



# ARCHITECTING A SOUNDSCAPE:

A SPATIAL INTERFACE FOR DESIGNING  
A DYNAMIC SONIC ENVIRONMENT

Alex Scarlatos, Margaret Schedel (faculty advisor)



# RESEARCH AREA

Interaction design for sound mixing and experimentation.

Target Audiences:

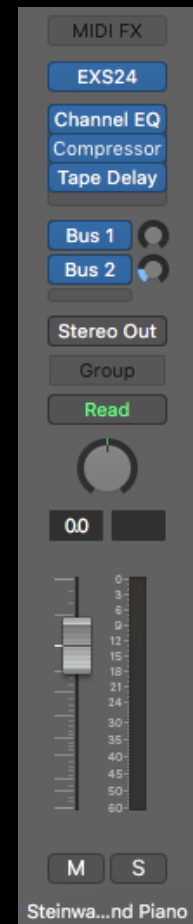
Novices and professionals interested in rapid prototyping.

# MOTIVATION

Modern tools for sound design are very good at controlling audio with high precision and efficiency.

But the workflow is complex, difficult to grasp, and time-consuming.

The workspace can grow very quickly and become difficult to manage.



Images From: Logic Pro X, Apple

# MOTIVATION

Working these effects requires detailed knowledge of what each parameter does.

While this is useful if you want very tight control over your sound and are being very precise about your mix, the existing interfaces are highly unintuitive, especially for sound novices.



Image From: Logic Pro X, Apple

# MY SOLUTION

What could a better tool look like?

Observations:

- Because of everyday experiences, people can associate physical variables with sound (ex: things will sound muffled on the opposite side of a wall).
- A commonly known fact of visualization is that planar variables are the strongest.

Conclusion: develop a spatial interface.

# Interface Idea

Imagine you are looking down into a room.

There is a microphone in the room, and you hear every sound that reaches it.

You are able to place objects in this room that make sound. As these objects move closer to the microphone, they become louder.

The room can have walls passing through it, creating sonic barriers in the space. Sounds will pass through and bounce off these barriers, changing as they do. You are able to draw walls freely, using various materials that will affect the sounds differently.

