Timothy Astacio

February 21st, 2023

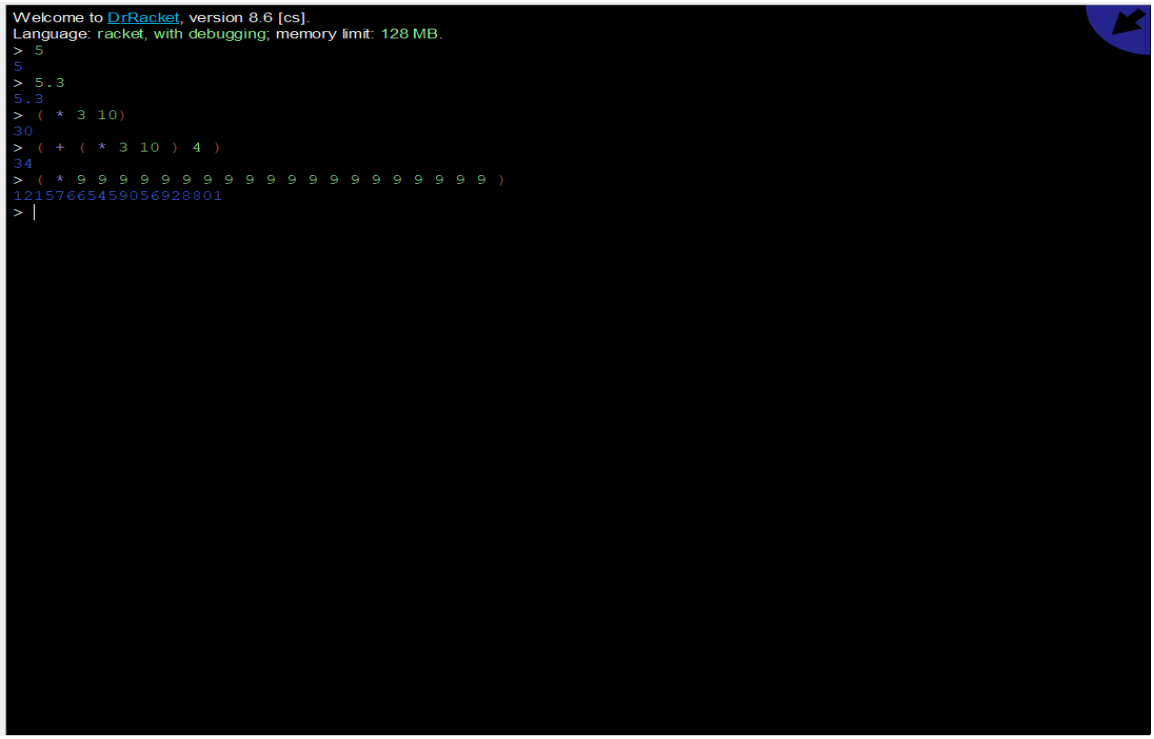
CSC 344 Programming Languages

Racket Assignment 2: Interactions, Definitions, Applications

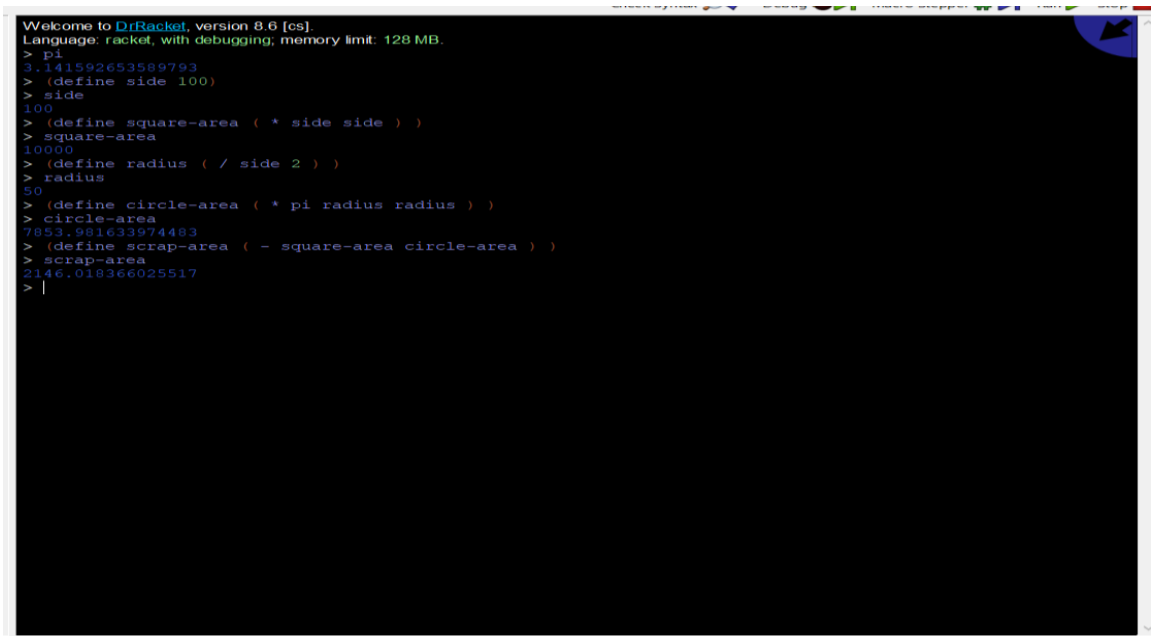
Learning Abstract: The goal of this assignment is to help improve one's understanding of the Racket program and how its syntax works. This assignment will cover