Second Racket Programming Assignment

Abstract

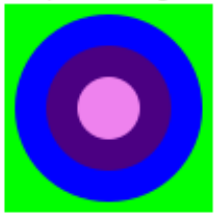
For this assignment I demonstrated the ability to generate and manipulate images using the 2htdp/image library in DrRacket. I was able to write a function to print each permutation of colored disks. Using recursive functions, I was able to print out patterns of integers and patterns of colored dots. I was able to mimic the artwork of Frank Stella by overlaying concentric shapes using the image library.

Task 1: Permutation of Randomly Colored Stacked Dots

```
#lang racket
(require 2htdp/image)

(define (tile background large medium small)
  (define side 100)
  (define small-radius(/ 30 2))
  (define medium-radius(/ 60 2))
  (define large-radius(/ 90 2))
  (define backg (square side "solid" background))
  (define small-disk (circle small-radius "solid" small))
  (define medium-disk (circle medium-radius "solid" medium))
  (define large-disk (circle large-radius "solid" large))
  (overlay small-disk medium-disk large-disk backg)
 )
```

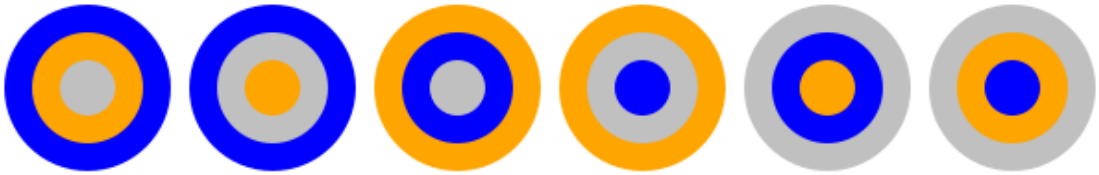
```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (tile "red" "orange" "yellow" "green")
```

```
> (tile "green" "blue" "indigo" "violet")
```

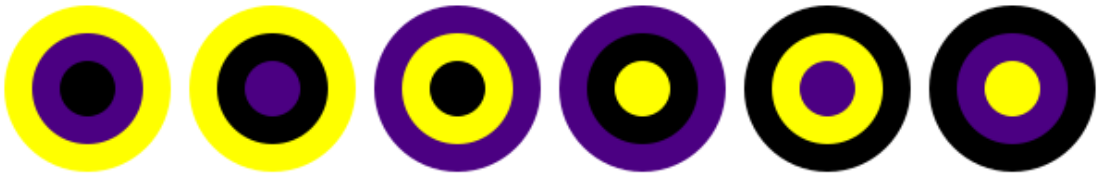


```
(define (dots-permutations color-1 color-2 color-3)
  (beside
   (tile "white" color-1 color-2 color-3)
   (tile "white" color-1 color-3 color-2)
   (tile "white" color-2 color-1 color-3)
   (tile "white" color-2 color-3 color-1)
   (tile "white" color-3 color-1 color-2)
   (tile "white" color-3 color-2 color-1)
   )
  )
```

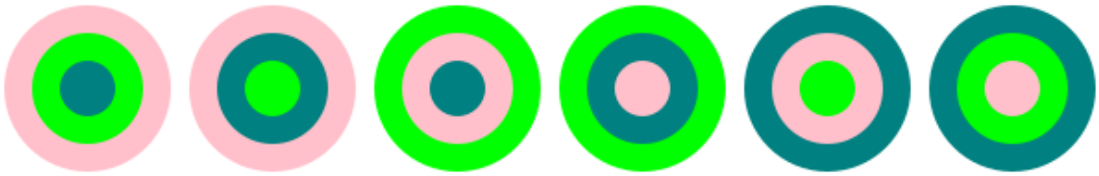
```
> (dots-permutations "blue" "orange" "silver")
```



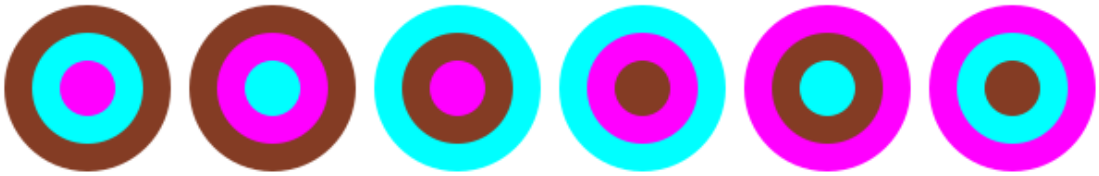
```
> (dots-permutations "yellow" "indigo" "black")
```



```
> (dots-permutations "pink" "green" "teal")
```



```
> (dots-permutations "brown" "cyan" "magenta")
```



```
>
```

