

---

## Second Racket Programming Assignment Specification

By JIUN KIM

---

---

### Learning Object

In this assignment, I learned new functions such as cond, above, beside from the 2htdp/image library. The main object of the assignment was using the recursion from our lesson, using it to create more shapes.

I also used recursion for my creation. I decide to use underlay/offset and isosceles-triangle. from 2htdp/image library.

---

### Task 1 - Permutations of Randomly Colored Stacked Dots

---

```
#lang racket
(require 2htdp/image)
(define (tile color1 color2 color3 color4)
  (define square1 (square 100 "solid" color1))
  (define circle1 (circle 45 "solid" color2))
  (define circle2 (circle 30 "solid" color3))
  (define circle3 (circle 15 "solid" color4))
  (overlay circle3 circle2 circle1 square1)
)
(define (dots-permutations color1 color2 color3)
  (beside
    (tile "white" color1 color2 color3)
    (tile "white" color1 color3 color2)
    (tile "white" color2 color1 color3)
    (tile "white" color2 color3 color1)
    (tile "white" color3 color1 color2)
    (tile "white" color3 color2 color1)
  )
)
```

---

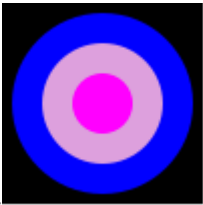
### The Demo

---

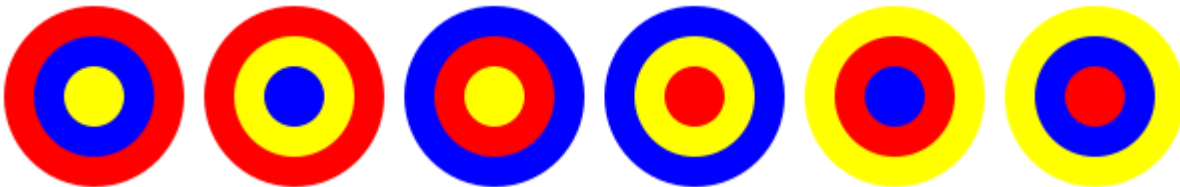
```
> ( tile "red" "cornflowerblue" "cyan" "teal" )
```



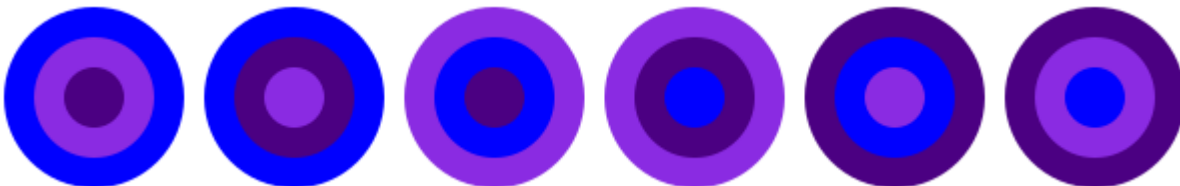
```
> ( tile "black" "blue" "plum" "magenta" )
```



```
> ( dots-permutations "red" "blue" "yellow" )
```



```
> ( dots-permutations "blue" "blueviolet" "indigo" )
```



```
>
```

---

## Task 2 - Number Sequences

---

```
( define ( sequence n1 n2 )
```

```
  ( cond
```

```
    ( ( <= n2 n1 )
```

```
      ( display n2 )
```

```
      ( display " " )
```

```
( sequence n1 ( + 1 n2 ) )  
 )  
 )  
 )
```

```
( define ( natural-sequence n )  
  ( cond  
    ( ( = n 0 )  
      ( display "\n" )  
    )  
    ( ( >= n 1 )  
      ( sequence n 1 )  
    )  
  )  
 )  
 )
```

```
( define ( copies letter times )  
  ( cond  
    ( ( > times 0 )  
      ( display letter )  
      ( display " " )  
      ( copies letter ( - times 1 ) )  
    )  
  )  
 )
```

```
( define ( special-sequence n1 n2 )  
  ( cond  
    ( ( <= n2 n1 )  
      ( copies n2 n2 )  
      ( special-sequence n1 ( + n2 1 ) )  
    )  
  )  
)  
  
( define ( special-natural-sequence n )  
  ( cond  
    ( ( = n 0 )  
      ( display "\n" )  
    )  
    ( ( >= n 1 )  
      ( special-sequence n 1 )  
    )  
  )  
)
```

