Contest 15.3: A Game of Tones

Start date: 25 September 2016
Due: 2 October 2016, 23:59

Task 1:

Being masters of sound manipulation, you are to use your creativity to design some awesome-sounding sounds.

You may submit *one* sound. The sound should be submitted as its own function (following the naming convention described below) such that the function returns a sound that can be played using the `play` function.

The sound must be between **1 and 20** seconds long.

Your function for this contest must return a sound within a time limit of **two** minutes. Please note that if your sound-generating function does not terminate within the stipulated limit, beat will not be patient enough to wait till the end. So use `sound` instead of `sourcesound` to save your precious time. It is strongly recommended to use the existing sound primitives and other utility functions already provided to you:

- `sine_sound`
- `sawtooth_sound`
- `triangle_sound`
- `square_sound`
- `silence`
- `adsr`
- `stacking_adsr`
- `trombone`
- `piano`
- `violin`
- `bell`
- `cello`
- `midi_note_to_frequency`
- `letter_name_to_midi_note`
- `letter_name_to_frequency`
- `string_to_list_of_numbers`

Rules of Submission

- Be creative.
- **No play functions is allowed.**
- Strictly follow the naming convention outlined in the template.

- All functions should be self-contained.

Evaluation of Submissions

- Each student will be randomly assigned to 9 or 10 entries for voting. (This is to ensure that each submission receives an equal number of votes.)
- Each student can assign a score of 1 to 10 for each entry. The scores assigned to all 10 entries must sum up to 55. (The simplest way to achieve this is to rank the submissions from 1 to 10.) For students that are assigned 9 entries only, the total scores must sum up to 54.
- The final score for a submitted entry is /(s(v,t) = v - 2^t/50 /), where v is the `normalized_voting_score` (max 100), and t is the maximum of number of tokens in your program and **the length of longest string** in your program. (semicolons, operators, parentheses except comments are considered as tokens)
- `normalized_voting_score = sum_of_scores / number_of_voters / 10 * 100`

Submission

To submit your work to the Source Academy, place your program in the "Source" tab of the online editor within the mission page, save the program by clicking the "Save" button, and click the "Submit" button. Please ensure the required function from each Task is included in your submission. Note that submission is final and that any mistakes in submission requires extra effort from a tutor or the lecturer himself to fix.

IMPORTANT: Make sure you've saved the latest version of your work by clicking the "Save" button before finalizing your submission!