Racket Programming Assignment #1: First Interactions

Learning Abstract

This assignment features the Racket programming language, all of the computations occur within the interactions pane of the DrRacket PDE. In the first three parts of this assignment, I mimicked the solutions pages to learn about the basic numeric computations in Lisp, then found the area of the blue tile with a red dot, then finally rendered the image of the tile. The last two parts of the assignment featured an image consisting of 5 concentric circles which I rendered the image and then computed the area of the blue circles.

Interaction: 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
> 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:

The blue and red tile problem: A tile of side 200 is blue, except for a centered red disk of radius one-third of the side of the tile. What is the area of the tile which is blue?

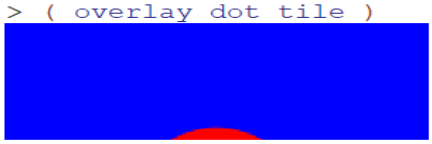
```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( define side-of-tile 200 )
> ( define diameter-of-dot ( / side-of-tile 3 ) )
> ( define radius-of-dot ( / diameter-of-dot 2 ) )
> ( define total-tile-area ( expt side-of-tile 2 ) )
> ( define red-dot-area ( * pi ( expt radius-of-dot 2 ) ) )
> ( define blue-tile-area ( - total-tile-area red-dot-area ) )
> 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: 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 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" ) )
> tile
```

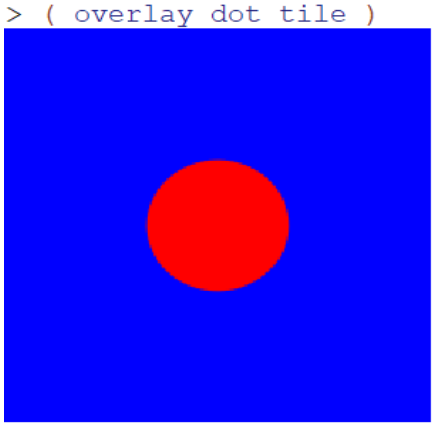
```
> ( define dot ( circle radius-of-dot "solid" "red" ) )
> dot
```

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



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

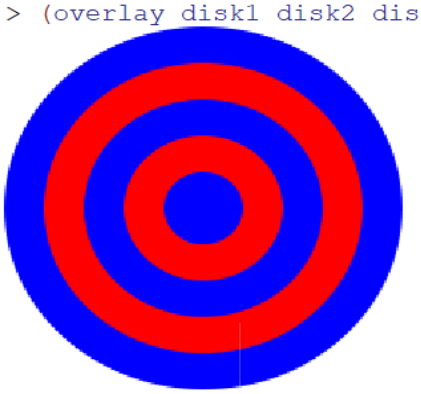
```
> ( overlay dot tile )
```



```
>
```

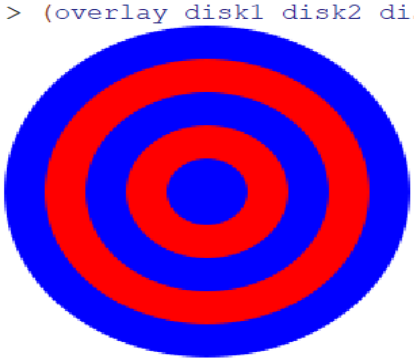
Interaction: Painting the blue and red concentric disk image

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (require 2htdp/image )
> (define radius 20 )
> (define disk1 ( circle radius "solid" "blue" ))
> (define disk2 ( circle (* radius 2 ) "solid" "red" ) )
> (define disk3 ( circle (* radius 3 ) "solid" "blue" ) )
> (define disk4 ( circle (* radius 4 ) "solid" "red" ))
> (define disk5 ( circle (* radius 5 ) "solid" "blue" ))
#<procedure:>>
#<procedure:>>
#<procedure:>>
#<procedure:>>
#<procedure:>>
#<procedure:>>
> (overlay disk1 disk2 disk3 disk4 disk5)
```



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

```
> (define disk2 ( circle (* radius 2 ) "solid" "red" ) )
> (define disk3 ( circle (* radius 3 ) "solid" "blue" ) )
> (define disk4 ( circle (* radius 4 ) "solid" "red" ))
> (define disk5 ( circle (* radius 5 ) "solid" "blue" ))
#<procedure:>>
#<procedure:>>
#<procedure:>>
#<procedure:>>
#<procedure:>>
#<procedure:>>
> (overlay disk1 disk2 disk3 disk4 disk5)
```



```
> ( define disk1-area ( * pi ( expt radius 2 ) ) )
> ( define disk2-area ( * pi ( expt ( * radius 2) 2 ) ) )
> ( define disk3-area ( * pi ( expt ( * radius 3) 2 ) ) )
> ( define disk4-area ( * pi ( expt ( * radius 4) 2 ) ) )
> ( define disk5-area ( * pi ( expt ( * radius 5) 2 ) ) )
> ( + ( - ( + ( - disk5-area disk4-area) disk3-area ) disk2-area )
disk1-area )
18849.55592153876
>
```