Racket Assignment #2: Interactions, Definitions, Applications

Abstract

In this assignment, I have a chance to practice some rather basic Racket programming. I'll engage in a variety of interactions, create a number of function definitions, and solve computational issues using a combination of recycled code, original creations, and rearranged existing code.

Task 1: Interactions - Scrap of Tin

Arithmetic Expressions

Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.

```
> 5
5
> 5.3
5.3
> ( * 3 10 )
30
> ( + ( * 3 10 ) 4 )
34
> ( * 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 )
12157665459056928801
>
```

Solve a Simple Problem (Area of Scrap)

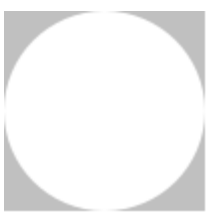
```
> pi
3.141592653589793
> ( define side 100 )
> side
100
```

```
> ( define square-area ( * side side ) )
> square-area
10000
> ( define radius ( / side 2 ) )
> radius
50
> ( define circle-area ( * pi radius radius ) )
> circle-area
7853.981633974483
> ( define scrap-area ( - square-area circle-area ) )
> scrap-area
2146.018366025517
>
```

Rendering an Image of the Problem Situation

```
> ( require 2htdp/image)
> ( define side 100 )
> ( define the-square ( square side "solid" "silver" ) )
> the-square
```

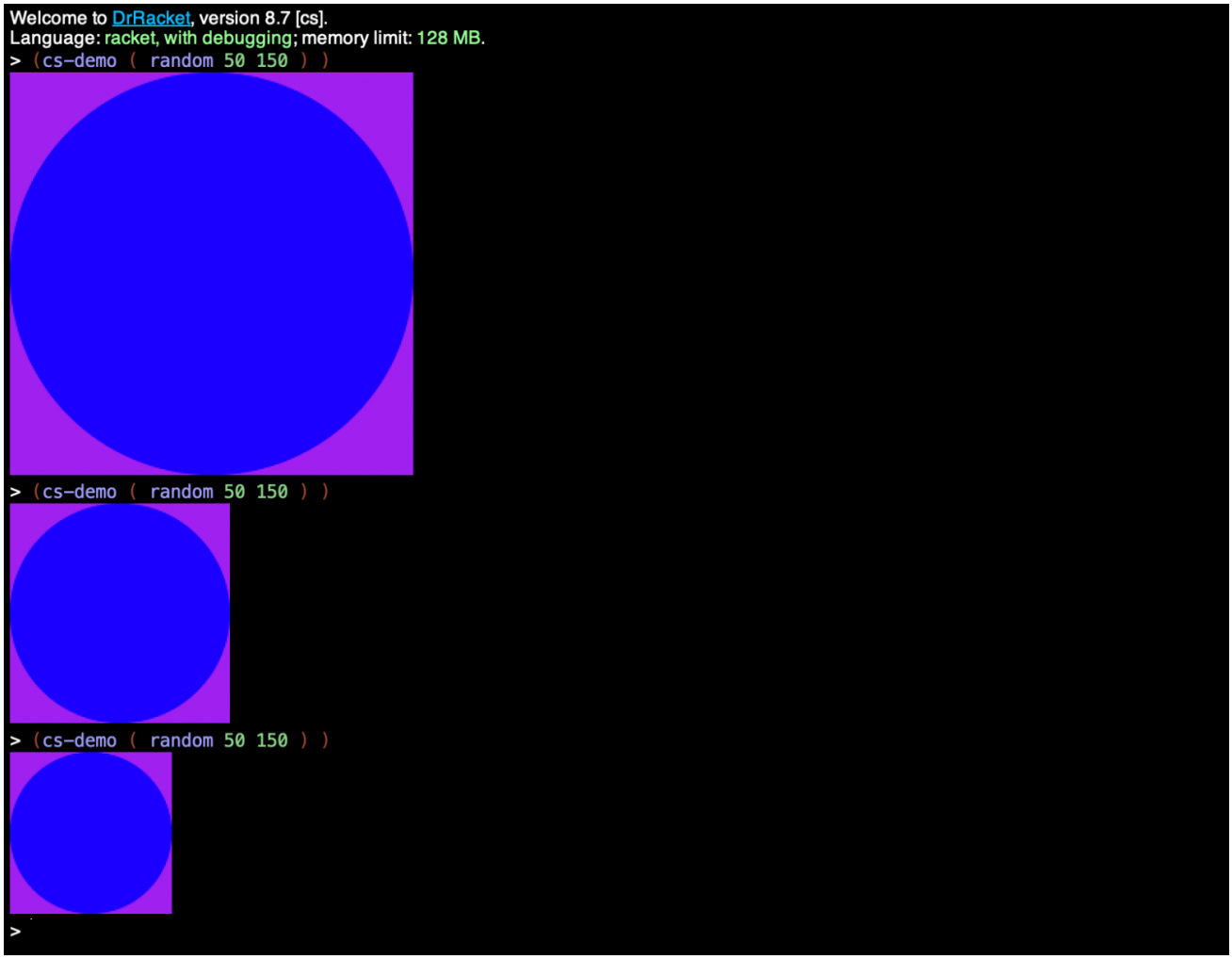
```
> ( define radius ( / side 2 ) )
> ( define the-circle ( circle radius "solid" "white" ) )
> ( define the-image ( overlay the-circle the-square) )
> the-image
```



