Lab 9: Trees & Fractals

Instructions:

This worksheet serves as a guide and set of instructions to complete the lab.

- You must use the starter file, found here, to get credit for the lab.
- Additionally, here is the workbook that serves as a walkthrough for the lab.
- All material was sourced from the CS10 version of The Beauty and Joy of Computing course.

Submitting:

You will need to fill in the blocks under the starter file and submit to Gradescope.

- To receive full credit, you will need to complete the required blocks, and the required blocks must pass all tests from the autograder in Gradescope.
- For instructions on how to submit to labs to Gradescope, please see this document.

Please note, you must use the starter file, and you must NOT edit the name of any of the required blocks. Failing to do either for these will result in the autograder failing.

The name of the file may be incorrect. Please use the links provided.

Objectives:

Today you will continue to build your knowledge of recursion and understand how we can use recursion to move across data structures (trees, linked lists). We will start by navigating through visual fractals. By the end of lab you will:

- Explore the idea of representing a problem in terms of itself.
- Practice planning and coding recursive blocks.
- Create your own fractals

Required Blocks:

** It is highly recommended that you work with a partner for this assignment.

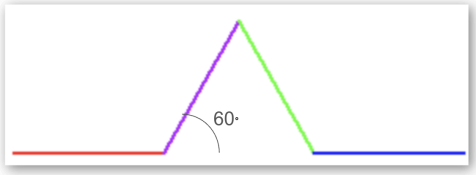
- DO NOT CHANGE THE COLOR OF YOUR PEN
- Block 1: snowflake segment levels: (level) size: (size)
    - You can use only recursion. No HOFs or iteration allowed.
- Block 2: complete snowflake segment levels: (level) size: (size)
    - You must use Block 1 in your solution
    - Recursion is not used for this question
- Block 3: c-curve segment levels: (level) size: (size)
    - You can use only recursion. No HOFs or iteration allowed.

Important Topics mentioned in the Workbook:

For better understanding of the lab we highly recommend going through these workbook pages! These topics are best reviewed in order and as you complete the lab.

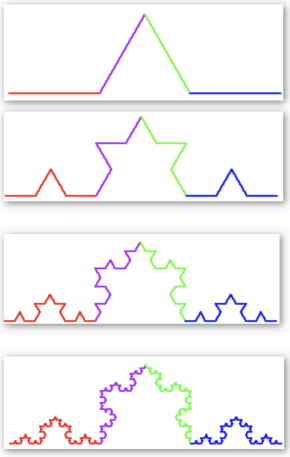
- Recursion
- Recursive Tree Part 2
- Recursive Tree Part 3
- Base Case and Recursive Case
- Input Sliders!
- Self Test: What Was Changed?
- Vary your tree
- Random Tree
- Snowflake
- C-Curve

Block 1: snowflake segment levels: (level) size: (size)

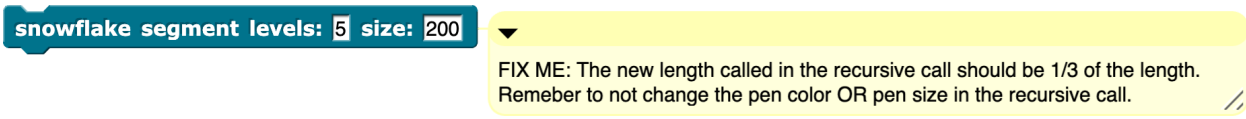
- Objective:
  - Build a function that can create one side of a koch's snowflake
  - 
- Note
  - You must recursion, no HOFs or iteration
  - Note: Each recursive call is one-third the size of the caller
  - **Tip for fractals: The sprite must end in the correct direction (angle!!) at the end**
  - 

- Inputs:
  - Level = any number

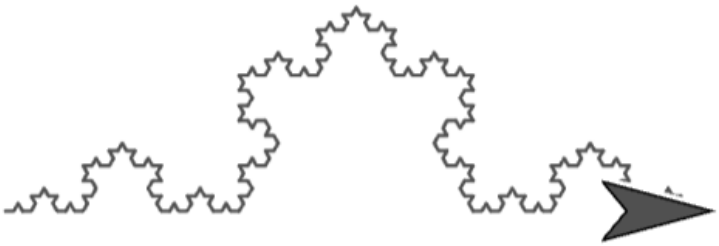
        - Should be a positive number that indicates the level (2-5 shown below)



    - Size = any number
        - Should be any positive number indicates the length of an entire snowflake segment
- Output:
    - None! It is a procedure / non-pure function, so it only displays a visual effect on the stage and does not output any values
        - We are working on a command block!
- Examples:
    -