Task 2: Number Sequences

```
(define (natural-sequence number)
(cond
  ((= number 0)
   (display " ")
   )
  ((> number 0)
     (natural-sequence (- number 1))
     (display number)(display " ")
     )))
(define (copies to-copy number)
  (cond
    ((= number 0)
     (display " ")
     )
    ((> number 0)
     (copies to-copy(- number 1))
     (display to-copy)
     (display " ")
     )))
(define (special-natural-sequence number)
  (cond
    ((= number 0)
     (display " ")
     )
    ((> number 0)
     (special-natural-sequence(- number 1))
     (copies number number)
     )))
```

```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (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 13)
 1 2 3 4 5 6 7 8 9 10 11 12 13
> (natural-sequence 30)
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
> (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 "CSC344" 20)
 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344 CSC344
> (copies "oswego" 10)
 oswego oswego oswego oswego oswego oswego oswego oswego oswego oswego
```

```
> (special-natural-sequence 5)
  1  2 2  3 3 3  4 4 4 4  5 5 5 5 5
> (special-natural-sequence 20)
  1  2 2  3 3 3  4 4 4 4  5 5 5 5 5  6 6 6 6 6 6  7 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 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 13  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 15 15  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 17 17 17  18 18 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  20 20 20 20 20 20 20 20 20 20 20 20 20 20  ↵
20 20 20 20 20 20
> (special-natural-sequence 17)
  1  2 2  3 3 3  4 4 4 4  5 5 5 5 5  6 6 6 6 6 6  7 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 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 13  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 15 15  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 17 17 17
```

```
> (special-natural-sequence 30)
  1  2 2  3 3 3  4 4 4 4  5 5 5 5 5  6 6 6 6 6 6  7 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 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 13  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 15 15  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 17 17 17  18 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  20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20  21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21
21 21 21  22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
22  23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25
25  26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26
26 26 26  27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27
27 27 27 27 27 27  28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28
28 28 28 28 28 28 28 28 28 28  29 29 29 29 29 29 29 29 29 29 29 29 29 29
29 29 29 29 29 29 29 29 29 29 29 29 29 29  30 30 30 30 30 30 30 30 30
30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
>
```

Task 3: Hirst Dots

```
#lang racket
(require 2htdp/image)
(define (rgb-value)(random 256))
(define (random-color)(color (rgb-value)(rgb-value)(rgb-value)))
(define side 40)
(define radius(/ 30 2))
(define (tile)(square side "solid" "white"))
(define (dot)(circle radius "solid" (random-color)))
(define (tile-dot) (overlay (dot) (tile)))

(define (row-of-dots number)
   (cond
    ((= number 0)
     empty-image
     )
    ((> number 0)
     (beside (row-of-dots(- number 1))
     (tile-dot))
)))
(define (rectangle-of-dots number number2)
  (cond
    ((= number 0)
     empty-image
     )
    ((> number 0)
     (above (rectangle-of-dots(- number 1)number2)
     (row-of-dots number2))
)))
(define (square-of-dots number)
  (rectangle-of-dots number number)
)
```

Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 128 MB.

```
> (hirst-dots 10)
```

```
> (hirst-dots 4)
```

