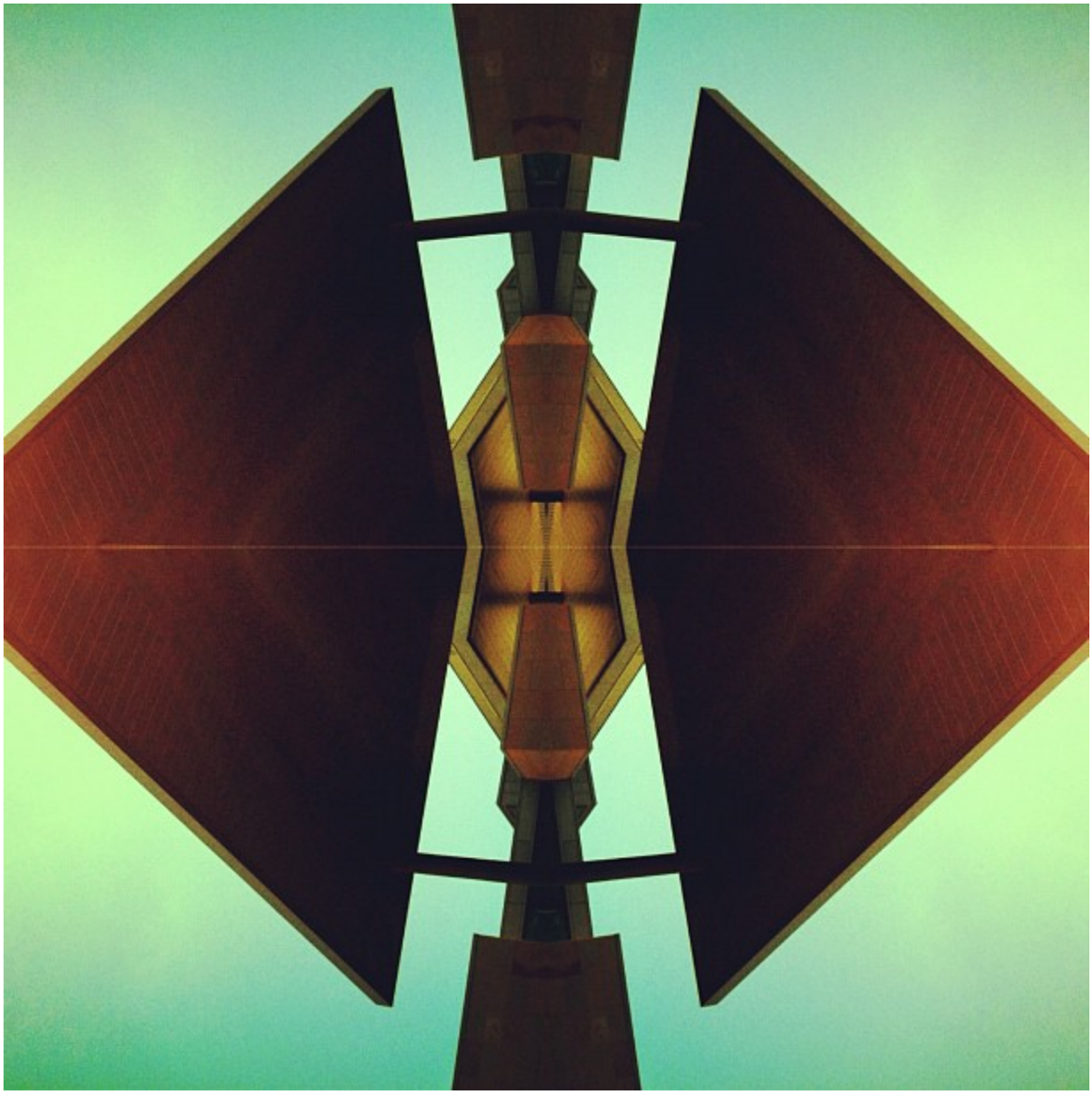
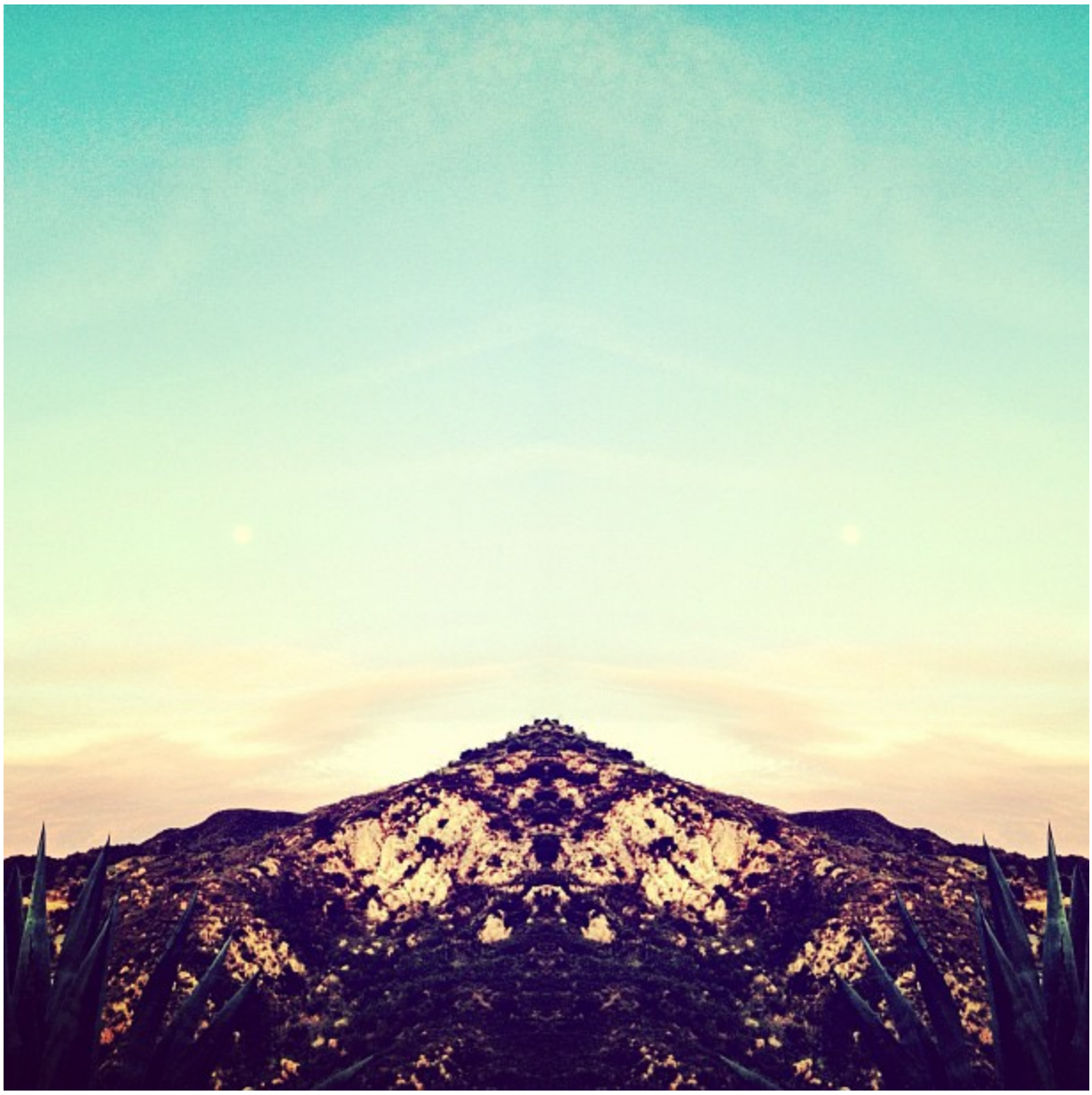
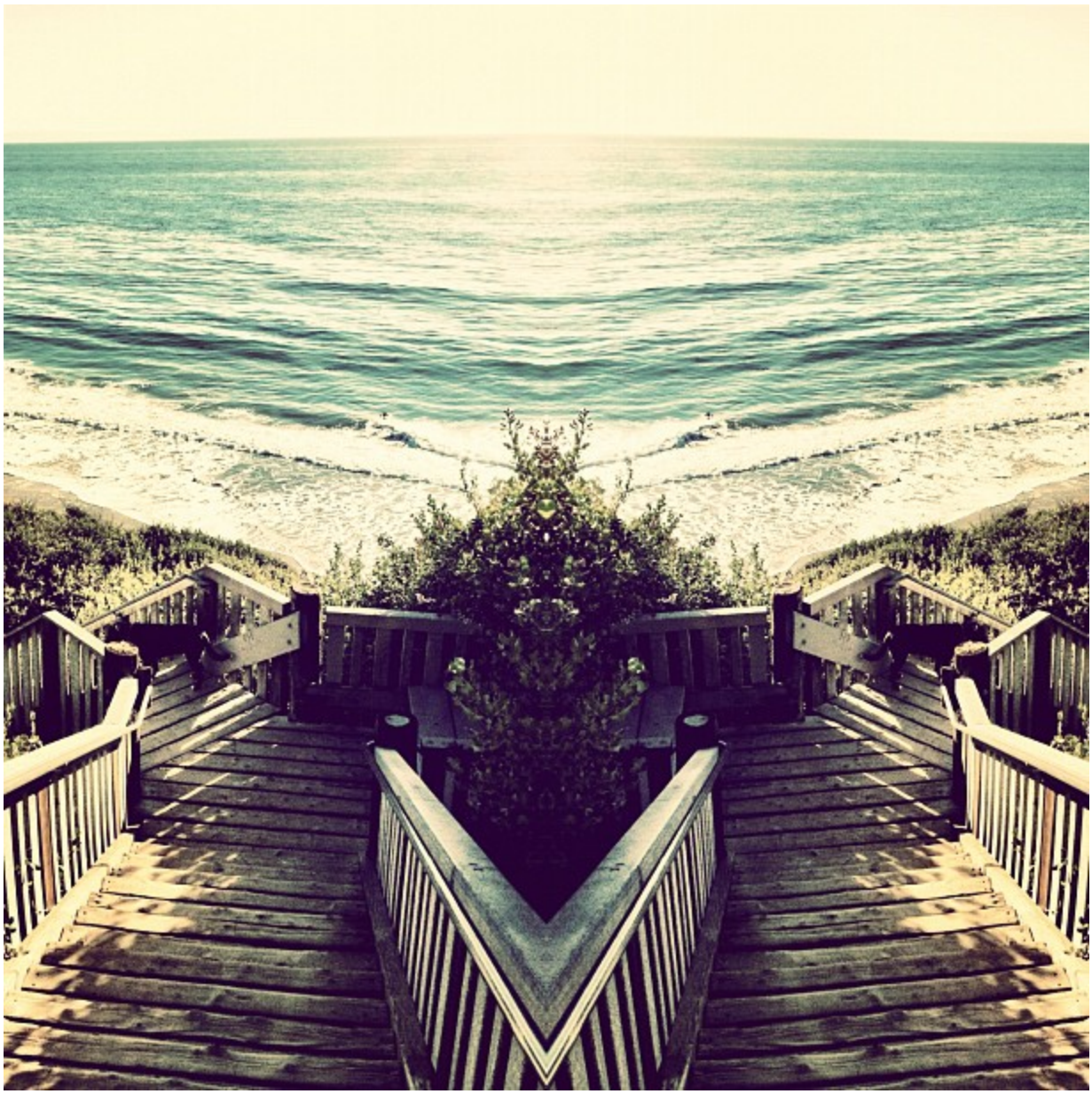
RJ Duran
CCS120 (W2012) - Symmetry and Aesthetics in Contemporary Physics
Art Project 2
2/17/12

