Racket Assignment #2: Interactions, Definitions, Applications

Written by David Hennigan

Learning Abstract

In this assignment tasks were completed which helped further my understanding of Racket. A variety of functions revolving around shapes were used. The problems in this assignment allowed for a great deal of expression and creativity.

Task 1: Interactions – Scrap of Tin

Arithmetic Expressions

> 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 9)

12157665459056928801

Solve a Simple Problem (Area of Scrap)

> pi

3.141592653589793

> side

. . side: undefined;

 cannot reference an identifier before its definition

> (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))

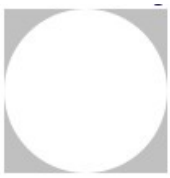
> scrap-area

2146.018366025517

Rendering an Image of the Problem Situation

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

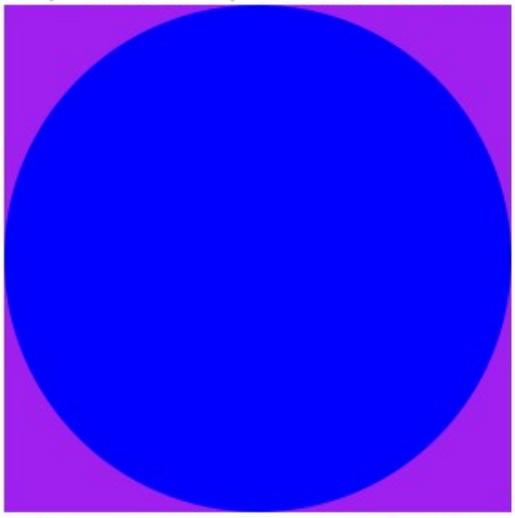
```
> ( define radius ( / side 2 ) )
> ( define the-circle ( circle radius "solid" "white" ) )
> ( define the-image ( overlay the-circle the-square ) )
> the-image
```



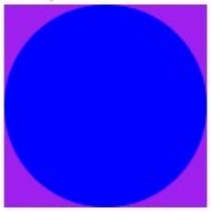
Task 2: Definitions – Inscribing/Circumscribing Circles/Squares