the interactions, definitions, and applications in the racket programming language. The first task requires us to do a simple copy and paste while the remaining tasks help engage the student more with the programming language by testing their problem solving skills.

Task 1 Interactions – Scrap of Tin

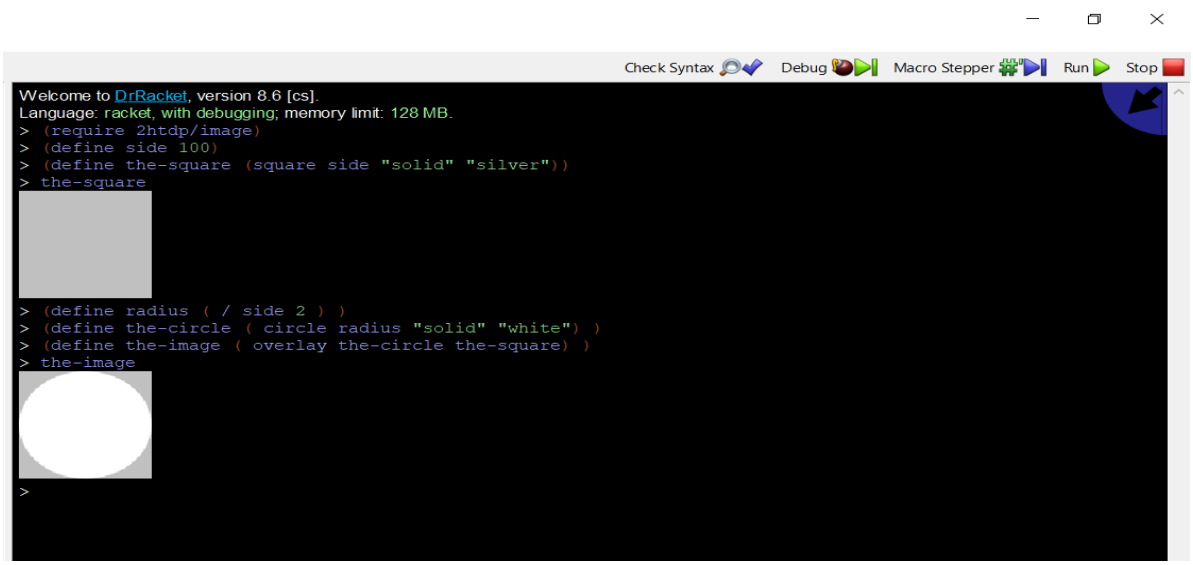
Arithmetic Expressions Demo:



