

DROP SPIN FADE

Drop Spin Fade is the documentation of work completed so far on a collaborative project between Owen Lloyd and [Chris O'Shea](#).

Introduction

We were invited to take part in the [Future of Sound](#) tour, showcasing the 3d sound imaging system made by [Martyn Ware's](#) company [Illustrious](#). This system takes up to 16 audio inputs and positional data, enabling artists & composers to create soundscapes over a multi-layered speaker array.

Our aim was to work on a interactive composition system, and to present our work so far on each stage of the tour. The virtual space that would represent the physical audio space, using gestural video game technology as input. We wanted to create a playful and game like experience, with no goals or objectives. We began to play with the metaphor of the hands, what possibilities are there if you could sculpt, stretch, push and spin sounds.

Images

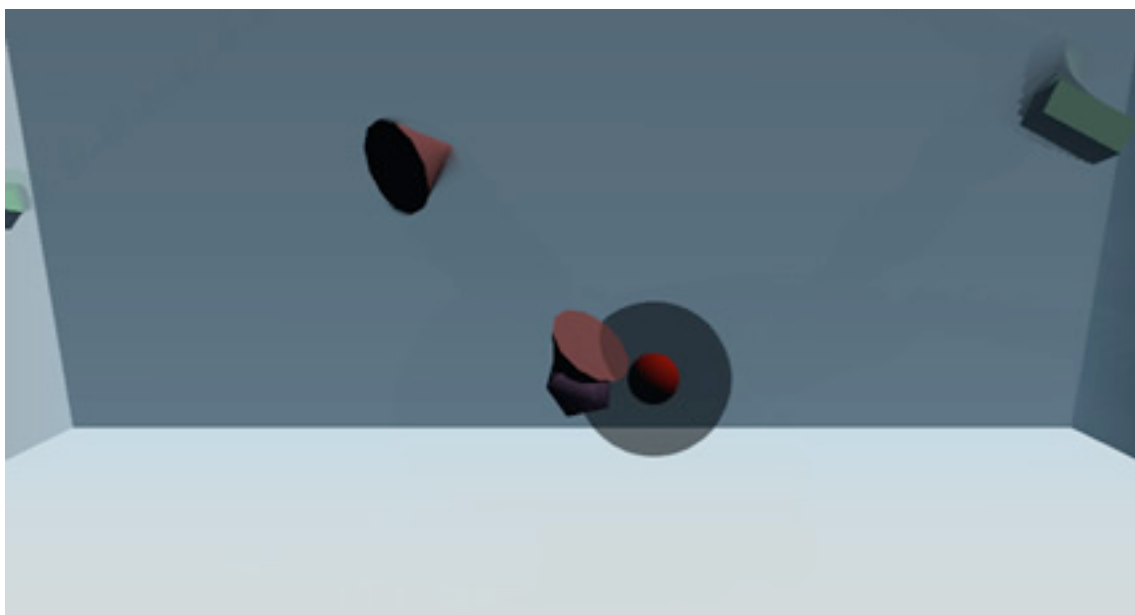
Process images can be found [here](#)

Input



[Gametrak](#) is a game controller, from [In2Games](#), that accurately tracks the position of your hands. The benefits are that it is low cost and low latency. By reading the HID data, then calculating the angles and distances, it is possible to use this hand tracking device outside of the games it was intended for.

Prototype1



This version was shown at FACT in Liverpool and at the Millennium Galleries in Sheffield. It was the initial prototype to get sounds moving around a 3D space with the [Illustrious](#) system and sound wise, the functionality was basic.

In this version the user could use their hands to push simple primitive geometric objects around a virtual 3D space. These objects had predefined linear sound loops attached, thereby providing spatial panning for the sounds over 16 channels. The cubed space represents the speaker space, with the listener standing in the centre.

Prototype 2



This version was shown at Goldsmiths College in London and was a more involved prototype using a granular playback engine for the sound. The granular engine allows more interaction with the sound and a direct connection between sound and object visuals. Instead of primitive shapes, the mesh shapes were modulated by the amplitude of the sounds they contained.

The rate of rotation of each object now controls the rate of playback for the sound contained, this applies bi-directionally, with reverse spin scrubbing through the sound backwards. The size of the object is also changeable and relates to pitch, large objects are pitched low and small pitched high.

There is also a degradation system affected by object collisions whereby sounds can collide a number of times with walls or other objects before they disappear. As they collide the sounds degrade in terms of their granular playback with each collision affecting their grain size, density and smear. There is also a synthesized collision sound, used sparsely, to underline this event from time to time.

Where next?

The direction we want to take this project in is one where the user has a more direct role in the creation of audio content. Users will be able to zoom into a sound object to compose melodies and rhythms, either through predefined or random paths, and with either fixed or free tempo. Within an object they will be able to load their own sound files for creation of sequences.

Moving out a level will take you to the effects level, creating affect paths on the sequence space. These paths will create xyz coordinate data that will be used to modulate effect parameters, cutoff, delay time, reverb time etc

We then zoom out again to the top level, which is the wireframe representation of what is going on within. This can then be moved around the space and positioned in a mix, fades can be created by sending an object slowly away for them to fade back in as they return and more subtle control will be built in.

A limitation of Gametrak is the single foot pedal button. We have now begun to focus on the Nintendo Wii controller, giving us more options for finer interaction and built in feedback.

Technical details

The visuals and interactive system were programmed in Lingo using Adobe Director. The sound engine programmed in Max/MSP. Communication between the two was via Open Sound Control, and position data sent to Illustrious via MIDI ([see sketch](#)).

Programming of Gametrak input was using the Joystick Xtra for Director. Prototype 3 interfacing the Wii controller is using [GlovePie](#) data sent over OSC.