Abstract

From rock concerts to musicals, multi-channel speaker systems create complex performances by surrounding the audience within a three-dimensional soundscape. However, such productions often require teams of people to create. Building an intuitive program that handles such computation and is controlled by a joystick gifts the solo artist yet another tool for artistic expression.

Manifold-Interface Amplitude Panning

MIAP is a suite of externals created by Zachary Seldess [1] which implements CueStation’s audio spatialization tool SpaceMap in the programming language Max [2][3].

SpaceMaps are graphical interfaces which map speakers on a manifold that is arbitrary from the physical layout of a space. An amplitude panning algorithm is used to compute speaker gains between a group of three nodes [3]. The picture below shows how SpaceMaps work.

Max is a visual programming language used to create music and multimedia programs [2].Patches are created by sending messages to and from a variety of objects on the canvas. The possibilities with Max are limitless.

The Process

Routing the Joystick

By far the most code intense parts of the program are the patches that route the joystick. Max must read the values of each element and manipulate them for use with MIAP. For example, the joystick position is composed of two elements, an x and y value, ranging from 0 to 1023 but the map ranges from -1000 to 1000. Furthermore, different program states may alter how the joystick is routed.

File IO and Audio IO

Like the joystick, various files and audio signals must be routed into Max in order for the program to work. These elements are coded separately from the joystick which makes the program more flexible. For example, one could change the audio from .mp3 files to MIDI data without altering the rest of the program.

Creating a GUI

Creating an effective user interface required a creativity that exceed standard coding practices. After determining which elements needed to be present in Max’s presentation mode they were orientated in a way that was both functional and visually pleasing. This process also determined which parameters needed to be controlled by the joystick.

The Product

- Single Map
- Dual Map, Same Attributes
- Dual Map, Different Attributes

Element

- X/Y Position
- Start/Stop Trajectory
- Change Virtual Node type
- Trajectory Rate +/- 50
- Traj. Repetitions +/- 1
- Reset trajectory offset/scale
- Trajectory x/y offset +/- 50
- Trajectory x/y scale +/- 0.5
- Audio input volume
- Change trajectories

Conclusion and Next Steps

Overall, I believe the programs I created do an excellent job of giving the user control of MIAP with a joystick. Although the code under the hood is complex, the GUI in Max’s presentation mode abstracts it away letting the artist’s creativity shine through.

Future work might do and answer the following:
- Create a program that controls four and eight maps
- Can the same SpaceMap be used in a studio and a theatre with 70+ speakers?
- What other instruments could a musician use to control the SpaceMap? Which ones work best?

Acknowledgments and References

Special thanks to Zachary Seldess for creating the MIAP externals. Copyright(c) 2014 Zachary Seldess.


Work supported by the National Science Foundation (NSF) Award #1560410