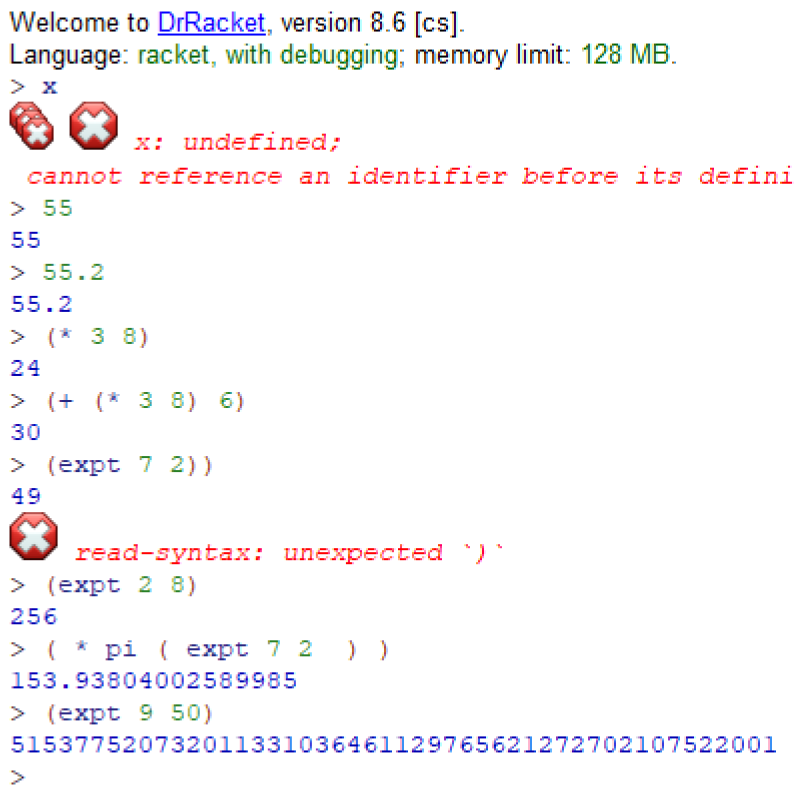
Racket Programming Assignment #1: First Interactions

Learning Abstract

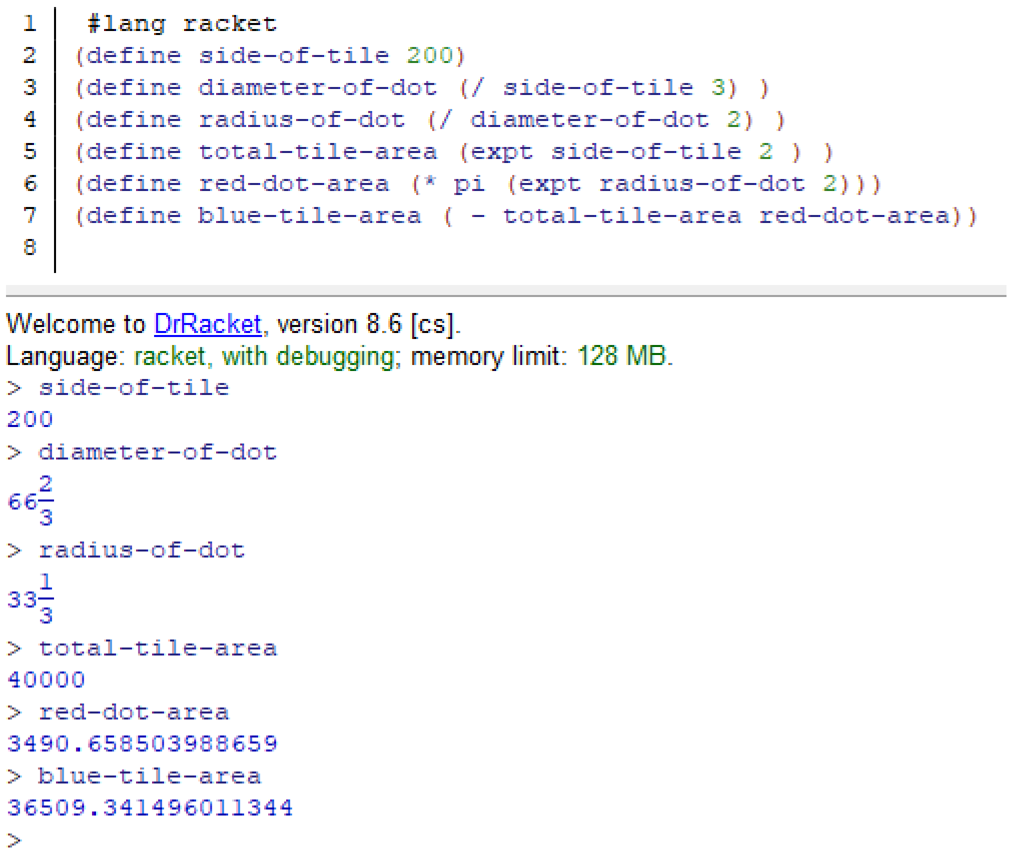
This assignment features relatively simple interactions in the Racket programming language. In fact, all of the computations take place within the interactions pane of the DrRacket PDE. In the first part of this assignment I learned a little bit about numeric computations in Lisp. The next two parts of the assignment featured a square tile which was blue except for a centered red dot. In the second part of the assignment I mimicked the solution of the problem of finding the area of the tile which was blue. In the third part I mimicked the computational rendering of the tile. The last two parts of the assignment featured an image consisting of 5 concentric squares. In the fourth part of this assignment I rendered the image. In the fifth part I computed a percentage based on the concentric squares image. Throughout the problem solving parts of this assignment the concept of binding values to variables was a predominant theme.

