

# “white” versus “brown” noise

Jeremy Mates

May 12, 2018

## white noise

is simply random. In the context of a roguelike given a grid of arbitrary size white noise may be achieved by placing objects randomly on the grid. The file `noise.lisp` details the complete code; only snippets will be shown here.

```
(defparameter *rows* 20)
(defparameter *cols* 72)
(defparameter *board* (make-array (list *rows* *cols*) ...))

(dotimes (n 100))
  (setf (aref *board* (random *rows*) (random *cols*)) #\x)))
```

This code produces (or should produce) no apparent pattern.

```
...xxx.....x..x.....x.....x.....
.....x...x.....x.....x.....
.....xx.....x.....x.....x.....
...x.....x...x.x.....x.....x...x...x...
.....xx.....x.....
.x.....x.....x.....x.....
...x.....x.....x.....xx.....
.....x.....x.....x.....x.....
...x.....x...x...x.....x.....x.....
.....x.....x.....x.....x.....
.x...x.....xx..x.....x...x.....
.....x...x.....x.....x.x.....x...
.x.x.....xx..xx.x.....x...x.....
.....x.....
x...x.....x...x.....x.....
...x.....x..x.....x.....
.....x.....
.....xxx.....x.....x.....x.....x.....
.....x...x.....x...x.....x...x...
```

## brown noise

by contrast will locate each new object relative to the previous object. Care must be taken that the objects do not walk off of the grid (unless they wrap around) and that there is no bias (unless such is desired) that makes the walk favor a particular direction.

```
(defun if-legal (new max)
  (and (>= new 0) (< new max) new))

(defun nearby (x max range)
  (do ((new nil))
      ((numberp new) new)
      (setf new (if-legal (+ x (- (random range)
                                   (truncate (/ range 2)))) max))))

(let ((r (random *rows*)) (c (random *cols*)))
  (dotimes (n 100)
    (setf (aref *board* r c) #\x)
    (setf r (nearby r *rows* 5))
    (setf c (nearby c *cols* 9)))))
```

A notable feature here is that the randomization is different for the rows and columns; rows move  $\pm 2$  and columns  $\pm 4$ . Neither this nor the previous implementation care if an object overwrites another.

```
.....XX.X....X.....X.....
.....X..XX..X...XX.X.X....X.....
.....XXX.XX.....X.....
.....XX.XX.....XX.....
.....X..X..X.....X..X.....
.....X.....
.....X.....XXX.....
.....XXXX.....
.....X..X.X.X....X.....
.....X.X.XX...X.....
.....X.....X.....X.....
.....X.XX.....X.....X.....
.....XX.....
.....X.X.....X.....
.....X.....XX.....
.....X..X.....X..X.....
.....X.....X..X.X.....
.....X.....X..X.X.....
.....X..XX.....
.....X...XXXX.X.....
```

This is obviously quite different from the white noise; moreover, repeated runs of the white noise version will look more or less the same while the brown noise will vary greatly depending on the starting point and exact sequence of random values.

With multiple object types the differences between white and brown noise can be made most notable; instead of x the following place plants, fungus, and rocks.

```

.....#.....P.....f..f.....#P..f.....P..#..P
.....#...P.....#.....P...P..PP.....f.....P..#...P..#..f
.....P..P..P.....#f.f.f...f.....#...#P...#..f..#.....
f.P.P..#..f#...#.....P.....#...#.....f.P.fP...P...#f.....f.f#..f...
...#..P.f...#..P.....P.....f.....P.....f.f...f.....
#.....Pf.....#.....#...f.f.P.....f...P
.....#.....P.f#...#..f.....#...#..f.#f...f.....f...P
..P...Pf.....ff.....f#.....ff..##.f.....#.....
.P.....#..#...P.....f.....f.....f.P...#ff.#..#...f...
..f.P...#f.....fP.....f.....#.....fP...#.....
...f.....P...f.....f.P...#...P.....P.P..f..P#.....P.....
...#..f#.....#.....f..f...fP.....f.....P..#..#.....#..##.....#..P
#ff.....#.....#.....f.....#.....
.....P.....#..P.....P...P..f.....#.....#.....
..#f#..#.....f...P..f...P...P...P.P.P...P..P...#.....f.....#
.....P...f.....P...##...P#...#...P..#..#.....fP.....#...
#f...f.....#..P.....f#...#..f#P...#.....#.....PP...f...#
.P.....f.#.....f...P..P.f..f.#...###...f.....f..f
.PP.#PP.#.....##...#P.#.....fP..#
.#...#.....#..P...f.....f.....#..P...P...P.f.P..f

```

and a brown noise version

```

.....f.....ff...f.ff.....#...#####.....
.....f.....ff.....##.....##.##.##.
...f...f...f...f...#..#..#..#####.##.###.....
.....f.....#...#...#..##...###.##.....
f...f...ff.....#.....#...###.##.###.###.....
.....f.....#...#..#..#...#..#..#.....f.....
.f.....P...#..#..##..#..#####..#.....
.....P...P.....###...##...#.....f.....
.P.P.f..P.....P.....P..P..##..#..#...#.....f.....
.P..P.....P.....#..#..##.....#.....f.f.f.....
...P..fP.....P.P.PP.P###.....#.....f..f.....
..P...P.PP.fP...PP..P.....#.....f.....
...PP.f..f.....f..Pf.....PP.....ff.f.....f.....
...PP..PP...f.....P..P.Pf.....f.f.f.....f..f..f.....
P.P...P.....f..f.f...f.....f.....fff.f.....
.P.PPPP.....fP...f...f..f...ff.....f.f.f.f..f.fff.....
..PPP...P.....f.....f.....f.f.f.ff.f..f.....
PPPPPP..P.....P..PP.....f.....f.f.f.....f..f.....
PPPPPP.PP.....f.f...f..ff...f.....
P.P.PPP...P.....fff.....

```

Other complications are possible under brown noise. The randomization rules could vary depending on the object type being placed, or might even vary over time—close randomization punctuated by the occasional long leap would be an obvious thing to try.