
CSC 344 Second Racket Programming Assignment Solution

Learning Abstract: This assignment explores the use of iteration as well as recursion in both a visual sense (using the 2htdp/image library) and a numerical one. The first task aimed to create an interesting array of stacked colored circles using no repetitive constructs. Recursion and iteration were however, used in the remainder of the assignment to show off some intriguing numerical lists, Hirst dots, a visual creation imitating Frank Stella's work, and a completely original creation.

First Task: Permutations of Randomly Colored Stacked Dots

> Definitions Pane

```
#lang racket
(require 2htdp/image)
(define (tile bg base middle center)
  (overlay(circle 15 "solid" center) (circle 30 "solid" middle)
           (circle 45 "solid" base) (square 100 "solid" bg))
)

(define (dots-permutations first second third)
  (beside(tile "black" first second third) (tile "black" first third second)
         (tile "black" second first third) (tile "black" second third first)
         (tile "black" third first second) (tile "black" third second first))
)
```

> Interactions Pane

```
> (tile "beige" "peru" "burlywood" "navajo white")
```



```
> (tile "honeydew" "olive drab" "dark sea green" "mint cream")
```



```
> (dots-permutations "dark slate gray" "light slate gray" "light steel blue")
```



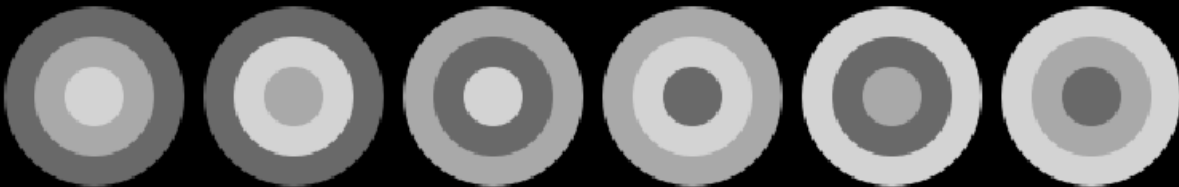
```
> (dots-permutations "maroon" "pale violet red" "pink")
```



```
> (dots-permutations "dark magenta" "violet" "thistle")
```



```
> (dots-permutations "dim gray" "dark gray" "light gray")
```



```
>
```

Second Task: Number Sequences

> Definitions Pane

```
#lang racket
(define (natural-sequence n)
  (cond
    ((> n 0)
     (natural-sequence (- n 1))
     (display n) (display " "))
    )
  )
)

(define (copies value num)
  (cond
    ((> num 0)
     (copies value (- num 1))
     (display value) (display " "))
    )
  )
)

(define (special-natural-sequence x)
  (cond
    ((> x 0)
     (special-natural-sequence (- x 1))
     (copies x (- x 1))
     (display x) (display " "))
    )
  )
)
```

> Interactions Pane

```
> (natural-sequence 5)
1 2 3 4 5
> (natural-sequence 18)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
> (natural-sequence 1)
1
> (natural-sequence 21)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
> (copies "a" 11)
a a a a a a a a a a a
> (copies 9 9)
9 9 9 9 9 9 9 9 9
> (copies "j" 10)
j j j j j j j j j j
> (copies 12 12)
12 12 12 12 12 12 12 12 12 12 12 12
> (special-natural-sequence 5)
1 2 2 3 3 3 4 4 4 4 5 5 5 5
> (special-natural-sequence 20)
1 2 2 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6 6 7 7 7 7 7 7 8 8 8 8 8 8 8 8 9 9 9
9 9 9 9 9 9 10 10 10 10 10 10 10 10 10 11 11 11 11 11 11 11 11 11 11 11 12
12 12 12 12 12 12 12 12 12 12 12 13 13 13 13 13 13 13 13 13 13 13 13 14 14
14 14 14 14 14 14 14 14 14 14 14 14 14 15 15 15 15 15 15 15 15 15 15 15 15
15 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 17 17 17 17 17 17 17
17 17 17 17 17 17 17 17 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20
> (special-natural-sequence 8)
1 2 2 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6 6 7 7 7 7 7 7 8 8 8 8 8 8 8
> (special-natural-sequence 0)
>
```

