Racket Programming Assignment 1

Learning Abstract

This assignment features relatively simple interactions in the Racket programming language. All interactions were done in the console of Racket. 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 computational rendering of the tile. In the third part I mimicked the solution of the problem of finding the area of the tile which was blue.

Interaction: Simple Numeric Processing

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> y
y: undefined;
 cannot reference an identifier before its definition
> 44
44
> 32.2
32.2
> (* 4 2)
8
> (+ (* 3 2) 4)
10
> (expt 4 4)
256
> (* pi (expt 4 2))
50.26548245743669
> (expt 9 50)
515377520732011331036461129765621272702107522001
>
```

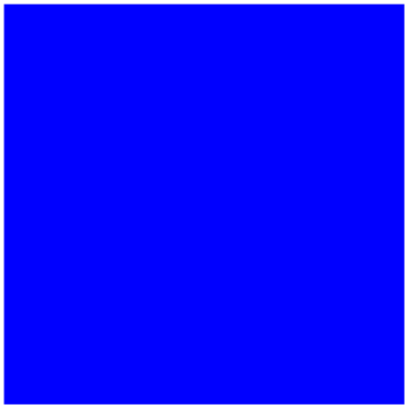
Interaction: Solution to the blue and red tile area problem

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (define sideLength 200)
> (define dotDiameter (* 1/3 sideLength))
> (define dotRadius (* 1/2 dotDiameter))
> (define dotArea (* pi (expt dotRadius 2)))
> (define squareArea (- (expt sideLength 2) dotArea))
> sideLength
200
> dotDiameter
66 2/3
> dotRadius
33 1/3
> dotArea
3490.658503988659
> squareArea
36509.341496011344
>
```

Interaction: Painting the blue and red tile

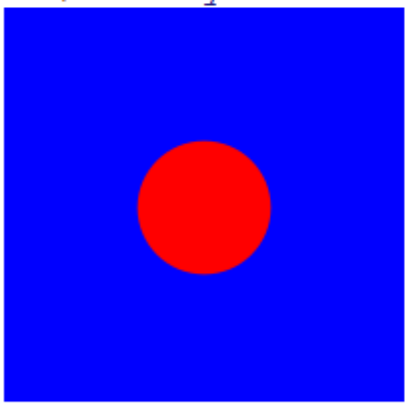
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

```
> (require 2htdp/image)
> (define sideLength 200)
> (define dotDiameter (* 1/3 sideLength))
> (define dotRadius (* 1/2 dotDiameter))
> (define tile (square sideLength 'solid 'blue))
> tile
```



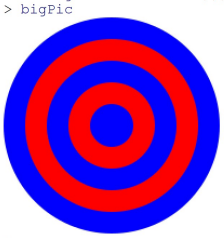
```
> (define dot (circle dotRadius 'solid 'red))
> dot
```