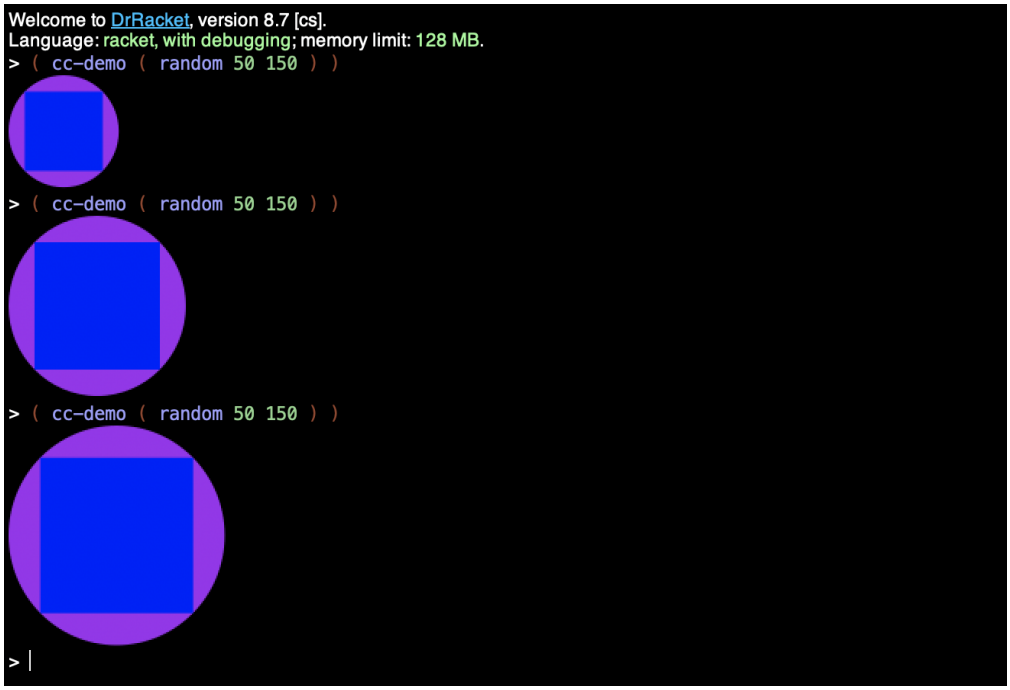
```
>
```

Task 2: Definitions - Inscribing/Circumscribing Circles/Squares

cs-demo

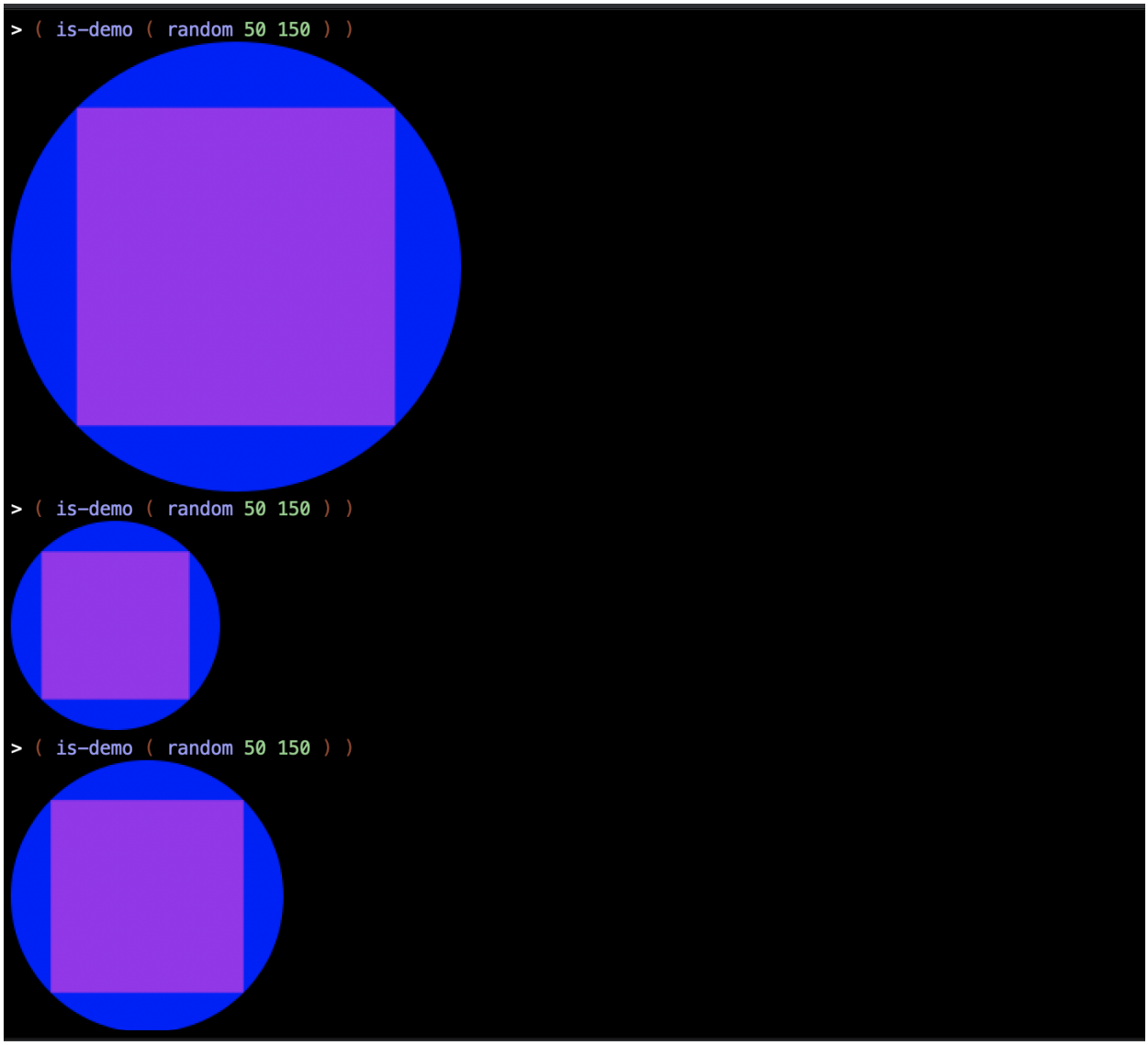


cc-demo



ic-demo

is-demo


> (is-demo (random 50 150))
> (is-demo (random 50 150))
> (is-demo (random 50 150))

The Code

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

( define ( cs n ) ( * n 2) )

( define ( cc n )
   ( sqrt (+ (* (/ n 2) (/ n 2)) (* (/ n 2) (/ n 2))))
   )

( define ( ic n ) (/ n 2 ) )

(define ( is n )
   ( sqrt (* ( * n n ) 2 ) )
   )

( define (cs-demo n)
( define the-square ( square (cs n) "solid" "purple" ) )
(define the-circle ( circle n "solid" "blue" ) )
    (overlay the-circle the-square ) )


( define (cc-demo n)
( define the-circle ( circle (cc n) "solid" "purple" ) )
( define the-square ( square n "solid" "blue" ) )
   (overlay the-square the-circle ) )

( define (ic-demo n)
( define the-circle ( circle (ic n) "solid" "purple" ) )
( define the-square ( square n "solid" "blue" ) )
   (overlay the-circle the-square ) )

