Transforming the Creative Practices of Composers and Sound Designers with VirDAW, the Virtual Reality Digital Audio Workstation

Theresa Jean Tanenbaum (co-PI), Vincent Olivieri (co-PI), Kevin Anthony, JoVia Armstrong, Jack Bueermann, Garrett Gagnon, and Omar Costa Hamido.

INTRODUCTION

The Virtual Reality Digital Audio Workstation, or VirDAW, is a research project with the ultimate goal of developing a fully-functional Digital Audio Workstation (DAW) for the Virtual Reality (VR) environment. This paper outlines the early stages and current status of the production of the project, including preliminary conceptual research, initial software development, conceptual reassessment, and current prototyping targets. It also identifies key goals for the next few phases of the research.

VirDAW is an interscholastic collaborative research project helmed by faculty in Informatics (in the Donald Bren School of Information and Computer Sciences) and Drama (in the Claire Trevor School of the Arts) at UC Irvine. Students on the project come from Computer Science, Computer Game Science, Sound Design, and Music and range in degree programs from BA up through PhD.

The VirDAW team began our research by exploring two key questions; the first connects tools and aesthetics. Our first question asks: how does reconceiving the function of a DAW in VR change the practices and aesthetic outcomes of designers? Research and common-sense show that the toolset an artist uses directly affects the aesthetics of the creation (Davies, 2003); anyone who has tried to replicate an oil painting with watercolors will confirm this. With that in mind, we want to explore how the particular experiential phenomena of VR (immersion, embodiment, and spatiality, which will be expanded on in a later section) can be exploited to allow the sound designer to engage with the DAW in new and unexpected ways. All current DAW applications (including ProTools, Logic, Digital Performer, Ableton Live, Audacity, Reaper, etc.) use the standard computer interface tools (mouse, monitor, and keyboard). DAW Graphical User Interfaces (GUIs) are skeuomorphic in that they are designed to represent their real-world counterpart. Industry leader ProTools provides excellent examples of this: the Edit Window visualizes tracks of audio much like they would be organized on magnetic tape, and the Mix Window resembles the channel strips of an analog mixing console. Even most DAW control surfaces skew skeuomorphic by representing channels as if on an analogue mixing console. In the DAW environment, skeuomorphism certainly has value (chiefly by utilizing a common set of understood visual cues that translate from platform to platform and decade to decade, which eases the learning curve for users when moving between tools), but what happens if skeuomorphism is exchanged for a new way of visualizing sound design tools, one that is based on functionality and comprehension, not on a century-old analogue toolset? How would the designer’s content change? What new paths of creativity are opened? If we fundamentally change the tool, how is the art impacted?

The second question specifically concerns spatialization and mixing in multi-channel environments. Current tools for mixing in 3D generally visualize in 2D forms, either as a control surface or through a video monitor. This forces the user to engage with the three dimensions either parametrically (wherein the user determines position by entering cartesian or radial coordinates into an interface) or via an orthogonal projection, in which a sound’s position is represented via visual cues placed inside a wire-frame outline of the space. These current tools are cumbersome and unintuitive to use. The simulated space of Virtual Reality provides an opportunity to rethink spatial mixing using a more indexical approach to sound's position in space where the simulated environment can contain manipulatable sound objects. When mixing in 3D becomes exciting and engaging as opposed to tedious and time-consuming, we believe designers will choose to mix in 3D more often and more deeply. What will that work sound like?

We believe that VirDAW will fundamentally re-shape sound creation and mixing by moving the interface into an environment where tools and techniques can fundamentally break free from their skeuomorphic histories and instead be developed in ways that encourage engagement and functionality. VirDAW will certainly become a useful tool for the theatrical sound designer, but it can also be used to mix for film, television, VR environments, video games, and music.

FIRST STEPS

Initial research into VirDAW generally fell into two categories. One was iterating sketches of how different aspects of a DAW could be visually represented within a VR environment. We sought out visualization metaphors that were easy to grasp from their appearance, and we tried to avoid using traditional skeuomorphic visualization techniques unless we could identify a fundamental reason that such techniques added valuable meaning to the interface.