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



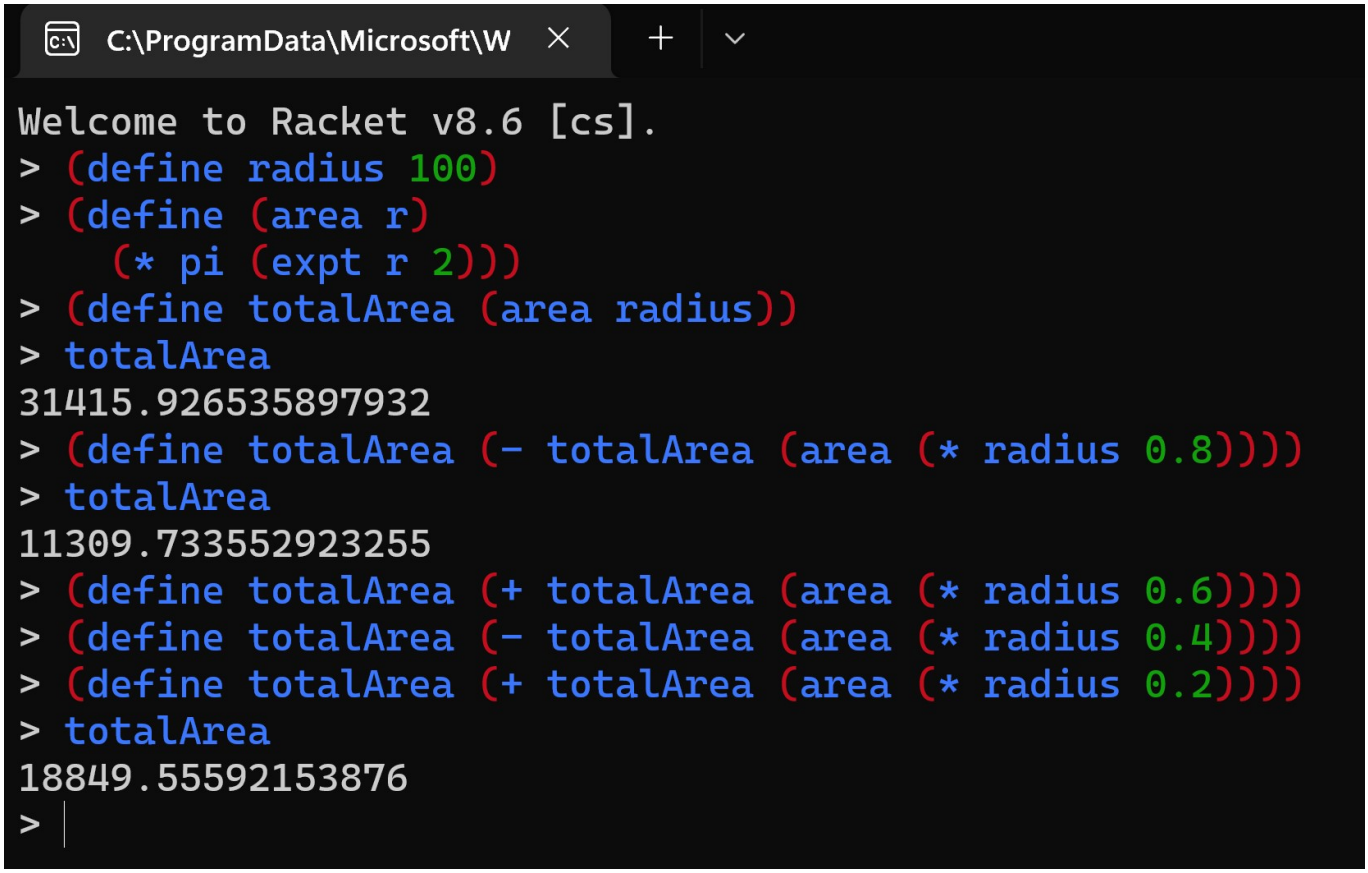
Interaction: Painting the blue and red concentric disks image

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket/base, with debugging; memory limit: 128 MB.
> (require 2htdp/image)
> (define sideLength 100)
> (define initCircle (circle (* sideLength 0.2) 'solid 'blue))
> (define twoCircles (overlay initCircle (circle (* sideLength 0.4) 'solid 'red)))
> (define bigPic (overlay twoCircles (overlay (overlay (circle (* sideLength 0.6) 'solid 'blue) (circle (* sideLength 0.8) 'solid 'red)) (circle
sideLength 'solid 'blue))))
> bigPic
```



```
>
```

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


C:\ProgramData\Microsoft\W
Welcome to Racket v8.6 [cs].
> (define radius 100)
> (define (area r)
(* pi (expt r 2)))
> (define totalArea (area radius))
> totalArea
31415.926535897932
> (define totalArea (- totalArea (area (* radius 0.8))))
> totalArea
11309.733552923255
> (define totalArea (+ totalArea (area (* radius 0.6))))
> (define totalArea (- totalArea (area (* radius 0.4))))
> (define totalArea (+ totalArea (area (* radius 0.2))))
> totalArea
18849.55592153876
>