# Computational Modeling of Social Behavior

Coding Challenges – Day 1

1. Taste the Rainbow. Your task involves the creation of a new NetLogo model called Color Spread. Create this model and save it. All patches will start as white, except a single patch of a second color.
(a) Write a 'go' procedure to get a color to spread from patch to patch. (There are many ways to do this. Pick one you like.) That is, a patch should change color if one or more of its neighboring patches are colored. The NetLogo primitives *neighbors* or *neighbors4* may be helpful.
(b) Create buttons to setup the simulation and launch the procedure.
(c) Add a plot that graphs the number of colored patches over time.
(d) **CHALLENGE**: Create a chooser to allow the user to select which color will spread.
(e) **CHALLENGE**: Create a switch called *rainbow-spiral?*. If it's switched to ON, your model should produce a spiral (either inward, starting in a corner, or outward, starting in the center) of spreading colors in which patches change color one-by-one. Each patch should have randomly chosen color.
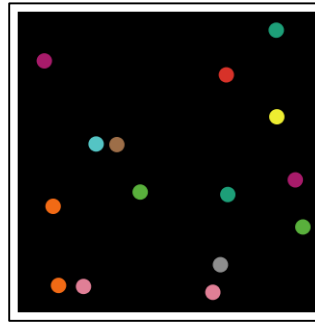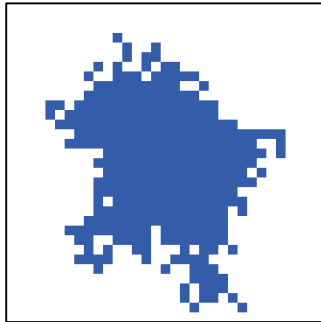
2. Super Smash Brothers. This task involves the creation of a new NetLogo model called Collisions, in which circular agents will move in space and "bounce" off walls and each other. Create this model and save it.
(a) Create turtles that move in a straight line until they collide with the edge of the grid or another turtle. Assign turtles the shape "circle." Agents should be initialized in random location with a random heading.
(b) Make the turtles interact with the boundaries of the world, and "bounce" off the walls. From physics, let the "angle of incidence" equal the "angle of reflection" (i.e., if a moving turtle hits the wall on a shallow angle, it should bounce off at a similarly shallow angle. To do this, consider that the turtle's *heading* is zero when the turtle is facing toward the top of the screen, and 90 when the turtle is facing right. If the turtle hits the right or left wall, it's heading can be set to its current heading's negative. If the turtle hits the top or bottom wall, its heading can be set to (180 – heading).
(c) When a turtle "hits" another turtle (gets within a certain distance of another turtle), have it "bounce." To do this, you can start by simply giving the turtle a new random heading. As a challenge, see if you can make the bounce more realistic.
(d) **CHALLENGE**: Create a plot of the cumulative number of collisions as a function of time.
(e) **CHALLENGE**: Have the turtles "break apart" when they collide. So that a colliding turtle will transform into four smaller turtles. When these smaller turtles collide, they will disappear.