---

## The Demo

---

```
> ( 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

> ( 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

> ( 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 6 6 6 6 6 7 7 7 7 7 7 8 8 8 8 8 8 8 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 11 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 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  
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 20

>

---

### Task 3 - Hirst Dots

---

```

( require 2htdp/image )
( define ( rgb-value ) ( random 256 ) )
( define ( random-color ) ( color ( rgb-value ) ( rgb-value ) ( rgb-value ) ) )

( define white-square ( square 50 "solid" "white" ) )
( define ( square-with-dot)
  ( define dot ( circle 15 "solid" ( random-color ) ) )
  ( overlay dot white-square )
)
( define ( row-of-dots n )
  ( cond
    ( ( = n 0 )      empty-
image
    )
    ( ( > n 0 )
      ( beside ( row-of-dots ( - n 1 ) ) ( square-with-dot ) ) )
    )
  )
)
( define ( rectangle-of-square r c )
  ( cond
    ( ( = r 0 )
      empty-image
    )
    ( ( > r 0 )
      ( above ( rectangle-of-square ( - r 1 ) c ) ( row-of-dots c ) )
    )
  )
)
( define ( hirst-dots n )
  ( rectangle-of-square n
n ) )

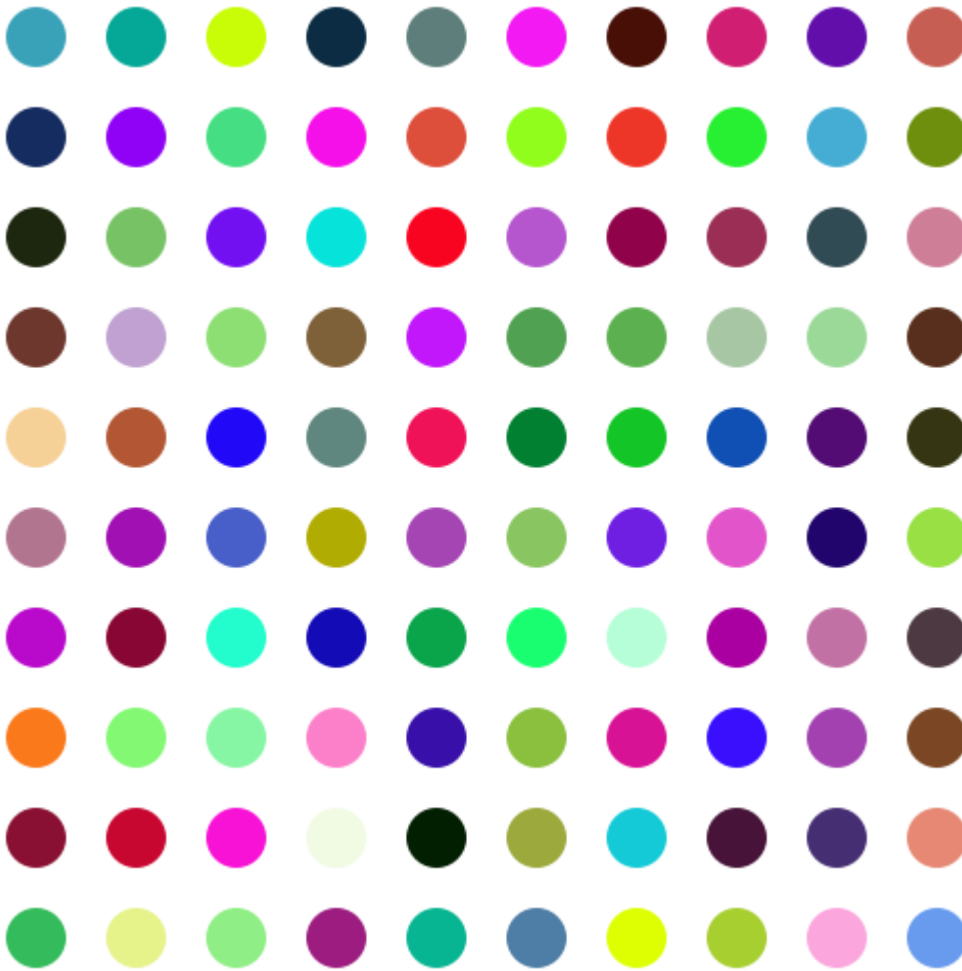
```

---

The Demo

---

> ( hirst-dots 10 )



( hirst-dots 4 )



.

---

## **Task 4 - Stella Thing**

---



**( require 2htdp/image )**

**( define ( stella side count color1 color2 )**

**( define delta ( / side count ) )**

**( paint-nested-circles-two 1 count delta color1 color2 )**

**)**

**( define ( paint-nested-circles-two from to delta color1 color2 )**

**( define side-length ( \* from delta ) )**

**( cond**

**( = from to )**

**( if ( even? from )**

**( circle side-length "solid" color1 )**

**( circle side-length "solid" color2 )**

**)**

**)**

**( ( < from to )**

**( if ( even? from )**

**( overlay**

**( circle side-length "solid" color1 )**

**( paint-nested-circles-two ( + from 1 ) to delta color1 color2 )**

**)**

**( overlay**

**( circle side-length "solid" color2 )**