cs-demo

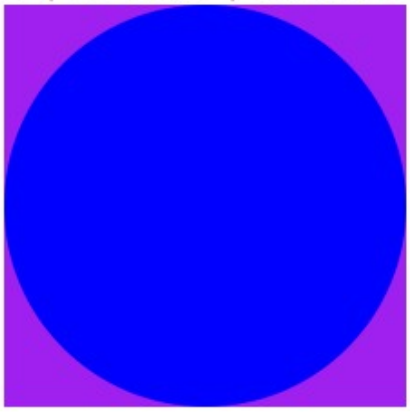
```
> ( cs-demo ( random 50 150 ) )
```



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



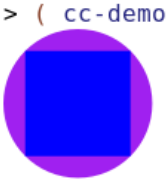
```
> ( cs-demo ( random 50 150 ) )
```



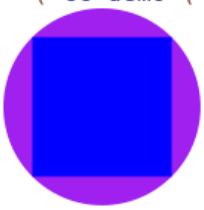
```
>
```

cc-demo

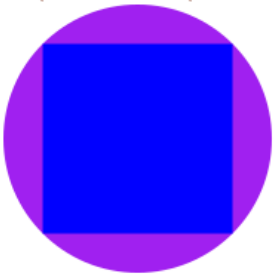
```
> ( cc-demo ( random 50 150 ) )
```



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



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



```
>
```

ic-demo

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

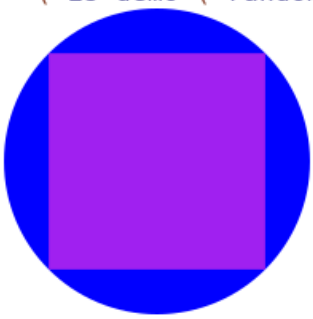
```
> ( ic-demo ( random 50 150 ) )
```

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

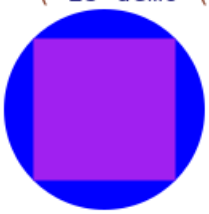
```
>
```

is-demo

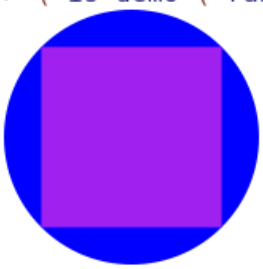
```
> ( is-demo ( random 50 150 ) )
```



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



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



```
>
```

The Code

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

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

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

( define ( ic side )
```

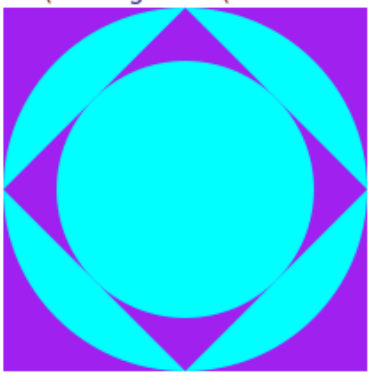
```
    ( / side 2 )
)

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

( define ( cs-demo radius )
   ( define c ( circle radius 'solid 'blue ) )
   ( define s ( square ( cs radius ) 'solid 'purple ) )
   ( overlay c s )
)

( define ( cc-demo side )
   ( define s ( square side 'solid 'blue ) )
   ( define c ( circle ( cc side ) 'solid 'purple ) )
   ( overlay s c )
)

( define ( ic-demo side )
   ( define s ( square side 'solid 'blue ) )
   ( define c ( circle ( ic side ) 'solid 'purple ) )
   ( overlay c s )
)

( define ( is-demo radius )
   ( define c ( circle radius 'solid 'blue ) )
   ( define s ( square ( is radius ) 'solid 'purple ) )
   ( overlay s c )
)
```

Task 3: Inscribing/Circumscribing Images

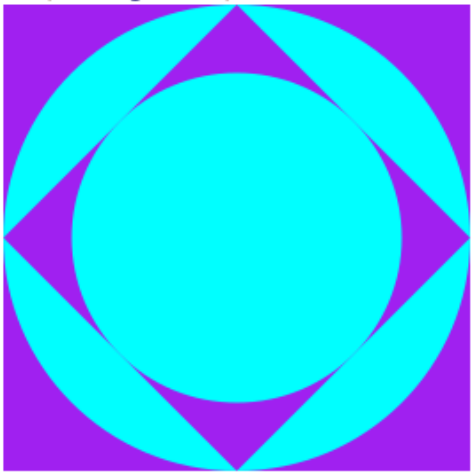
Image 1 Demo

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

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



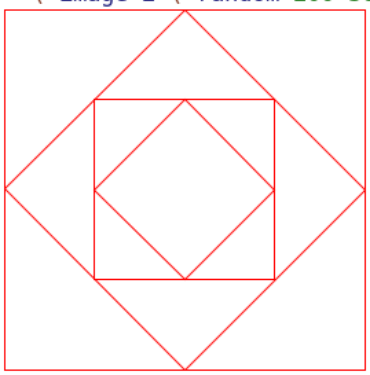
```
> ( image-1 ( random 200 300 ) )
```



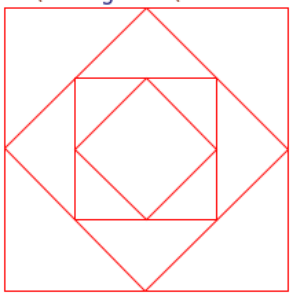
```
>
```

Image 2 Demo

