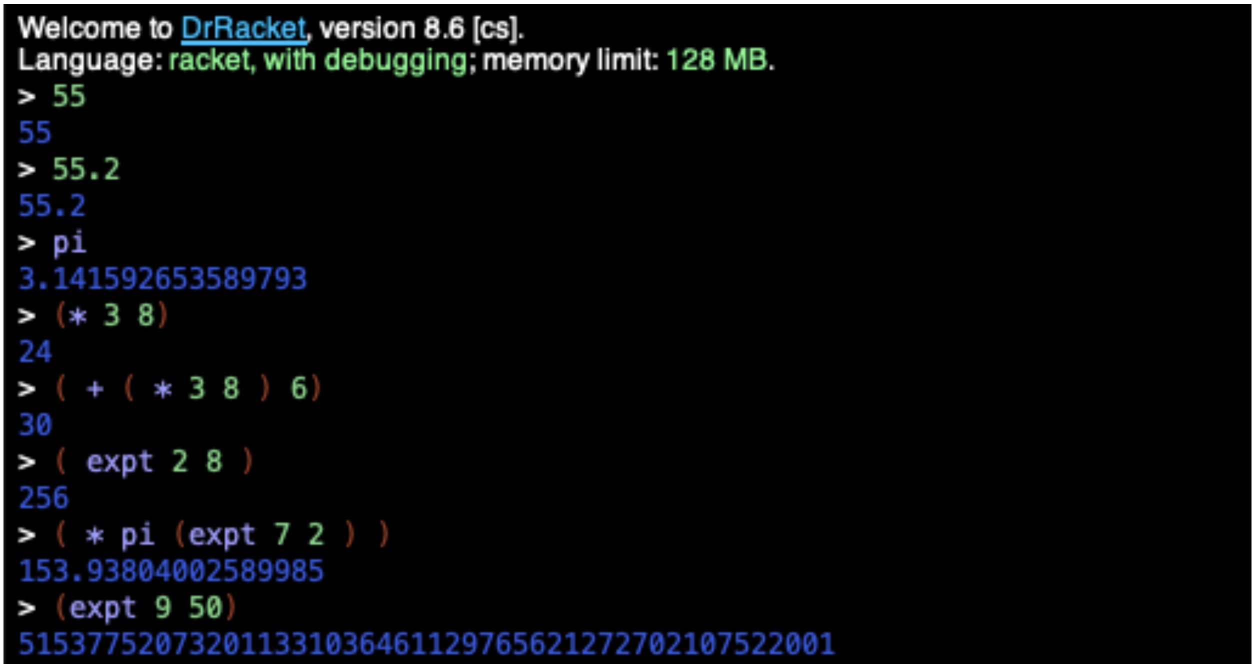
Racket Programming Assignment #1: First Interactions and Model

