
Racket Programming Assignment #1: First Interactions | By: Miguel Cruz

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

```
1 | ;-----
2 | ; Programming Assignment 1: First Interactions
3 |
4 | #lang racket
5 | ;; Part 1 (Simple Numeric Processing)
6 | 55
7 | 55.2
8 | pi
9 | ( * 3 8 )
10 | (+ (* 3 8) 6 )
11 | (expt 2 8)
12 | (* pi (expt 7 2))
13 | (expt 9 50)
```

Welcome to [DrRacket](#), version 8.3 [cs].

Language: **racket**, with debugging; memory limit: **128 MB**.

55

55.2

3.141592653589793

24

30

256

153.93804002589985

515377520732011331036461129765621272702107522001

>

Interaction: Solution to the blue and red tile area problem

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

Answer: The Area of the tile which is blue is 36509.34

```
15 ;; Part 2 (Solution to the blue and red tile area problem)
16 (define side-of-tile 200)
17 (define diameter-of-dot (/ side-of-tile 3))
18 (define radius-of-dot (/ diameter-of-dot 2))
19 (define total-tile-area (expt side-of-tile 2))
20 (define red-dot-area (* pi (expt radius-of-dot 2)))
21 (define blue-tile-area (- total-tile-area red-dot-area))
22
```

Welcome to [DrRacket](#), version 8.3 [cs].

Language: **racket**, with **debugging**; memory limit: **128 MB**.

> 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

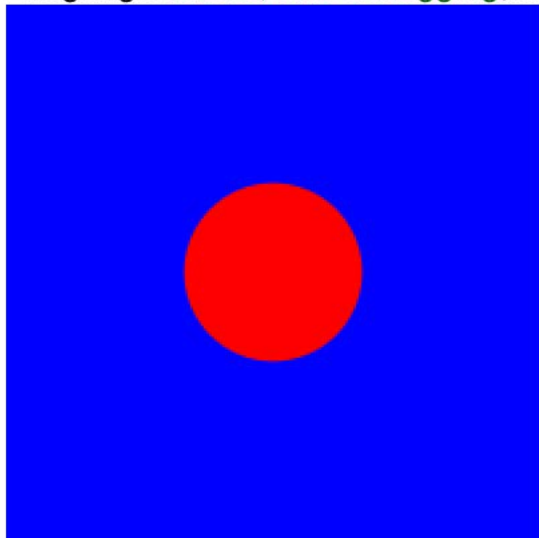
36509.341496011344

>

Interaction: Painting the blue and red tile

```
;; Part 3 Painting the blue and red tile
( 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"))
( define dot (circle radius-of-dot "solid" "red"))
( overlay dot tile )
```

Welcome to [DrRacket](#), version 8.3 [cs].
Language: racket, with debugging; memory limit: 128 MB.



>

Interaction: Painting the concentric squares image

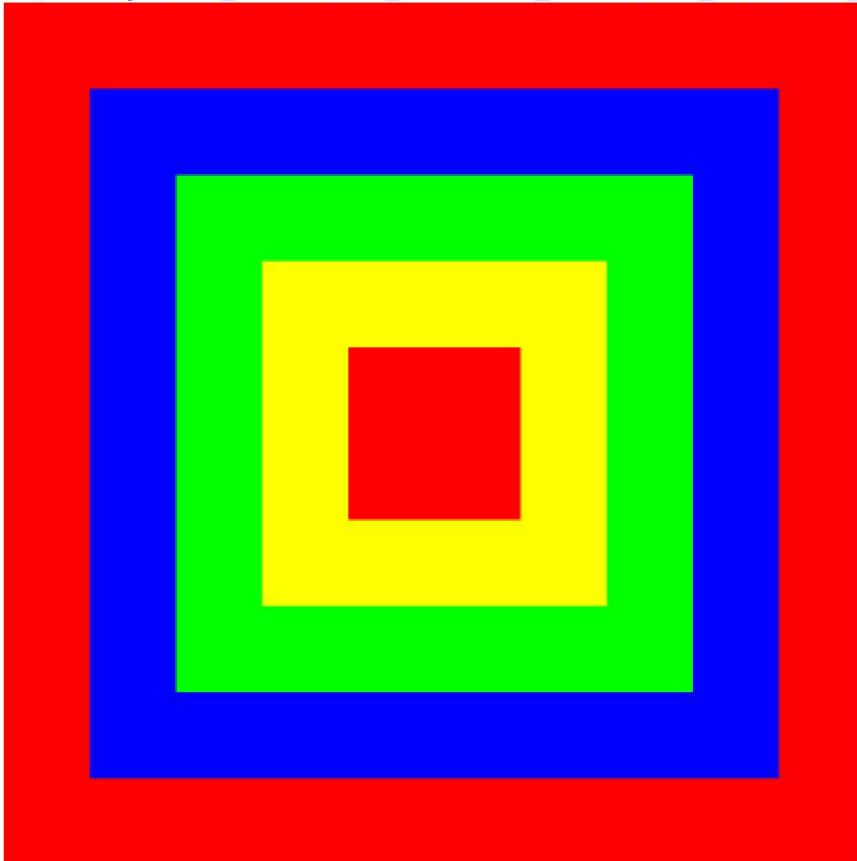
```
;; Part 4 Painting concentric squares image
( require 2htdp/image )
( define side-of-1st-tile 88.5 )
( define side-of-2nd-tile (* side-of-1st-tile 2 ))
( define side-of-3rd-tile (* side-of-1st-tile 3 ))
( define side-of-4th-tile (* side-of-1st-tile 4 ))
( define side-of-5th-tile (* side-of-1st-tile 5 ))

( define first_tile (square side-of-1st-tile "solid" "red"))
( define second_tile (square side-of-2nd-tile "solid" "yellow"))
( define third_tile (square side-of-3rd-tile "solid" "green"))
( define fourth_tile (square side-of-4th-tile "solid" "blue"))
( define fifth_tile (square side-of-5th-tile "solid" "red"))
```

Welcome to [DrRacket](#), version 8.3 [cs].

Language: [racket](#), with [debugging](#); memory limit: 128 MB.

```
> (overlay first_tile second_tile third_tile fourth_tile fifth_tile)
```



```
>
```

Interaction: Computing the percent of the concentric squares image which is red

```
;; Part 5 Computing the percent of the concentric squares image which is red
(define (square_area side)
  (* side side)
)

(define area_of_all (square_area side-of-5th-tile))
(define area_of_fourth_tile (square_area side-of-4th-tile))

(define bigger_red_area (- area_of_all area_of_fourth_tile))
(define smaller_red_area (square_area side-of-1st-tile))
(define total_red_area (+ bigger_red_area smaller_red_area))

(define percentage_of_red (* (/ total_red_area area_of_all) 100))
```

Welcome to [DrRacket](#), version 8.3 [cs].
Language: **racket**, with **debugging**; memory limit: 128 MB.

```
> area_of_all
195806.25
> area_of_fourth_tile
125316.0
> bigger_red_area
70490.25
> smaller_red_area
7832.25
> total_red_area
78322.5
> percentage_of_red
40.0
>
```

Answer: The area of the concentric squares which are red is 40%