**Requirements:**(100 pts total)

1) (***7 pts***) Four buttons on the UI that let the user ***choose a context***. Each button will trigger a nicely looped clip of realistic background sounds. These background sounds will help a user explore whether your sonification scheme would meld sonically with the environment. The button also controls settings in the code that determine how to present and mix the different communication events based on the environment, interruptibility, and social setting.

2) (***5 pts***) Check boxes/radio buttons representing the five types of events present in a stream (Twitter, Email, MissedCall, TextMessage, Email). The user can ***choose which event types she wants to hear/receive***. User can turn these on and off in real-time. The number of active event types will affect how your sonification scheme chooses to present and/or mix sounds.

3) (***3 pts***) There is a button/checkbox that turns a “heartbeat” sound on and off. This is a continuous unobtrusive sound of your own design that ***indicates the system is working***.

4) (***5 pts***) Three buttons on the UI that let the user choose ***which example event stream (JSON) to load***(the three example JSON streams will be provided by the instructors).

5) (***10 pts***) System does ***real-time sonification of the data from the event stream***

6) (***25 pts***) System has a ***dynamic sonification “engine”*** that chooses which events to sonify, how to sonify them, and how much information from the event to present, based on the current context and the number of active streams as determined by the check boxes. The scheme changes significantly and dynamically based on these factors

7) (***5 pts***) Your engine must utilize ***auditory icons and earcons***

8) (***5 pts***) Your engine must ***synthesize at least one of the sound cues dynamically*** in code, rather than by playing a premade sound file.

9) (***5 pts***)You must utilize ***text-to-speech***in some manner in at least one of the context modes.

10) (***10 pts***) ***Individual event sonifications*** are fitting for the event type and context.

11) (***10 pts***) Groups of simultaneous sonifications (and the “heartbeat” indicator) are ***mixed appropriately***.

12) (***10 pts***) A ***design document that explains the reasoning and motivations behind your design choices***. Describe how your sonification scheme works for each event type and context. Cite any related work/additional resources you used to inform your design choices.

**Extra Credit:**

You can receive up to 10 extra points if you implement a feature that allows you to review past events in a relatively short aggregated sonification scheme that the user can access to get an overall sense of what types of information have been received over a previous span of time (For example, the NR project had a mode where messages were spatialize around the user’s head like a clock face and could be scanned quickly to get an overall picture of the information)

Design Document – Midterm

Going through the UI and the buttons’ various sound designs first, the UI has the 5 checkboxes for the events on the far left, the contexts in the middle, and the heartbeat and 3 JSONS on the right. The Engine Java file makes the intelligent decisions on scheduling the events.

For the Working out context I added a sound file that depicts a running treadmill. There was an audio scratching sound that existed in the background so I shaved that off by using a low-pass filter at 60 dB and then normalized and lowered the gain on OcenAudio. For the walking sound clip it was super quiet when I downloaded it, so I normalized it which really emphasized the footsteps, I experimented with a low and high pass filter, the former muffled the footsteps and the latter made it sound like he was stepping on leaves, so I reverted those changes. The clip itself is supposed to depict a person wearing sneakers and walking on concrete. For the socializing clip I got a sound clip from a crowded café. I sound quality was pretty good, so I just normalized the clip and a low pass filter at 1478 Hz. The presentation audio clip is actually of a quiet classroom where students are breathing, moving chairs, using utensils, etc. The beginning of the clip was really silent and didn’t seem to sonify any events going on, so I cut that off, and then I increased the volume slightly because it seemed fairly quiet.

The heartbeat was synthetically made in the createHeartbeat() method of the contexts file. Though it doesn’t mimic the beating of a heart, it provides a low unobtrusive buzz through a very low frequency of a sine wave that is the product of two sine waves, the carrier wave and the input. I then lowered the gain to a level where the noise is just slightly noticeable. ([Source](https://www.premierguitar.com/articles/4_Unconventional_Effects_and_How_To_Use_Them?page=3) 1)([Source](http://www.computermusicblog.com/SonifyingProcessing/Sonifying_Processing_The_Beads_Tutorial.pdf) 2, pg. 28)

For the five events (Tweet, text, voicemail, phone call, and email) I used auditory icons. For twitter, I used the recognizable sound file of a tweet and the same for a text message. For voicemail, I had a robot lady do an elaborate “message received” procedure and utilized Bugs Bunny and AOL for my email auditory icon. For the phone call sound file I trimmed the playtime of the music file. For the messages that the events carried, I had the sender and message both be sonified through text to speech utilizing the ttslib Processing library.

The “engine” is primarily in the engine java file.