Racket Programming Assignment #1: First Interactions

Learning Abstract

This programming assignment is, as the name suggests, my first interaction with the Racket Programming Language. In the first three examples below I fiddle with the functions of the Interactions pane in racket to get a handle on the syntax of Racket (a), how to tie values to variables and perform operations with the variables (b), and how to render images using the 2htdp/image library (c). In the fourth interaction I render an image of concentric squares, mimicking a target (d). Finally for the last interaction I calculated the percent of red in the image from the previous interaction (e). Binding values to variables and using those variables in calculations was the predominant theme of this assignment.

Interaction (a): Simple Numeric Processing

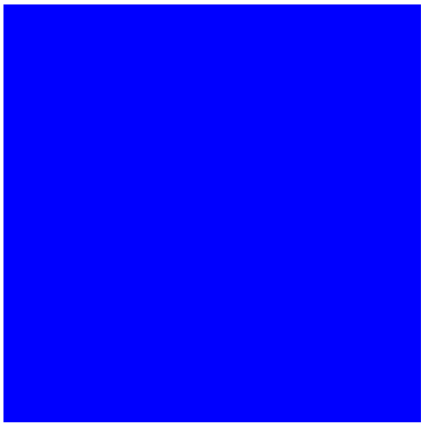
```
Welcome to DrRacket, version 8.3 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> x
x: undefined;
 cannot reference an identifier before its definition
> 55
55
> 55.2
55.2
> pi
3.141592653589793
> ( * 3 8)
24
> (+ (* 3 8) 6)
30
> ( expt 2 8 )
256
> ( * pi ( expt 7 2 ) )
153.93804002589985
> ( expt 9 50)
515377520732011331036461129765621272702107522001
```

Interaction (b): Blue square and red dot computation solution

```
Welcome to DrRacket, version 8.3 [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
>
```

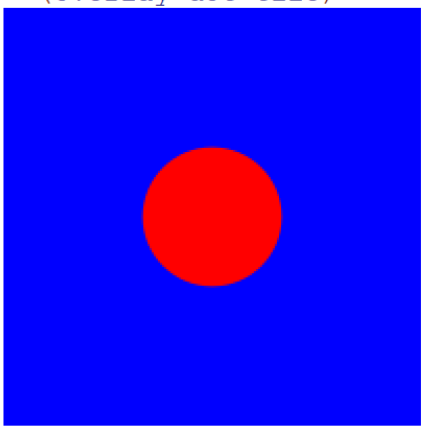
Interaction (c): Painting blue square and red dot

```
Welcome to DrRacket, version 8.3 [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
```

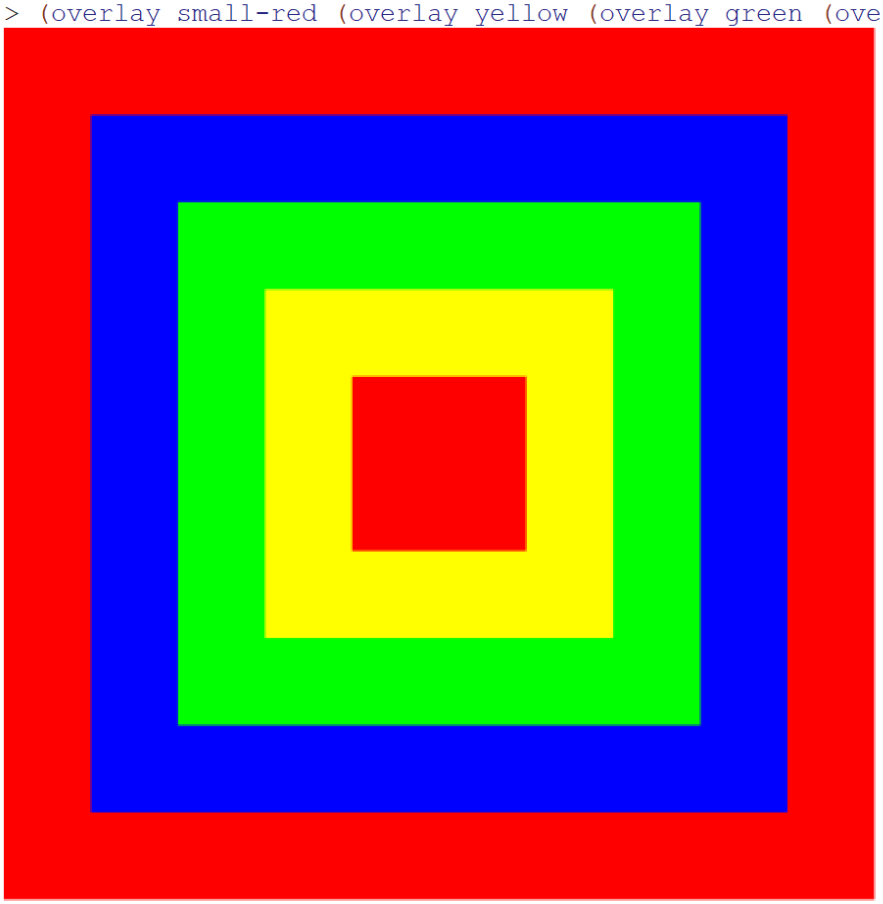
```
> (overlay dot tile)
```



```
>
```

Interaction (d): Painting the concentric square

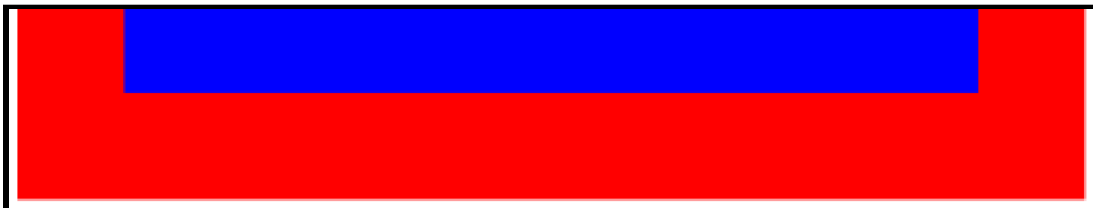
```
Welcome to DrRacket, version 8.3 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (require 2htdp/image)
> (define side-of-small-red 88.88)
> (define side-of-yellow (+ side-of-small-red side-of-small-red))
> (define side-of-green (+ side-of-yellow side-of-small-red))
> (define side-of-blue (+ side-of-green side-of-small-red))
> (define side-of-large-red (+ side-of-blue side-of-small-red))
> (define small-red (square side-of-small-red "solid" "red"))
> (define yellow (square side-of-yellow "solid" "yellow"))
> (define green (square side-of-green "solid" "green"))
> (define blue (square side-of-blue "solid" "blue"))
> (define large-red (square side-of-large-red "solid" "red"))
> (overlay small-red (overlay yellow (overlay green (overlay blue large-red))))
```



```
>
```

Interaction (e): Percent of red in concentric square image solution

Continued from previous interaction.



```
> (define total-area (expt side-of-large-red 2))
> (define blue-area (expt side-of-blue 2))
> (define red-ring-area (- total-area blue-area))
> (define small-red-area (expt side-of-small-red 2))
> (define red-total-area (+ red-ring-area small-red-area))
> (define percent-of-red (* 100 (/ red-total-area total-area)))
> total-area
197491.36
> blue-area
126394.47039999999
> red-ring-area
71096.8896
> small-red-area
7899.654399999999
> red-total-area
78996.544
> percent-of-red
40.0
>
```