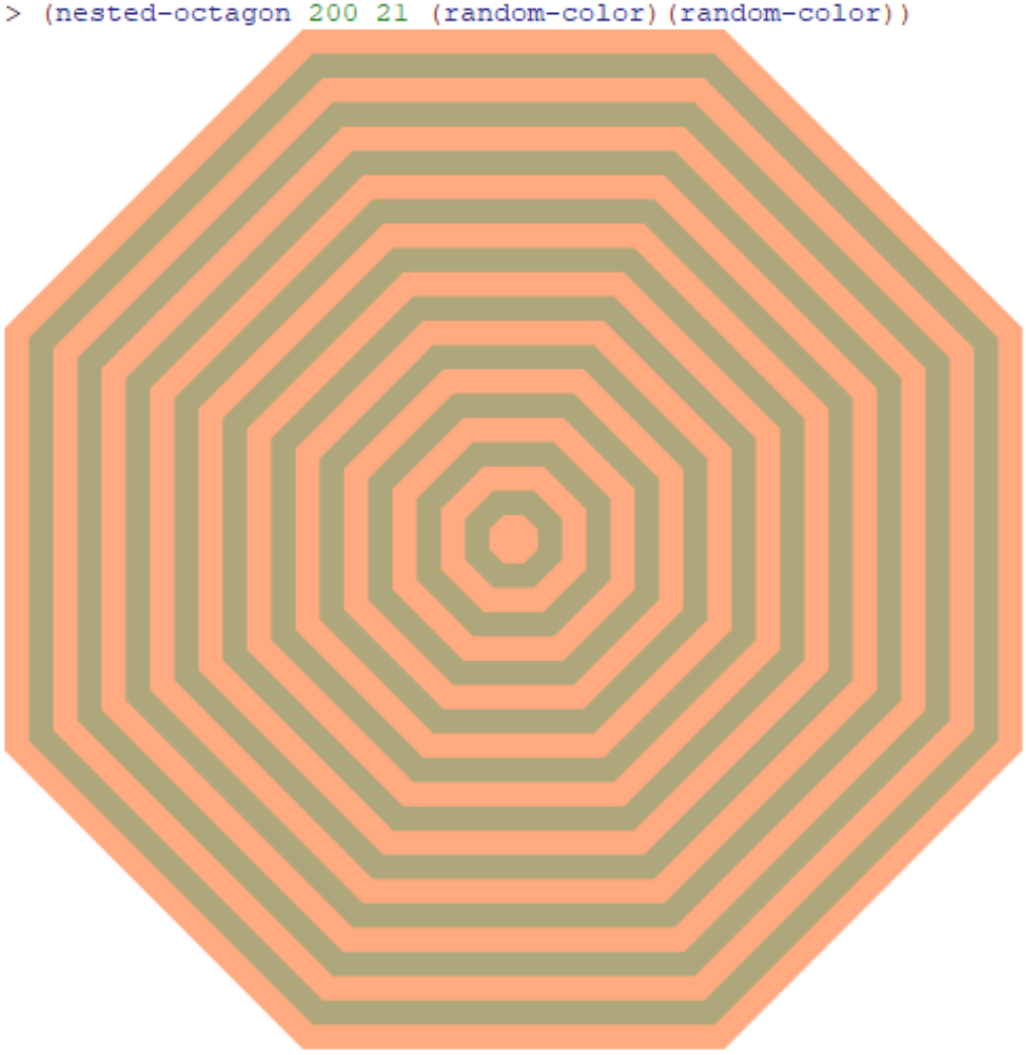
```
>
```

Task 4: Stella Thing

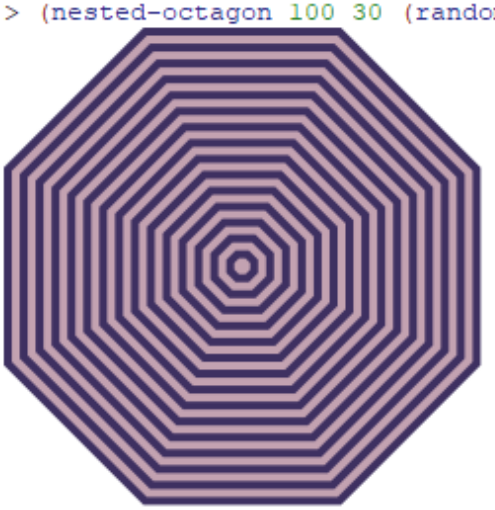
```
(define(octagon) (regular-polygon 100 8 "solid" (random-color)))

(define (nested-octagon side-length iterations color1 color2)
  (define delta ( / side-length iterations))
  (paint-nested-octagon 1 iterations delta color1 color2)
  )
(define (paint-nested-octagon from to delta color1 color2)
  (define length (* from delta))
  (cond
    ((= from to)
     (if (even? from)
      (regular-polygon length 8 "solid" color1)
      (regular-polygon length 8 "solid" color2)
      )
     )

  ((< from to)
   (if (even? from)
       (overlay
        (regular-polygon length 8 "solid" color1)
        (paint-nested-octagon (+ from 1) to delta color1 color2)
        )
        (overlay
        (regular-polygon length 8 "solid" color2)
        (paint-nested-octagon (+ from 1) to delta color1 color2)
        )
     )
    )
  )
)
```

Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 128 MB.

```
> (nested-octagon 200 21 (random-color)(random-color))
```



```
> (nested-octagon 100 30 (random-color)(random-color))
```



