



Lab Exercise 8: Recursion

Goal

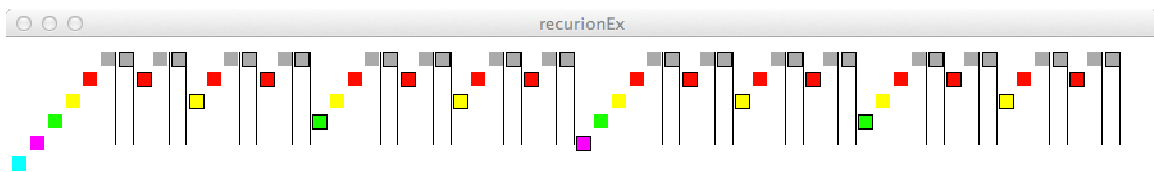
The objective is to understand how recursion works. The programming in this exercise is easy. The goal is to imagine how the computer follows the instructions.

Overview

You are given a recursive program that produces the following



and you will improve it so that it produces the following.



Then, you will explain in your own words what's happening.

Step 1: Start. Grab the Recursive Code posted with this lab page. Also, open a word processing document.

Step 2: Color the boxes. The first step is to modify the `paint()` function so that it colors each level differently. You choose the colors, but they must all be different, and consistent by level – no random colors this time! The `paint()` function will be just a series of if-statements, like you saw in the `keyPressed()` function. For example, it uses

```
if (level == 0) {
    return color(170,170,170);
}
```

and a separate if-statement for each level (up to, say, 7). Remember, copy/paste are your friends for programming similar sequences of code. *Check that each level has one color.*

Embellishment A. Next, **replace** the return statement in the `box()` function with the following code

```
stroke(0);
line(xdist-5, 20, xdist-5,150);
noStroke( );
```

This will draw the vertical lines. *Check your code and notice where the lines are drawn.*

In your document write a sentence or two to finish “The vertical line drawn in Embellishment A indicates visually the point in the recursive code where ...” The sentences are to say what’s happening in the code.

Embellishment B. Next, **replace** the two recursive calls in the `box()` function with the following code

```
noStroke( );  
box(level);  
stroke(0);  
box(level);  
noStroke( );
```

This will outline some boxes and leave others without outlines. Check out your code and notice which boxes are outlined and which ones are not. *Check your code and notice where the outlined boxes are drawn.*

In your document write a sentence or two to finish “The outlined boxes from Embellishment B indicate visually the point in the recursive code where ...”

Analysis. Check out the gray diagram above. Using your knowledge of recursion, and your observations/information from the two Embellishment steps, answer these questions: [Give answers in your document.]

1. The Processing program that drew the gray diagram calls the `box()` function how many times in total? _____
2. The box pointed at by the arrow is what color in the colored picture? _____
3. If the box pointed at by the arrow was drawn as a result of the call, `box(level)`, what was the value of *level* in the call? _____
4. Referring to the function that made the call of question 3, what color box did it draw (as shown in the colored picture)? _____
5. Referring to the function that made the call of question 3, which of the two `box()` calls was it? _____
6. As explained in Wednesday’s lecture, functions are often suspended in recursive programs while they perform recursive calls. At the moment that the `rect()` function was called to draw the rectangle referred to in question 2, what `box()` functions (if any) were suspended besides `box(6)`? _____

Wrap Up: You have taken a simple recursive program, and by adding lines and colors, you have tagged so that you can follow its behavior. This allows you to understand what it is doing.

Turn In: Paste your `.pde` code for `paint()` into your document, save it and submit it to the class dropbox.