

National University of Singapore
School of Computing
CS1101S: Programming Methodology (JavaScript)
Semester I, 2012/2013

Mission Sidequest 5-1
Wizard

Start date: 30 August 2012

Due: 05 September 2012, 23:59

Readings:

- Textbook Sections 1.3 to 1.3.1

In Missions 2 and 3, you learned to appreciate the beauty (and the vital importance) of abstraction. Without knowing how the runes were drawn and how primitive operations like `stack` were defined, you made use of them anyway to build up complex runes. To be more than a Padawan however, it is inevitable that you have to look under the hood, or even worse (or better), conjure up such primitives yourself.

In this side quest, we expose you to such wizardry. Complete this side quest successfully and you will get the “Pure Wizardry” achievement.

Information:

In the following, let R be the rectangular region bounded by the vertices $(0, 0)$, $(1, 0)$, $(1, 1)$, $(0, 1)$, which is the region displayed in the viewport.

You may want to use the following `test_curve` as input to test your functions.

```
var test_curve = function(t){  
    return make_point(t, 0.5 + (Math.sin(4 * (Math.PI * t)) / 2));  
}
```

This side quest consists of **two** tasks.

Task 1:

Implement `stack` with the same behavior as the `stack` function you encountered in Mission 2.

To be specific, `stack` takes as input two curves c_1 and c_2 that lie entirely within R . The output is a curve that also lies entirely within R , such that c_1 is visibly stacked over c_2 when drawn.

For example, `stack(test_curve, test_curve)` gives the output shown in the top right viewport of Figure 1 when drawn.

Task Files

- lib/list.js
- lib/misc.js
- lib/graphics.js
- lib/hi_graph.js
- sidequest_5-1_1.html
- **sidequest_5-1_1.js**

Task 2:

Implement `stack_frac`, again with the same behavior as the `stack_frac` function you came across in Mission 2.

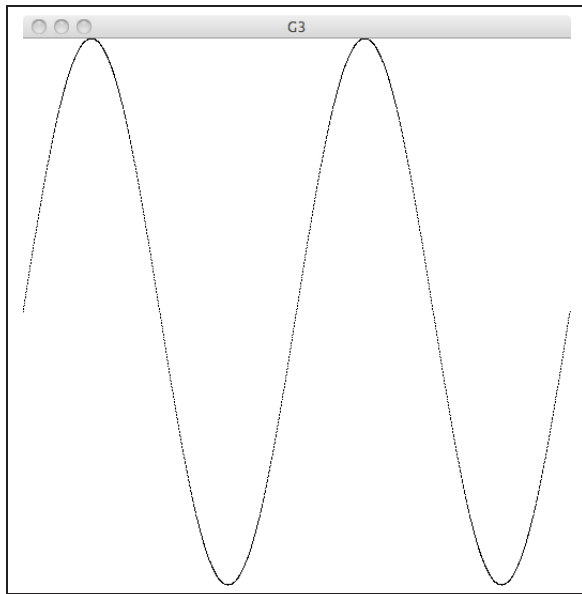
To be specific, `stack_frac` takes as input a fraction `frac` and two curves c_1 and c_2 that lie entirely within R . The output is a curve that also lies entirely within R , such that when drawn, c_1 is visibly stacked over c_2 with c_1 taking up `frac` of the display. For example, `stack_frac(1/5, test_curve, test_curve)` gives the output shown in the bottom left viewport when drawn.

