message and object protocol. Users, who are not initially alienated by the programming elements, must learn these Max
fundamentals, increasing the learning curve gradient. If using the jmod.sur.dbap~.maxhelp patch to spatialize sound, there are a number of interface restrictions. There is no apparent procedure for a user to follow, reflecting that Jamoma Modular assumes some sound spatialization knowledge. This is particularly apparent when a user is to place virtual speakers in the virtual space. This is achieved by clicking on the ‘p setup’ object where the user then types in speaker number + position coordinates into a message box, however none of this is documented, or explained. This unclear interface that depends on Max programming knowledge is not appropriate for such a vital element of the process. Another major setback revealed in the interface is the 2D map used to represent a 3D space. This metaphor is can hinder the conceptual and practical artistic exercise.

2.4.2 VBAP in Max

Villie Pulkki’s rendering concept VBAP has been transformed into a functional object within the Max environment. As the object itself does not have a UX to analyze, the object’s help patch (figure 17) will be assessed. To spatialize sound using the VBAP object, one must send it speaker position coordinates reflecting 2D or 3D arrays, a mono source audio file, and the intended panning position represented as azimuth and elevation coordinates. This tool only satisfies two of the five artistic requirements. A major drawback comes inherent in the algorithm. Although it allows for 2D and 3D spatialization, it requires speaker arrays that are equidistant to one listener, causing hardware and listener position inflexibility. Also, this tool has no appropriate automation or real time performance features, rendering it unsuitable for performance contexts.
In terms of the interface, it is clear that little consideration has been applied. Aside from the lack of procedural structure, this tool introduces the issue of language barriers. Words such as ‘azimuth’ and ‘elevation’ help define the exact location of a sound source. Unfortunately, this language is not shared outside the audio community and therefore hinders the usability of this product for many. Usability issues are again experienced when defining speaker placement. As with the Jamoma tool, users define speaker location with two coordinates in a Max message box. This is an unintuitive way to represent the location of speakers in a real space, and does not lend itself to the 3D spatialization concept. Furthermore, there is no visual cue in the GUI that
signifies the location of sound or speakers within the space. The lack of visual stimuli heightens the complexity level.

2.4.3 Spatium

Spatium is a toolbox of sound spatialization applications designed by Rui Penha. Spatium is free, open source, and created for artistically driven usage. Based on modular architecture, multiple Spatium interfaces can interact via OSC (Open Sound Control) communication. The suite of Spatium tools is made up from four renderers, five Max objects, ten interfaces, and two plugins. Spatium Panning and Spatium Ambi will be analyzed as they most closely adhere to the five requirements.

2.4.3.1 Spatium Panning

This standalone allows for the spatialization of a mono source file within a 2D speaker layout. A user defines the sound trajectory path which may move between up to twenty-four speakers. The Spatium Panning algorithm is based on equal intensity panning between two stereo pairs. When a sound moves towards the circumference of the 2D circle, it activates the two closest speakers and applies the sine or square root panning law (user defined.) One particularly intriguing attribute of Spatium Panning, is the renderer’s ability to record the user defined speaker path into a set of mono audio files, furthermore user sessions are savable. These features are helpful for installation and performance contexts. The interface also is very successful in terms of offloading real-time virtual source spatial information via the GUI. However, this application achieves just one of the five artistic requirements; it is software.
Unfortunately, Spatium Panning does not allow for 3D spatialization, and panning may only occur in predefined speaker set-up options. It also does not offer real-time functionalities or multi-source movement, both of which can hinder this product's performance ability. The interface is attractive, however lacks some user-friendly qualities, such as helpful visual cues, appropriate user documentation, and a coherent execution structure.