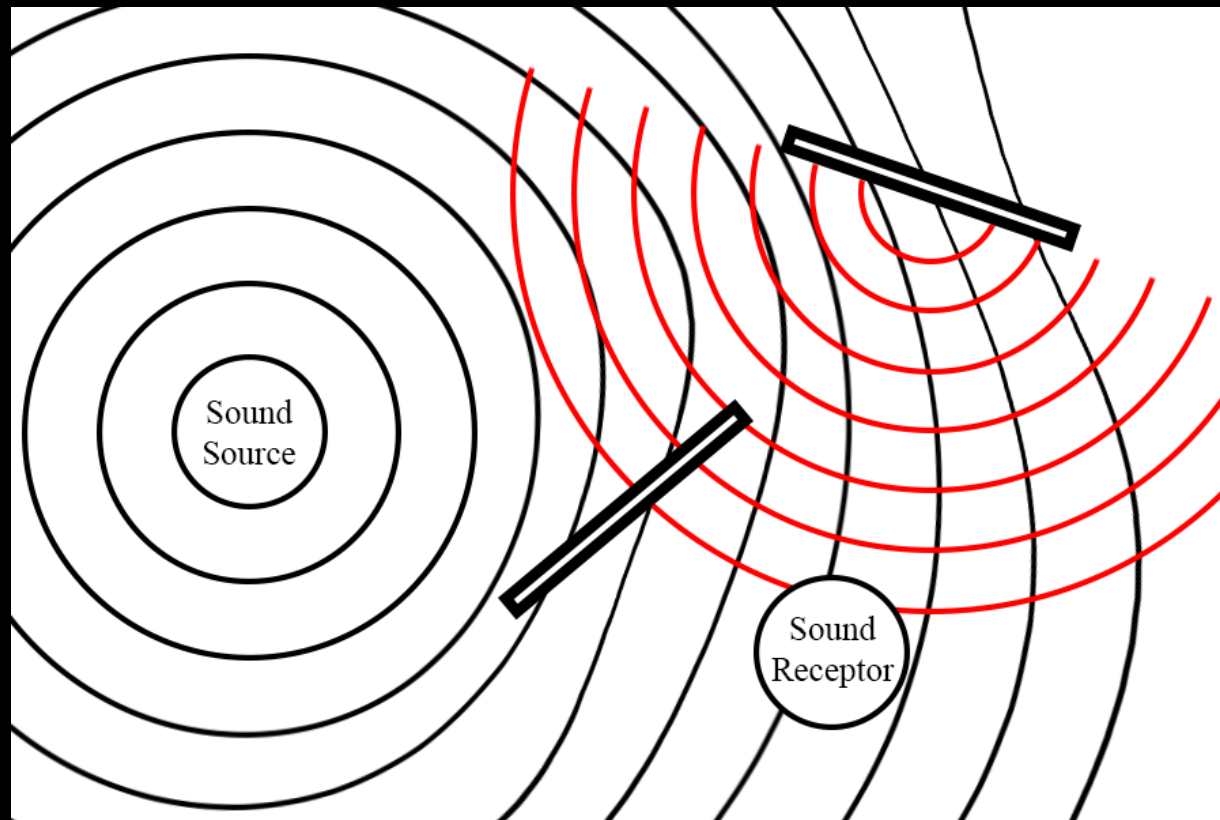
# Application in Music Production

The environment provides an easy to visualize, rapid prototyping environment for sound mixing. Engineers can hear changes as they draw and move objects. And all tracks and effects can be viewed at once, allowing for a quick analysis of a mix.



# Sound Interacting with Barriers

In reality, sound waves bend around barriers, and then normalize out over distance. Some sound will pass through these barriers because of resonance, and some will be reflected off the barriers.



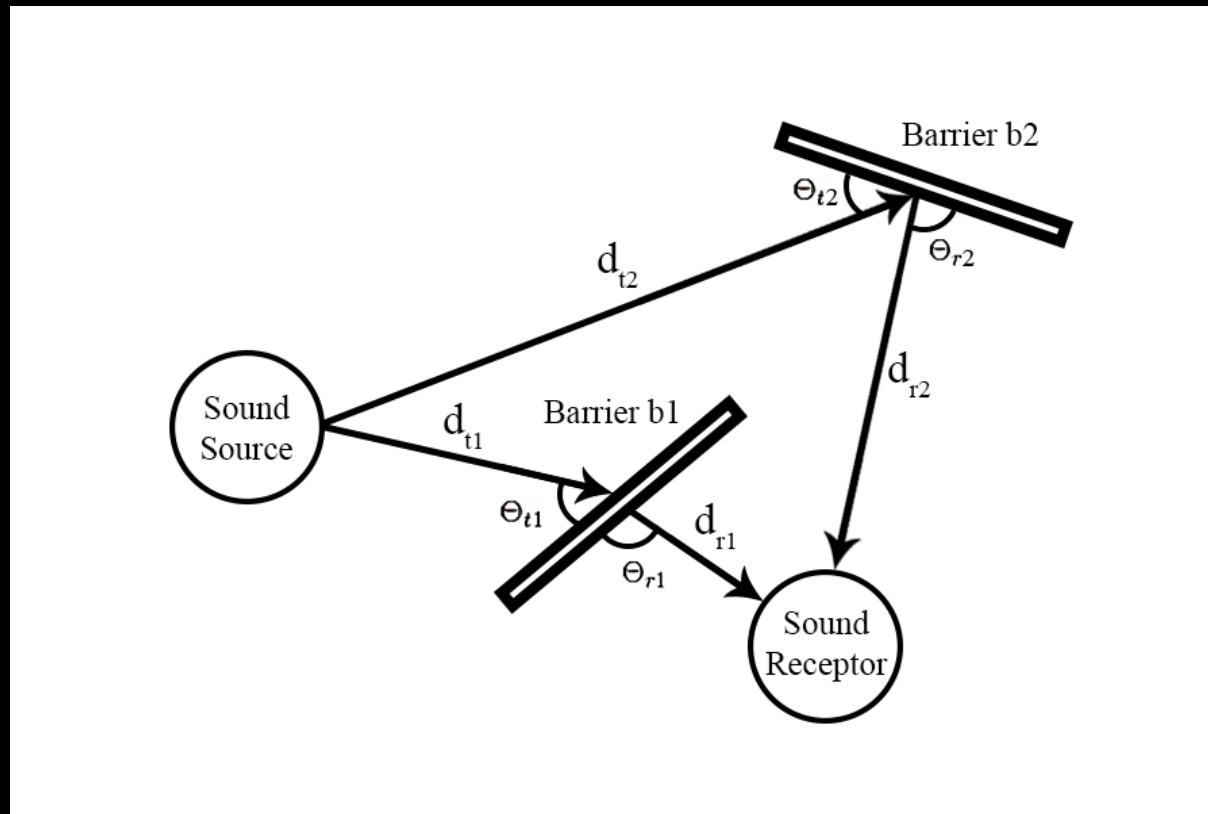




To simulate these properties, I had to make a few generalizations:

