Contest 3.2: 3D Runes

Start date: 03 September 2017
Due: 11 September 2017, 23:59

Readings:

- Textbook Sections 1.1.1 to 1.1.4

Background

Avenger Muru, the Graphics Master of the Academy, notices a bunch of enthusiastic Cadets conjuring beautiful runic images for the Grandmaster's carpet. He wonders if their creativity is constrained by the 2D nature of the carpet, and offers them another contest: to conjure the most beautiful 3D runic art possible. You hear of this contest and, wildly enthusiatic as always, decide to jump right in.

Task 1:

Avenger Muru likes the colour palette provided by the Grandmaster for his carpet, and allows its use as well:

Colour	Reference Image	Colour	Reference Image
`blue`		`brown`	
`green`		`indigo`	

Colour	Reference Image	Colour	Reference Image
`orange`		`pink`	
`purple`		`red`	
`yellow`			

For those who like surprises, a special color function called `random_color` is also given to you. Nobody knows which color it will transform the rune into, but certainly not the boring black and white.

An example on how to use the new colour functions:

```
var pic = purple(heart_bb);
show(make_cross(pic));
```

would display a purple `heart_bb`.

You may submit up to 3 separate 3D runes. Each rune should be submitted as its own function (following the naming convention described below) such that running the function will return a rune that can then be displayed with `show`, `hollusion` or `anaglyph`. Follow the given template:

```
// First entry
function yourname_3d_contest_0() {
}

// Second entry
function yourname_3d_contest_1() {
}

// Third entry
function yourname_3d_contest_2() {
}
```

More information on the contest, such as judging criteria, will be provided as the due date approaches.

Rules for submission

- Be creative.
- No call show, hollusion, anaglyph, clear and any other attempt to interact with the viewport is allowed.
- Strictly follow the naming convention outlined in the contest PDF description.
- For 3D contest, indicate whether your entry is a hollusion or anaglyph via a comment as the first line within the function.
- All functions should be self-contained.

Example of what is acceptable

```
function martin_henz_3d_contest_0() {
    // hollusion
    function favorite(index) {
        return index === 0 ? heart_bb : black_bb;
    }
    return overlay(favorite(0), favorite(1));
}
```

Example of what is not acceptable

```
function helper() {
    return star_bb;
}

// hollusion - this is not the correct line
// Not following naming convention
function my_3d_rune() {
    clear(); // Interacting with the output
    show(helper()); // Calling functions not defined within your entry
```

```
}
```

Evaluation of Submissions

- This contest (3.2) is for the 3D category of the rune contests.
- For each category, 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 `score(v,t) = v - 2^{t/50}`, where v is the normalized_voting_score (max 100), and t is the number of tokens in your program, including semicolons, operators, parentheses, but not including comments. The "acceptable" example above has 36 tokens.
- `normalized_voting_score = sum_of_scores / number_of_voters / 10 * 100`

Submission

To submit your work for this mission, copy the url on your browser and submit the url (there is a tab to submit a url; don't submit a file containing the url) to the IVLE Contest 3.2 submission folder. Strictly follow to the deadlines set at the start of this file.

IMPORTANT: Make sure that everything for your programs to work is on the left hand side and not in the interpreter! This is because only that program is preserved when opening the url you have given us.