2.4.3.2 Spatium Ambi

Spatium Ambi is an example of an Ambisonic encoder, or panner, that uses Ambisonic encoding principles to encode the spatial position of a sound, as per its angular relationship to the point of origin. However, such panners do not simulate any other part of the natural
environment, such as the distance, velocity or remembrance. Spatium Ambi allows for spatialization by encoding up to 16 monophonic sound sources into a virtual Ambisonic sound field. The distance and placement of the sources within the sound field are received via OSC, or input manually in the top-most section of the GUI (figure 18). The sound field is then decoded into a predefined loudspeaker arrangement. This product does not satisfy all five requirements. Due to the nature of Ambisonic reproduction, a truly flexible hardware set up is not achievable. Similarly to Spatium Panning, the interface is attractive, yet it poses numerous usability issues. In this product, the complexity of Ambisonic technology is mirrored in the interface through language use such as: ‘encode,’ ‘decode,’ ‘azimuth,’ ‘elevation’ and ‘radius.’ These words, which reflect critical functionality of the product, will mean little to those without an understanding of Ambisonic technology. Spatium Ambi is also not practical in performance scenarios due to its reliance on the inclusion of other applications from the Spatium family to send channel location information via OSC. This is due to the lack of interface space to do so manually, as seen in the top most box of the GUI, figure 18.

Figure 18, Penha et al (2013) GUI Spatium Ambi
This analysis of spatialization products affirms the argument that artists are facing creative disservice through a lack of appropriate tools for their practice. Suitable software is necessary to encourage and enable artistic sound spatialization.
3.0 spaceJam Implementation

Many spatial sound artists are currently experiencing major creative setbacks due to an inaccessibility of appropriate media authoring software. In reaction to this demand, spaceJam has been developed. spaceJam is an interactive multimedia authoring tool that is designed for the spatialization of audio in 3D spaces. spaceJam is reliant on user input via the GUI, which then informs software functionality. The GUI architecture is a non-modular, multi-frame view that attempts to guide users effortlessly through the sound spatialization procedure. Users are presented with three perspectives of a 3D virtual model that intentionally mimics their real room and up to 256 speaker channels, of which they may pan up to 256 virtual sound sources within. Gesture capture of sound source spatialization is a key feature of this tool, along with multi-channel audio file recordings that are based on source spatialization.

Implementation decisions during the development of spaceJam have been directly informed by the research conducted in section 2.0 of this paper. Usability is the design criteria for spaceJam, therefore the UCD process has structured its development. To comply with UCD principal number 1, ‘understand your target audience,’ research was implemented to recognize the needs of the target user and the five key requirements can be found in section 2.2 of this paper. UCD principal number 2: ‘designing for usability’ requires spaceJam to be developed according to these findings.

3.1 UX

This section details the UX of spaceJam. The current spaceJam prototype is available to users in a folder containing all relevant files for the spaceJam Max patch to function, and the audio
folder. Users place the audio files they wish to spatialize into the audio folder, and upon opening spaceJam the audio files will be loaded automatically into the program.

When a user opens spaceJam they are sent to the ‘Set It Up’ window (figure 19.) This screen is dedicated to options regarding spaceJam features, and this window is where users may virtually re-creation their real space. To aid usability, each window has numbered steps, in an attempt to further apply procedure for user ease.

Step 1: ‘user level’ motions a user to decide between the ‘beginner’ and ‘I’ve got the hang of it’ level. These settings modify the level of interface and feature complexity. For example, beginner mode will disable technical features such as OSC source movement, impose settings for individual speaker influences per source, and will turn on hints. The transition to the next level unlocks these features. Step 2: asks the user to turn hints on or off; hints are tooltips. Step 3 and 4 of this window are dedicated to the virtual space set up. Users define the shape and dimensions of the space, and then virtual speaker numbers. The four larger spheres in figure 19
resemble speakers. The balls within each speaker indicate the center of the speaker, and the radius size is indicative of influence size. The smaller sphere in figure 19 signifies the sound source uploaded from the audio folder. When users have completed all four steps, they may click the forward button at the bottom of the screen, which scrolls to the next window.

