A04 - Recursion

30 Possible Points

9/30/2023

Attempt 1

In Progress
NEXT UP: Submit Assignment

Add Comment

Unlimited Attempts Allowed
9/16/2023 to 10/7/2023

Details

Fundamentals

Assignment: A04

Learning Objectives

- Modify a recursive method
- Use core functionality from class StdDraw
- Implement a fractal

Overview

This is an individual assignment where you work on your own.
You will learn about fractals and deepen your understanding of recursion as you modify and implement recursive methods.

Fractals

What are fractals?

Previous (https://slcc.instructure.com/courses/915963/modules/items/21581574)

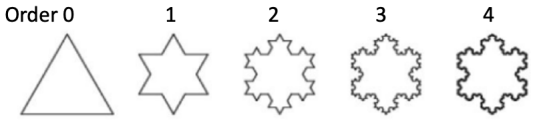
Submit Assignment

Next (https://slcc.instructure.com/courses/915963/modules/items/21581578)

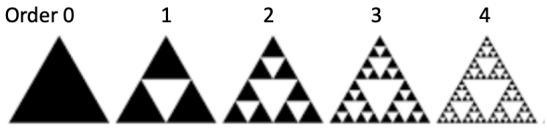
- Fractals are defined by a parameter called **order**, that indicates the complexity.

Examples of fractals:

- Koch snowflake:



- Sierpinski triangle:



- Fractal Tree

An order-n tree is a line with 2 smaller order-(n-1) trees starting at the end of the line. The smaller trees have a different **position**, different **size**, different **orientation**, and different **order**.

Watch a short video that shows a **Fractal tree generator** (https://www.youtube.com/watch?v=DTTZDMnyFco)

Instruction

Challenge A:

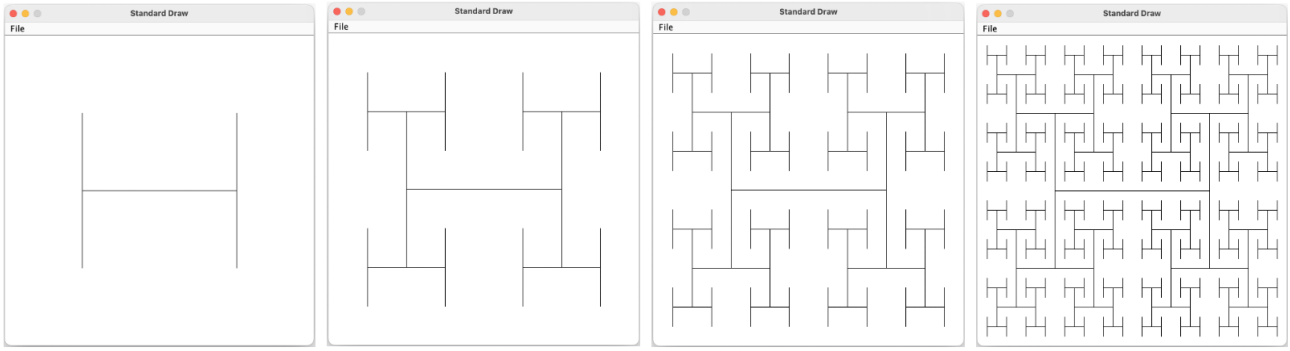
In this challenge, you will **modify a recursive method**.

- Create a package called **a04** and add the public class **Htree.**
- Copy the following code into Htree.java and import classes from algs4.jar as needed to make it compile. https://introcs.cs.princeton.edu/java/23recursion/Htree.java.html
- Modify the first line in the main method. Rather than reading in a number from the command line argument, assign the hard-coded value **1** to n, and run the program. You should see the upper-case letter H.
- Run the program three more times, assigning the values 2, 3, and 4 to n.
  Note, how each increase in n adds complexity by adding additional, smaller letter H-s at the four ends of the vertical lines.

