Project 2

Due date: Mar 10, 2020, 11:59pm

Objective

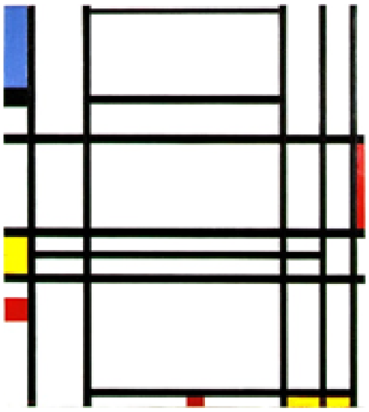
- Design and implement a recursive algorithm.
- Apply divide-and-conquer.

Project Overview[1]

Piet Mondrian (March 7, 1872 – February 1, 1944) was a Dutch painter who created numerous famous paintings in the early half of the previous century that consisted of a white background, prominent black horizontal and vertical lines, and regions colored with red, yellow and blue. Three examples are shown below:

Tableau I, 1921

Composition II in Red, Blue, and Yellow, 1930



Composition No. 10, 1939-1942

You will implement a Java program that uses recursion to generate pseudo-random art in a Mondrian style.

The Java project available on the course website contains a few code snippets that may be helpful to complete this project.

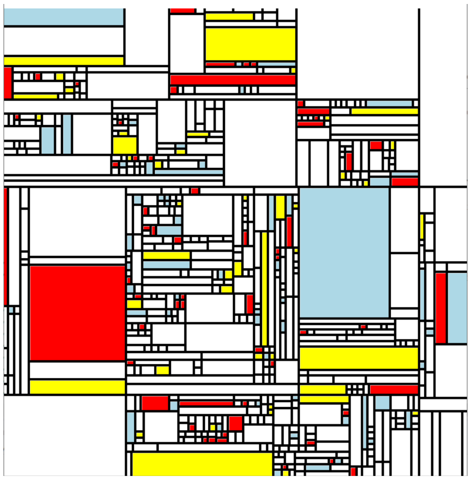
Minimum Requirements

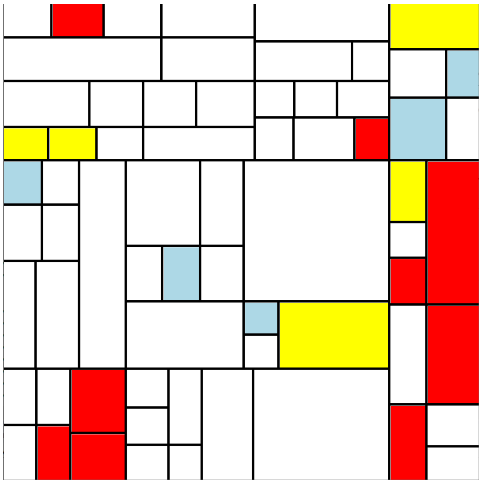
Implement a JavaFX application that uses a divide-and-conquer strategy to generate recursively pseudo-random art in a Mondrian style. Using a dialog, the application prompts the user first to enter a minimum region length. The program should then display a window that contains regions, each with a minimum height and width as entered by the user.

[1] This project has been derived from a project made available by Ben Stephenson, University of Calgary.

The following general strategy can be used to divide the problem into smaller subproblems: Randomly decide whether a region should be split horizontally and/or vertically into smaller regions, or should not be split at all. The location of a split should be chosen randomly. The final regions should be filled with a random color. Then generate Mondrian art for the resulting smaller regions.

Two images generated using an algorithm that meets the minimum requirements are shown below:





Expansions

Expand this algorithm to generate art with your own specific style provided that it is still at least vaguely Mondrian in style (meaning that it largely consists of horizontal and vertical lines and colored regions, at least the majority of which are rectangular). Ideas for customizing your work that you might want to consider include:

- Using lines of variable width
- Selecting a color that is influenced by the colors of neighboring regions.
- Changing the distribution of random region sizes.
- Using a patterned fill for some regions instead of only using solid fills.
- Occasionally splitting a region into something other than rectangles.
- Occasionally split a region into 3 smaller regions instead of 2 or 4.

Include a simple textfile in your submission, called readme.txt that outlines the expansions of the basic algorithm.

Submission

Rename the Eclipse project to `CS3151Project2`*`FirstNameLastName`* where *`FirstNameLastName`* is your first and last name. Include the readme file in your project folder. Submit a zipped solution to the course website before the due date on Mar 10, 11:59pm.

Grading

A project will receive 0 points if one or more of the following holds

- the code contains syntax errors
- the program crashes
- the program does not use recursion to generate the art.

A (≥ 90 pts.) Meets all requirements for B and applies at least two non-trivial expansions.

B (≥ 80 pts.) Meets all minimum requirements; in particular, uses recursion to create Mondrian art, uses randomization to set the sizes of the regions and the regions' colors. The regions are filled without overlapping the lines that separate the regions.

C (≥ 70 pts.) Generates Mondrian art using recursion; some calculation for the drawings may be inaccurate; some parts may not be randomized.

Three points will be deducted if your project does not follow the naming convention. To rename an Eclipse project, right-click the project name in the Eclipse Project Explorer and select the option `Refactor > Rename…`