( define (is-demo n)
( define the-circle ( circle n "solid" "blue" ) )
( define the-square ( square (is n) "solid" "purple" ) )
   (overlay the-square the-circle ) )
```

Task 3: Inscribing/Circumscribing Images

Image 1 Demo

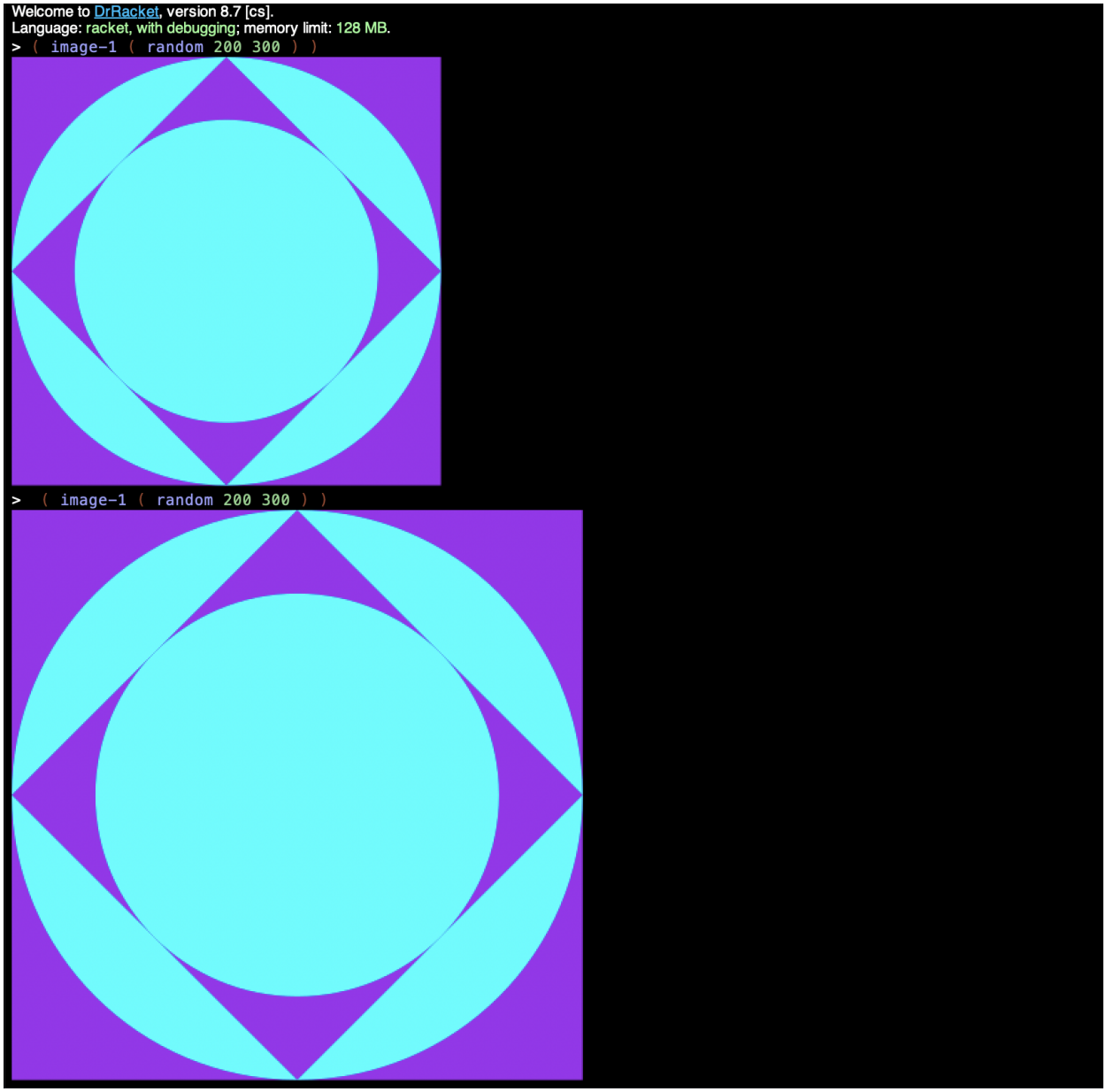