Previous (https://slcc.instructure.com/courses/915963/modules/items/21581574)

Submit Assignment

Next (https://slcc.instructure.com/courses/915963/modules/items/21581578)



- Copy Htree.java and save it in **a04** under the name **Chairs.java**
  - Modify the code to make the output look like chairs instead of H-s.
  - Update method names and comments as needed.
  - Add a loop to the main method in class Chairs to run the fractal three times with n = 1, n = 2, and n = 3. Display each fractal two seconds before showing the next.
    Hint: The class StdDraw has a method called **pause** that pauses the execution for the specified number of milliseconds.

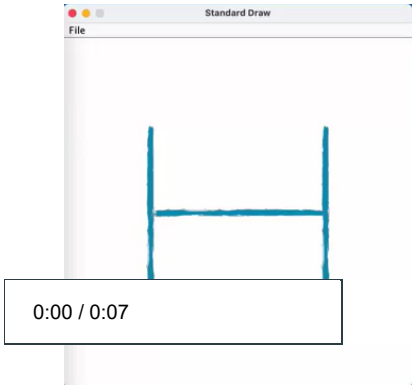
Expected output:

- Copy Htree.java and save it in **a04** under the name **HtreeBrushStroke.java**
  - Create a folder called **Resources** inside a04. Then download the following image files **BrushStrokeHorizontal.png** (https://slcc.instructure.com/courses/915963/files/149887658/preview) and **BrushStrokeVertical.png** (https://slcc.instructure.com/courses/915963/files/149887657/preview) and copy them into that folder:
  - Replace the three lines that form the H with images of brushstrokes.
    Here are two hints that might help you: ***hint1*** and ***hint2***
  - Update method names and comments as needed.
  - Add a loop to the main method in class HtreeBrushStroke to run the fractal three times with n = 1, n = 2, and n = 3.
    Display each fractal two seconds before showing the next.

Previous (https://slcc.instructure.com/courses/915963/modules/items/21581574)

Submit Assignment

Next (https://slcc.instructure.com/courses/915963/modules/items/21581578)

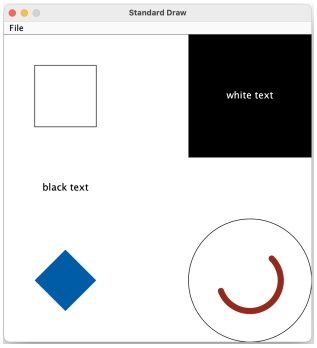


Challenge B:

Familiarize yourself with StdDraw.

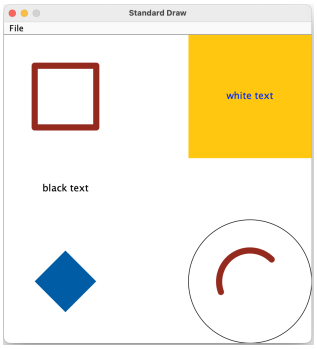
Notice that Htree drew the lines using the static method line of class StdDraw.
StdDraw has many methods that allow you to draw lines, outlines of shapes, filled shapes, and much more.

- Add the class **StdDraw** to package a04.
- Copy the following code into StdDraw.java and change the package name to a04.
  https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/StdDraw.java
  (https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/StdDraw.java)
- (https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/StdDraw.java) Sedgewick's team added a main method at the bottom of the class that introduces some of the key features of StdDraw.
  Run the code. You should see the following GUI:



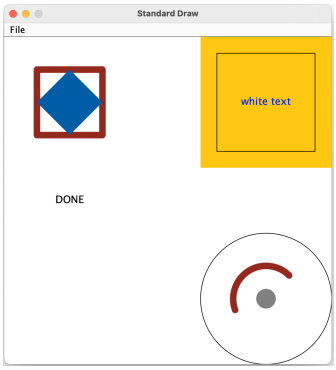
- Make the following changes to the main method to familiarize yourself with StdDraw.
  1. Change the outline of the square on the top left to a thicker red line
  2. Change the color of the square on the top right to orange, and the color of "white text" to blue.
  3. Chang the red arc at the bottom right to show (only) the remaining circle that has not been shown before.

