Intro to Generative Art
Problem Set 1

Harvard University
Taught by Sudhan Chitgopkar

This problem set is optional, but *highly encouraged*. You may work in groups and/or solicit outside help including but not limited to Generative AI tools. All outside sources/code used *must* be properly credited in the code documentation submitted.

1 A (Random) Walk Down Memory Lane

During lecture, we investigated how stochastic processes can be used to create powerful and interesting visualizations. *Modify the basic* ***`Walker`*** *object we created during lecture to exhibit some new properties of your choosing.* This may include (in order of increasing difficulty)

- position-dependent coloring
- diagonal walking functionality
- 3D random walkers
- territorial walkers (who avoid one another)
- self-avoiding walkers

To get started, recall the `Walker` object we created during lecture:

```
class Walker {
  protected int x, y;

  public Walker(int initX, int initY) {
    //set initial x and y values to given parameters
    x = initX;
    y = initY;
  } //Walker

  public void move() {
    //generate random num between 0 - 3 (inclusive) to determine next direction
    int nextDir = int(random(4));

    if (nextDir == 0) {
      //move right
      x += 1;
    } else if (nextDir == 1) {
      //move left
      x -= 1;
    } else if (nextDir == 2) {
      //move up
      y += 1;
    } else {
```

```
      //move down
      y -= 1;
    } //else

    x = constrain(x, 0, width-1);
    y = constrain(y, 0, height-1);
  } //move

  public void display() {
    point(x,y);
  } //display
} //Walker
```

2 What's All That Noise!

During lecture, we used the `random()` function to generate randomly distributed floating point numbers within an interval. However, randomness comes in all shapes and sizes. One such type of randomness is Perlin noise, which produces a visually organic sequence of numbers with applications in procedural terrain generation. Perlin noise can be generated directly in Processing using `noise()`. *Modify the above* ***`Walker`*** *object to use Perlin noise to guide its movement.* Compare the differences between a `Walker` using `random()` versus `noise()`. What do you see?

For reference, here are example outputs for each `Walker`.

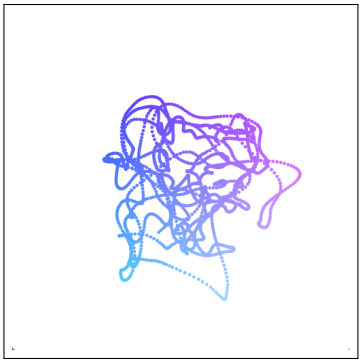

Figure 1: Perlin Walker

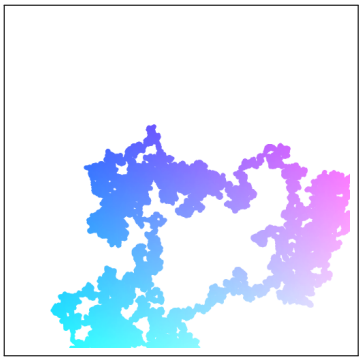

Figure 2: Random Walker