The second category of early research focused on identifying methods to provide control tools (which we term "widgets") within the VR environment. Designing for VR necessitates understanding the key ways in which VR is different from traditional computer use, and identifying those differences is essential before meaningful work can be done. We can distill those differences into three related concepts:

VR is immersive. While wearing VR goggles and headphones, aural and visual content surrounds the user, blocking out the rest of the world and enveloping the user in a media-rich experience. VirDAW surrounds the designer. Sound sources can be placed anywhere. Control points can be placed anywhere. Even listening positions can be placed anywhere in the VR environment. In fact, in VR, we can de-couple the listening position from the visualization position, so that the designer can move about in VirDAW, tweaking parameters, but still listening to the mix from whatever position they prefer.

VR is embodied. The current operational scale of VR is larger than keyboards, mice, and touchscreens, so developing for VR mandates specific thought about how interactivity translates from the real-world into virtual space. Small finger-level movements, such as typing, scrolling, or tracking, generally do not translate well into VR (though some hand controllers, such as the HTC Vive, have built-in trackpads and triggers). Movement at or above the hand-scale is more successful in VR. In order to facilitate the shift from finger-scale to body-scale movement, we conducted research into body-scale tools, including tools used to operate airplanes, submarines, and tractors. This research led us towards interface tools that require more physical movement, but even so, we must be considerate of the body-scale limitations of VR. Many gestures (extending arms, tilting heads) are sustainable by most people for short periods of time, but become uncomfortable and/or painful if held for longer periods of time; we must be mindful of these limitations when designing interfaces.

VR is spatial. It allows designers to simulate world space at many scales simultaneously and move seamlessly through those scales in ways that are impossible in the physical world. This is a tremendous opportunity to re-imagine how we think about navigation in a DAW. When developing VirDAW, we need to build specific tools that allow the user to move within and between environments, and those tools need to be intuitive and rewarding to use.

Keeping these considerations in mind, regarding both DAW functional needs and VR design space considerations, helped us clarify and remain focused on the specific set of questions related to developing VirDAW's user interface.

DETERMINING TOOLSETS

The VirDAW team is currently deep in the process of developing functional prototypes for demonstration purposes. If we want to deliver VirDAW to market, we will need to build from scratch with a team of developers, but since our current need is to develop playable prototypes, we looked to existing software applications that we can use to expedite the development process. When considering tools for prototype development, we evaluated them according to a clear set of criteria. The principle criteria was workflow; we wanted to avoid bottlenecking the entire project because only one team member had a required skillset, so we sought out software packages that our research team was familiar with or that had a shallow learning curve. Other criteria included a VR space that we could customize, a DAW back end that had components we could untether from skeuomorphic conventions, the ability to collect data points as well as play audio, and the ability to deliver sound in multiple audio formats (standard stereo, binaural, multi-channel,etc.). After considering a number of VR platforms, we concluded that the right solution for the VirDAW team at this phase of development was to use separate software applications for GUI and back-end development. Though both of the GUI platforms we considered could theoretically process the audio in the way we would need to, since our research team had more experience using other toolsets, we elected to use separate tools for GUI and back-end processing.

We considered three different VR systems for development. Microsoft's Hololens (https://www.microsoft.com/en-us/hololens) was rejected due to purchase cost and lack of available inventory. The Oculus Rift (www.oculus.com) was a strong contender, but ultimately the team chose to develop for the HTC Vive (https://www.vive.com/us/), primarily because we had access to a number of Vive headsets and could program and iterate in multiple labs simultaneously. For the GUI, the VirDAW team considered both the Unity (https://unity.com/) and Unreal (https://www.unrealengine.com/) game engines. Both engines are free to developers (they only collect revenue upon distribution) and both offer native VR development support. The decision to commit to Unity ultimately came down to the fact that the developers on the team overwhelmingly preferred Unity.