Third Task: Hirst Dots

> Definitions Pane

```
#lang racket
(require 2htdp/image)
(define (random-color)
  (color (random 256) (random 256) (random 256))
  )

(define (tile)
  (overlay (circle 15 "solid" (random-color)) (square 50 "solid" "black"))
  )

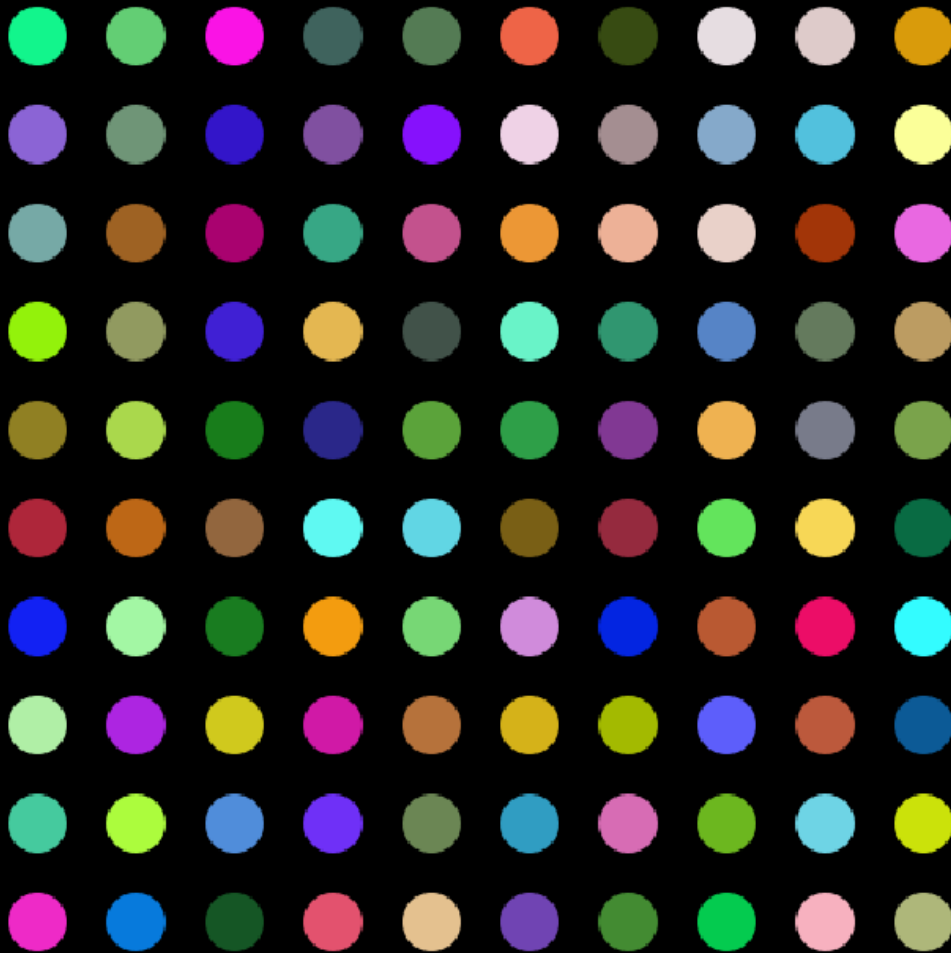
(define (row-of-circles c)
  (cond
    ((= c 0)
     empty-image
    )
    (> c 0)
    (beside (row-of-circles (- c 1)) (tile))
  )
  )

(define (rectangle-of-circles w h)
  (cond
    ((= w 0)
     empty-image
    )
    (> w 0)
    (above (rectangle-of-circles (- w 1) h) (row-of-circles h))
  )
  )

(define (hirst-dots s)
  (rectangle-of-circles s s)
  )
```

> Interactions Pane

> (hirst-dots 10)



> (hirst-dots 4)



Fourth Task: Stella Thing

> Definitions Pane

```
#lang racket
(require 2htdp/image)
(define (random-color)
  (color (random 256) (random 256) (random 256))
)

(define (nested-circles radius count c1 c2)
  (define step (/ radius count))
  (paint-nested-circles 1 count step c1 c2)
)

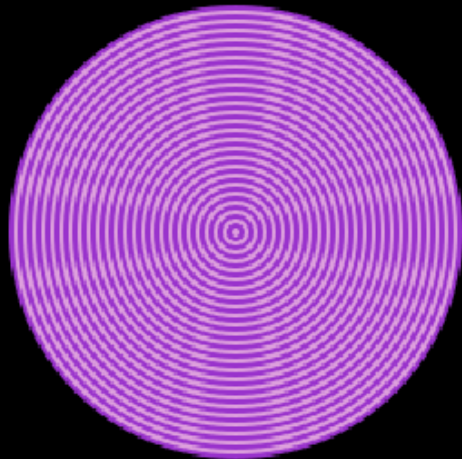
(define (paint-nested-circles from to step c1 c2)
  (define r (* from step))
  (cond
    ((= from to)
     (if (even? from)
         (circle r "solid" c1)
         (circle r "solid" c2))
     )
    (< from to)
    (if (even? from)
        (overlay
         (circle r "solid" c1)
         (paint-nested-circles (+ from 1) to step c1 c2))
        )
    (overlay
     (circle r "solid" c2)
     (paint-nested-circles (+ from 1) to step c1 c2))
    )
  )
)
)
```

> Interactions Pane

```
> (nested-circles 200 20 "sea green" "light green")
```



```
> (nested-circles 100 50 "dark orchid" "plum")
```



Fifth Task: Creation

> Definitions Pane

```
#lang racket
(require 2htdp/image)
(define (creation size intensity color)
  (cond
    ((= intensity 0)
     empty-image)
    ((> intensity 0)
     (creation size (- intensity 1) color)
     (scene+curve (square size 0 "black")
                  (random size) (random size) (random size) (random size)
                  (random size) (random size) (random size) (random size) color)
     )
  )
)
```

> Interactions Pane

```
> (creation 200 50 "blue")
> (creation 150 300 "pink")
```