The ‘Speaker Placement’ window allows for the placement of virtual speakers in the virtual space. The speaker channel is to be first specified, and its placement within the virtual space may be manipulated via the top/bottom, left/right, and forward/back sliders. Common language as opposed to technical language (x, y, z axis) has been chosen to represent speaker placement to encourage usability. The section ‘speaker influence’ allows a user to define the radius of each speaker, which creatively aids source individuality. The sound source will not emanate until it has entered the speaker radius, and the distance between sound source and speaker center is inversely proportional the speaker gains. A large speaker radius correlates to a large area of potential source audibility, which is inversely proportional to a speakers gain gradation size.

Figure 20, spaceJam GUI ‘Speaker Placement’ window
The final window allows users to spatialize the sound source. In this window, users may pan, record, playback, and bounce audio files to the disk. To spatialize a sound source, step 1 prompts the user to select an audio source (figure 21), and chosen source file name will appear below.

Figure 21 reveals a scenario where three audio files (the smaller spheres) have been uploaded and spatialized around the virtual room. The controllers to move sound sources are displayed in the box below the virtual room screen labeled ‘Controllers’ (figure 21). To move a source on the x and y axis, a user holds down the command key and expresses direction with their mouse. For movement on the z axis, they hold down alt key and again direct movement with the mouse. Audio movement may be recorded when spaceJam is in record mode. Record enable is performed by clicking the record button, or
presses ‘r’ on the keyboard. When in record mode, the record button flashes red to convey an active mode (figure 22.) To play back gesture capture, users click the button in the playback section or hit ‘p.’ This button flashes when playback is in session, and the spatialization for all sound sources will playback (figure 23.) Additional options regarding playback speed, and pause / resume may be accessed in the playback section too. Users have the option to record the audio scene to file. This recording outputs a multi-channel file (defined by speaker amount), where audio is based on spatialization positions. The record button flashes (figure 24) when recording is in progress. The three buttons mentioned here start the file from the beginning, regardless of functionality.

To upload new files to spaceJam without exiting, the user can place the new audio files into the audio folder and click ‘reload sounds’ as seen in figure 22. Reset buttons are available in the playback and record to file sections, which wipe spatialization recordings and record to file settings.

3.2 Technical Implementation

spaceJam has been developed using Max, a visually based graphical programming language that is primarily devoted to multimedia. Max is an appropriate environment as it has a large range of suitable features for interactive multimedia applications, and has the ability to create standalone applications easily. Standalones do not require the purchase of Max, just the download of free ‘Max Runtime’ software. This means potential users will not face monetary restrictions when using spaceJam.
3.2.1 Spatialization Algorithm

spaceJam’s spatialization algorithm will affect the product’s ability to provide multidimensional sound spatialization, and flexible hardware setups. Section 2.3 details a number of possible spatialization algorithms, each retaining their own set of advantages and disadvantages in regards to sound standard and technical hardware/software requirements. In light of these factors and in recognition of the target user group, the algorithm implemented will be based on Lossius’ DBAP algorithm. The DBAP algorithm has been selected due to its unrestricted assumptions regarding listener position and speaker layout. This flexibility is of major artistic value as the system may be dynamically re-defined as per logistical, technological, temporal, spatial, performer, and/or location-based issues, that may well occur during the artistic process.

As outlined previously, DBAP is a panning based sound spatialization method for relaying sound over multichannel systems in arbitrary 2D or 3D arrays. The relative amplitude of the speakers is inversely proportional to the distance from source to speaker, and additional coefficients are then applied to ensure spatial spread. DBAP assumes that the source retains constant intensity, and that all speakers are active at all times during calculation. The chosen algorithm has been influenced by DBAP; however, spaceJam will instantiate a modified, simplified version. Speaker gains are defined according to distance from speaker to source and speaker influence. The gain of each speaker will be found via the following equation:

\[ v_i = 1 - \left( \frac{d_i}{r_i} \right) \]

Figure 25: spaceJam Algorithm \( v_i \) = volume of the \( i \)th speaker, \( d_i \) = distance between speaker and virtual source, \( r_i \) = \( i \)th speaker radius size.
Distance from source to speaker is still calculated via:

\[ d_i = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2 + (z_i - z_s)^2} \]

