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: 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 )
*3: undefined;
 cannot reference an identifier before its definition
> (* 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

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
```

$66\frac{2}{3}$

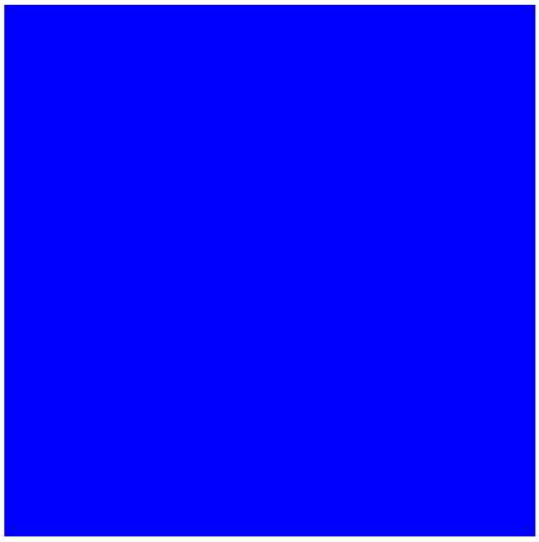
```
> radius-of-dot
```

$33\frac{1}{3}$

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

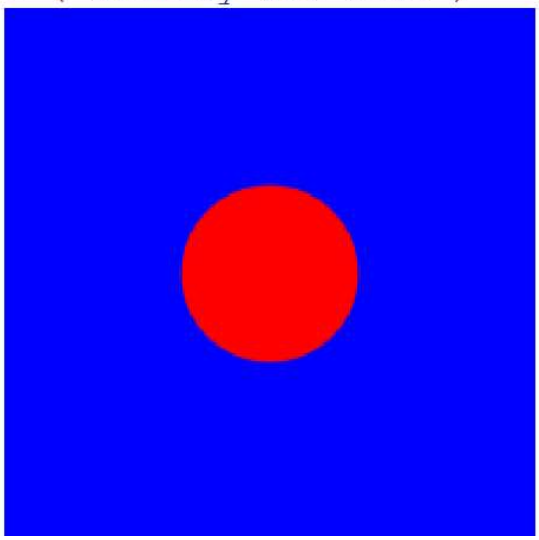
Interaction: Painting the blue and red tile

```
> ( require 2htdp/image)
> ( define tile ( square side-of-tile 'solid 'blue))
> tile
```



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

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

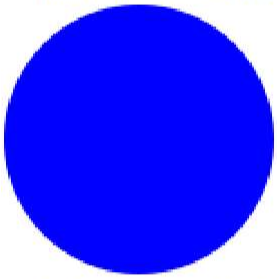


Interaction: Painting the blue and red concentric disks image

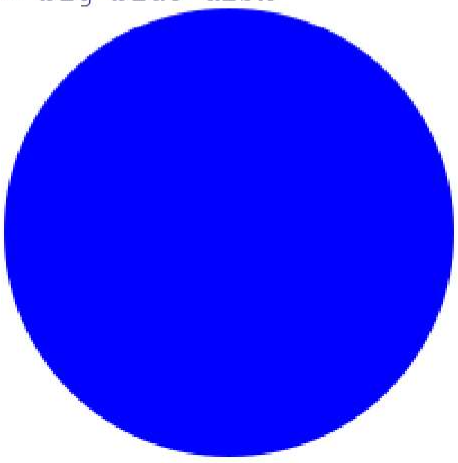
Welcome to DrRacket, version 8.6 [cs].
Language: Determine language from source; memory limit: 128 MB.

```
> (require 2htdp/image)
> (define rad-of-bigblue-disk 100)
> (define rad-of-bigred-disk ( - rad-of-bigblue-disk 20))
> (define rad-of-medblue-disk ( - rad-of-bigred-disk 20))
> (define rad-of-medred-disk ( - rad-of-medblue-disk 20))
> (define rad-of-smallblue-disk ( - rad-of-medred-disk 20))
> (define sm-blue-disk ( circle rad-of-smallblue-disk 'solid 'blue))
> (define med-blue-disk ( circle rad-of-medblue-disk 'solid 'blue))
> (define big-blue-disk ( circle rad-of-bigblue-disk 'solid 'blue))
> sm-blue-disk
```

```
> med-blue-disk
```

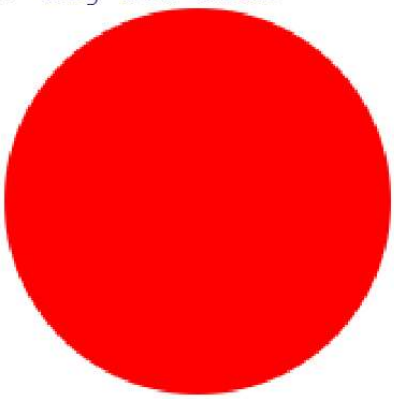


```
> big-blue-disk
```

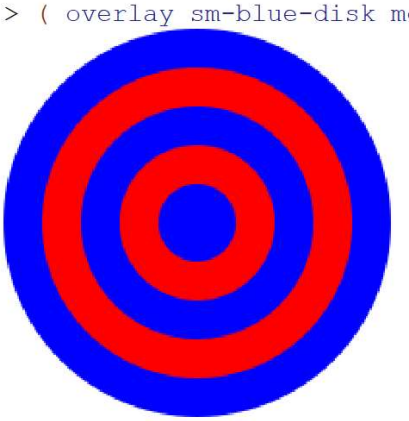


```
> (define med-red-disk ( circle rad-of-medred-disk 'solid 'red))
> (define big-red-disk ( circle rad-of-bigred-disk 'solid 'red))
> med-red-disk
```

```
> big-red-disk
```



```
> ( overlay sm-blue-disk med-red-disk med-blue-disk big-red-disk big-blue-disk )
```



```
>
```

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

---

```
> ( define bigblue-area ( * pi ( expt rad-of-bigblue-disk 2 )))
> ( define medblue-area ( * pi ( expt rad-of-medblue-disk 2 )))
> ( define smblue-area ( * pi ( expt rad-of-smallblue-disk 2 )))
> ( define bigred-area ( * pi ( expt rad-of-bigred-disk 2 )))
> ( define medred-area ( * pi ( expt rad-of-medred-disk 2 )))
> ( define blue-total-area ( + ( + bigblue-area medblue-area) smblue-area))
> ( define red-total-area ( + bigred-area medred-area))
> ( define blue-area ( - blue-total-area red-total-area ))
> blue-area
18849.555921538755
>
```