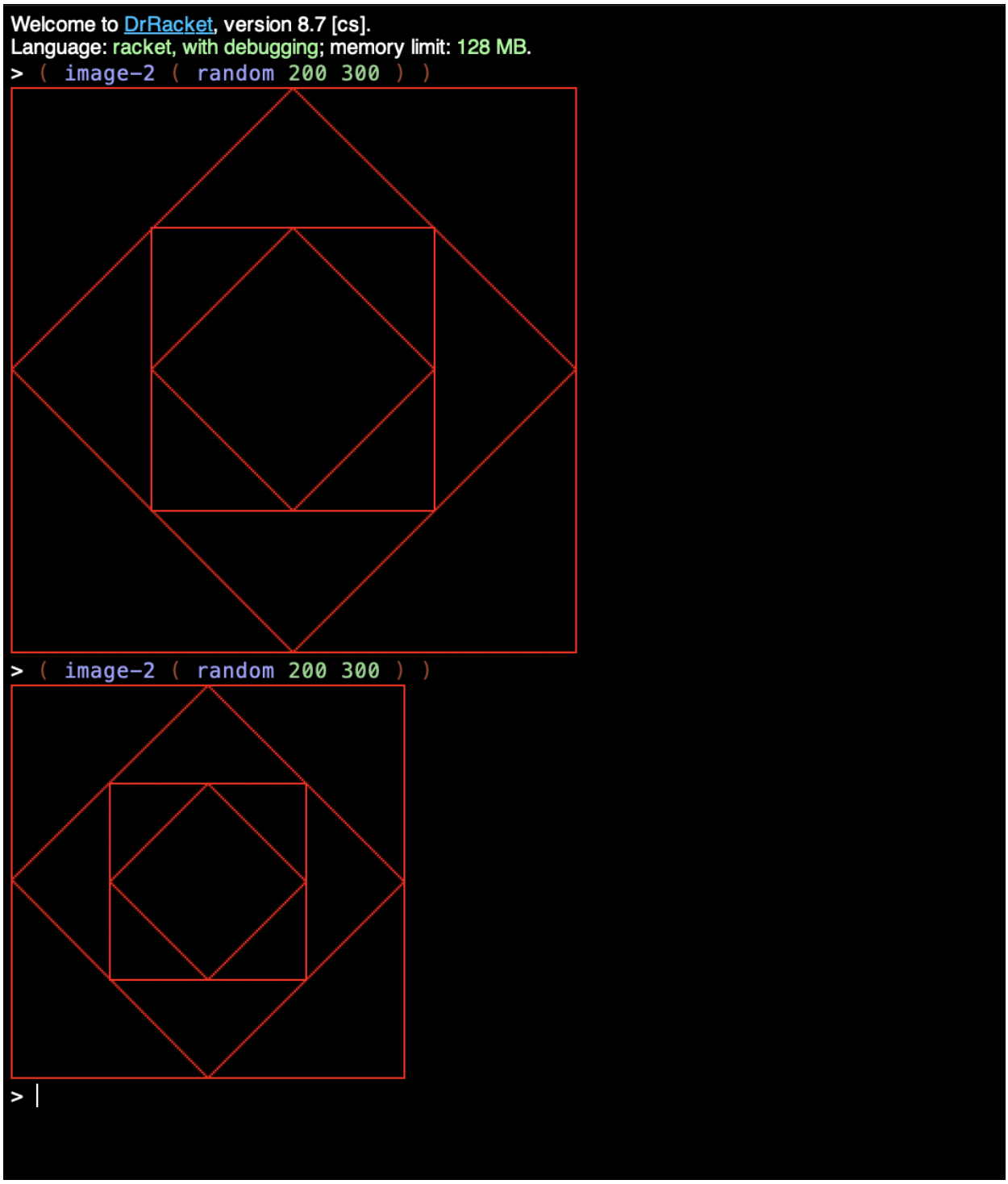
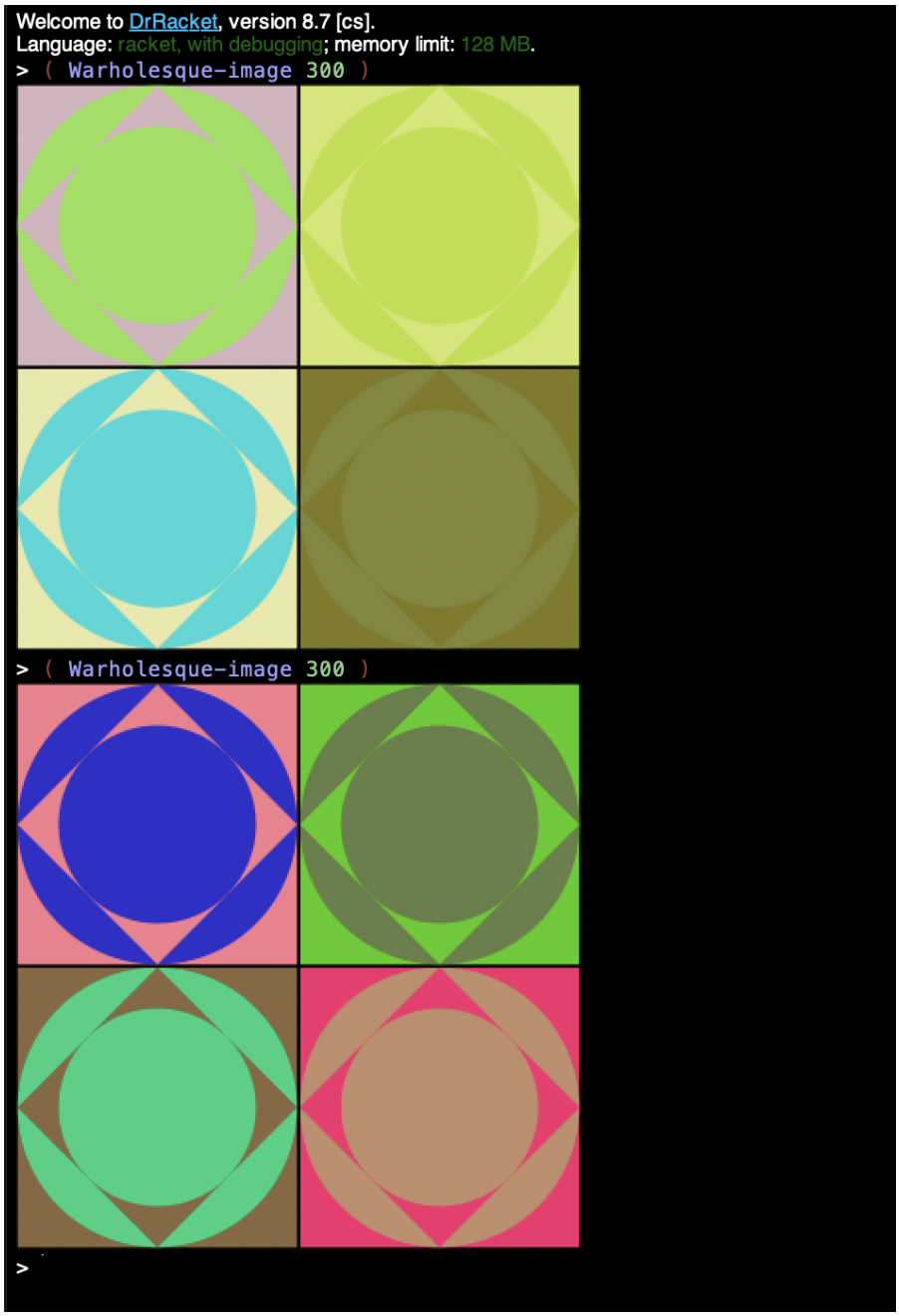


Image 2 Demo


Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (image-2 (random 200 300))
> (image-2 (random 200 300))
>

Warholesque Image



The Code

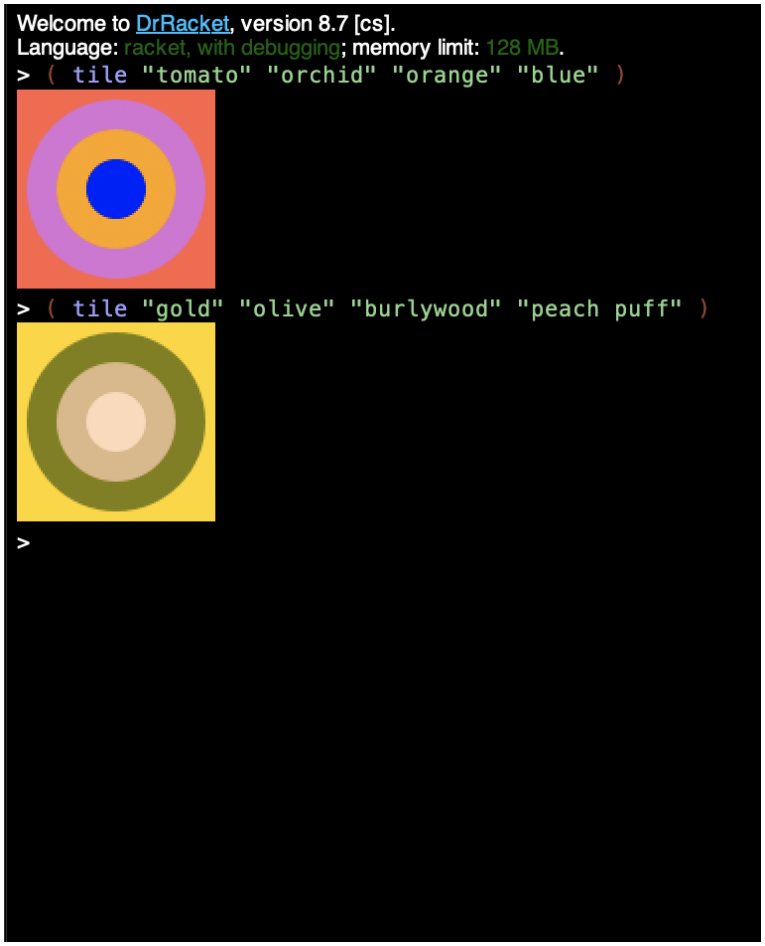
```
( define ( image-1 n )
   (overlay ( circle ( ic n) "solid" "cyan" )
             ( rotate 45 ( square (cs (ic n) ) "solid" "purple"
))
            ( circle ( cc (cs (ic n) ) ) "solid" "cyan" )
            ( square (cs ( cc (cs (ic n) ) )) "solid" "purple" )
            ) )