Figure 26: Lossius (2007) \( d_i = \) distance between speaker and virtual source, \((x_s, y_s, z_s) = \) virtual source coordinates, \((x_i, y_i, z_i) = \) speaker positions

The gain factor of each speaker is still found with the above calculation; however, the calculation is only applied when a source intersects the perimeter of speaker influence (speaker radius \( r_i \)). The speaker gain is 0.0 outside the circumference, and 1.0 in the center. It is a similar concept to the speaker weights \( (w_i) \) of DBAP. The larger the speaker influence, the larger the area of potential audibility a speaker has. Speaker influence size is inversely proportional to a speakers gain gradation from perimeter to center, i.e. minimum to maximum gain. For an inverse relationship between gain and distance, the ratio found from \( (d_i / r_i) \) is then subtracted from 1. Distance from source to speaker is clipped at 1 to guarantee a gain of 0 outside the speaker radius.

The \( k \) coefficient of DBAP has been removed from spaceJam, for simplicity and reduction of computational expense. It was not considered an absolute necessity, as the sound quality was deemed acceptable by the author in the prototyping stage. However, users will assess this opinion during the preliminary testing.

3.2.1 Implementation

This section will discuss some other notable functionalities of spaceJam. Three tables of data define the speaker positions within spaceJam’s virtual environment. Each table represents either position, radius size, or colour information. The position table stores speaker number + x + y +
z co-ordinates. The radius table holds speaker number + size, and the colour table holds speaker number + red + green + blue + white numbers that combine to provide the colour variables. The number of speakers described by the user defines the row number of each table. Cell information is modified and viewed via JavaScript code living within the Max js object. This code takes in function calls posed by the user interface, and updates table information accordingly. The code automatically outputs new information informing speaker characteristics in the GUI.

Audio sources are represented virtually as individual spheres in spaceJam. The position of sources directly affect speaker gains, thus this information is always tracked. To allow for gesture capture and recording, first the audio source number is verified and record button must be is turned on. The co-ordinates from the handle object (the MAX object that allows interactive movement of objects on the screen; in this case, the sound source sphere) are found in real time. The co-ordinate output is then sent into the spaceJam equation (figure 25, figure 26) to find speaker gains. Simultaneously the file playback object (sfplay) outputs a number indicating time elapsed. This, along with the source position information, is sent into an event sequencer. This sequencer times virtual source position with the audio file. If the ‘playback’ button is selected in the GUI, the file will playback and the sequencer will output the recorded source positional information. These sound source positions are sent back into the equation to allow for correct speaker gains.
3.3 Other Implementation Decisions

Without listing all implementation choices, this section attempts to discuss the more profound choices made during development.

3.3.1 Non-Modular

The choice to have a non-modular system is based on the theory that it will reduce complexity and learning curves. A modular structure assumes at least three essential prerequisites: the user is educated in the concept of a modular based system, the user has familiarized themselves with the module library to appropriately build their own application, and the user is able to program within the given environment. In light of these requirements, an integrated approach is applied to the interface design.

3.3.2 Multi-Frame View

Sound spatialization tools discussed in section 2.4 are all missing one major element: a sense of procedure. Sound spatialization can be a complex undertaking with many different physical, technical and mathematic variables. The multi-frame view approach leads users through the product in a practical manner. This enforced procedure will not only lower learning curves, but also reduce the amount of unnecessary features within each window, providing a sleeker GUI layout. Additionally, it will inherently educate users on the procedure required to spatialize sound.
3.3.3 User defined levels

spaceJam is been developed to promote usability, therefore, this product should be accessible to a wide scope of users with differing practical and technical backgrounds. By allowing users to modify the level of interface and feature complexity, the program will make reduced assumptions of user knowledge and comfort level. For beginners, the software will perform three functions: 1) disable highly technical features such as OSC, 2) impose general settings for technically complex ideas, such changing all speaker radius sizes per sound source and 3) automatically turn on ‘hints’.

