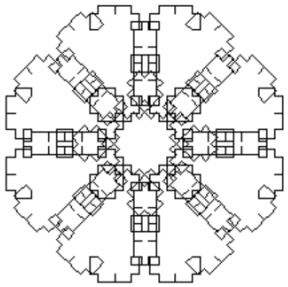
Computational Problem Solving CSCI-603
Islands Lab 3

6/3/2021



In your programming assignment you will be developing a graphical application that draws patterns in the turtle window using recursive techniques.

1 Implementation

You will develop a program called `island.py` which draws fractal islands in a turtle's canvas using recursion. A fractal island is similar to an equilateral, equiangular polygon in that each side of the polygon is identical. Each side of the island is drawn in counter-clock wise manner.

1.1 Polygon Side Patterns

Your program must be able to draw fractal islands using two different fractal curves.

1.1.1 Fractal curve 1

This fractal curve is a slightly modification of the pattern from the problem-solving session. Modify the `draw_side` function from the problem-solving session to generate the following fractal curve. This function must receive as parameters the side's length `n` and the level `l`. It should draw one side of the island and returns its total length.

- If the level is 1, the function should draw a straight line of length `n`.
- If the level is greater than 1, the second segment is drawn using two recursive calls to `draw_side` with the level one less. Assume the angle is a fixed value of $60°(\pi/3$ radians).

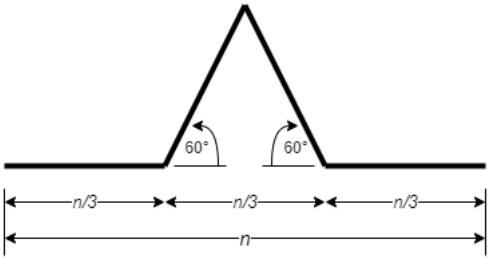


Figure 1. Specification for fractal curve 1 at level 2.

Table 1 shows such a fractal curve with levels from 2 to 5.

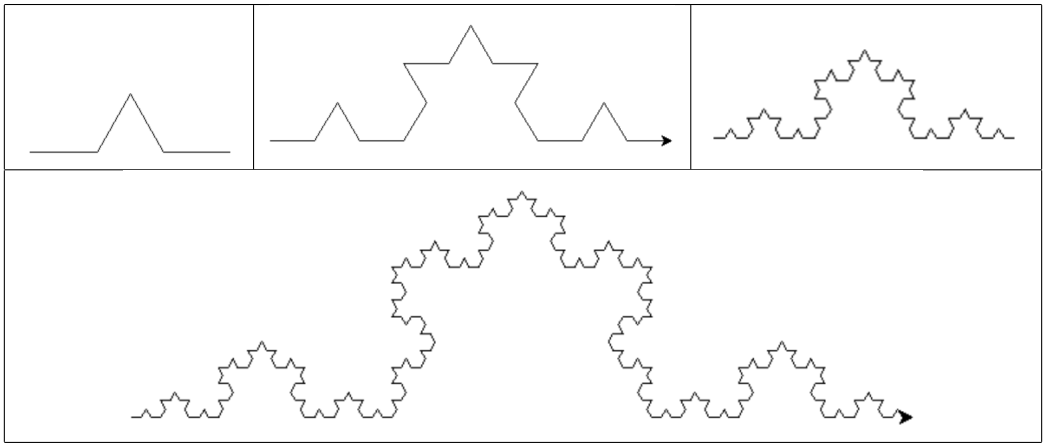


Table 1: Example Fractal Curve 1 Figures for `level` values 2 to 5.

1.1.2 Fractal curve 2

Create another function that receives as parameters the side's length `n` and level `l`. It should draw one side of the island and returns its total length.

- If the level is 1, the function should draw a straight line of length `n`.
- If the level is 2, the straight line gets replaced by the two sides of an isosceles triangle.

