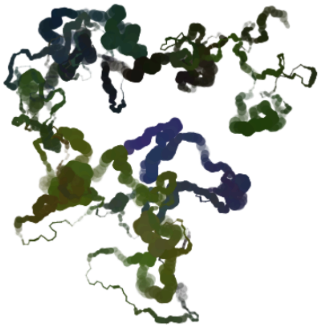
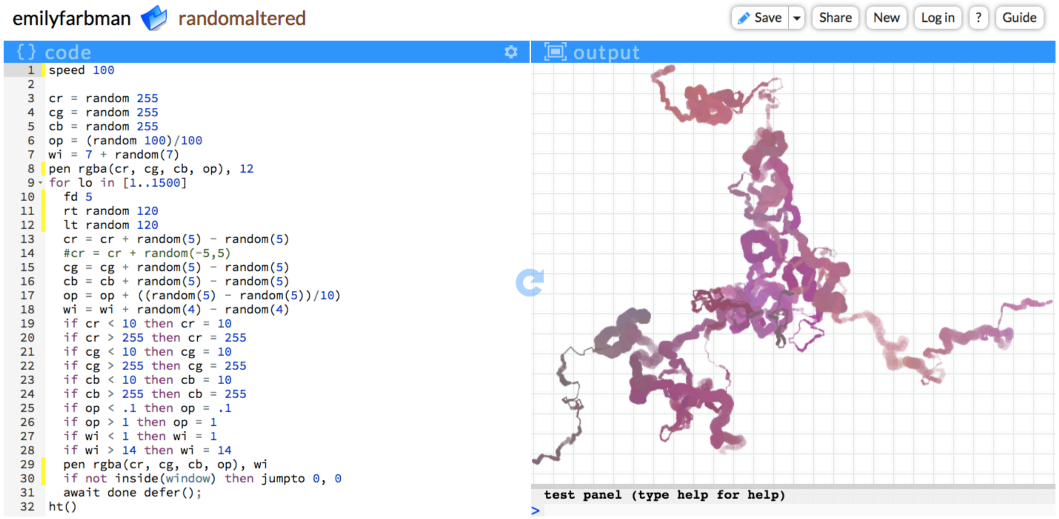
Interpreting Coded “Paintings”

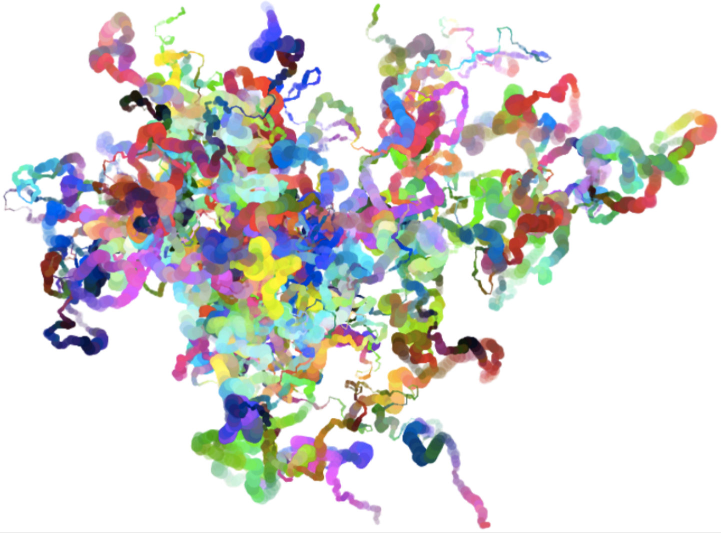
What can we interpret about code through observation and repetition of a randomized program?





```
speed 100

cr = random 255
cg = random 255
cb = random 255
op = (random 100)/100
wi = 7 + random(7)
pen rgba(cr, cg, cb, op), 12
for lo in [1..1500]
  fd 5
  rt random 120
  lt random 120
  cr = cr + random(5) - random(5)
  #cr = cr + random(-5,5)
  cg = cg + random(5) - random(5)
  cb = cb + random(5) - random(5)
  op = op + ((random(5) - random(5))/10)
  wi = wi + random(4) - random(4)
  if cr < 10 then cr = 10
  if cr > 255 then cr = 255
  if cg < 10 then cg = 10
  if cg > 255 then cg = 255
  if cb < 10 then cb = 10
  if cb > 255 then cb = 255
  if op < .1 then op = .1
  if op > 1 then op = 1
  if wi < 1 then wi = 1
  if wi > 14 then wi = 14
  pen rgba(cr, cg, cb, op), wi
  if not inside(window) then jumpto 0, 0
  await done defer();
ht()
```



Students were apprehensive when I told them we would be doing “coding” work during an art class. My objectives were to treat the visual product of the code like a work of art and interpret it based on the variations in the results. After viewing and capturing several events then reading and discussing the code, we tried to demystify the language of the code (above right), understand where the variables were, and make changes based on where we wanted to see visual and aesthetic improvements (results center right). The student takeaways were many and the following class a student volunteered the results of her adapted code (below right) that had run all night.