

# The Most Awesome Auditory Tone Experiment Ever

Title contains no bias whatsoever.

## Methods

We constructed a computer program for the Linux/GNU 2.6.18-5-686 kernel Debian etch that generated tones using the sound card in a specified sequence to test whether or not the victim could perceive distinctions between said tones. The purpose of these tests were to determine Just Noticeable Difference (JND) and the absolute threshold where the stimulus is perceived 50% of the time.

## Materials

Koss titanium headphones  
Gateway M675 laptop  
Sigmatel® Soft Audio 6-channel AC97 revision 2.3 codec (STAC9758)  
Desk  
Pen  
Data table spreadsheet  
The computer program (see Appendix A)

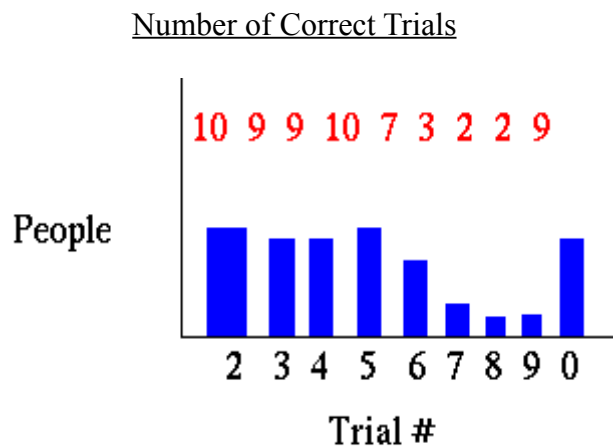
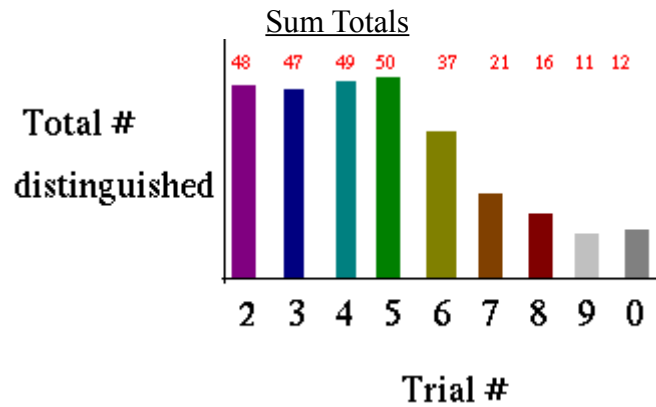
## Results

The Truth

Test number	First tone	Second tone
0	1300 Hz	1300 Hz
2	1300 Hz	1200 Hz
3	1300 Hz	1400 Hz
4	1300 Hz	1350 Hz
5	1300 Hz	1250 Hz
6	1300 Hz	1325 Hz
7	1300 Hz	1315 Hz
8	1300 Hz	1310 Hz
9	1300 Hz	1305 Hz

Reported Tone Distinguishments

	Test->	2	3	4	5	6	7	8	9	0
Person										
1		3	4	3	4	4	1	2	1	3
2		4	4	4	4	4	4	4	4	4
3		4	3	4	4	4	2	4	4	4
4		4	4	4	4	0	0	0	0	0
5		4	4	4	4	0	1	0	0	0
6		4	2	3	4	2	0	3	0	1
7		2	3	3	2	0	0	0	0	0
8		4	4	2	3	3	1	0	0	0
9		4	4	4	4	4	4	0	0	0
10		4	4	4	4	4	4	2	0	0
11		2	2	4	4	2	0	0	0	0
12		2	1	2	4	2	1	0	0	3
13		4	4	4	3	4	0	0	0	0
14		4	4	4	2	4	3	1	2	0



## Analysis / Discussion

The results show that the number of completely 'correct' perceptions is maximized at trials 2 (1300 & 1200 Hz), 5 (1300 & 1250 Hz), and that on average just as many people who were able to get 2 & 5 were able to get 3 & 4 which have similar differences in tonal frequency. Between tests 8 & 9 there was a constancy of the number of people that were able to get it correct, suggesting that typically a 5 Hz difference in tone cannot be perceptually differentiated. (In fact, one student that performed the most accurately mentioned she does piano and has to have a good ear. Or did she merely start playing because she had a good ear to begin with?). Out of the 66 trials for each test, it seems that the absolute threshold for the class is around trial 6 where there was a drop from 1350 Hz to 1325 Hz to test against 1300 Hz. After trial 5 the graph takes a curve down (concave up) suggesting rapidly reducing ability to differentiate.

## Conclusion

Our experiment tested for JND by successfully testing increasingly smaller tonal differences. Additionally absolute threshold was detected by trial 7 where most participants were either completely sure of each or not (with one or two glitches appearing). These glitches are due to perceptual cues that make people think that they know what they sensed, or the torment of thoughts that lead one to 'reason' out what they had perceived, which is a process that can possibly dilute the results. An effective alternative would be to implant microelectrodes into the auditory cortex of ~~the rats the victims~~ the students to detect neural spikes and correlate the information to the tones played- for this reason the lab didn't completely effectively detect perception, but rather translated/opinionated perception that has been filtered through the prefrontal cortex for a few moments.

With this new knowledge of perception, we can further understand how it is that people are unable to hear certain sounds, while still clearly and crisply distinguishing other tones and ranges of audio. Maybe this will serve as incentive to turn down the headphones (ahem, Puckett).

The most notable knowledge-gain has been the wide-range of perception per each stimulant, suggesting that any form of stimulation (inc. education) would have to be fixed appropriately to each student, listener, and so on. The experimental methods were sound because the computer could generate the same tones each time as well as the ability to test multiple times with minimal preparation (move hand, check pen ink, etc.).