For this assignment I built two representations of reflective symmetry in the visual and numeric domains. The first project was completed over the course of a few weeks utilizing my iPhone camera and several image manipulation apps on the phone. Most of the images explore the shape, space and forms found in nature reflected across a vertical axis in the center.







The second project was an exploration of the symmetry of numbers sequences and color patterns. To do this I wrote a simple program in Processing (Processing.org). I explored three number sequences and made a visual representation using only these three sequences. The symmetries seen here are mirror symmetries. In sequence 1 the numbers start at 1 and go to 20, at 20 it proceeds back to 1. Sequence 2 only counts up from 0 to 2 and back down before reversing direction again. Sequence 3 is a simple on/ off or toggle symmetry. It's basically the symmetry of anything digital.

sequence 1
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1,

sequence 2
0, 1, 2, 1, 0, 1, 2, 1, 0, 1, 2, 1, 0, 1, 2, 1, 0, 1, 2, 1, 0, 1, 2, 1, 0, 1, 2, 1, 0, 1, 2, 1,

sequence 3
0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,

These sequences were generated using the three pieces of code:

```
 println("sequence 1");
  int x = 0;
  int n = 20;
  for (int i = 0; i < 2*n - 1; i++) {
    if (i < n) {
      x++;
    }
    else {
      x--;
    }
    print(x + ", ");
  }
  print("\n");

  println("sequence 2");
  x = 0;
  n = 2;
  for (int j = 0; j < 4*n; j++) {
    x = 0;
    for (int i = 0; i < 2*n; i++) {
      print(x + ", ");
      if (i < n) {
        x++;
      }
      else {
```

```
      x--;
    }
  }
}
print("\n");

println("sequence 3");
x = 0;
n = 20;
for (int i = 0; i < 2*n - 1; i++) {
  x = i % 2;
  print(x + ", ");
}
print("\n");
```