( define ( image-2 n )
   ( define sq1 ( is ( / n 4 ) ) )
   ( define dsq1 ( rotate 45 ( square sq1 "outline" "red" ) ) )

   ( define sq2  ( cs ( cc sq1) ) )
   ( define dsq2 ( square sq2 "outline" "red" ) )

   ( define sq3 ( cs ( cc sq2 ) ) )
   ( define dsq3 ( rotate 45 ( square sq3 "outline" "red" ) ) )

   ( define sq4 ( cs ( cc sq3 ) ) )
   ( define dsq4 ( square sq4 "outline" "red" ) )
   ( underlay dsq4 dsq3 dsq2 dsq1 )
  )


( define ( Warholesque-image n )


      (define (rgb)
      (random 256))
      ( define ( sc ) ( color (rgb) (rgb) (rgb) ) )

      ( define sc1 ( sc ) )
      ( define cc1 ( sc ) )
      ( define sc2 ( sc ) )
```

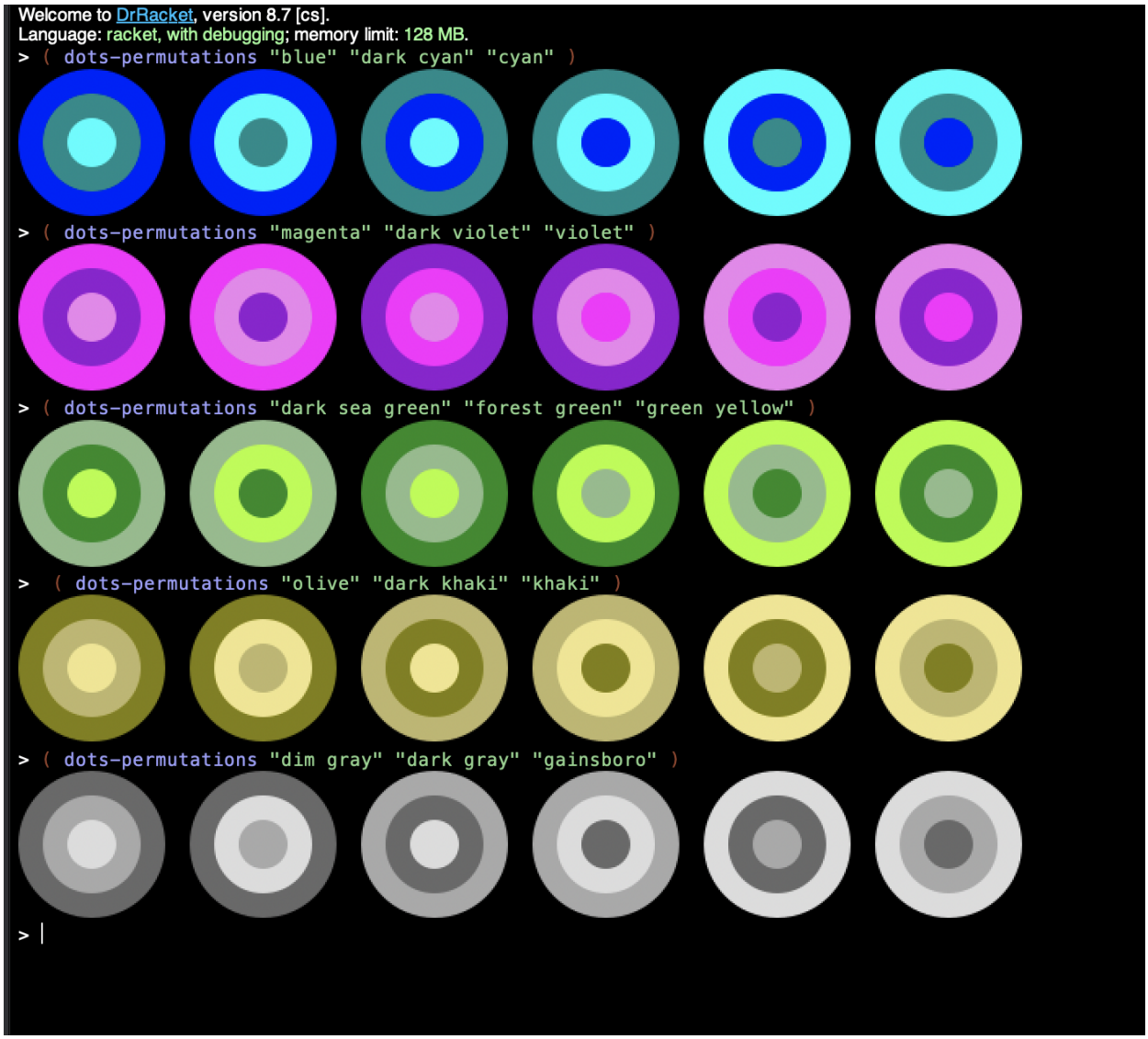
```
      ( define cc2 ( sc ) )
      ( define sc3 ( sc ) )
      ( define cc3 ( sc ) )
      ( define sc4 ( sc ) )
      ( define cc4 ( sc ) )

   ( define squareSide ( / n 2 ) )


   ( define squareSide2
      ( - squareSide 45 ) )

   ( define ( BackSquare n )
      (  square n "solid" "black") )



( define ( shapes1 squareSide )
   ( define border ( square squareSide "outline" "black" ) )

   (overlay( circle ( ic squareSide2) "solid" sc1 )
    ( rotate 45 ( square (cs (ic squareSide2) ) "solid" cc1 ))
            ( circle ( cc (cs (ic squareSide2) ) ) "solid" sc1 )
             ( square (cs ( cc (cs (ic squareSide2) ) )) "solid"
cc1 )
             border )
     )