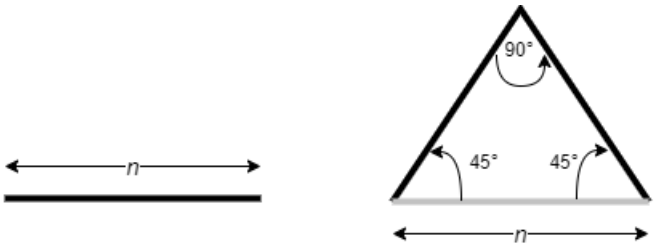


Figure 2. Specification for fractal curve 2 at level 1 and 2.

- If the level is 3, the two lines are now the hypotenuse of their own triangle, and are replaced by their two sides of the triangle. And so on.

Table 2 shows such a fractal curve with levels from 2 to 5, and level 8.

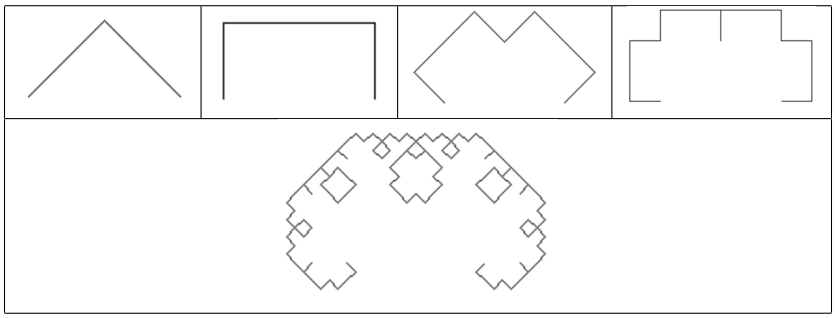


Table 2: Example Fractal Curve 2 Figures for `level` values 2-5 and 8

1.2 Specification

Your program will have a `main` function that does the following:

```
Prompt the user for the number of sides the island will have (positive integer).
If the number of sides is not a valid integer value, produce an error message
and ask again.

Prompt the user for the length of one side (positive float) in pixels
(technically, its level-1 equivalent length). If the length is not a
valid float value, produce an error message and ask again.

Prompt the user for number of levels (positive integer). If the
level is not a valid integer value, produce an error message
and ask again.

Set up the turtle window.

Call the recursive function that draws the island using curve 1 for each side
and print the island's total length.

Pause the window and wait for the user to hit the enter key.

Reset the turtle window.

Call the recursive function that draws the island using curve 2 for each side
and print the island's total length.
```

1.3 Input Validation

All user input must be error-checked, with error messages of this form displayed when needed.

`Value must be a` *type*. `You entered` '*string-value*'.

The user is given an unlimited number of chances to enter a legal value. The program only need check for correct type, not correct range of value, e.g. non-negative.

To do the error-checking, look at the `fullmatch` function in the `re` package. Also study the form of *regular expression patterns*. For example, `r"[0-9]+"` is a pattern matched by all strings only containing decimal digits. (As an alternative, and if you have experience with exceptions, you can handle the exception thrown when numeric conversion fails.)

1.4 Contraints

- Both curve functions must draw a fractal curve with the given length and level, and return the total length of all the lines drawn. The resulting drawing and total length may be different. That's ok.
- When your program draws the figure, it may be off center. This is valid.
- You are not allowed to use global variables to compute the island's total length.

1.5 Useful tips

- Use the `input` built-in function to pause the program between displaying the island using curve 1 and 2.
- A variety of capabilities from the turtle library will help produce the drawing.
  - `reset()`: Reset the window by clearing everything and setting the default state. Use this in between the two drawings.
  - Turtle can very slow to animate, but it is useful when you are developing and debugging. To turn off the animation, use the function `tracer()` before starting to draw. After drawing, use the function `update()` to refresh the turtle's window and display the drawing.

    ```
    turtle.tracer(0, 0)
    # call the drawing recursive function here
    turtle.update()
    ```

    Note: You are not allowed to use commands like `goto`, `setPos`, or similar. You must move the turtle by using the commands `left`, `right`, `forward`, and `backward`.