Lab partner performance: A+ vampire slayer

No	Points	Name
1	161761	(Grayson) died in The Dungeons of Doom on level 2 B building.
Killed by Puckett.	-	Max HP: 116

## Appendix A

```
/* Convert text to beeps. */
/* Modifications by Bryan Bishop 2007 ...
originally as a bible-to-beep program. */
/* Originally: beep.c - something i wrote while bored at work */

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/ioctl.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <linux/kd.h>

#include <signal.h>

#include <linux/vt.h>

#include <stdio.h>

void play_note(int fdesc, int frequency, int duration);
void interrupt();

int fd;
FILE *fileMine;

int main(int argc, char *argv[]) {
    //FILE *fileMine;

    int intDone;
    char charInput;
    int intTheCharInput;
    char *charSecond;
    // printf("hello.\n");

    if((fd = open("/dev/console", O_WRONLY | O_APPEND|O_CREAT)) < 0)
    {
        printf("ERROR: could not open /dev/console\n");
        exit(1);
    }
    int input;
    while(1) {
        printf("Press a number associated with a test.\n> ");
        //while(input == ' ') { fgets(input,1,stdin); }
```

```
//input[0] = getchar();

scanf("%d", &input);

if (input == 1) {
    play_note(fd, 1300, 200);
} else if (input == 2) {
    play_note(fd, 1300, 200);
    usleep(30000);
    play_note(fd, 1200, 200);
} else if (input == 3) {
    play_note(fd, 1300, 200);
    usleep(30000);
    play_note(fd, 1400, 200);
} else if (input == 4) {
    play_note(fd, 1300, 200);
    usleep(30000);
    play_note(fd, 1350, 200);
} else if (input == 5) {
    play_note(fd, 1300, 200);
    usleep(30000);
    play_note(fd, 1250, 200);
} else if (input == 6) {
    play_note(fd, 1300, 200);
    usleep(30000);
    play_note(fd, 1325, 200);
} else if (input == 7) {
    play_note(fd, 1300, 200); // 1300 - 1315

    usleep(30000);
    play_note(fd, 1315, 200);
} else if (input == 8) {
    play_note(fd, 1300, 200);
    usleep(30000);
    play_note(fd, 1310, 200);
} else if (input == 9) {
    play_note(fd, 1300, 200);
    usleep(30000);
    play_note(fd, 1305, 200);
} else if (input == 0) {
    play_note(fd, 1300, 200);
    usleep(30000);
    play_note(fd, 1300, 200);
    /*play_note(fd, 500, 100);
    play_note(fd, 300, 100);
    play_note(fd, 900, 100);
    play_note(fd, 534, 100);
    play_note(fd, 1500, 100);
    play_note(fd, 320, 100);
    play_note(fd, 780, 100);
    play_note(fd, 990, 100);*/
} else {
    printf("Help! I'm trapped on the internet!\n\n");
}
```

```
    }
    input=0;
}

/*
    if(argc == 2) {
        fileMine = fopen(argv[1], "r");

        intDone = 0;
        while(intDone == 0) {

            charInput = getc(fileMine);
            if(charInput == EOF) intDone=1;
            else {
                // printf("Hello. The charInput is %c\n",
charInput);
                // sprintf(charInput, "%d",
intTheCharInput);
                // sprintf(intTheCharInput, "%c",
charInput);

                intTheCharInput = getFreq(charInput);

                signal(SIGINT, (void *)interrupt);
                //sprintf(charSecond, "echo", charInput);
                // printf("%c", charSecond);

                // if(charInput != '\n') execl((char *)'',
charSecond, ); // write(fd, charInput, 1); // fprintf(fd, "%c",
charInput); // putchar( charInput);

                //printf("--> %c\n", charInput);

                play_note(fd, intTheCharInput, 200);

                printf("--> %c\n", charInput);

                //printf("Playing note with frequency
%d\n", intTheCharInput*5);
            }

        }

        fclose(fileMine);
    } else {
        printf("usage: %s <file to read>\n", argv[0]);
    }
*/
exit(0);
}

/*
int main(int argc, char *argv[]) {
    int frequency = 1000; // in ?hz
    int duration = 2000; // in ms
    // int fd;
```

```
    int temp = 0;

    unsigned long int count = 0;

    char **endptr = NULL;

    if ((fd = open("/dev/console", O_WRONLY)) < 0) {
        printf("ERROR: could not open /dev/console.\n");
        exit(1);
    }

    if (argc == 3) {
        temp = (int)(strtoul(argv[1], endptr, 10));
        if (endptr == NULL)
            frequency = temp;

        endptr = NULL;

        temp = (int)(strtoul(argv[2], endptr, 10));
        if (endptr == NULL)
            duration = temp;

        signal(SIGINT, (void *)interrupt);
        play_note(fd, frequency, duration);
    }
    else
        printf("usage: %s <frequency> <duration>\n", argv[0]);

    exit(0);
} */

void play_note(int fdesc, int frequency, int duration) {
    if (frequency <= 0)
        frequency = 1;

    ioctl(fdesc, KIOCSOUND, 1190000 / frequency);
    usleep(duration * 1000);
    ioctl(fdesc, KIOCSOUND, 0);
}

int getFreq(char bob) {
    /*
    if(bob == '!') return 100;
    if(bob == '"') return 140;
    if(bob == '#') return 160;
    */
    return 114+((int)bob * 5);
}

void interrupt() {
    printf("Terminating...");
    ioctl(fd, KIOCSOUND, 0);
    fclose(fileMine);
}
```

```
    exit(0);  
    #if (SYSTEM == UNIX && FLAVOR == ATT)  
        signal(SIGINT, (void *)interrupt);  
    #endif  
}
```