Interaction-1: Simple Numeric Processing



```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> x
x: undefined;
 cannot reference an identifier before its definition
> 55
55
> 55.2
55.2
> (* 3 8)
24
> (+ (* 3 8) 6)
30
> (expt 7 2))
49
read-syntax: unexpected `)`
> (expt 2 8)
256
> ( * pi ( expt 7 2   ) )
153.93804002589985
> (expt 9 50)
515377520732011331036461129765621272702107522001
>
```

Interaction-2: Solution to the blue and red tile area problem

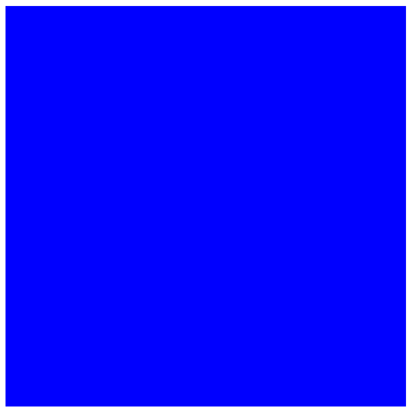


```
 #lang racket
(define side-of-tile 200)
(define diameter-of-dot (/ side-of-tile 3) )
(define radius-of-dot (/ diameter-of-dot 2) )
(define total-tile-area (expt side-of-tile 2 ) )
(define red-dot-area (* pi (expt radius-of-dot 2)))
(define blue-tile-area ( - total-tile-area red-dot-area))
```

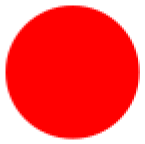
```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> side-of-tile
200
> diameter-of-dot
66 2/3
> radius-of-dot
33 1/3
> total-tile-area
40000
> red-dot-area
3490.658503988659
> blue-tile-area
36509.341496011344
>
```

Interaction-3: Painting the blue and red tile

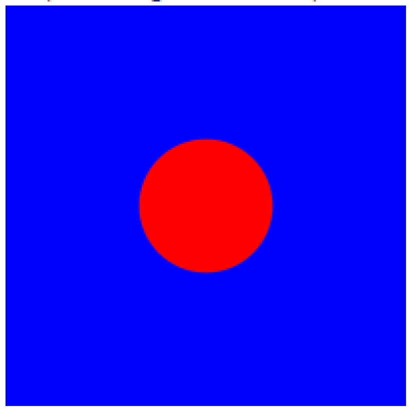
```racket
(require 2htdp/image)
(define side-of-tile 200)
(define diameter-of-dot (/ side-of-tile 3) )
(define radius-of-dot (/ diameter-of-dot 2) )
```



