



Morgan Gomez

## The Inspiration:

As I was listening to music while looking for inspiration for the project, I had the idea to represent audio visually in a creative way. Whether it be words spoken by the user, a conversation between two people, or the sound of a metal container dropping on the ground, I wanted to be able to create a unique representation of what that would look like visually over a period of time.. Every time a sound is registered by the computer's microphone, instead of restarting the drawing, it is layered upon the pre-existing visuals, creating a layered history of the sounds registered since the user pressed the start button in Processing.

Each component of the visuals are attached to the microphone, so that depending on the amplitude, frequency, volume, along with other factors that can be analyzed by the microphone, different visuals will appear. For instance, the colors of the circles and sine waves are determined by a combination of the frequencies given from the left and right sides of the stereo input, and the opacity of the circles is determined by the amplitude, producing a wide range of possibilities depending on the sound registered by the mic. The position and size of these objects is also determined by these factors as well, creating an image that resembles abstract art.

## Use:

After the user imports the Minim library, all the user has to do is press start and make any type of noise they want. This can be loud, soft, abrupt, drawn out, essentially any type of sound. From there, visual representations of the user's audio input is visualized on screen through circles, sine waves, and geometric shapes corresponding to the different ranges of input. If the

user decides to redo their creation, they can press “R” on the keyboard and that will wipe the screen of visuals and the user can start over.

## Limits:

With the amount of analysis that the program has to do every second, I realized that the sine waves began to get less distinct as more input had to be analyzed to create the image. When first starting the project, the sine waves were the first objects created, and their colors were strong and vibrant. As I added the geometric shapes and a variety of circles that had to be created using the microphone’s data, the quality of the initial sine waves declined, sometimes resulting in sine waves that more closely resemble tears in the page rather than waves. Because the draw function runs multiple times per second, and within the draw function are for-loops, this slows the program and causes distorted sine waves. This can be seen in the second image created, as the sine wave in the middle of the image, while still visually appealing, is not what was supposed to be created.

```
int clipped = 0;
void draw(){

  fill(0);
  text("To reset, press R", 10, 15);

  //show the user that it is recording
  for(int i = 0; i < in.left.size() - 1; i++) {
    float freqMix = in.mix.get(int(i));
    float freqLeft = in.left.get(int(i)) ;
    float freqRight = in.right.get(int(i));
    float amplitude = in.mix.level();
```

In the two example done by different users, you can clearly see the differences in tone and volume of the different individuals as in the first example, there is more blank space and white circles, which shows a softer tone of voice compared to the second image, which shows the dark geometric shapes that only are expressed only with loud sounds. Both create beautiful works of art, but it can clearly be seen how the two individuals went about creating their pieces in completely different ways.