3.3.4 Affordance

The HCI concept of affordance has been applied to spaceJam when possible. Affordance refers to controls that visually advocate their functionality (Norman 1988); a button affords clicking, and a scroll bar affords scrolling (Lee & Boling 1996.) Motivational and affordance elements have been added, such as: flashing colour animation for buttons when activated, the animation of the spaceJam logo when users are spatializing sound sources, and the animation of the forward and back buttons when they clicked.
4.0 Preliminary Testing

Research suggests that current sound spatialization products do not adequately support spatial sound artistic practice. In response to this need, spaceJam has been developed with a heightened focus on UX for the media authoring process. To understand if spaceJam is correctly abiding to artistic needs, a prototype of spaceJam was tested during a pilot study. This goal of this preliminary testing was to gage product usability, and general user attitude toward the software.

The definition of usability in reference to interactive multimedia software has been discussed in section 2.0 of this paper. Lee (1999) has established the five dimensions of usability that may be assessed in user testing scenarios.

1. Learnability
2. Performance effectiveness
3. Flexibility
4. Error tolerance and system integrity
5. User satisfaction.

It is unproductive to test all five dimensions simultaneously (Reed 1992), therefore the first preliminary test will assess learnability and user satisfaction. Learnability refers to ‘the ease with which new or occasional users may accomplish certain tasks (Lindgard 1994,)’ and ‘user satisfaction’ aims to articulate the users emotional response to the system.

4.1 Participants

Ten graduate students from New York University voluntarily participated in the pilot study. This participant number does not provide statistically significant evidence, however it is a
reasonable number of participants for preliminary testing. Participants in this study were earning masters degree in either music technology or integrated digital media, 90% of which considered themselves audio artists or musicians. When asked to label their personal experience with multi channel audio, 40% beginner, 40% intermediate and 20% advanced. 100% of participants saw artistic benefit in spatializing multichannel sound; such benefits included: ‘adding realism’, ‘extending virtual scenes’ and ‘creating envelopment and immersion’. However, only 30% of participants had ever used any spatialization tools or techniques for multi channel audio, and two thirds of the 30% predominantly used DAW panning automation or pan-pots on mixing boards. The remaining 70% either felt alienated by the technical implications (predominantly from software applications,) or had not been in a situation that required 2+ audio channels.

4.2 Testing environment

The pilot study was held in the NYU Music Technology research lab. The lab is fit with 20-channel speaker array in a spherical formation. 4 Channels were used in this experiment:

<table>
<thead>
<tr>
<th>Channel #</th>
<th>Azimuth°</th>
<th>Elevation°</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel 1</td>
<td>315°</td>
<td>0°</td>
</tr>
<tr>
<td>Channel 2</td>
<td>45°</td>
<td>0°</td>
</tr>
<tr>
<td>Channel 3</td>
<td>135°</td>
<td>0°</td>
</tr>
<tr>
<td>Channel 4</td>
<td>225°</td>
<td>0°</td>
</tr>
</tbody>
</table>

Channels were externally marked for the participants to note. The spaceJam prototype in the form of a Max standalone was accessible on the desktop computer in research lab. A number of
audio files were added to the ‘spaceJam’ folder on the desktop. All audio files were limited to 30 seconds for uniformity.

4.3 Procedure

The test was designed to provide feedback on product usability and to gather a general response toward the product. To gain this knowledge, users would have to directly engage with the product as the intended end user would. Participants were asked to first watch a three-minute video that introduced them to spaceJam, and the exact steps needed to perform spatialization. Participants were then asked to complete two tasks of increasing difficulty. There were no time constraints for the testing iterations. After completion of the tasks, users were then asked to fill out a ‘user evaluation’ survey.

4.3.1 Task 1

An audio file was placed into the audio folder for the user, and the user opened to the ‘Set It Up’ window. They were asked to re-create the research lab in their virtual room screen, and create 2 speaker channels, representing channels 1 and 2 in the research lab. Participants then placed their virtual speakers within their virtual space, and made each source have an influence of 0.9. Spatialization between channels 1 and 2 was then asked of: ‘move the sound source from the center of speaker 1 to the center of speaker 2 three times. Take two seconds to move between each speaker.’ Participants then viewed their spatialization, and recorded the audio to the desktop.

