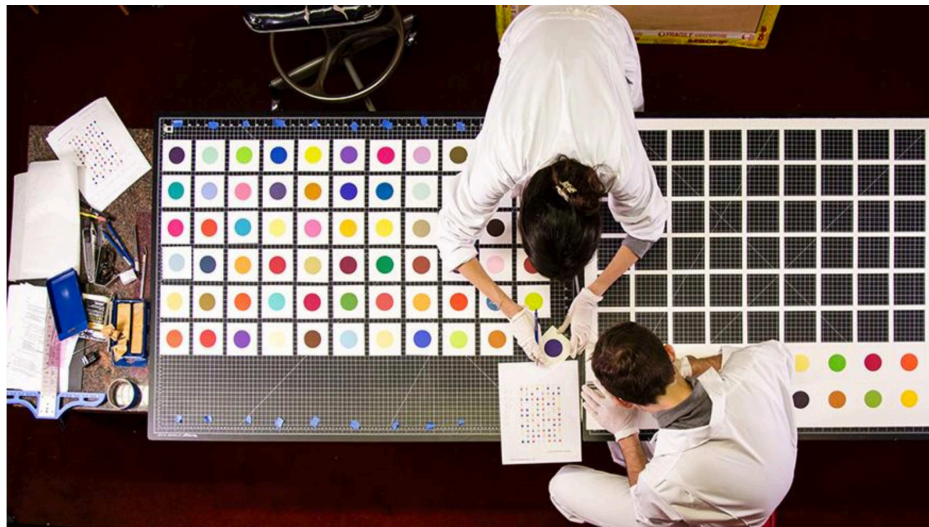

Second Racket Programming Assignment Specification



Working within the DrRacket PDE, do each of the programming tasks. Then please do the document compilation/posting task.

Task 1 - Permutations of Randomly Colored Stacked Dots

Programming constraint: For this part of your assignment, you are not permitted to use any form of repetition (recursion/iteration) or any form of conditional statement (e.g., if, cond).

1. Please study the “Colorful Little Checkerboard” program, and the way it was developed, in Lesson 2. This program is intended to serve as a resource for you to consult as you engage in the programming of the “tiles” decorated with randomly colored stacks of dots featured here.
2. Write a program called `tile` which takes four parameters, each presumed to represent a color, which creates an image representing a square tile of side 100 with background defined by the first color, on which are concentrically piled a disk of diameter 90 of the second color, a disk of diameter 60 of the third color, and a disk of diameter 30 of the fourth color. If the words are overwhelming, just look at the examples presented in the accompanying demo.
3. Write a program called `dots-permutations` taking three parameters, each presumed to represent a color, which creates a row of tiles representing the permutations of three colors, where each permutation is rendered as a stack of dots of diameters 90, 60, and 30. Please look to the accompanying demo for clarification.
4. Please generate a demo that is just like the demo provided, **except that the colors are different throughout the demo**. For a list of available colors, look here:
https://docs.racket-lang.org/draw/color-database___.html

The Demo

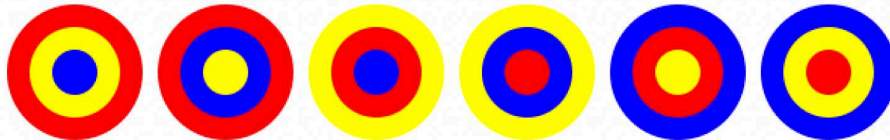
Welcome to [DrRacket](#), version 8.1 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (tile "black" "cornflowerblue" "cyan" "teal")



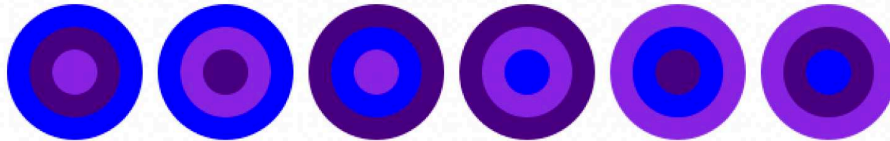
> (tile "silver" "blue" "plum" "magenta")



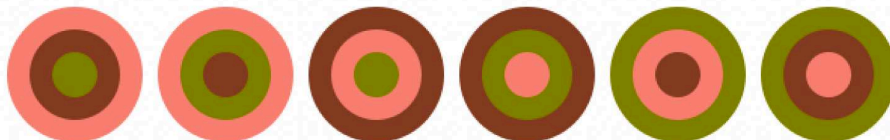
> (dots-permutations "red" "yellow" "blue")



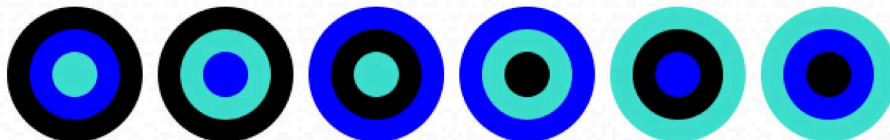
> (dots-permutations "blue" "indigo" "blueviolet")



> (dots-permutations "salmon" "brown" "olive")



> (dots-permutations "black" "blue" "turquoise")