Deciding on back-end processing, where the playing and processing of sound files would occur, was a much more time-consuming process. Initially, the team was excited about Tracktion Engine (https://www.tracktion.com/develop/tracktion-engine), an open-source DAW toolkit for rapid development, but upon further inspection, the support documents were nonexistent; since no one on our team actually knew the Traktion products in the first place, we set Tracktion Engine aside and continued our search. Next, we looked at Wwise by Audiokinetic (https://www.audiokinetic.com/products/wwise/). Wwise is commonly used to implement sound design and music into video games, and it coordinates with Unity seamlessly. However, it did not have a feature set that was robust enough for our purposes without significant coding, so we continued our search even further. We next considered Ableton Live (https://www.ableton.com/), a popular DAW, but as we started thinking through the implementation of our design ideas, we realized that using any existing DAW, even for prototype development, would force us into thinking about DAW development in a 2D environment; so, we set it aside. Ultimately, the VirDAW team committed to Cycling '74's Max (https://cycling74.com/). Max is object-oriented, so team members can build and edit modules without having command-line coding experience. Max also does not default to grouping or representing sound skeuomorphically), so VirDAW's development can proceed without having to be backwards-compatible with a traditional DAW. Finally, all of the MFA and PhD members of our research team had varying levels of familiarity with Max, which let us avoid a bottlenecking problem.

Once Unity and Max were settled up on as the GUI and back-end applications, we needed a way to get the programs to speak to one another. We purchased OSC simpl (https://assetstore.unity.com/packages/tools/input-management/osc-simpl-53710) from the Unity Asset Store, which lets Unity and Max send Open Sound Control (OSC) messages back and forth between applications.

MODULE PROTOTYPES

Prototype development for VirDAW is currently focused on a handful of modules, each designed to engage the user in a common DAW feature that has been reconceived for the VR environment. In this section of the paper, we will introduce some of these modules and discuss our approach to the VR reconception.

REVERBERATION - Most software reverb tools provide an interface that combines parameters (ie, RT-60, LPF frequency, mix levels), programmatic preset names (ie 'Big Hall' or 'Tiled Bathroom'), and a 2D visualization. The primary method for the user to interface with the software is by adjusting some of the parameters with skeuomorphic knobs or faders. While this is a useful way for computers to think about reverb, it bears no resemblance to the way humans experience reverb. VirDAW replaces the parametric reverb interface with a Reverb Room. The designer can build a room in whatever shape they choose. They can add or move walls, adjust the curvature of surfaces to change the shape of the room, or cover surfaces with different materials (ie. shag carpet, maple panels, grass). Then, the designer can place the sound source and the listener within the room. VirDAW performs all the necessary calculations to determine the reverberation properties. Instead of reducing reverb to a series of computer-friendly digital parameters, VirDAW reconceives the entire process in a human-centered design that more closely matches the physical reality of how sound activates a space.

MIXING - Tools for mixing in DAW environments generally represent each track's volume level with a skeuomorphic fader embedded in the GUI. Instead of simply representing this same information in a track-based visualization within VR, VirDAW presents the designer with an open virtual expanse in which the designer may place sound emitters. Each emitter can represent a track or group of tracks, and the position of the emitter affects its aural performance. Volume is affected by distance from the emitter; a greater distance between the emitter and the receiver results in a lower volume in the mix. Reconceiving the practice of mixing from a track-based parametric system to an embodied simulation system lets the designer visualize the mix in an intuitive way that conventional DAWs do not support.

SPATIALIZATION - The process of placing and moving sound objects in most DAWs involves either using a skeuomorphic knob to represent position in the panoramic field (ie 'pan pots' for 2D sound delivery systems) or, for 3D sound delivery systems, using a 2D visualizer. Position information is entered either via parametric data fields or by manipulating a virtual emitter within an orthogonal projection of a 3D space onto a 2D screen. These approaches are neither intuitive nor easy to use, resulting in an under-application of 3D auralization. VirDAW approaches spatialization by empowering the designer to manually place sound elements within the virtual expanse, positioning each sound element in the 3D virtual world. The virtual position of the sound relative to the listener determines its position in the spatialized field. Finally, because the output format and the spatialization within the virtual environment are independent from each other, one mix (ie positions of emitters in a virtual space) can be rendered into stereo, binaural, 5.1, 7.1, ATMOS or any number of other standard or custom multichannel formats.

