Assignment 7 – due 5/25/2017 at 11:59pm

Topic: Spatial Audio & Project Proposal

For the programming part of this assignment, you should use JavaScript with the Chrome browser. Other programming environments or browsers may also work, but will not be supposed by the teaching staff in labs, on piazza, and in the office hours. Students may use the computers in Packard 001, or their own laptops or desktop computers.

Please document all your answers, plots, and insights of tasks 1 and 2 in a single pdf file containing all requested results. Task 3, the project proposal, should be submitted as a separate pdf that is 2 pages long.

Please read Lab7.pdf before you start. You need to run an HTTP server on your local machine to start this homework. You will not need to connect Arduino to your machine except Task 2's Part C and Part D.

Task 1 of 3 (10 points)

Interaural level (amplitude) and time differences are the main cues we use to localize sound sources. The starter code includes an intuitive platform for performing a few simple audio experiments with your browser.

When the provided html file is opened, both audio channels play monaural music or a sound at equal gain without any delay between the two stereo channels. The panel on the top right allows you to change the gain ratio of the two channels and the delay of the sound and select a sound source among the provided audio files. If the gain ratio value is set to $\alpha$, the gain of the right and left channels are set to $2\alpha$ and $2(1-\alpha)$, respectively. The delay value defines the time delay of the left channel with respect to the right channel in milliseconds. If the delay value is set to negative, the right channel is delayed with respect to the left channel by its absolute time. Please use headphones or earphones to complete this task. We find that this experience is best with closed eyes.

Part A: (3pts)

In the first task, you will control the gain ratio without changing the delay (delay should be set to 0). By changing the gain values from 0 to 1, you should feel the sound sources moving from the left to the right. Report the gain ratio you felt the interaural level difference. Is there any difference between sound sources? Can you locate the sound source?

Part B: (3pts)

In the second task, you will control the delay without changing the gain ratio (ratio should be set to 0.5). Describe the experience you felt in your own words. Can you locate the sound source?

Part C: (4pts)

In this task, you will control both controls simultaneously. Find a set of values that you felt the sound source is located at the front right and the back right. If you couldn't find such values, report its reason.

Task 2 of 3 (40 points)
We'll now explore stereo sound and the improved localization and improvement of audio in a virtual environment, based on delay and attenuation of the human head, known as the Head Related Transfer Function (HRTF). The starter code comes preloaded with one of the HRTF data collected by the CIPIC lab at UC Davis (http://interface.cipic.ucdavis.edu/sound/hrf.html), as well a class to manage the convolution of the streaming audio samples with the Head Related Impulse Response (HRIR). In this task, we ask you to implement HRIR lookup based on the orientation of your head and apply the HRIR to multiple sound sources in the virtual world.

**Part A: (5pts)**

The CIPIC's HRIR dataset is sampled at unevenly spaced points based on the interaural coordinate system\(^1\). We need to know where along this space our sound source lies in order to find the best HRIR to apply. In **task2.js**, function `updateHRIR()`, calculates the azimuth `az`, elevation `el`, and Euclidian distance `d` from the sound source to the listener position. The sound and listener coordinates are in the world space, which has the same coordinate system of Three.js/WebGL. Update the variables `el`, `az`, and `d` with the resulting values. Report the azimuth, elevation, and distance to the default sound source in your writeup.

**Part B: (15pts)**

With a known location of the sound source in the interaural space, we can find the Head Related Impulse Response for both the left and the right ear that, when convolved with the audio sample, will provide the correct attenuation and delay to perceptually originate at the dedicated point in 3D space.

Since the HRIR dataset is sampled on a coarse sampling grid, we need to approximate the true HRIR from the dataset. To achieve this, we use the binary search method to find the closest sampling point from the true azimuth and elevation that we computed in Part A. In **hrir_container.js**, we have provided a function prototype `binarySearch(array, val)`. This function should return the element of `array` which is closest to `val`. Implement this function to perform the binary search.

At this point, you can listen to the scene in your headphones and the sound should perceptually originate from a 3D point in space. As in the previous homework, you can "walk" in the world by pressing W/A/S/D keys on your keyboard. If you're getting tired of hearing drum beats, we provide other sounds in the audio/bensound directory.

**Part C: (10pts)**

A static sound source isn't that interesting. Let's use the IMU and the orientation tracking we developed in Homework 5 to build a soundscape dependent on the head orientation. We have provided Arduino starter code with proper quaternion implementation to continuously stream quaternions. If you haven't connected Arduino, re-open the server after you connect your Arduino to your machine as described in the Lab 5 Writeup and the video. The starter code updates the `listenerQuaternion` based on the IMU.