```
> ( image-2 ( random 200 300 ) )
```



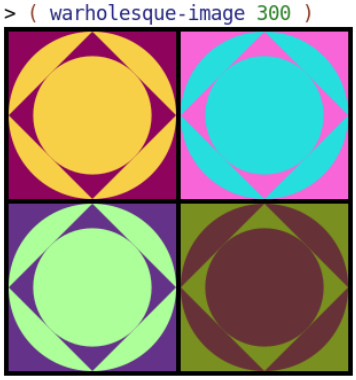
```
> ( image-2 ( random 200 300 ) )
```



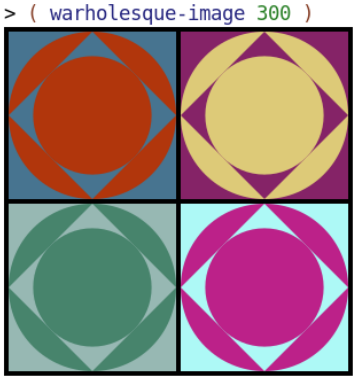
```
>
```

Warholesque Image

```
> ( warholesque-image 300 )
```



```
> ( warholesque-image 300 )
```



```
>
```

The Code

```
( define ( image-1  side )
   ( define outer-s ( square side 'solid 'purple ) )
   ( define outer-radius ( ic side ) )
   ( define outer-c ( circle outer-radius 'solid 'cyan ) )
   ( define inner-side ( is outer-radius ) )
   ( define inner-s ( square inner-side 'solid 'purple ) )
   ( define inner-s-rotated ( rotate 45 inner-s ) )
   ( define inner-c ( circle ( ic inner-side ) 'solid 'cyan ) )
   ( define inner-shape ( overlay inner-c inner-s-rotated ) )
   ( define outer-shape ( overlay outer-c outer-s ) )
   ( overlay inner-shape outer-shape )
)

( define ( image-2 side )
   ( define outer-s ( square side 'outline 'red ) )
   ( define side-outer-rotated ( is (/ side 2 ) ) )
   ( define rotated-outer-s ( rotate 45 ( square side-outer-rotated 'outline 'red ) ) )
   ( define inner-side ( is (/ side-outer-rotated 2) ) )
   ( define inner-s ( square inner-side 'outline 'red ) )
   ( define side-inner-rotated ( is (/ inner-side 2 ) ) )
   ( define rotated-inner-s ( rotate 45 ( square side-inner-rotated 'outline 'red ) ) )
   ( define inner ( overlay rotated-inner-s inner-s ) )
   ( define outer ( overlay rotated-outer-s outer-s ) )
   ( overlay inner outer )
)

( define (warholesque-image outer-side )
   ( define ( image-part  side )
```

```
  ( define ( rgb ) ( random 0 256 ) )
  ( define color-1 ( color ( rgb ) ( rgb ) ( rgb ) ) )
  ( define color-2 ( color ( rgb ) ( rgb ) ( rgb ) ) )
  ( define outer-s ( square side 'solid color-1 ) )
  ( define outer-radius ( ic side ) )
  ( define outer-c ( circle outer-radius 'solid color-2 ) )
  ( define inner-side ( is outer-radius ) )
  ( define inner-s ( square inner-side 'solid color-1 ) )
  ( define inner-s-rotated ( rotate 45 inner-s ) )
  ( define inner-c ( circle ( ic inner-side ) 'solid color-2 ) )
  ( define inner-shape ( overlay inner-c inner-s-rotated ) )
  ( define outer-shape ( overlay outer-c outer-s ) )
  ( overlay inner-shape outer-shape )
)
( define outer-horz-rect ( rectangle (- (/ outer-side 2) 6) 4 'solid 'black ) )
( define inner-horz-rect ( rectangle (- (/ outer-side 2) 6) 2 'solid 'black ) )
( define outer-vert-rect ( rectangle 4 (/ outer-side 2 )  'solid 'black ) )
( define inner-vert-rect ( rectangle 2 (/ outer-side 2 ) 'solid 'black ) )
( define inner-side ( - ( / outer-side 2) 6 ) )

( define top-left ( image-part inner-side) )
( define top-left-with-border ( beside outer-vert-rect ( beside ( above outer-horz-rect ( above top-left
      inner-horz-rect ) ) inner-vert-rect ) ) )