DYNAMICS - VirDAW's approach to working with dynamics in the 3D environment takes advantage of the early research we conducted in body-scale interactives. Instead of the traditional visualization of a compression curve as a bent diagonal line, we represent compression as a clamp that applies pressure to the audio waveform. Compression ratio and threshold are adjusted with a virtual interface that resembles a standpipe shut-off handwheel, which has the added reward of creating a stronger metaphor for communicating the process of sound compression.

SYNTHESIZER/SAMPLER - The VirDAW team is also developing a synthesizer/sampler application that uses the 3D environment as a musical instrument. The instrument functions by using triggers to execute events. There are two kinds of triggers in the VirDAW instrument: 'balls' move throughout the space with vector trajectories, and 'pulses' emit a sphere outward from a center point. Whenever a particular trigger comes in contact with a particular container module, an event (a note, a sample, a parameter change, etc.) occurs. There are four kinds of containers. 'Surfaces' allow balls to bounce off of them, changing the ball's direction; surfaces allow the designer to create a linear or loopable sequence of notes or events. 'Membranes' are essentially permeable surfaces; events occur as the trigger passes through the membrane, but the trajectory of the trigger is unchanged. 'Balls' can be events as well as triggers. Finally, 'fields' are 3D surfaces; they can contain a single event (such as enabling a low-pass filter), or they can contain a linear sequence of events (such as a filter sweep). The physical orientation of the field relative to the trigger can be manually adjusted, which provides the designer with the ability to easily adjust the timing of fields within the 3D environment. Building a synthesizer/sampler instrument within the VR module allows the VirDAW team to use 3D space not just in terms of sound localization, but also as a musical instrument.

NAVIGATION - A fully-featured DAW in a VR environment will need an intuitive and speedy way for the sound designer to navigate both within one environment (ie moving around within the reverb chamber) and between virtual environments (moving from the reverb chamber to the dynamics environment). Fortunately, these questions involve the greater VR research community, and the VirDAW team is able to draw on current research to inform its choices. Currently, movement within one environment is performed both through physical movement and teleporting. The team has not settled on an inter-environment movement methodology yet, but we have been following McVeigh-Schultz's use of "world pops" as an interaction metaphor (McVeigh-Schultz et al., 2018) with great enthusiasm.

NEXT STEPS

VirDAW is still in its early stages of development. We are focusing on developing a handful of key features (which we outlined in the previous section) as a series of standalone working prototypes with the belief that these features will demonstrate to users that VirDAW is a compelling way to engage with a DAW that can lead to innovative discoveries for sound designers and composers.

As the prototypes become playable, we plan to invite members of the sound and music industries to experience VirDAW with the goal of raising awareness and development support. Our current functional model (Unity and Max sending OSC data back and forth) is adequate for demonstration purposes, but any serious development of VirDAW into a standalone product will require a team of software engineers to re-code the content from the ground up.

The ultimate goal of VirDAW is to release a fully-featured professional-grade product into the marketplace.

CONCLUSION

As immersive sound systems become more common in all aspects of experiential sound design, the 2D mixing environment has been trying mightily to serve the needs of designers and composers mixing in 3D. However, forcing 3D design tools into a 2D visualization environment adds comprehension and effort costs for the user that deter designers from using the tools to their fullest potential. VirDAW seeks to provide the designers and composers with a new way to manipulate and create sound, in part by moving the whole experience into VR, in which the 3D environment can be accurately represented, and in part by re-envisioning the various DAW toolsets in ways that connect deeply to their functionality as opposed to replicating the decades-old traditional skeuomorphic visualization.

REFERENCES

Davies, D. (2003). Medium in art. *The Oxford Handbook of Aesthetics*, 181–191.

McVeigh-Schultz, J., Kreminski, M., Prasad, K., Hoberman, P., & Fisher, S. S. (2018). Immersive Design Fiction: Using VR to Prototype Speculative Interfaces and Interaction Rituals within a Virtual Storyworld. *Proceedings of the 2018 Designing Interactive Systems Conference*, 817–829.