Learning Abstract

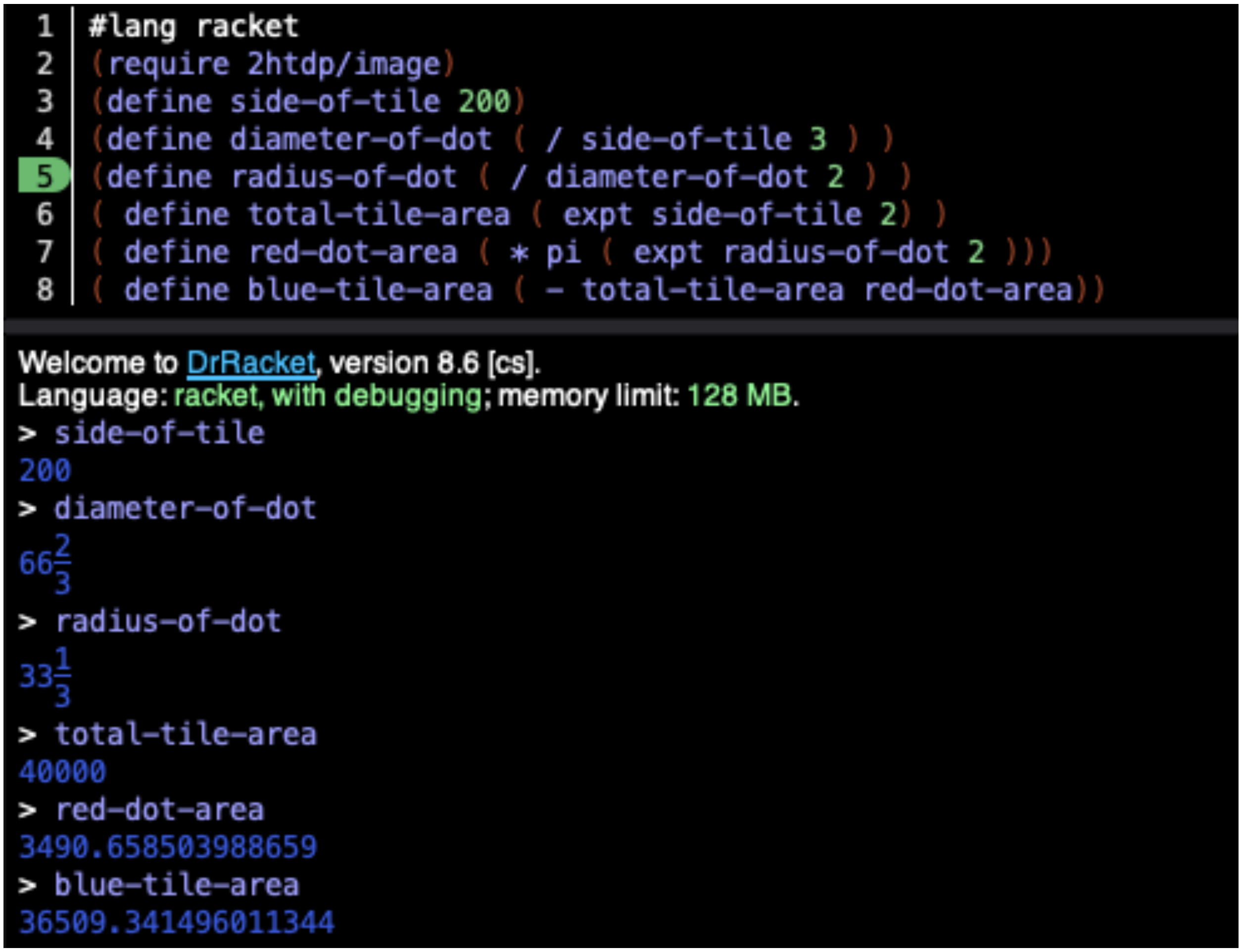
In this assignment, I have learned the basic mathematical codes for the programming software called DrRacket as well as creating, coloring and modifying each shape based on the lessons that the professor provided. As a result, numeric computations based on Lisp, creating a square tile with a red dot in the circle and finding the area of the remaining blue area in a square tile, and creating the blue and red concentric disks images and finding the area of the concentric disks image which is blue, were done based on the lessons in the class.

Interaction: Simple Numeric Processing



```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> 55
55
> 55.2
55.2
> pi
3.141592653589793
> (* 3 8)
24
> ( + ( * 3 8 ) 6)
30
> ( expt 2 8 )
256
> ( * pi (expt 7 2 ) )
153.93804002589985
> (expt 9 50)
515377520732011331036461129765621272702107522001
```

Interaction: Solution to the blue and red tile area problem



```racket
#lang racket
(require 2htdp/image)
(define side-of-tile 200)
(define diameter-of-dot ( / side-of-tile 3 ) )
(define radius-of-dot ( / diameter-of-dot 2 ) )
( define total-tile-area ( expt side-of-tile 2) )
( define red-dot-area ( * pi ( expt radius-of-dot 2 )))
( define blue-tile-area ( - total-tile-area red-dot-area))
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> side-of-tile
200
> diameter-of-dot
```