( define ( shapes2 squareSide )
   ( define border ( square squareSide "outline" "black" ) )
   (overlay ( circle ( ic squareSide2) "solid" sc2 )
             ( rotate 45 ( square (cs (ic squareSide2) ) "solid"
cc2 ))
            ( circle ( cc (cs (ic squareSide2) ) ) "solid" sc2 )
             ( square (cs ( cc (cs (ic squareSide2) ) )) "solid"
cc2 )
            border )
```

```
     )

( define ( shapes3 squareSide )
   ( define border ( square squareSide "outline" "black" ) )
   (overlay ( circle ( ic squareSide2) "solid" sc3 )
             ( rotate 45 ( square (cs (ic squareSide2) ) "solid"
cc3 ))
            ( circle ( cc (cs (ic squareSide2) ) ) "solid" sc3 )
             ( square (cs ( cc (cs (ic squareSide2) ) )) "solid"
cc3 )
            border )
     )


( define ( shapes4 squareSide )
   ( define border ( square squareSide "outline" "black" ) )
   (overlay ( circle ( ic squareSide2) "solid" sc4 )
             ( rotate 45 ( square (cs (ic squareSide2) ) "solid"
cc4 ))
            ( circle ( cc (cs (ic squareSide2) ) ) "solid" sc4 )
             ( square (cs ( cc (cs (ic squareSide2) ) )) "solid"
cc4 )
            border )
     )

( overlay
  ( above ( beside ( shapes1 squareSide) (shapes2 squareSide) )
          ( beside ( shapes3 squareSide ) (shapes4 squareSide) )
  ( BackSquare n )
)

)
```

Task 4: Permutations of Randomly Colored Stacked Dots

Demo





Code

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

( define ( rgb)
   (random 256 ) )

( define squareSide 100 )
( define radius1 (/ 90 2 ) )
( define radius2 (/ 60 2 ) )
```

```
( define radius3 (/ 30 2 ) )

 ( define (shc) ( color (rgb) (rgb) (rgb) ) )
 ( define color1 (shc) )
 ( define color2 (shc) )
 ( define color3 (shc) )
 ( define color4 (shc) )

( define ( tile color1 color2 color3 color4 )
     ( define the-square ( square squareSide "solid" color1 ) )
     ( define circle1 ( circle radius1 "solid" color2 ) )
     ( define circle2 ( circle radius2 "solid" color3 ) )
     ( define circle3 ( circle radius3 "solid" color4 ) )
     ( underlay the-square circle1 circle2 circle3 ) )


( define ( dots-permutations color1 color2 color3)
     ( define (tile1  color1 color2 color3 )
            ( define circle1 ( circle radius1 "solid" color1 ) )
            ( define circle2 ( circle radius2 "solid" color2 ) )
            ( define circle3 ( circle radius3 "solid" color3 ) )
            ( underlay circle1 circle2 circle3 )  )

   ( define (tile2  color1 color2 color3 )
            ( define circle1 ( circle radius1 "solid" color1 ) )
            ( define circle2 ( circle radius2 "solid" color3 ) )
            ( define circle3 ( circle radius3 "solid" color2 ) )
            ( underlay circle1 circle2 circle3 )  )

   ( define (tile3  color1 color2 color3 )
            ( define circle1 ( circle radius1 "solid" color2 ) )
            ( define circle2 ( circle radius2 "solid" color1 ) )
            ( define circle3 ( circle radius3 "solid" color3 ) )
            ( underlay circle1 circle2 circle3 )  )

   ( define (tile4  color1 color2 color3 )
            ( define circle1 ( circle radius1 "solid" color2 ) )
            ( define circle2 ( circle radius2 "solid" color3 ) )
```

```
        ( define circle3 ( circle radius3 "solid" color1 ) )
        ( underlay circle1 circle2 circle3 )  )

( define (tile5  color1 color2 color3 )
        ( define circle1 ( circle radius1 "solid" color3 ) )
        ( define circle2 ( circle radius2 "solid" color1 ) )
        ( define circle3 ( circle radius3 "solid" color2 ) )
        ( underlay circle1 circle2 circle3 )  )

( define (tile6  color1 color2 color3 )
        ( define circle1 ( circle radius1 "solid" color3 ) )
        ( define circle2 ( circle radius2 "solid" color2 ) )
        ( define circle3 ( circle radius3 "solid" color1 ) )
        ( underlay circle1 circle2 circle3 )  )

( define space ( square 15 "solid" "black" ) )

( beside ( tile1 color1 color2 color3 ) space
        ( tile2 color1 color2 color3 ) space
        ( tile3 color1 color2 color3 ) space
        ( tile4 color1 color2 color3 ) space
        ( tile5 color1 color2 color3 ) space
        ( tile6 color1 color2 color3 ))
)
```