Task Files

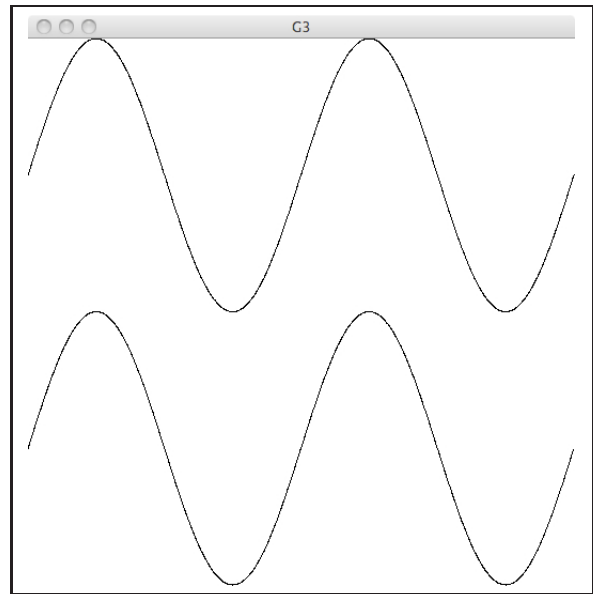
- lib/list.js
- lib/misc.js
- lib/graphics.js
- lib/hi_graph.js
- sidequest_5-1_2.html
- **sidequest_5-1_2.js**

Submission

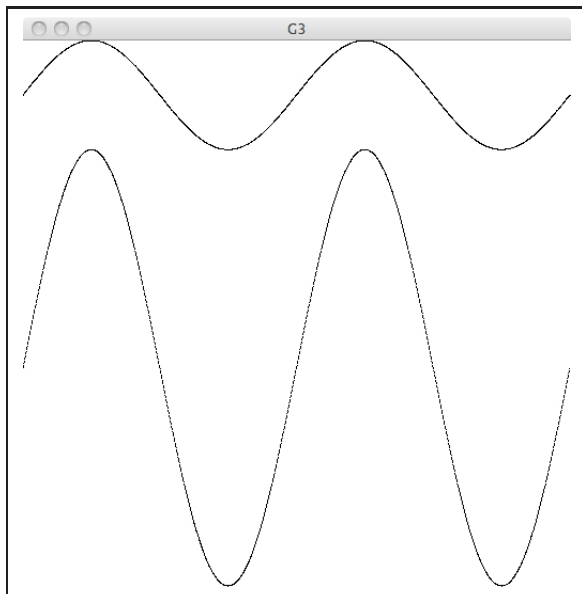
To submit your work to the Academy, copy the contents from the template file(s) into the box that says "Your submission" on the mission page, click "Save Code", then click "Finalize Submission". Note that submission is final and that any mistakes in submission requires extra effort from a tutor or the lecturer himself to fix.



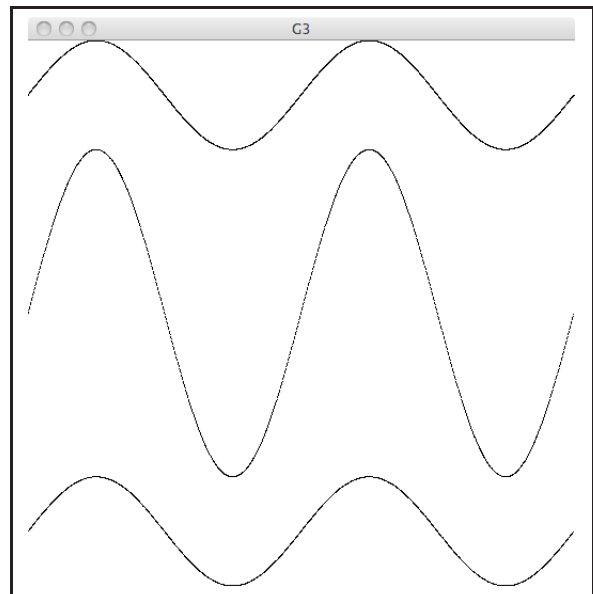
`draw_points_on(2000)(test_curve);`



`draw_points_on(4000)`
`(stack(test_curve, test_curve));`



`draw_points_on(4000)`
`(stack_frac(1/5, test_curve, test_curve));`



`draw_points_on(6000)`
`(stack_frac(1/5, test_curve,`
`stack_frac(3/4, test_curve, test_curve));`

Figure 1: Examples.