Solve a Simple Problem (Area of Scrap)



Rendering an Image of the Problem Situation



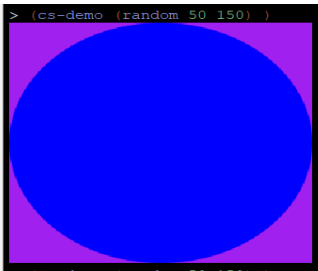
Task 2: Definitions – Inscribing/Circumscribing Circles/Squares

The Code for Task 2:

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

(define (cs radius )
  ( * radius 2 ) )

(define ( cc side-length)
  (sqrt ( * (/ side-length 2) (/ side-length 2) 2 ) ) )

(define (ic square-side)
  ( / square-side 2 ) )

(define (is cir-radius)
  (/ (* cir-radius 2 ) (sqrt 2 ) ) )

(define (cs-demo cs-Demo-Radius)
  (define cr (circle cs-Demo-Radius "solid" "blue" ) )
  (define sq (square (cs cs-Demo-Radius) "solid" "purple" ) )
  (overlay cr sq) )

(define (cc-demo cc-Demo-Side)
  (define cr (circle (cc cc-Demo-Side) "solid" "purple" ) )
  (define sq (square cc-Demo-Side "solid" "blue" ) )
  (overlay cr sq) )

(define (ic-demo ic-Demo-Side)
  (define cr (circle (ic ic-Demo-Side) "solid" "red"))
  (define sq (square ic-Demo-Side "solid" "green" ) )
  (overlay sq cr) )