Extend the `updateHRIR()` to incorporate the `listenerQuaternion` into the calculation. Since the starter code is loading the Three.js library, you can use their functions to help your computation.

Put on your headphones and move the IMU along the same axis of rotation that you move your head. Listen to how the sound changes depending on the distance and head rotation to the source.

**Part D: (10pts)**

---

\(^1\) The interaural coordinate system is one of the popular coordinate systems in psychoacoustics studies (http://interface.cipic.ucdavis.edu/sound/tutorial/psych.html).
To complete this task, you may need to understand some part of Web Audio API\(^2\). Web Audio API is one of the great features in HTML5 which allows us to process audio in an easy and efficient way. In short, Web Audio API requires you to construct a graph to process audio signals. The most simple graph would consist of two nodes: a source node and a destination node (two nodes / one edge). The source node, which is encapsulated in the Speaker class in our homework, streams audio signals to connected nodes. The destination node receives audio signals and play the signals on your headphones/speakers. Another simple graph would have multiple source nodes all of which directly connect to the destination node. In this case, all of the audio signals are played on your headphones simultaneously. You can set up multiple input sound source nodes and connects all of them to the destination node (five nodes / four edges).

The HrirConvolver class is encapsulating multiple processing and has an input node and an output node to be easily usable as if it is like an Web Audio API's AudioNode. The graph we have been using has one source node, one HrirConvolver node, and one destination node which are connected sequentially. As you can see in the starter code, you can establish a connection between two nodes by a function connect().

Extend `updateHRIR()` to support three sound sources. We recommend an array of sound sources and HrirConvolver nodes, constructed in the same manner that the original sound source but at a different position. Additionally, we have included a variety of sound files in the starter files. Assign a different audio track to the new sound source and listen to the scene.

**Task 3 of 3** (50 points)

Write a 2-page proposal for your course project. The project is coming up next and you will have two weeks to complete it. This is not a lot of time, so we’d like you to make a very clear and concrete plan for your project before you dive into it. This applies equally to students who are enrolled in the 3-unit and the 4-unit version of EE 267.

The proposal will be graded based on two criteria: **creativity** and **feasibility**.

Creativity can mean different things. If you choose to pursue the default project option, i.e. building a virtual environment, creativity includes scene layout and complexity, interactivity, and the quality of the general concept for your scene. For more hardware-focused or other types of projects, the creativity aspect includes how useful, novel, complex, etc. your idea is.

The feasibility aspect is important since we want you to be realistic. You have two weeks with only a few team members, so pitching projects that are clearly impossible would be ill advised. Usually, students overestimate how much can be accomplished in this short amount of time, so keep it simple and focused on one or a few key aspects or ideas.

An ideal project will pick a simple idea and explore it well, resulting in a demo of your results. The idea could be a type of user interface, a type of interaction, an input or output device, a specific perceptual effect that you want to explore, or some other concrete, well defined project. Focus on that one aspect, do your research by reading papers & online resources to understand and communicate your proposed idea to us in the proposal. Over the next two weeks, build a small demo to validate, test, or explore your ideas. The demo doesn’t have to be fancy; it’s better to focus on the key idea you are pitching.

The demo and evaluation of your idea is the only homework you will have after this week, we want you to propose a responsible, well researched project for review now.

---

\(^2\) The first section would be helpful for you to understand the concept of Web Audio API. Ref: https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API
You can use JavaScript and WebGL for the project or you can also use Unity or any other environment. If you want to work with the IMU or positional tracking, perhaps all your work will be done with the Arduino. We have a few professional HMDs (Oculus Rift, HTC Vive), depth cameras (Intel RealSense, Microsoft Kinect), and other hardware pieces (flex sensors, actuators, …) in the lab that you could borrow for your project. **However, if external hardware is critical for your project to succeed, make sure to discuss this with the teaching staff BEFORE you submit the proposal to make sure you will actually have access to it.**

Here are several important parts of the proposal that we will look for:

1. A clear motivation or problem statement. What are you trying to do? Articulate your objectives using absolutely no jargon.
2. A brief summary of previous work to demonstrate that you understand what’s already out there. How is it done today? What are the limits of current practice?
3. A clear statement on what is new, different, or creative in your approach.
4. A timeline with the tasks and milestones necessary to complete the project. Don’t embark on an open-ended mission - know when you will be done. Write down the specific end goal and the milestones necessary to get there. Associate a realistic time with each milestone.

**Questions?**

First, Google it! It’s a good habit to use the internet to answer your question. For 99% of all questions, the answer is easier found online than asking us. If you can’t figure it out this way, post on piazza and definitely make sure you attend the lab on Fridays (in Packard 001).