

Racket Programming Assignment #1: First Interactions

Learning Abstract

In this assignment, the interaction with images and usages of arithmetic operations are used in the programming IDE called DrRacket. The purpose of this assignment is to help the students understand the functions of the Racket programming language and training to become the better programmers. In the first section, some numbers and some operations were done to help understand how numeric processing works in Racket. In the second section, the solution to find the blue and red tile was done. We use the radius, side of the tile to compute the area of the rest of the blue area in the tile. Thirdly, the image function was imported the blue tile and the red dot was painted. In the fourth section of the assignment, we used the skills we did in task 1,2 and 3 and made concentric circles by overlaying on top of each other with different sizes and colors. Finally, the computational methods were used to find the total area of the blue circles.

Interaction-1: 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-2: Solution to the blue and red tile area problem

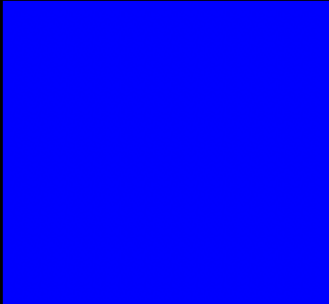
```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( define side-of-tile 200 )
> ( define diameter-of-dot ( / side-of-tile 3 ) )
> ( define radius-of-dot ( / diameter-of-dot 2 ) )
> ( define total-tile-area ( expt side-of-tile 2 ) )
> ( define red-dot-area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


```
1 | #lang racket
2 |
3 |
```

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

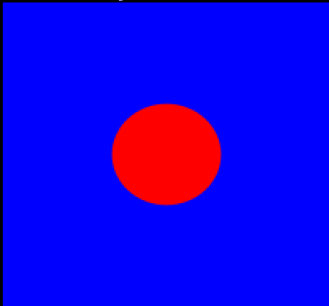
```
> ( require 2htdp/image )
> ( define side-of-tile 200 )
> ( define diameter-of-dot ( / side-of-tile 3 ) )
> ( define radius-of-dot ( / diameter-of-dot 2 ) )
> ( define tile ( square side-of-tile "solid" "blue" ) )
> tile
```



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



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



```
>
```

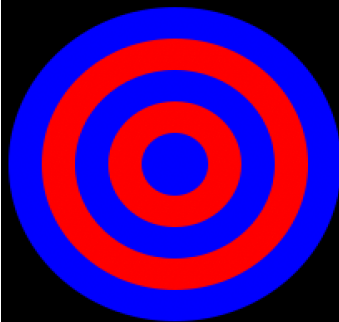
Interaction-4: Painting the blue and red concentric disks images

```
1 #lang racket
2 ;Program to generate concentric circles.
3
4 (require 2http/image)
5 (define diameter-of-the-circle 200)
6
7 (define radius-of-the-circle (/ diameter-of-the-circle 2))
8 (define circles (circle radius-of-the-circle "solid" "blue"))
9
10 (define second-radius-of-the-circle (- radius-of-the-circle 20))
11 (define circles1 (circle second-radius-of-the-circle "solid" "red"))
12
13 (define third-radius-of-the-circle (- radius-of-the-circle 40))
14 (define circles2 (circle third-radius-of-the-circle "solid" "blue"))
15
16 (define fourth-radius-of-the-circle (- radius-of-the-circle 60))
17 (define circles3 (circle fourth-radius-of-the-circle "solid" "red"))
18
19 (define fifth-radius-of-the-circle (- radius-of-the-circle 80))
20 (define circles4 (circle fifth-radius-of-the-circle "solid" "blue"))
21
22 (define circles-area (* pi (expt radius-of-the-circle 2)))
23 (define circles1-area (* pi (expt second-radius-of-the-circle 2)))
24 (define circles2-area (* pi (expt third-radius-of-the-circle 2)))
25 (define circles3-area (* pi (expt fourth-radius-of-the-circle 2)))
26 (define circles4-area (* pi (expt fifth-radius-of-the-circle 2)))
27
28 (define blue-circles-area (- circles-area circles1-area))
29 (define blue-circles2-area (- circles2-area circles3-area))
30
31 (define total-blue-circles-area(+ blue-circles-area blue-circles2-area circles4-area))
```

Welcome to [DrRacket](#), version 8.6 [cs].

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

> (overlay circles4 circles3 circles2 circles1 circles)



>

Interaction-5: Computing the area of the concentric disks image which is blue

```
1 #lang racket
2 ;Program to generate concentric circles.
3
4 (require 2htdp/image)
5 (define diameter-of-the-circle 200)
6
7 (define radius-of-the-circle (/ diameter-of-the-circle 2))
8 (define circles (circle radius-of-the-circle "solid" "blue"))
9
10 (define second-radius-of-the-circle (- radius-of-the-circle 20))
11 (define circles1 (circle second-radius-of-the-circle "solid" "red"))
12
13 (define third-radius-of-the-circle (- radius-of-the-circle 40))
14 (define circles2 (circle third-radius-of-the-circle "solid" "blue"))
15
16 (define fourth-radius-of-the-circle (- radius-of-the-circle 60))
17 (define circles3 (circle fourth-radius-of-the-circle "solid" "red"))
18
19 (define fifth-radius-of-the-circle (- radius-of-the-circle 80))
20 (define circles4 (circle fifth-radius-of-the-circle "solid" "blue"))
21
22 (define circles-area (* pi (expt radius-of-the-circle 2)))
23 (define circles1-area (* pi (expt second-radius-of-the-circle 2)))
24 (define circles2-area (* pi (expt third-radius-of-the-circle 2)))
25 (define circles3-area (* pi (expt fourth-radius-of-the-circle 2)))
26 (define circles4-area (* pi (expt fifth-radius-of-the-circle 2)))
27
28 (define blue-circles-area (- circles-area circles1-area))
29 (define blue-circles2-area (- circles2-area circles3-area))
30
31 (define total-blue-circles-area(+ blue-circles-area blue-circles2-area circles4-area))
```

Welcome to [DrRacket](#), version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (overlay circles4 circles3 circles2 circles1 circles)



> blue-circles-area
11309.733552923255
> blue-circles2-area
6283.185307179586
> circles4-area
1256.6370614359173
> total-blue-circles-area
18849.55592153876
>