README

TEAM MEMBER:

Shenghan Chen, Jincheng Xu, Siyang Wang

Q1: Approximately how many hours it took you to finish this assignment:

2.5hours (pls add your time on top of this)

Q2: Your overall impression of the assignment. Did you love it, hate it, or were you neutral? One word answers are fine, but if you have any suggestions for the future let me know.

I really loved it, this assignment has helped me understand many concepts that I overlooked during the lecture. I think it is very comprehensive -- the line intersecting face, using recursion/loop to find images of sources, and use of scene graph, transformation to world coordinate, sounds and impulse response handling -- it contains many interesting facets, which otherwise we would not be able to appreciate and comprehend completely had we not do this assignment. Also, this is one of the project cool to show around since it is very presentable. The fact that we can use web GUI and inspect element to test is very helpful, as it provides Integrated Development Environment usually not seen in scripting/loosely bounded language coding.

Q3: Any other concerns that you have. For instance, if you have a bug that you were unable to solve but you made progress, write that here:

**Section by Section:**

Core Techniques

Image Source Generation:

Path Extraction:

Impulse Response Generation:

In this section, we compute the impulse response of the sound traveled through different paths. We do this by first calculating the time it takes for the sound to travel through each routes, and calculating the intensities heard by the receiver for each path. The intensities decays can be obtained by multiplying the attenuation/decay using rcoeff at each faces the path hits, as well as 1/((1+distance traveled)^p).

Since in path, the first element is receiver, and the last element is the source, two of which without the rcoeff. To handle this special case, we calculate the distance of last path segment first before we enter into the for loop.

Once we have the time required for each path as well as the decays, we have the impulse response. We stored information in array, the size of which is the longest time of path (s) \* sample rate (44100/s). We then store each impulse response on its corresponding index, which in this case is (each time for arrival) \* (44100). When the index falls in between bins, we used a nearest index technique, basically assigning the impulse to the new index where the original index is closest to.