>

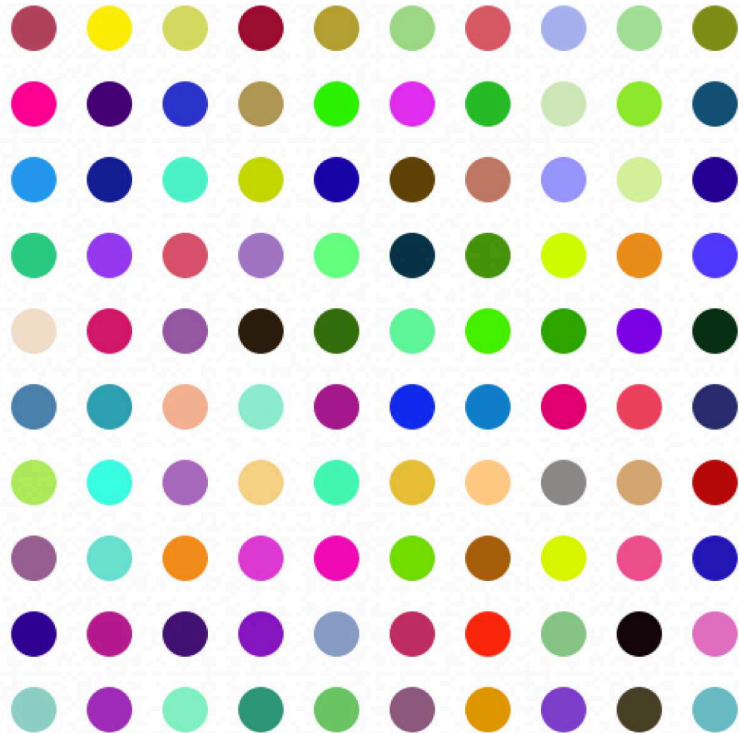
Task 3 - Hirst Dots

Programming constraint: For this part of your assignment, you are not permitted to use any form of iterative construct. Rather, you are required to use recursion.

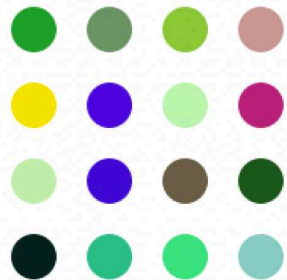
1. Please study the “Squares of Squares” portion of Lesson 2. The programs featured in that section of the lesson are intended to serve as a resource for you to consult as you engage in this programming activity. Furthermore, you might find it beneficial to keep the first problem of this assignment in mind as you think about how to accomplish what you need to do for this problem..
2. Write a function called `hirst-dots` to draw square arrangements of Hirst dots that are consistent with the accompanying demo. Please arrange for the diameter of the dots to be 30 pixels, and for each dot to be 20 pixels from its nearest dots. Please note that the parameter is presumed to be a non-negative integer, and that the number of dots in an image is the square of the value of the parameter.
3. Please generate a demo that displays a 10x10 grid of Hirst dots, and a 4x4 grid of Hirst dots.

The Demo

Welcome to [DrRacket](#), version 8.1 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (hirst-dots 10)



> (hirst-dots 4)



>

Task 4 - Stella Thing

Programming constraint: For this part of your assignment, you are not permitted to use any form of iterative construct. Rather, you are required to use recursion.

1. Please study the “Channeling Frank Stella” portion of Lesson 2. The programs featured in that section of the lesson are intended to serve as a resource for you to consult as you engage in the programming for this problem.
2. Write a function called `stella` to display graphical images in the spirit of Frank Stella subject to the following constraints:
 - (a) Your program must be based on a shape other than either of those that I used in the Stella section Lesson 4. Thus, you must not use either a square or a star for it. Perhaps you would like to use a circle, or an ellipse, or a wedge, or a triangle, or rhombus, or a regular polygon, or a star-polygon. Maybe something else! Please find your way to the documentation for the `2htdp/image` library and find some functionality with which to do the deed:
<https://docs.racket-lang.org/teachpack/2htdpimage.html>
 - (b) Your program must be consistent with one of the two varieties of Stella that were presented in Lesson 4. That is, speaking in the terminology dropped in the lesson, you must do a monochromatic shape, or a two tone shape.
3. Please generate a sound demo for your program, one that displays at least two different images in the family of images that your program produces.

Task 5 - Creation

Programming constraint: For this part of your assignment, you are not permitted to use any form of of iterative construct. Rather, you are required to use recursion for any repetition that you would like to accomplish.

1. Please define a function called `my-creation` to produce an interesting image using some of the more involved functionality in the `2htdp/image` library. Perhaps use some of the more elaborate `overlay` commands (maybe `overlay/offset` or `overlay/xy`), or maybe some of the rotating, scaling, flipping, cropping, and framing functionality. Again, here is the place to go to find some useful functionality:
<https://docs.racket-lang.org/teachpack/2htdpimage.html>
2. Please generate a sound demo for your program, one that displays your image.

Task 6 - Document Compilation/Posting

Craft a nicely structured document that contains:

1. A nice title, indicating that this is your second Racket assignment.
2. A nice learning abstract, which artfully says something about the `2htdp/image` library, the generation of visual permutations on a set of three discs, recursive programming, number sequences, renderings that channel a couple of famous modern artists, and a visual creation of your own.
3. A section that provides the code and a demo for Task 1, the task involving permutations of colored disks.
4. A section that provides the code and a demo for Task 2, the task involving number sequences.
5. A section that provides the code and a demo for Task 3, the task featuring Hirst dots.

6. A section that provides the code and a demo for Task 4, the task featuring a Stella design.
7. A section that provides the code and a demo for Task 5, the task featuring your own visual creation.

Post your document to you web work site.

Due Date

Please complete your work on this assignment, and post your work to your web work site, by the end of the day on Wednesday, September 15, 2021.