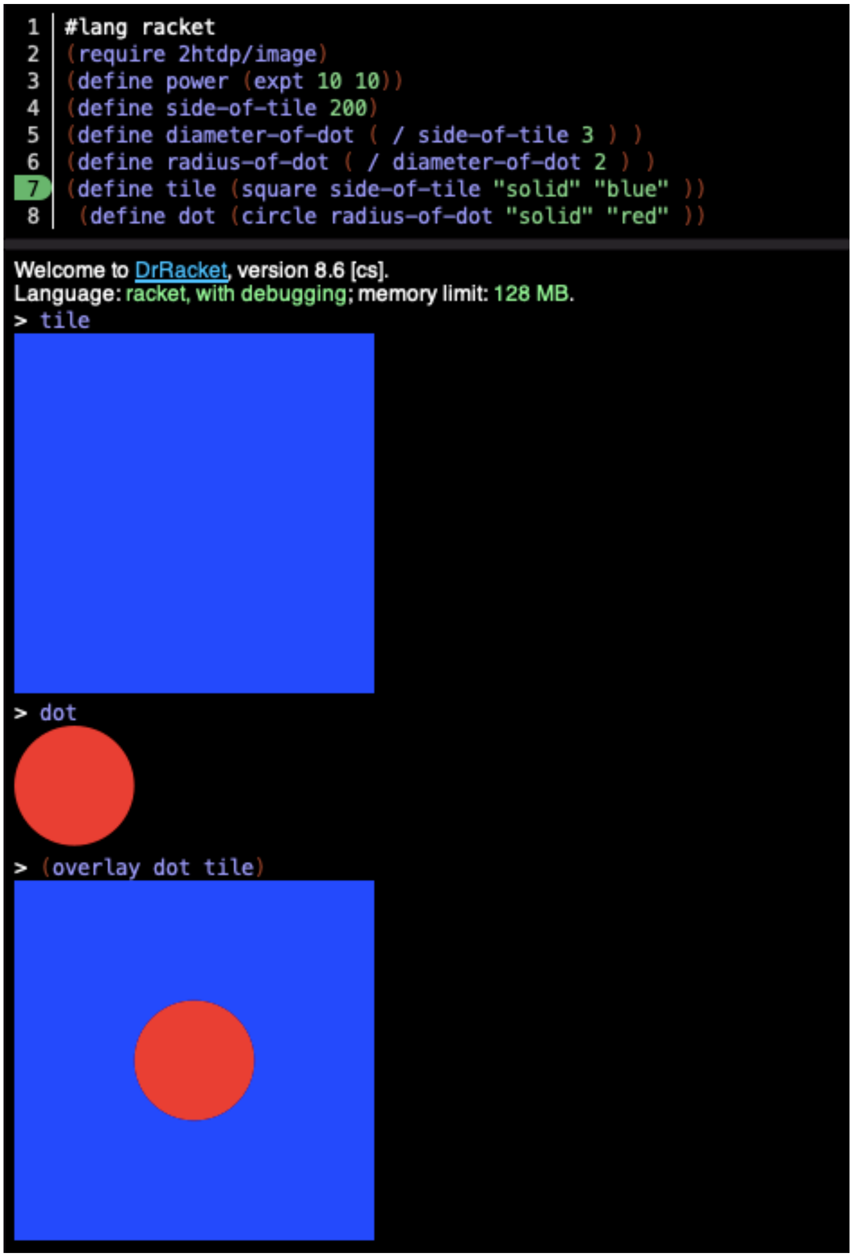
$66\frac{2}{3}$

```
> radius-of-dot
```

$33\frac{1}{3}$

```
> total-tile-area
40000
> red-dot-area
3490.658503988659
> blue-tile-area
36509.341496011344
```