  At this point, the GUI should look like this:



- Continue the modifications of the main as described below:
  1. Add a thin black outline of a square on top of the yellow square. It should have the same center as the yellow square but shorter sides as shown in the GUI below. It should be big enough to comfortably fit the text.
  2. Add a small gray disc in the center of the circle at the bottom right.
  3. Move the blue diamond in the center of the red outline of a square (top left).
  4. Change the black text read "DONE."

  At this point, the GUI should look like this:

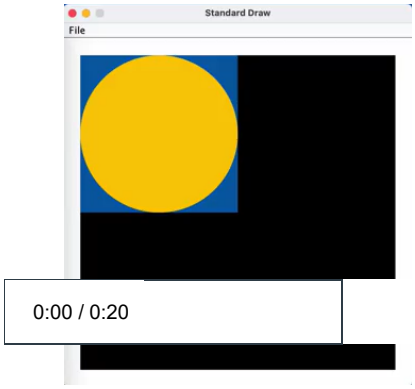


Challenge C:

In this challenge, you will **write a recursive method that draws a fractal**.

- Add the class **MyFractal** to package a04.
  This is your chance to be creative and have fun.
- Create your own fractal based on one or more filled shapes (not outlines)
  As the order (n) increases, repeated instances of the same shapes are added, and the complexity increases. The added shapes are smaller and in different locations.
  The following video shows an example of a fractal that fulfills the basic requirements.
  For those of you who like a challenge, consider including an image or changing the orientation or color of the added shapes.
- Draw your fractal 6 times starting with n = 1 and increasing n until n = 6.
  Display each fractal three (3) seconds before showing the next.
  Hint: The class StdDraw has a method pause that pauses the execution to the specified number of milliseconds

Previous (https://slcc.instructure.com/courses/915963/modules/items/21581574)
Submit Assignment
Next (https://slcc.instructure.com/courses/915963/modules/items/21581578)



Submission

(https://community.canvaslms.com/t5/Student-Guide/How-do-I-embed-images-from-Canvas-into-the-Rich-Content-Editor/ta-p/356)

- Create a video that is 2 - 4 minutes long.
  Follow the **Guidelines for Assignment/CE Recordings** (https://slcc.instructure.com/courses/915963/pages/guidelines-and-expectations)
- After the title page and a brief introduction do the following:
  - Show the code of class Chairs
    Run the main method that displays the chair of order 1, 2, and 3.
  - Show the code of class HtreeBrushStroke
    Run the main method that displays the H of order 1, 2, and 3.
  - Show the code of the main method of class StdDraw after implementing all specified modifications.
    Run it.
  - Show the code of the class MyFractal.
    Run the main method that displays your fractal of orders 1 to 6.

- Embed the video **and** submit the file MyFractal.java.

View Rubric

Previous (https://slcc.instructure.com/courses/915963/modules/items/21581574)

Submit Assignment

Next (https://slcc.instructure.com/courses/915963/modules/items/21581578)

2420 - A04

Criteria	Ratings				Pts
Text Upload Office 365 More					
Style | Best Practices | Video view longer description	**3 pts** **Well done.**	**2 pts** **Close**	**1 pts** **Needs Work**	**0 pts** **No Marks**	/ 3 pts
Chairs.java view longer description	**5 pts** **Well done.**	**4 pts** **Close**	**3 pts** **Getting There**	**0 pts** **No Marks**	/ 5 pts
HtreeBrushStroke.java view longer description	**5 pts** **Well done!**	**4 pts** **Close**	**3 pts** **Getting There**	**0 pts** **No Marks**	/ 5 pts
StdDraw.java view longer description	**5 pts** **Well done!**	**4 pts** **Close**	**3 pts** **Getting There**	**2 pts** **Needs Work**	**0 pts** **No Marks** / 5 pts
MyFractal view longer description	**12 pts** **Full Marks**		**0 pts** **No Marks**		/ 12 pts

Total Points: 0

Previous (https://slcc.instructure.com/courses/915963/modules/items/21581574)

Submit Assignment

Next (https://slcc.instructure.com/courses/915963/modules/items/21581578)