(define (is-demo is-Demo-Radius )
  (define sq (square (is is-Demo-Radius) "solid" "green"))
  (define cr (circle is-Demo-Radius "solid" "blue") )
  (overlay sq cr) )
```

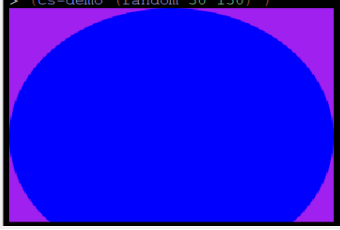
All the cs, cc, is, ic Demos are located below:

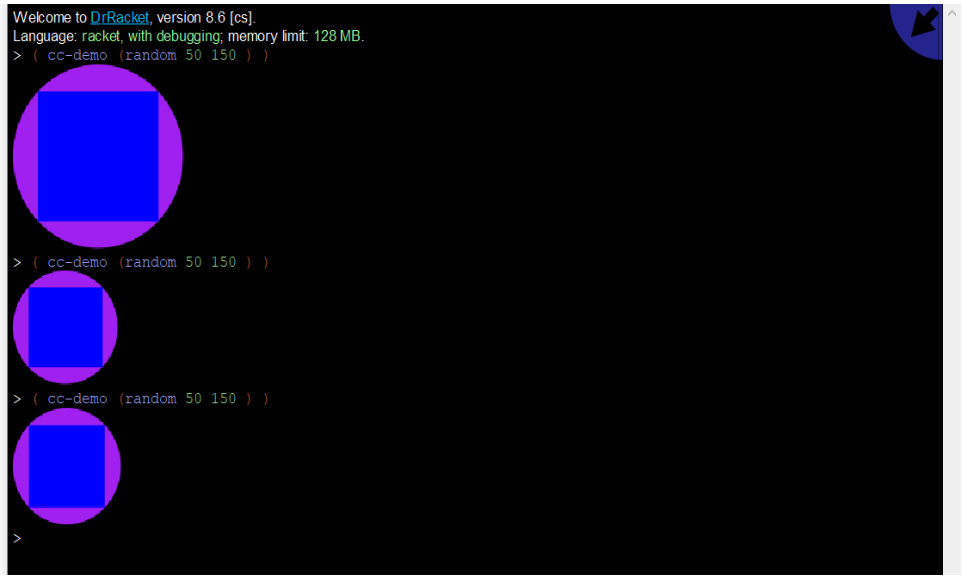
```
> (cs-demo (random 50 150) )
```

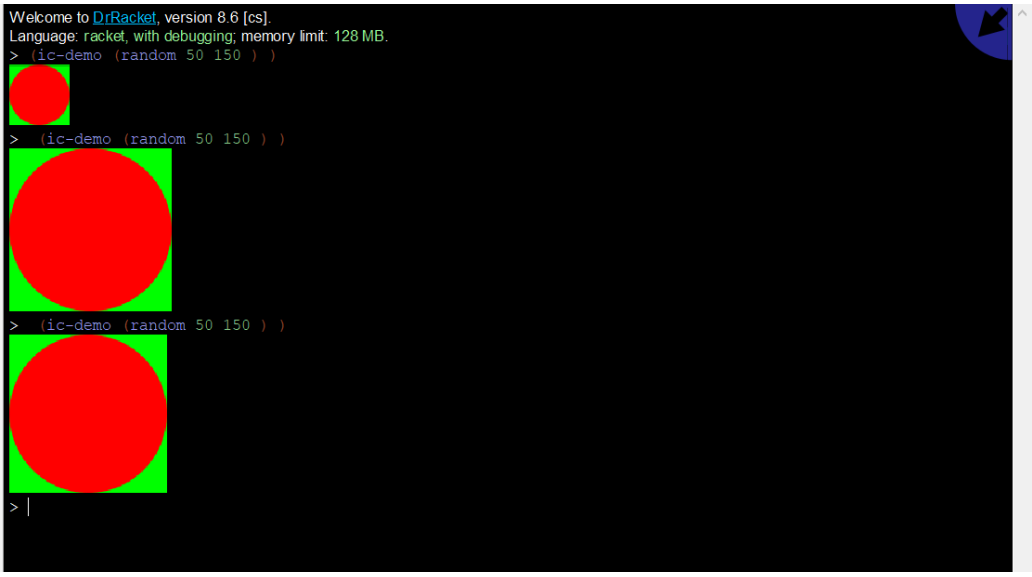


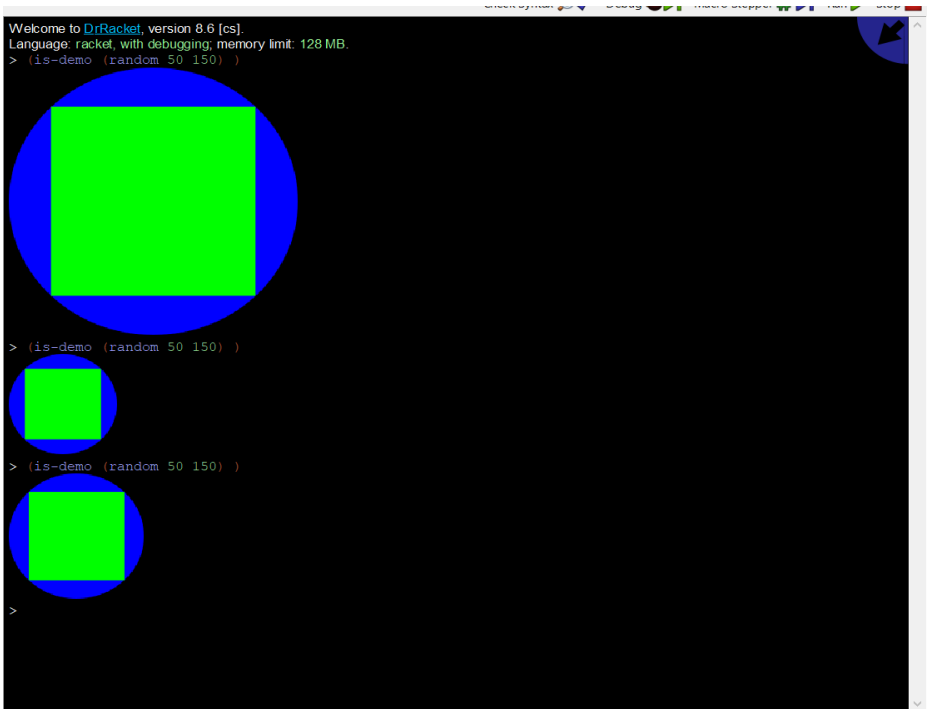
```
> (cs-demo (random 50 150) )
```