---

20 Task 1 Instructions, Preliminary Testing of spaceJam
Task 1 required participants to only spatialize on the x-axis, and did not have to switch between controllers. Specific timing and placement was instructed to gain an understanding of controller usability.

4.3.2 Task 2

Task 2 was purposefully more difficult and encompassed all the steps needed to spatialize multidimensional, multichannel and multi-source sound using spaceJam. Participants were asked to upload an extra audio file, and add an additional two speaker channels to their virtual space. After virtually placing channel 3 and 4, and creating an influence of 1.0, users were then asked to spatialize each sound source in a circular motion around the speaker array three times in 30 seconds (the limit of the audio track), playback the spatialization and then record to the desktop.

Task 2 called for two more advanced exercises: first participants had to drag specific files into the audio folder and upload them to spaceJam. Secondly, task 2 required panning in a circular motion between the given four channels, necessitating seamless transition between the two controllers. Completing three circular repetitions before the end of the 30-second audio clip heightened the challenge.

4.4 Results

Participant reaction to spaceJam was revealed via the user evaluation survey. This question and answer style survey was presented to participants directly after they completed both tasks.
<table>
<thead>
<tr>
<th>Question</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ‘How easy was spaceJam to use on a scale 1 – 10? (difficult = 1 and easy = 10)’</td>
<td>Mean response: 7.5 out of 10, standard deviation: 0.97.</td>
</tr>
<tr>
<td>2. ‘Did you feel you improved significantly between task iterations’</td>
<td>100% yes</td>
</tr>
<tr>
<td>3. ‘How would you rate spaceJam on learnability? Would learning the software be time consuming and difficult (1), or you think the learning curve is small (10)?’</td>
<td>Mean response: 8.95, standard deviation: 1.2</td>
</tr>
<tr>
<td>4. ‘Did the interface layout make sense to you Y/N?’</td>
<td>100% yes</td>
</tr>
<tr>
<td>5. ‘What Aspect of the interface (if any) do you believe hindered spaceJam’s usability?’</td>
<td>90% of participants claimed the viewing window hindered their usability experience. 20% of participants felt the speaker influence factor was confusing.</td>
</tr>
<tr>
<td>6. ‘How easy were the controllers to use out of 10?’</td>
<td>Mean response: 7.35 out of 10, standard deviation: 2.0</td>
</tr>
<tr>
<td>7. ‘Did you need to use the hints?’</td>
<td>0% used the hints</td>
</tr>
<tr>
<td>8. ‘Did the interface appeal to you?’</td>
<td>90% yes</td>
</tr>
<tr>
<td>9. ‘What was you impression of the sound quality? 1 = bad, 10 = good’</td>
<td>Average 9.3, standard deviation 0.67.</td>
</tr>
</tbody>
</table>
10. ‘Do you feel there were features missing in spaceJam?’

5/10 participants felt it lacked features. 3 participants wanted the option of external controllers to control the sound source. 1 participant wanted the option of audio effects, and 1 participant wanted DAW integration.

11. ‘If you had to spatialize sound in the future, would you use spaceJam over other products?’

100% yes.

12. ‘How could spaceJam be improved?’

90% of participants felt the spaceJam viewing window could be improved.
5.0 Discussion and Future Work

spaceJam, is a sound spatialization tool developed for use within the artistic community. To ensure needs of the artistic community are met, the development process was closely guided by research of end user requirements. Furthermore, a preliminary test was conducted to assess usability and provide user feedback. Lee (1999) reveals the five dimensions for usability testing. Although only two dimensions were officially tested, all dimensions will be discussed here for the benefit of future spaceJam development and testing iterations.

