

アルゴリズムとデータ構造入門 第九回課題

1029-24-9540 山崎啓太郎

February 7, 2013

1 square-limit

```
1 (define right-split (lambda (painter n)
2   (if (= n 0)
3       painter
4       (let ((smaller (right-split painter (- n 1))) )
5         (beside painter (below smaller smaller) )))))
6
7 (define up-split (lambda (painter n)
8   (if (= n 0)
9       painter
10      (let ((smaller (up-split painter (- n 1))) )
11        (below painter (beside smaller smaller) )))))
12
13 (define corner-split (lambda (painter n)
14   (if (= n 0)
15       painter
16       (let ((up (up-split painter (- n 1)))
17             (right (right-split painter (- n 1))) )
18         (let ((top-left (beside up up))
19               (bottom-right (below right right))
20               (corner (corner-split painter (- n 1))) )
21           (beside (below painter top-left)
22                   (below bottom-right corner) ))))))
23
24
25 (define square-limit (lambda (painter n)
26   (let ((quarter (corner-split painter n)) )
27     (let ((half (beside (flip-horiz quarter) quarter)))
28       (below (flip-vert half) half)))))
```

2 letter-lambda 色付

```
1 (define filled-letterlambda (vertexes->painter
2   (list
3     (make-vect .45 .60) (make-vect .25 .20)
4     (make-vect .25 .20) (make-vect .20 .20)
5     (make-vect .20 .20) (make-vect .20 .10)
6     (make-vect .20 .10) (make-vect .30 .10)
7     (make-vect .30 .10) (make-vect .50 .50)
8     (make-vect .50 .50) (make-vect .70 .10)
9     (make-vect .70 .10) (make-vect .80 .10)
10    (make-vect .80 .10) (make-vect .80 .20)
11    (make-vect .80 .20) (make-vect .75 .20)
12    (make-vect .75 .20) (make-vect .40 .90)
13    (make-vect .40 .90) (make-vect .30 .90)
14    (make-vect .30 .90) (make-vect .30 .80)
15    (make-vect .30 .80) (make-vect .35 .80)
16    (make-vect .35 .80) (make-vect .45 .60) )
17    #t ))
```

実行例

```
(load "init.lsp")
(load "lambda.scm")
(set-color '#xfccc77)
(filled-letterlambda frm1)
```

出力

letter-lambda.jpg を添付してあります。

3 letter-lambda を square-limit に適用

実行例

```
(load "init.lsp")
(load "lambda.scm")
(load "square-limit.scm")
(set-color '#xfccc77)
((square-limit filled-letterlambda 5) frm1)
```

出力

square-limit.jpg を添付してあります。

