Structure and interpretation of Computer programs

Module

Sound

1. adsr(attack_ratio: number, decay_ratio: number, sustain_level: number, release_ratio: number) Return: function(sound - sound)
   这个函数是一个包络生成器（envelope generator），用于调整音频信号的振幅。它接受四个参数：attack_ratio（攻击比例）、decay_ratio（衰减比例）、sustain_level（持续水平）和release_ratio（释放比例）。
   当将这个包络应用到一个音频信号上时，它会返回一个新的音频信号，其振幅根据参数进行修改。具体地说，相对振幅会在攻击阶段线性地从0增加到1，然后在衰减阶段从1降低到持续水平，保持在该水平上直到释放阶段，最后在释放阶段中振幅会衰减回0。
   这个函数的返回值是一个函数（Envelope），它接受一个音频信号作为输入，然后返回经过包络调整后的新音频信号。通过调整包络参数，可以实现不同的音频效果，如渐入渐出（fade in/out）或音频包络塑形（envelope shaping）。
2. bell(note: number(MIDI note), duration: number). 钟 Return: sound
   Sound resulting bell Sound with given pitch and duration
3. cello(note: number(MIDI note), duration: number). 大提琴 Return: sound
   Sound resulting cello Sound with given pitch and duration
4. consecutively(list_of_sounds: List). Return: sound
   创建给定sound list的连续新sound
5. get_duration(sound: Sound) Return: number(duration)
6. get_wave(sound: Sound) Return: wave function of the Sound
7. init_record(). Return: string "obtaining recording permission"
   获取设备默认麦克风的使用权限，初始化录音
8. is_sound(anything_to_be_checked) Return: true if x is a Sound, false otherwise
   Checks if the argument is a Sound
9. letter_name_to_frequency(note: string) Return: the corresponding frequency
   Example: letter_name_to_frequency("A4"); // Returns 440
   根据函数的参数和示例，可以推测这里的 `letter_name` 是指音符的字母名称表示。音符的字母名称是指音乐中使用的字母符号来表示特定音符，通常以英文字母 A 到 G 来表示音阶中的音符。
   在音乐理论中，字母名称用于表示音符的基本音高。字母名称后面可能还会带有一个数字表示音符所在的八度（octave）。在示例中，"A4"代表A音符在第4个八度上的音高。
   因此， `letter_name_to_frequency` 函数应该是将音符的字母名称作为输入，并返回相应的频率值。在示例中，它将接受"A4"作为参数，并返回对应的频率440。这意味着"A4"音符的频率为440赫兹（Hz）。
10. letter_name_to_midi_note(note: string) Return: the corresponding midi note
11. make_sound(wave: Wave, duration: number) Return: sound

12. midi_note_to_frequency(note: number) Return: number
13. noise_sound(duration: number) Return: sound
14. phase_mod(freq: number, duration: number, amount: number). Return: soundtransformer
    返回一个SoundTransformer，它使用它的参数用给定的声音调制给定频率和持续时间的(载波)正弦波的相位。用低频声音调制会产生颤音效果。用频率与正弦波频率相当的声音调制会产生更复杂的波形。
15. piano(note: number(MIDI note), duration: number) Return: sound
16. play(sound: Sound)
17. play_concurrently(sound: Sound)
    使用计算机的声音设备在当前正在播放的任何声音之上播放给定的声音。
18. play_wave(wave: Wave, duration: number)
    使用计算机播放给定声音，并且计算机设备不能有任何正在播放的声音
19. record(buffer: number)
    这个函数是一个录音器，用于在指定的缓冲时长内进行录音。它接受一个参数 `buffer`，表示录音的缓冲时长（以秒为单位）。
    函数返回一个无参数的停止函数 `stop`。调用 `stop()` 函数将停止录音，并返回一个声音承诺（sound promise），它是一个无参数函数，用于获取录制的声音。 总结：
    - 调用 `init_record()` 进行初始化。
    - 调用 `record()` 并传入缓冲时长 `buffer` 来开始录音，返回一个停止函数 `stop`。
    - 调用 `stop()` 停止录音，并获得一个声音承诺 `promise`。
    - 在下一个查询中，调用 `promise()` 来获取录制的声音，并使用 `play()` 函数播放。
20. sawtooth_sound(freq: number, duration: number)
21. silence_sound(duration: number) Return: sound
22. simultaneously(list_of_sounds: List) Return: sound
23. sine_sound(freq: number, duration: number) Return: sound
24. square_sound(f: number, duration: number) Return: sound 不是很确定什么意思
25. stacking_adsr(waveform: SoundProducer, base_frequency: number, duration: number, envelopes: List) Return: sound 跟#1 有联系
26. stop()
    Stops all currently playing sounds.
27. triangle_sound(freq: number, duration: number) 三角铁 Return: sound
28. trombone(note: number, duration: number) 长号 Return: sound
29. violin(note: number, duration: number)