5.0.1 Learnability

In terms of user deciphered product learnability, a very positive response was felt. Every participant felt improvement between iterations, and almost 90% felt the product would be easy to learn. These numbers are supported by the fact that no participants utilized the hints feature. This is likely due to the enforced procedure indicated by the numbered steps within every window. These made the process clear enough that no additional help was required. Source movement controllers scored a moderately score 7.35 out of 10, indicating they are somewhat easy to learn. spaceJam’s overall usability scored an average of 7.5 out of 10. This is a moderate result that can be improved upon. 90% of participants noted the biggest drawback to the products usability was the virtual space window. Users were confused with the three-perspective layout of the virtual room, and had trouble understanding the virtual space in relation the real space they were mimicking. A number of improvements to the virtual space interface have been envisioned, and range from easy to more advanced adjustments:
1. Segment the virtual space window. This may better define each perspective. Segmentation will imitate popular 3D modeling viewing paradigms, such as rhino3D and Sketchup.

2. Reduce the perspective views to just one or two. This will allows for larger views of a perspective.

3. Allow the perspective angle to be user defined. This will assign more control to the user.

4. Allow the user to insert an icon representing themselves in relation to the space. This may create a better understanding of the user position in relation to the real and virtual space.

5. Add spatial computing capabilities. Products such as structure sensor will allow the user to capture their real space and import it to their virtual space interface.

6. Allow for CAD files to be importable. This further extends point number five. Many 3D room-modeling products have this capability to provide a more robust portrayal of the space. This will allow users a better understanding of the virtual space.

These improvements have their own set of advantages and disadvantages ranging from computational expense, hardware cost and effectiveness. However, for the next iteration of testing, the first three points can be easily incorporated to the current system and tested.

5.0.2 Performance Effectiveness

Performance effectiveness was tested through a sound quality rating. This outcome was very positive with an average score of 9.3 out of 10. However, spaceJam could benefit with the addition of a spatial distribution factor. A spatial distribution is a dynamic co-efficient that
spreads a sound source based on the distance relationship between the source and speakers. It is artistically useful when differentiating sound source roles, such as ambient vs. direct.

5.0.3 Flexibility

When asked if the product was lacking any features, three opinions were offered: controller flexibility, DAW integration and extra audio effects. spaceJam has OSC compatibility to control sound source positions, although this feature was not tested in the pilot study. More specific hardware pieces that can integrate easily within spaceJam include: the leap motion and the XBOX 360 controller. These controllers are useful as one can move on the x, y, and z axes simultaneously. Each additional control option will positively grow the products performances features. DAW integration is an important and useful next step as common DAWs like ProTools and Ableton Live have insufficient multi-channel panning options. Additional audio effect features were considered during the development of spaceJam. However, as this product is first and foremost sound spatialization tool. It was decided that these additions would be best dealt with in software designed specifically for such signal processing implementations.

4. Error Tolerance and System Integrity

During the preliminary testing a small number of bugs were found, affecting system integrity. These include:

- If a participant switched from a shortcut to button clicks to activate the record feature, the feature was not consistently responsive.

- The radius influence factor was not changing between sound source file playback.
- All sound files gains snapped to the position of sound source 1 until their own position recording was enforced. These must be tended to before the next round of testing.

5.0.5 User Satisfaction

A very positive feedback was felt with respect to user satisfaction. 100% of users said they would use spaceJam over other sound spatialization tools, 100% of users felt the layout made sense, and 90% of users felt the interface was appealing.
6.0 Conclusion

spaceJam is a novel media-authoring interface created for the ill-supported spatial sound artistic community. A strong focus on usability within the target user group was applied to the development process, leading to product accessible to both an artist and a novice. A prototype of spaceJam was tested during the preliminary testing. This revealed a very positive reaction to the tool. However, it became clear the strategy used to create the virtual room interface was lacking. A number of improvements to spaceJam have been suggested, and should be tested during the next iterations of spaceJam evaluation testing.
References


Pulkki, V. (1997) Virtual Sound Source Positioning Using Vector Base Amplitude Panning, Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, HUT, Finland page 1