1. If a receptor is on the opposite side of a barrier from a sound source, the amount of the original signal received will decrease as it approaches the barrier. This relates to wave diffraction in physics.
2. In the same situation, while approaching the barrier, the amount of resonant sound received increases.
3. If the sound source and receptor are on the same side of a barrier, the amount of reflected signal received will increase as either node approaches the barrier.
4. If the barrier is infinitely thin and it is directly pointing at a sound source, no sound will interact with it. So, as the angle between a barrier and a node decreases, the above phenomena will be experienced less.

Simply by knowing the distances and angles between barriers and nodes, we can estimate how much a sound will be affected on its way to the receptor.



Each barrier is assigned an 'intensity' (a relative amount that it will affect sound), which increases as sources and receptors move closer and become more perpendicular to it.

# Formal Algorithm

*Output =*

$$\sum_{ss \in (\text{all sound sources})} \quad ss.\text{sourceData} * \text{distanceScaler} * (1 - D) \\ + \text{effectedData}(ss)$$

$$D = \sum_{m \in (\text{all materials})} D_m$$

*effectedData(ss) =*

$$\sum_{m \in (\text{all materials})} \left( \begin{array}{l} m.\text{diffractionFilter}(ss.\text{sourceData}) * \\ D_m = \log \left( \sum_{b \in (\text{barriers of material } m \text{ separating } ss \text{ and } r)} \text{intensity}_b \right) \\ + \\ m.\text{reflectionFilter}(ss.\text{sourceData}) * \\ R_m = \log \left( \sum_{b \in (\text{barriers of material } m \text{ aside } ss \text{ and } r)} \text{intensity}_b \right) \end{array} \right)$$

$$\text{intensity}_{\text{transmitted}} = |\sin(\theta_t)| * f(d_t)$$

$$\text{intensity}_{\text{received}} = |\sin(\theta_r)| * f(d_r)$$

$$\text{intensity}_b = \text{intensity}_{\text{transmitted}} * \text{intensity}_{\text{received}} * \text{b.materialScaler}$$

$f(x) = 1 / (ax + 1)$ : will approach 0 as x approaches infinity, a larger 'a' speeds the decline.  $\sin(\theta) = 1$  when the line between the node and barrier is perpendicular to the barrier, and it is 0 when the line is parallel. As a result, it acts as a perfect scaler for orientation. Absolute value of  $\sin(\theta)$  ensures that  $\theta > 180$  and  $\theta < 180$  are treated equally.

While the algorithm achieves the goal of being spatially dependent (rather than relying on pre-defined axes), it is not perfectly reflective of what happens in reality. For example, barriers do not affect each other by filtering sound already filtered by other barriers. Doing so would be computationally expensive, and would prevent the program from providing dynamic audio feedback. For future versions, I'm considering utilizing raycasting to better prioritize barriers.



DEMO

# OBSERVATIONS

I tested the app on several friends, as well as at URECA on an iPad.

## Positives:

- People seemed to be drawn in by the interface and enjoyed experimenting with different sound sources and effects.
- Some also enjoyed how it was modeled around physical acoustics.

## Negatives:

- The iPad couldn't handle the computations, and slowed down significantly with just a few items on screen.
- The whole app might be too computationally intensive.

# FUTURE WORK

- Add more user control
  - Give emitters the ability to ignore certain barriers.
  - Draw paths for emitters and receptors to follow.
- Make more useful for sound mixing & musical performance
  - Add more control over the effect parameters.
  - Allow dynamic swapping of audio files.
  - Add filters with more interesting effects.
  - Barriers will be able to affect musical parameters of generative emitters, like the tonality of a chord.
- Add in network aspect
  - Social network interface where users can share, view and edit each others creations.