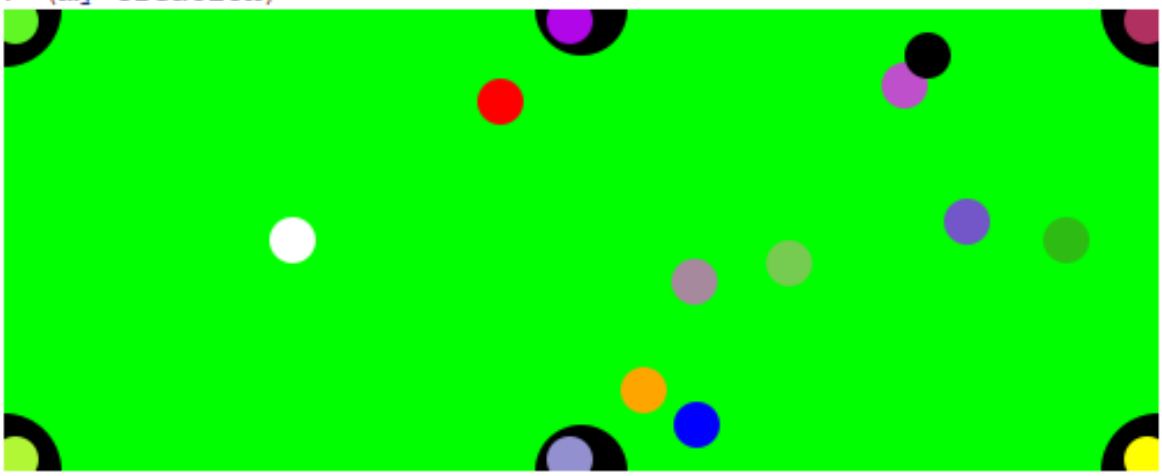
Task 5: My creation

```
#lang racket
(require 2htdp/image)
(define (rgb-value)(random 256))
(define (random-color)(color (rgb-value)(rgb-value)(rgb-value)))
(define pooltable( rectangle 300 100 "solid" "green"))
(define billiards
  (place-image
   ( circle 10 "solid" "white")
   125 100
   (place-image
    ( circle 10 "solid" "red")
    215 40
      (place-image
      ( circle 10 "solid" "blue")
      300 180
        (place-image
        ( circle 10 "solid" "orange")
        277 165
          (place-image
           ( circle 10 "solid" (random-color))
           460 100
             (place-image
              ( circle 10 "solid" "black")
                400 20
                (place-image
                 ( circle 10 "solid" (random-color))
                  340 110
                   (place-image
                   ( circle 10 "solid" (random-color))
                   390 33
                   (place-image
                    ( circle 10 "solid" (random-color))
                     417 92
```


(circle 10 "solid" (random-color))
299 118
(place-image
(circle 10 "solid" "yellow")
495 195
(place-image
(circle 10 "solid" "maroon")
495 05
(place-image
(circle 10 "solid" (random-color))
05 05
(place-image
(circle 10 "solid" (random-color))
245 05
(place-image
(circle 10 "solid" (random-color))
05 195
(place-image
(circle 10 "solid" (random-color))
245 195
(place-image
(circle 25 "solid" "black")
0 0
(place-image
(circle 25 "solid" "black")
0 200
(place-image
(circle 20 "solid" "black")
250 0
(place-image
(circle 20 "solid" "black")
250 200
imported

```
                                05 05
                                 (place-image
                                  ( circle 10 "solid" (random-color))
                                  245 05
                                    (place-image
                                     ( circle 10 "solid" (random-color))
                                     05 195
                                     (place-image
                                      ( circle 10 "solid" (random-color))
                                      245 195
                                       (place-image
                                        ( circle 25 "solid" "black")
                                        0 0
                                        (place-image
                                         ( circle 25 "solid" "black")
                                         0 200
                                         (place-image
                                          ( circle 20 "solid" "black")
                                          250 0
                                          (place-image
                                           ( circle 20 "solid" "black")
                                           250 200
                                           (place-image
                                            ( circle 25 "solid" "black")
                                            500 0
                                            (place-image
                                             ( circle 25 "solid" "black")
                                             500 200
                                             ( rectangle 500 200 "solid" "green"))))))))))))))))))))))))

(define (my-creation)
  (overlay billiards pooltable))
```

```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (my-creation)
```



```
>
```