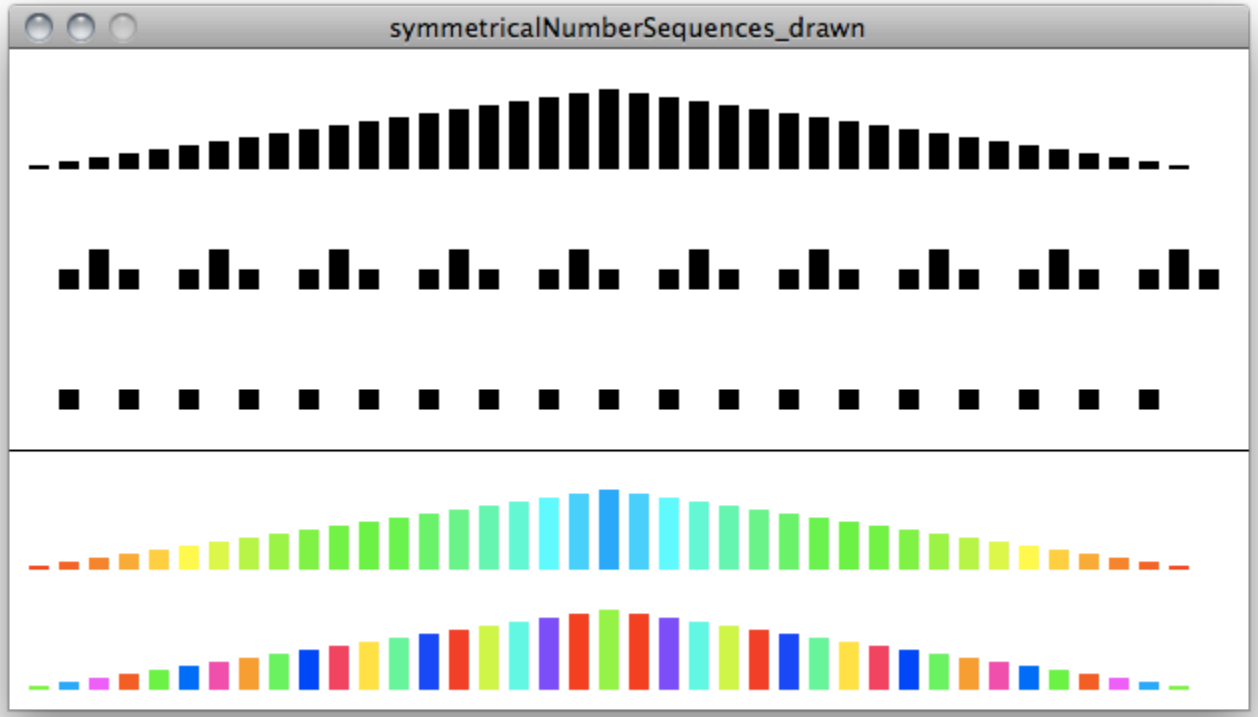
The color sequence uses number sequence 1 for the shape but utilizes a different pattern for coloring individual bars. In processing there is a function called fill() which is used to fill the shapes following it with a color specified within the parenthesis. The code for the second color sequence looks like this:

```
x = 10;
y = 320;
val = 0;
n = 20;

for (int i = 0; i < 2*n - 1; i++) {
  if (i < n) {
    val++;
  }
  else {
    val--;
  }

  fill((val % (map(mouseX, 0, width, 1, 10)))*100, 100, 100);
  rect(x + i*15, y - val*scale, 10, val*scale);
}
```

Basically, this says “fill the rectangle with the remainder of my value val divided by something that has a hue of 100 to 1000, a saturation of 100%, and a brightness of 100%.” The color here is specified in HSB notation. Below is a screen shot of the program running. The colors in the first color sequence are fixed and the second sequence changes with the movement of the mouse in the horizontal direction.


symmetricalNumberSequences_drawn