LEVEL ASSISTANT — A SONIFICATION-BASED SPIRIT LEVEL APP

Justin Niestroj

Institute of Systematic Musicology
University of Hamburg
Hamburg, Germany
`bbb3870@studium.uni-hamburg.de`

ABSTRACT

The idea of sonification is the use of sound to transfer information. This explicitly excludes the use of speech. For the Sonic Tilt Competition, this concept for an auditory "Level Assistant" was envisioned. This concept, unlike others before it, builds on the preexisting knowledge of acoustical parking assistants (a common form of sonification) in order to make the app as intuitive as possible. Through the implementation of intuitive directional cues and a known concept of beeping faster when more towards the extreme, it is hoped that many more people will find sonification-based apps useful for everyday work.

1. LINK TO APK FILE

The APK file of the Level Assistant app can be found under `https://github.com/Tiltification/sonic-tilt/tree/levelassistant/build/app/outputs/flutter-apk`.

2. INTRODUCTION

Sonification is hard. We as humans have very few areas in our lives, where sound is the primary giver of information. And when it is, the information is usually aided with the help of visual elements. Thus, existing sonification for a level often struggled to be intuitive and useful. This was a repeated criticism of Tiltification [1] despite overall positive ratings [2]. The idea of "Level Assistant" was to try and find an existing use of sonification and build on that in order to be able to draw on existing knowledge and intuition of the users. The most common use of sonification that most people know of is the parking assistant. The closer the driver is to an object outside the car, the faster the beeping will be. This is intuitive, as faster beeping is perceived as more urgent [3].

3. LEVEL ASSISTANT

Building on the idea of the parking assistant, "Level Assistant" uses the concept of beeping to sonify the level. Originally, acoustic car parking assistants only communicated the distance of the closest obstacle by means of the inter-onset interval, which is a common sonification strategy [4]. Modern assistants add directional cues through spatial audio and/or direction-dependent timbres [5]. The logo see 1 is inspired by the visual aids of parking assistants, combined with the black and white circles sometimes found in bullseye levels.

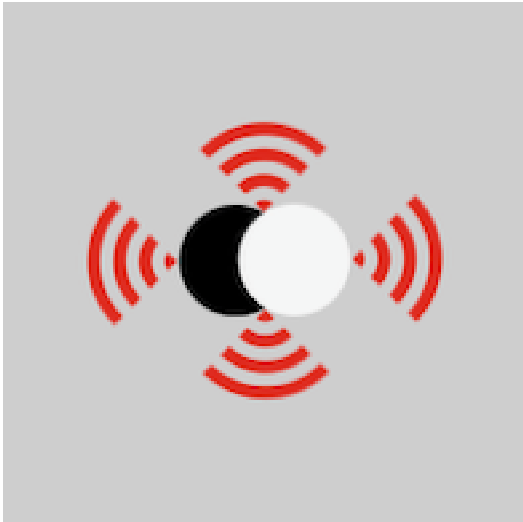

Figure 1: "Level Assistant" Logo

"Level Assistant" uses two sounds, one for each axis. The L/R (Left/Right) axis is represented by a cowbell or metallic beeping, and the T/B (Top/Bottom) axis is represented by a woodblock or wooden beeping. These different timbres are easily distinguished, and timbre is known to be rather categorical [6]. The more extreme the tilt is (up to 45◦) the faster the beeping will be. Once the user reaches the 0° level, the beeping will stop and a "leveling sound" will play ("done"). In order to determine in which direction the level is tilting, both sounds have three variations: High, normal, and low pitch. The "normal" or center pitch is heard when close to 0°. Again, "Level Assistant" draws on human intuition to determine which direction is associated with high or low. Most of the world has a basic knowledge of a piano keyboard and even if not, a lot of pop music is mixed with the piano's low notes being left and the high notes being right. Examples include "idontwannabeyouannymore" from Billie Eilish and "drivers licence" from Olivia Rodrigo. That is why the metallic beeping of the L/R axis will beep lower when tilted left and higher when tilted right. The intuitive direction for the T/B axis is even clearer. Tilted up will be

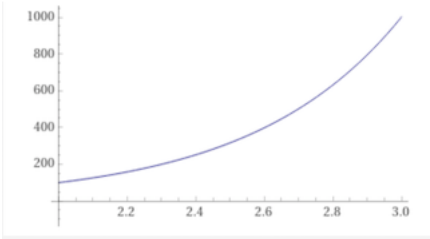

Figure 2: Metronome Speed Plot

higher and tilted down lower. This way, the only non-intuitive part of the level is what sound is what side. This though should easily be found out and remembered.