```
> (cs-demo (random 50 150) )
```




Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (cc-demo (random 50 150))
> (cc-demo (random 50 150))
> (cc-demo (random 50 150))
>


Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (ic-demo (random 50 150))
> (ic-demo (random 50 150))
> (ic-demo (random 50 150))
>



Task 3: Inscribing/Circumscribing Images

The code for Task 3:

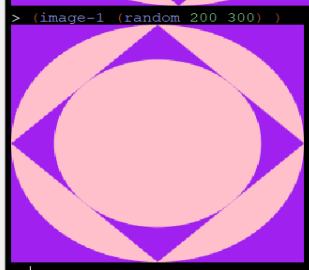
```
#lang racket

(define (is-demo is-Demo-Radius )
  (define sq (square (is is-Demo-Radius) "solid" "green"))
  (define cr (circle is-Demo-Radius "solid" "blue") )
  (overlay sq cr) )

(define (image-1 sqSide)
  (define cr1 (circle (ic sqSide) "solid" "pink" ) )
  (define sq1 (square sqSide "solid" "purple" ) )
  (define cr2 (circle (ic (is (ic sqSide))) "solid" "pink"))
  (define sq2 (rotate 45 (square (is (ic sqSide)) "solid" "purple")))
  (overlay cr2 sq2 cr1 sq1))

(define (image-2 sqSide)
  (define sq1 (square sqSide "outline" "violet"))
  (define sq2 (rotate 45 (square (is (ic sqSide)) "outline" "violet")))
  (define sq3 (square (is (ic (is (ic sqSide)))) "outline" "violet"))
  (define sq4 (rotate 45 (square (is (ic sq3-Side)) "outline" "violet")))
  (define sq3-Side (is (ic (is (ic sqSide)))))
  (overlay sq4 sq3 sq2 sq1))