```
( paint-nested-circles-two (+ from 1) to delta color1 color2 )  
)  
)  
)  
)  
)
```

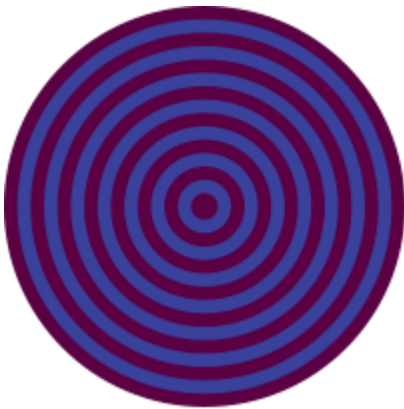
```
( define ( random-color ) ( color ( rgb-value ) ( rgb-value ) ( rgb-value ) ) )  
( define ( rgb-value ) ( random 256 ) )
```

---

### The Demo

---

```
( stella 100 15 ( random-color ) ( random-color ) )
```



```
( stella 50 7 ( random-color ) ( random-color ) )
```



## Task 5 - Creation

---

```
( require 2htdp/image )
```

```
( define ( ball radius )
```

```
  ( circle radius "outline" "black" )
```

```
)
```

```
( define ( arm length )
```

```
  ( rectangle length ( * 0.2 length ) "solid" "brown" )
```

```
)
```

```
( define ( hat width )
```

```
  ( above
```

```
    ( rectangle ( * 0.6 width ) ( * 0.4 width ) "solid" "black" )
```

```
    ( rectangle width ( * 0.3 width ) "solid" "black" )
```

```
  )
```

```
)
```

```
( define ( face radius )
```

```
  ( underlay/offset
```

```
    ( ball radius )
```

```
    0 -1
```

```
    ( underlay/offset ( circle ( * 0.25 radius ) "solid" "black" )
```

```
      -30 0
```

```
      ( circle ( * 0.25 radius ) "solid" "black" )
```

```
    )  
  )  
)  
  
( define ( my-creation n )  
  ( above  
    ( hat ( * 1.2 n ) )  
    ( overlay  
      ( isosceles-triangle ( * 0.3 n ) 330 "solid" "yellow" )  
      ( face ( * 0.5 n ) )  
    )  
  )  
  
  ( beside  
    ( arm n )  
    ( ball ( * 0.75 n ) )  
    ( arm n ) )  
  ( ball n )  
)  
)
```

---

## The Demo

---

```
( my-creation 50 )
```