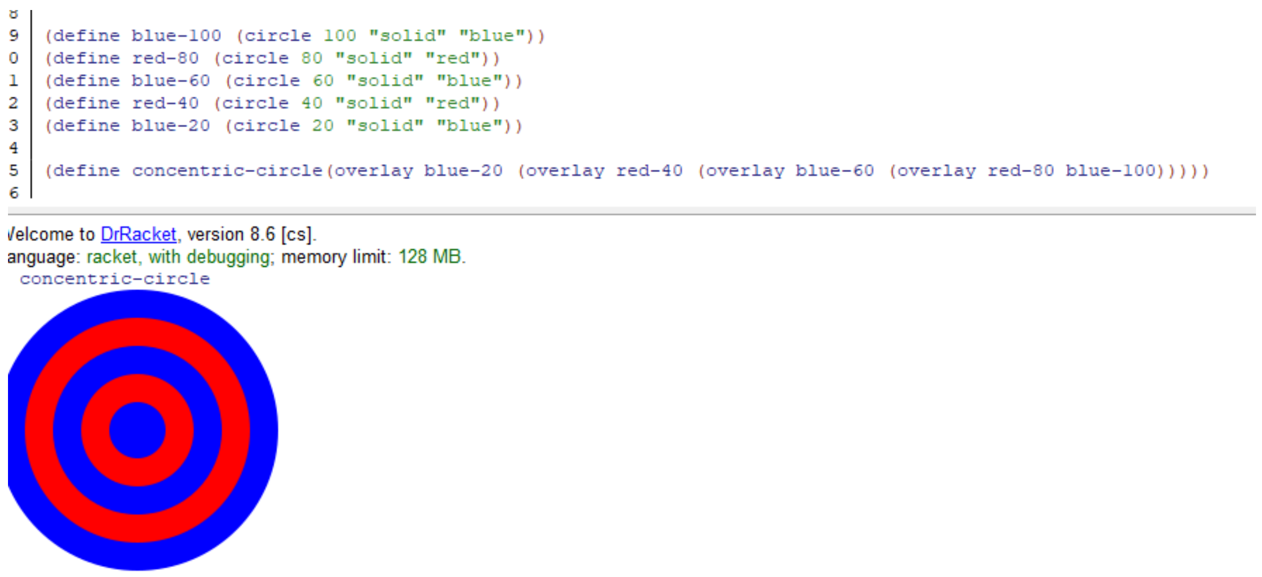
```racket
> (define dot (circle radius-of-dot "solid" "red"))
> dot
```



```racket
> (overlay dot tile)
```



Interaction-4: Painting the blue and red concentric disks images

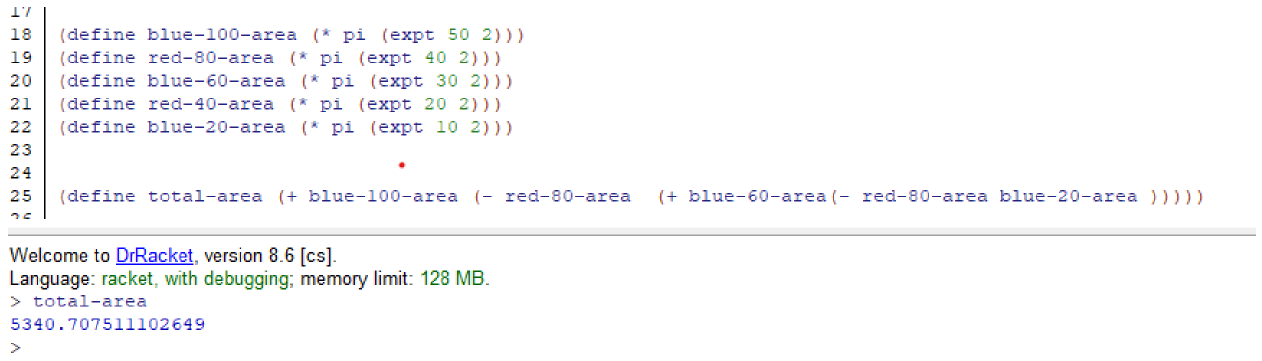


```
(define blue-100 (circle 100 "solid" "blue"))
(define red-80 (circle 80 "solid" "red"))
(define blue-60 (circle 60 "solid" "blue"))
(define red-40 (circle 40 "solid" "red"))
(define blue-20 (circle 20 "solid" "blue"))

(define concentric-circle(overlay blue-20 (overlay red-40 (overlay blue-60 (overlay red-80 blue-100)))))
```

```
elcome to DrRacket, version 8.6 [cs].
anguage: racket, with debugging; memory limit: 128 MB.
 concentric-circle
```

Interaction-5: Computing the area of the concentric disks image which is blue



```
(define blue-100-area (* pi (expt 50 2)))
(define red-80-area (* pi (expt 40 2)))
(define blue-60-area (* pi (expt 30 2)))
(define red-40-area (* pi (expt 20 2)))
(define blue-20-area (* pi (expt 10 2)))


(define total-area (+ blue-100-area (- red-80-area  (+ blue-60-area(- red-80-area blue-20-area )))))
```

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> total-area
5340.707511102649
>
```