National University of Singapore
School of Computing
CS1101S: Programming Methodology
Semester I, 2013/2014

Contest 3-2
Beautiful Runes

Start date: 23 August 2013
Due: 3 September 2013, 23:59

Readings:

- Textbook Sections 1.1.1 to 1.1.4

Background:

You have became adept as a JFDI initiate but so are many others like yourself. With everyone attempting to prove themselves superior, it is certain unhealthy rivalry will form amongst the fresh initiates.

But the masters have already foreseen this problem through the many generations of JFDI knights they had trained. Initially masquerading as a rumour, news of the annual rune conjuring contest quickly became the hottest of discussion topics.

With exquisite and intricate winning runes being displayed prominently in the grand hall and the hustle and bustle of preparation, you barely managed to get hold of an instructor to get the details. Clearly, it was not intended for all initiates to participate but only those possessing true passion and are pure of essence. Are you?

Task 1:

This contest represents the 2D runes segment of the annual rune conjuring contest which you may participate in.

Being masters of rune manipulation, you are to use your creativity and design some cool-looking runes. Simply define your runes such that they may be displayed with the `show` function.

You may submit up to 3 separate 2D runes. Each rune should be submitted as its own function (following the naming convention described below) such that by running the JediScript function, the rune will be created and displayed in the viewport.

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

Task Files

- lib/list.js
- lib/misc.js
- lib/graphics.js
- lib/runes.js
- contest_3-2_1.html
- **contest_3-2_1.js**

Submission

To submit your work to the Academy, code your submission within the box that says "Your submission" on the mission page. Name your (up to) three submissions using the function names yourname_2d_contest_x where x is from 0 to 2. Click "Save Code", then click "Finalize Submission" when you are done. Note that submission is final and that any mistakes in submission requires extra effort from a tutor or the lecturer himself to fix.