

# Mission Sidequest 5.1: Wizard

Start date: 6 September 2017

**Due: 12 September 2017, 23:59**

Readings:

- Textbook Sections 1.3 to 1.3.1

In Missions 2 and 3, you learnt the beauty and vital importance of abstraction. Without knowing how runes were drawn, or how primitive operations like `stack` were defined, you made use of them to build up complex runes. However, it is inevitable that you have to look under the hood, or even worse (or better), create such primitives yourself.

In this side quest, we expose you to such wizardry.

## Information

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

You may want to use `test_curve` as input to test your functions, defined below:

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

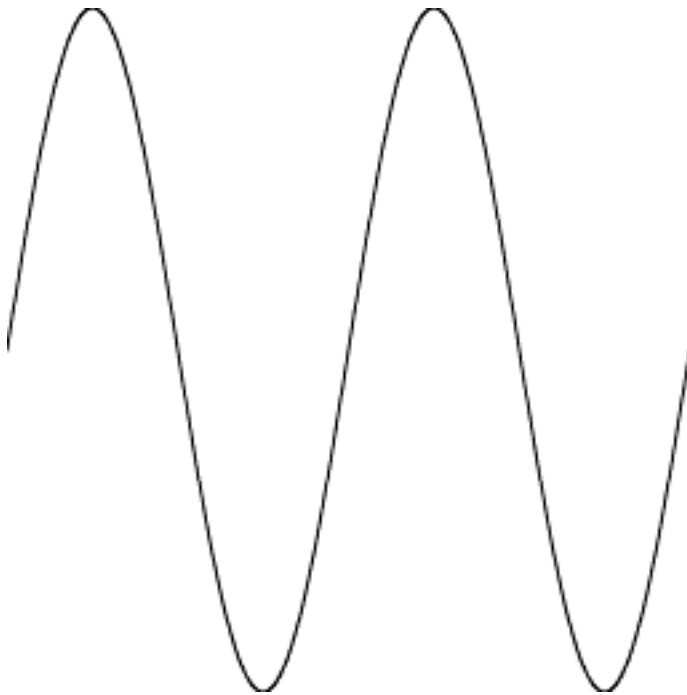
This side quest consists of **two** tasks.

## 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 `c1` and `c2` that lie entirely within  $R$ . The output is a curve that also lies entirely within  $R$ , such that `c1` is visibly stacked over `c2` when drawn.

For example, `stack(test_curve, test_curve)` gives the output shown below when drawn.



```
(draw_points_on(2000))(test_curve);
```

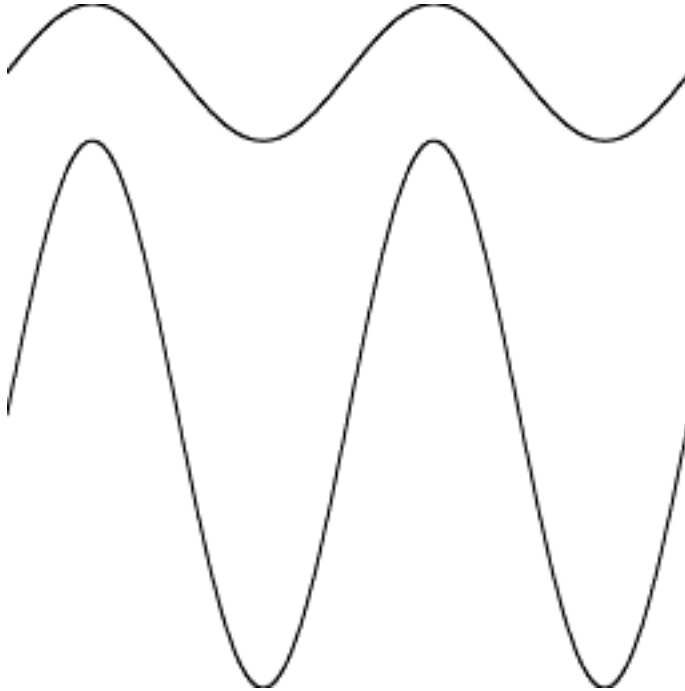


```
(draw_points_on(4000))(stack(test_curve, test_curve));
```

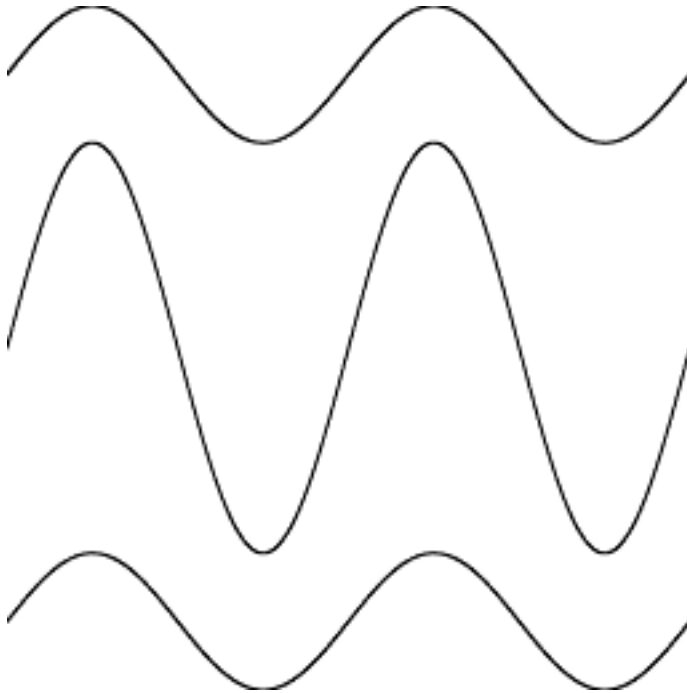
## Task 2:

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

`stack_frac` takes as input a fraction `frac` and two curves `c1` and `c2` that lie entirely within `R`. The output is a curve that also lies entirely within `R`, such that when drawn, `c1` is visibly stacked over `c2` with `c1` taking up `frac` of the display. For example, `stack_frac(1/5, test_curve, test_curve)` shows:



```
(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))));
```

## Submission

To submit your work for this mission, copy the url on your browser and email it to your respective Avengers. 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 pane on the right! This is because only that program is preserved in the url you have emailed to us.