( define ( Warholesque-image CanvasSide )
 ( define ( image-1 CanvasSide )
 ( define sqSide ( / CanvasSide 2 ) )
 ( define (random-color) ( color ( random 0 256 ) ( random 0 256 ) ( random 0 256 ) ) )
 ( define circleColor ( random-color ) )
 ( define squareColor ( random-color ) )
 ( define border ( square ( + 2 sqSide ) "solid" "black" ) )
 ( define cr1 ( circle ( ic sqSide ) "solid" circleColor ) )
 ( define sq1 ( square sqSide "solid" squareColor ) )
 ( define sq2 ( rotate 45 ( square ( is ( ic sqSide ) ) "solid" squareColor ) ) )
 ( define cr2 ( circle ( ic ( is ( ic sqSide ) ) ) "solid" circleColor ) )
 ( overlay cr2 sq2 cr1 sq1 border )
 )
 ( define border1 ( square ( + 6 CanvasSide ) "solid" "black" ) )
 ( overlay
 ( above
 ( beside ( image-1 CanvasSide ) ( image-1 CanvasSide ) )
 ( beside ( image-1 CanvasSide ) ( image-1 CanvasSide ) )
 )
 border1
 )
)
```

The demos for all Images and Warholesque

Image 1 Demo:

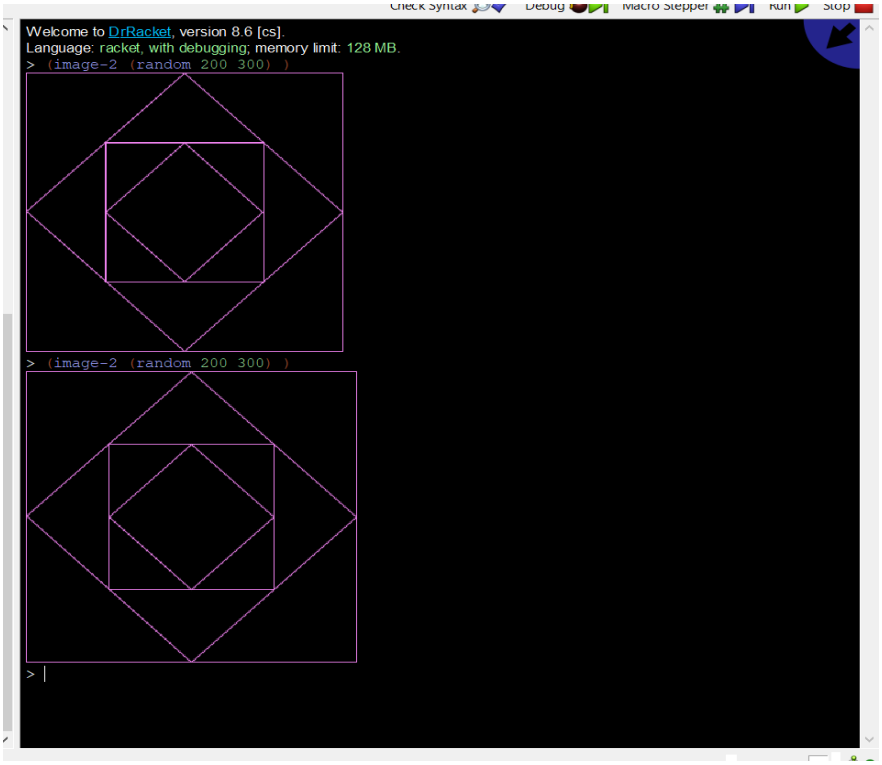
```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (image-1 (random 200 300) )
```