4. IMPLEMENTATION

The project was written and envisioned in Pure Data. In the following, this will explain how the code is written and executed. The code is also documented in the Pure Data Patch.

The patch is divided into three parts. The center part is the general cycle. A metro object is used to count upwards and a cycle object that cycles to 4 gives our 4 values. These values are the four stages of the code which are executed after each other in a set time frame (500 ms). The change and select objects output the four values 0, 1, 2, 3. These correspond to the four stages:

$$0 = \text{L/R axis Start} \tag{1}$$

$$1 = \text{L/R axis Stop} \tag{2}$$

$$2 = \text{T/B axis Start} \tag{3}$$

$$3 = \text{T/B axis Stop} \tag{4}$$

The 2nd and 3rd part are the two axis: The L/R and T/B axis work the same. The first part of the axis patch is the metro object speed: The absolute value of the sensor outputs that lie between -0.5 and 0.5 determine playback speed and only communicate the distance along the axis, not the corresponding side. Then all values receive a negative sign using the input *(-1) (-0.5-0—0.5) and multiplied (*2) (-1-0-1). Then that value adds (+3) (2-3-2). Because of this, values now generate between 100 and 1000 when used in the exponential function 10 pow $2 - 3$. These values are now used to determine the speed of the metronome (see Fig. 2).

Example: The extreme value L: -0.5
Abs= 0.5
*(-1)= (-0.5)
*2=(-1)
+3 = 2
10 pow 2 = 100

The value of 100 is then used to trigger the metro object every 100 ms. The other extreme is to trigger the metro object every 1000 ms and because of the exponential function, the values between the two extremes will be more exact the closer one is to the center.

The second part is the selection of high, normal, and low sound. This is done by using an if expression with $f1 representing the input:
expr (if($f1 < -0.1, 0, if$(f1>0.1, 2, 1))).
It then outputs a value and selects which bang to trigger. The amount of bangs is then triggered by the metro object output speed, as discussed earlier.

The last part is the "done" part. This sound is triggered when the level reaches a pre-determined value close to $0/0$ on both axes. The two inputs send a value to the expression, with $f1 being L/R and the $f2 being the T/B:
expr (if($f1+$f2<-0.05, 0, if($f1+$f2>0.05, 2, 1)))
If the expression outputs 0 or 2, nothing happens, and the general cycle continues. If it outputs 1, it triggers the stop of the general cycle and triggers the playing of the "done" sound file, which is the leveled sound. As soon as the expression outputs 0 or 2 again, it will trigger the general cycle. Unfortunately, the "done" sound can still be triggered multiple times in quick succession, making fine adjustments hard. The goal of having a"cooldow" time, a time after the triggering of the sound, before it can be triggered again, could not yet be realized and will be worked on in a future version.

5. REFERENCES

[1] M. Asendorf, M. Kienzle, R. Ringe, F. Ahmadi, D. Bhowmik, J. Chen, K. Huynh, S. Kleinert, J. Kruesilp, Y. Lee, X. Wang, W. Luo, N. Jadid, A. Awadin, V. Raval, E. Schade, H. Jaman, K. Sharma, C. Weber, H. Winkler, and T. Ziemer, "Tiltification — an accessible app to popularize sonification," in *Proc. 26th International Conference on Auditory Display (ICAD2021)*, Virtual Conference, June 2021, pp. 184–191. [Online]. Available: https://doi.org/10.21785/icad2021.025

[2] T. Ziemer and N. M. Jadid, "Recommendations to develop, distribute and market sonification apps," in *The 27th International Conference on Auditory Display (ICAD 2022)*, Virtual Conference, June 2022. [Online]. Available: https://doi.org/10.21785/icad2022.003

[3] E. J. Hellier, J. Edworthy, and I. Dennis, "Improving auditory warning design: Quantifying and predicting the effects of different warning parameters on perceived urgency," *Human Factors*, vol. 35, no. 4, pp. 693–706, 1993.

[4] G. Dubus and R. Bresin, "A systematic review of mapping strategies for the sonification of physical quantities," *PLOS ONE*, vol. 9, no. 4, p. e96018, 12 2013.

[5] A. Lundkvist, R. Johnsson, A. Nykänen, and J. Stridfelt, "3d auditory displays for parking assistance systems," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.*, vol. 10, pp. 17–23, 04 2017. [Online]. Available: https://doi.org/10.4271/2017-01-9627

[6] S. Barrass, "A perceptual framework for the auditory display of scientific data," in *International Conference on Auditory Display*, Santa Fe, Nov 1994, pp. 131–145. [Online]. Available: http://hdl.handle.net/1853/50821