    -

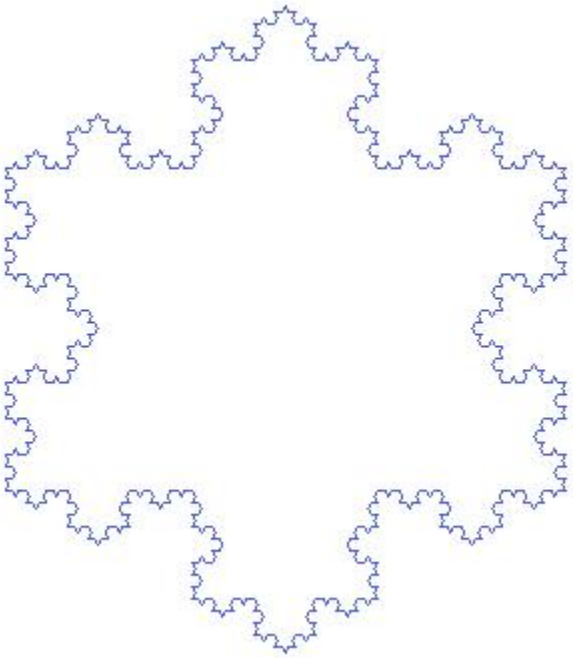


    -

Block 2: complete snowflake segment levels: (level) size: (size)

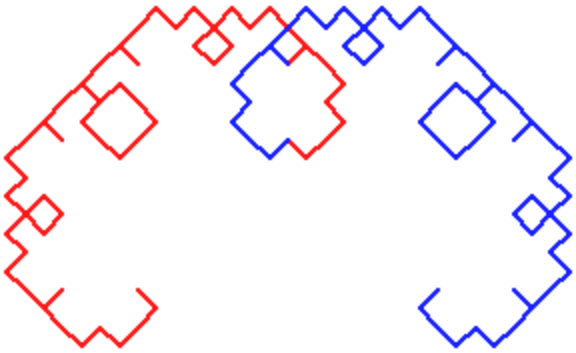
- Objective:
    - Build a function that will create a koch snowflake. The entire snowflake consists of three copies of a fractal, arranged in a triangular shape. Then at every successive level, it repeats the previous level 4 times.
- Notes:
    - You must use the previous block, snowflake segment levels: (level) size: (size), in your solution

    - Recursion is **not** used for this question
- Inputs:
    - Level = any number
        - Should be a positive number that indicates the level
    - Size = any number
        - Should be any positive number indicates the length of an entire snowflake segment
- Output:
    - None! It is a procedure / non-pure function, so it only displays a visual effect on the stage and does not output any values
- Examples:
    -

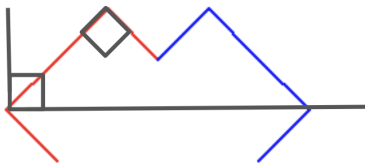


Block 3: c-curve levels: (level) size: (size)

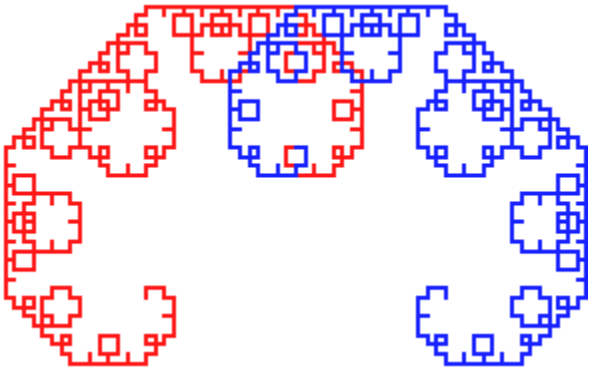
- Objective:
    - Build a function that will create a fractal called a c-curve. The fractal should recurse a c-curve shape to create a complex fractal. Please look at this workbook page for more instructions.
    -



- Notes:
  - You must use recursion. No HOFs.



- Inputs:
  - Level = any number
    - Should be a positive number that indicates the level
  - Size = any number
    - Should be any positive number indicates the total length of the fractal
- Output:
  - None! It is a procedure / non-pure function, so it only displays a visual effect on the stage and does not output any values
- Examples:



- **THE COLOR IS TO SHOW THE DIFFERENT PARTS OF THE C-CURVE.

You can always check the validity of your solutions by using the local autograder. Remember to submit on Gradescope!