```
> (image-1 (random 200 300) )
```

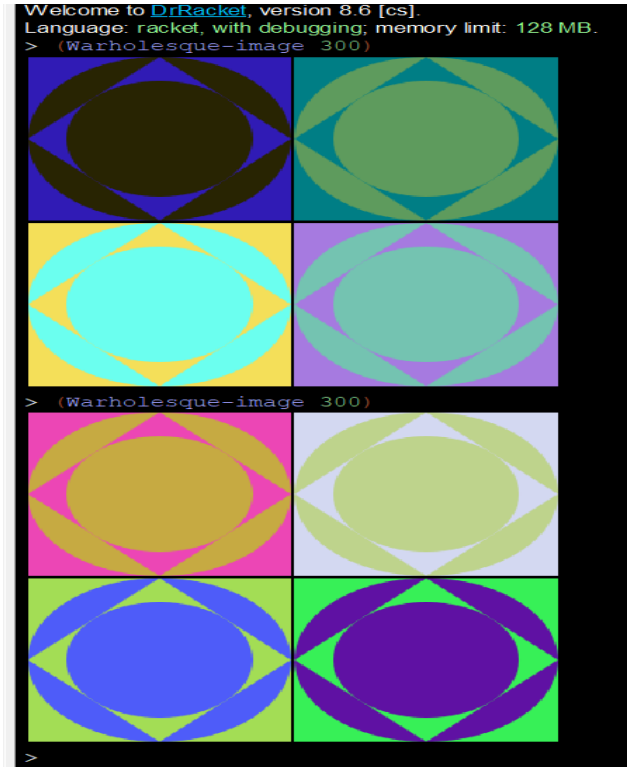


```
>
```

Image 2 Demo:



Warholesque Image:



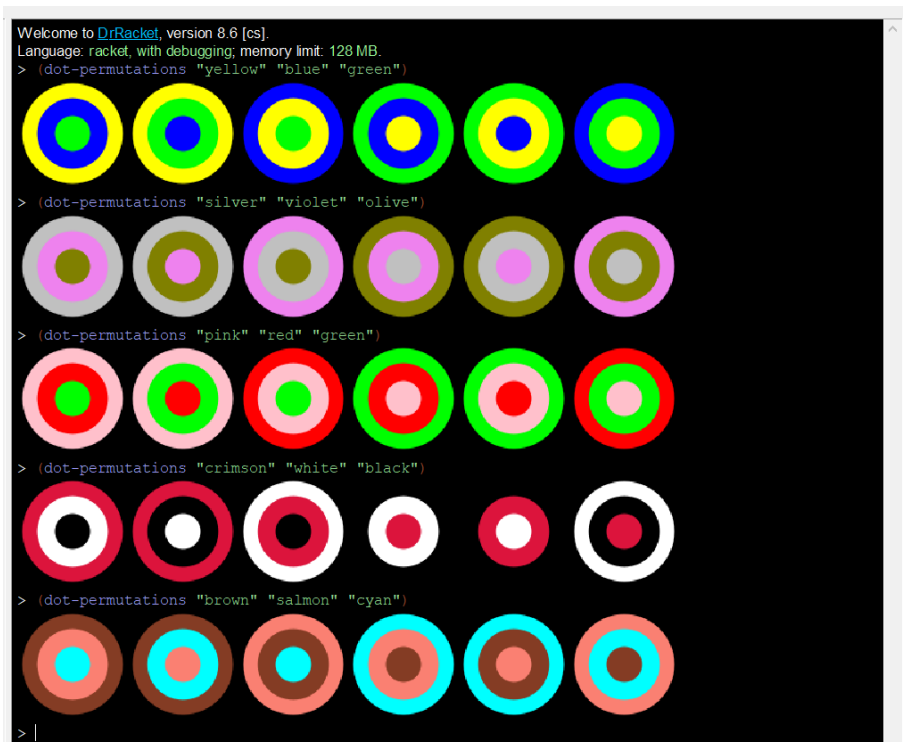
Task 4: Permutations of Randomly Colored Stacked Dots

The Code for this task:

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

(define (ic side-length)
  (/ side-length 2))

(define (cr2 radius)
  (/ ( / ( * radius 2 ) (sqrt 2) ) 2 ) )

(define (cr3 radius)
  (/ (/ ( * (cr2 (cr2 radius ) ) 2 ) (sqrt 2 ) ) 2 ) )

(define (tile sqColor crColor cr2Color cr3Color )
  (define radius 45 )
  (define sq (square ( * radius 2.2 ) "solid" sqColor) )
  (define cir1 (circle radius "solid" crColor ) )
  (define cir2 (circle (cr2 radius) "solid" cr2Color ) )
  (define cir3 (circle (cr3 radius) "solid" cr3Color) )
  (overlay cir3 cir2 cir1 sq))

(define (ColorMix color1 color2 color3)
  (define radius 45)
  (define sq (square (* radius 2.2) 0 color1) )
  (define cir1 (circle radius "solid" color1 ))
  (define cir2 (circle (cr2 radius) "solid" color2) )
  (define cir3 (circle (cr3 radius) "solid" color3))
  (overlay cir3 cir2 cir1 sq) )

(define (dot-permutations color1 color2 color3)
  (beside (ColorMix color1 color2 color3) (ColorMix color1 color3 color2)
          (ColorMix color2 color1 color3) (ColorMix color3 color2 color1)
          (ColorMix color3 color1 color2) (ColorMix color2 color3 color1) ) )
```

Demos for Title task and Permutations task:


Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (tile "blue" "violet" "teal" "cyan")
> (tile "silver" "pink" "purple" "red")
>


Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (dot-permutations "yellow" "blue" "green")
> (dot-permutations "silver" "violet" "olive")
> (dot-permutations "pink" "red" "green")
> (dot-permutations "crimson" "white" "black")
> (dot-permutations "brown" "salmon" "cyan")
>