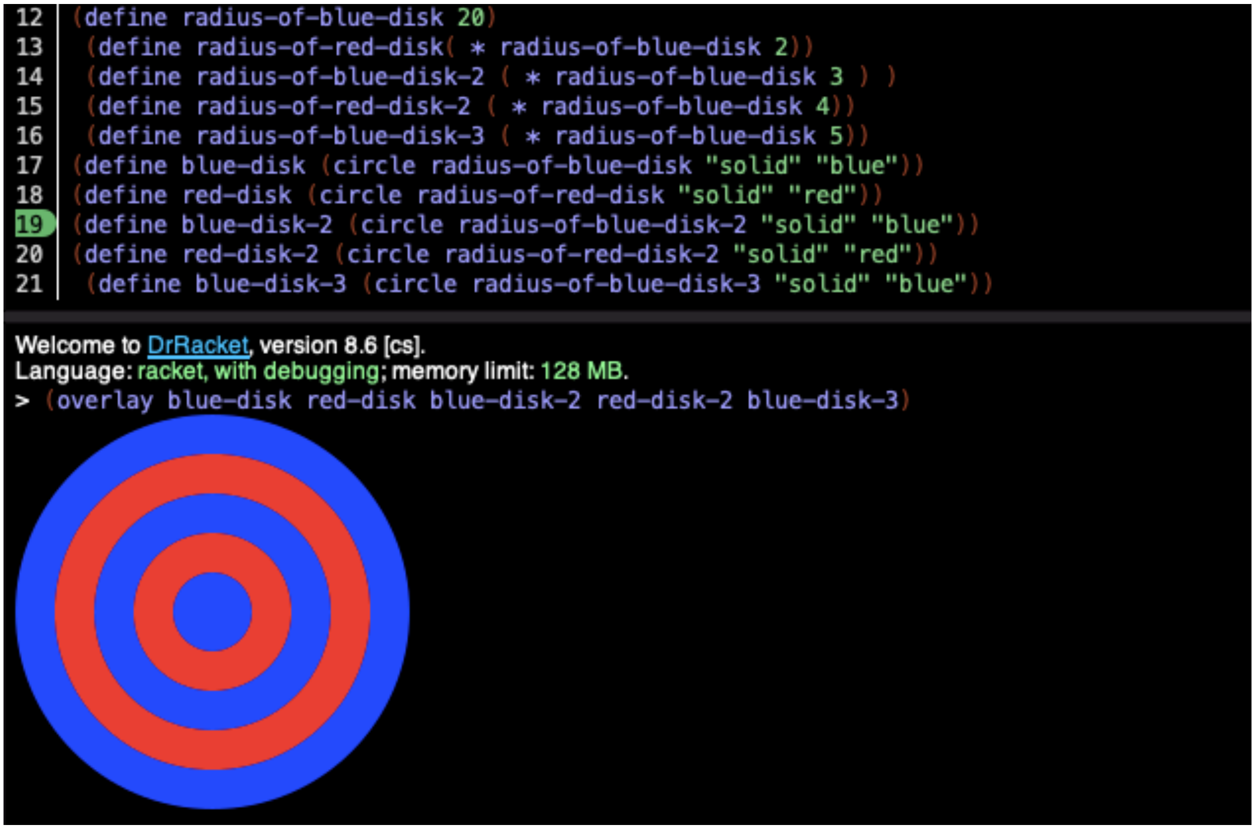
Interaction: Painting the blue and red tile



```
#lang racket
(require 2htdp/image)
(define power (expt 10 10))
(define side-of-tile 200)
(define diameter-of-dot ( / side-of-tile 3 ) )
(define radius-of-dot ( / diameter-of-dot 2 ) )
(define tile (square side-of-tile "solid" "blue" ))
 (define dot (circle radius-of-dot "solid" "red" ))
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> tile
> dot
> (overlay dot tile)
```