1.6 Sample Run

What follows is a sample run of your program for a specific set of parameter values. If you actually run your completed program, the following text should appear in the console. The only user-entered information is after the colons. The turtle will draw the figure shown in Tables 1 and 2, level 3.

```
Number of sides: 1
Length of initial side: 100
Number of levels: 3
Curve 1 - Island's length is 177.7777777777778 units.
```

```
Hit enter to continue...
Curve 2 - Island's length is 199.99999999999994 units.
Bye!
```

Find below more sample drawings for your reference:

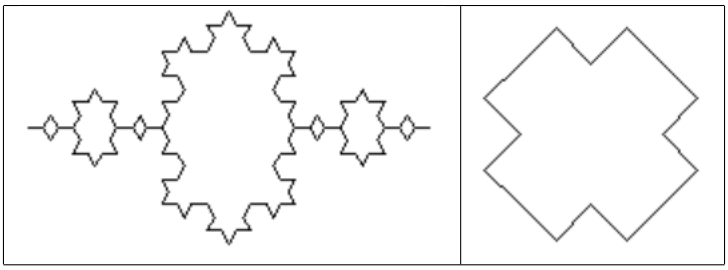


Table 3: Table 3. Islands with 2 sides, side's length 200, and level 4.

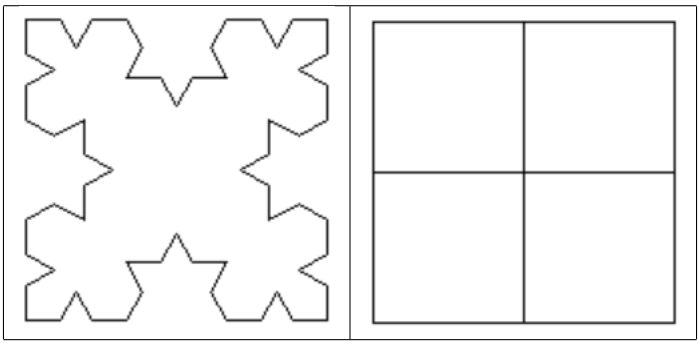


Table 4: Table 4. Islands with 4 sides, side's length 150, and level 3.

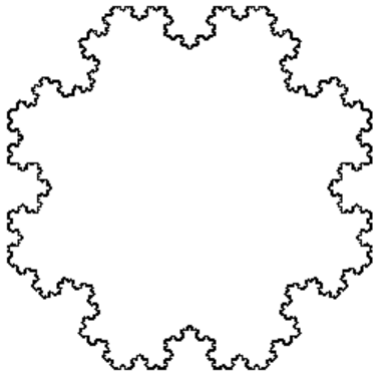


Figure 3. Island using curve 1 with 8 sides, side's length 100, and level 6.
Island's length is 3371.19341563786 units.

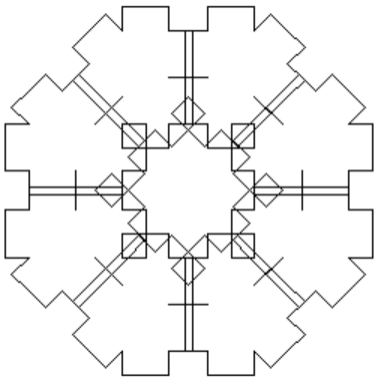


Figure 4. Island using curve 2 with 8 sides, side's length 100, and level 6.
Island's length is 4525.483399593902 units.

2 Grading

The assignment grade is based on these factors:

- 20%: Attendance at problem-solving and results of problem-solving teamwork
- 5%: Proper user input (including order)
- 5%: Error checking and re-prompting
- 20%: Accurate drawing of the island using curve 1 (15%) and compute its length (5%)
- 20%: Accurate drawing of the island using curve 2 (15%) and compute its length (5%)
- 5%: The program pauses between drawing the two islands
- 5%: The program resets the turtle window when drawing the second island
- 10%: Good design practices
- 10%: The code follows the style guidelines on the course web site.

3 Submission

Submit your `island.py` file to the MyCourses assignment before the due date.