( define bottom-left ( image-part inner-side) )
( define bottom-left-with-border ( beside outer-vert-rect ( beside ( above inner-horz-rect ( above
      bottom-left outer-horz-rect ) ) inner-vert-rect ) ) )

( define top-right ( image-part inner-side) )
( define top-right-with-border ( beside inner-vert-rect ( beside ( above outer-horz-rect ( above top-
      right inner-horz-rect ) ) outer-vert-rect ) ) )
```

```
  ( define bottom-right ( image-part inner-side) )
  ( define bottom-right-with-border ( beside inner-vert-rect ( beside ( above inner-horz-rect ( above
        bottom-right outer-horz-rect ) ) outer-vert-rect ) ) )


  ( define left-side ( above top-left-with-border bottom-left-with-border ) )
  ( define right-side ( above top-right-with-border bottom-right-with-border ) )
  ( beside left-side right-side )
)
```

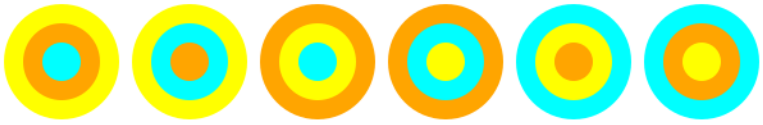
Task 4: Permutations of Randomly Colored Stacked Dots

Demo

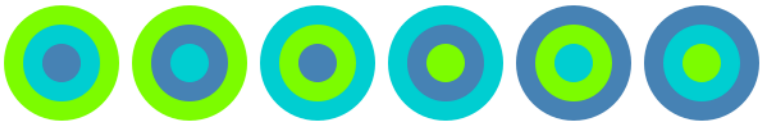
```
> ( tile "Dark Sea Green" "Medium Turquoise" "Black" "Honeydew" )
```

```
> ( tile "Dodger Blue" "Light Sky Blue" "Crimson" "Orchid" )
```

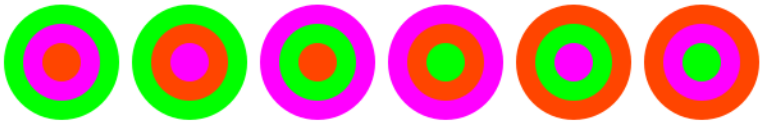
```
> ( dots-permutations 'yellow 'orange 'cyan )
```



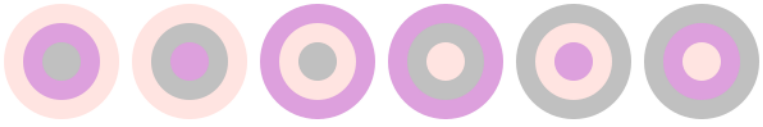
```
> ( dots-permutations "Lawn Green" "Dark Turquoise" "Steel Blue" )
```



```
> ( dots-permutations "Lime" "Magenta" "Orange Red" )
```



```
> ( dots-permutations "Misty Rose" "Plum" "Silver" )
```



```
>
```

Code

```
#lang racket


( require 2htdp/image )


( define ( tile background-color c1-color c2-color c3-color )
   ( define tile-background ( square 100 'solid background-color ) )
   ( define c1 ( circle 45 'solid c1-color ) )
   ( define c2 ( circle 30 'solid c2-color ) )
   ( define c3 ( circle 15 'solid c3-color ) )
   ( overlay c3 c2 c1 tile-background )
)


( define ( dots-permutations color1 color2 color3 )
   ( define dot1 ( tile 'white color1 color2 color3 ) )
   ( define dot2 ( tile 'white color1 color3 color2 ) )
   ( define dot3 ( tile 'white color2 color1 color3 ) )
   ( define dot4 ( tile 'white color2 color3 color1 ) )
   ( define dot5 ( tile 'white color3 color1 color2 ) )
   ( define dot6 ( tile 'white color3 color2 color1 ) )
   ( beside dot1 dot2 dot3 dot4 dot5 dot6 )
)
```