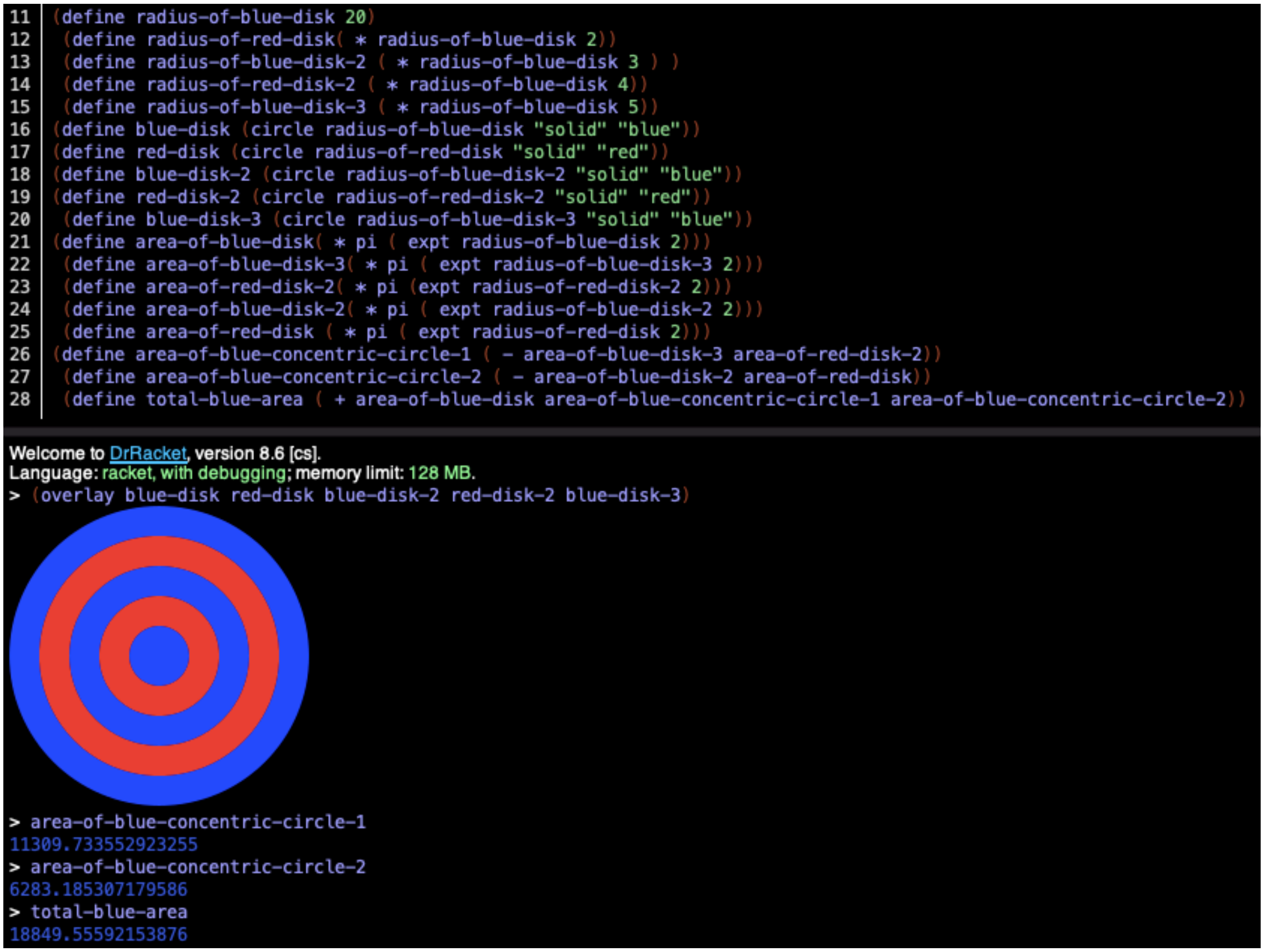
Interaction: Painting the blue and red concentric disks images



```
(define radius-of-blue-disk 20)
 (define radius-of-red-disk( * radius-of-blue-disk 2))
 (define radius-of-blue-disk-2 ( * radius-of-blue-disk 3 ) )
 (define radius-of-red-disk-2 ( * radius-of-blue-disk 4))
 (define radius-of-blue-disk-3 ( * radius-of-blue-disk 5))
(define blue-disk (circle radius-of-blue-disk "solid" "blue"))
(define red-disk (circle radius-of-red-disk "solid" "red"))
(define blue-disk-2 (circle radius-of-blue-disk-2 "solid" "blue"))
(define red-disk-2 (circle radius-of-red-disk-2 "solid" "red"))
 (define blue-disk-3 (circle radius-of-blue-disk-3 "solid" "blue"))
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (overlay blue-disk red-disk blue-disk-2 red-disk-2 blue-disk-3)
```

Interaction: Computing the area of the concentric disks image which is blue



```
(define radius-of-blue-disk 20)
 (define radius-of-red-disk( * radius-of-blue-disk 2))
 (define radius-of-blue-disk-2 ( * radius-of-blue-disk 3 ) )
 (define radius-of-red-disk-2 ( * radius-of-blue-disk 4))
 (define radius-of-blue-disk-3 ( * radius-of-blue-disk 5))
(define blue-disk (circle radius-of-blue-disk "solid" "blue"))
(define red-disk (circle radius-of-red-disk "solid" "red"))
(define blue-disk-2 (circle radius-of-blue-disk-2 "solid" "blue"))
(define red-disk-2 (circle radius-of-red-disk-2 "solid" "red"))
 (define blue-disk-3 (circle radius-of-blue-disk-3 "solid" "blue"))
(define area-of-blue-disk( * pi ( expt radius-of-blue-disk 2)))
 (define area-of-blue-disk-3( * pi ( expt radius-of-blue-disk-3 2)))
 (define area-of-red-disk-2( * pi (expt radius-of-red-disk-2 2)))
 (define area-of-blue-disk-2( * pi ( expt radius-of-blue-disk-2 2)))
 (define area-of-red-disk ( * pi ( expt radius-of-red-disk 2)))
(define area-of-blue-concentric-circle-1 ( - area-of-blue-disk-3 area-of-red-disk-2))
 (define area-of-blue-concentric-circle-2 ( - area-of-blue-disk-2 area-of-red-disk))
 (define total-blue-area ( + area-of-blue-disk area-of-blue-concentric-circle-1 area-of-blue-concentric-circle-2))
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (overlay blue-disk red-disk blue-disk-2 red-disk-2 blue-disk-3)

> area-of-blue-concentric-circle-1
11309.733552923255
> area-of-blue-concentric-circle-2
6283.185307179586
> total-blue-area
18849.55592153876
```