

Neighborhood specification

This program is a tool for studying how intolerance leads to segregation (people living separated by type) and how tolerance leads to integration (people all living together). It is simplistic, of course, but I hope it helps us all think about how to get along with each other

You have a world which is a rectangular grid of houses in a torus. (Torus explained below). Each house has a person of some race in it (color coded with red and blue) or it is empty (white). In a single round, each person has ONE chance to stay where they are or move to a vacant house, according to whether they are happy with their 8 current neighbors. If a person decides to move, they move randomly to any vacant house in the whole world, without regard to the neighbors there. The simulation repeats this for many turns and we see where everyone goes.

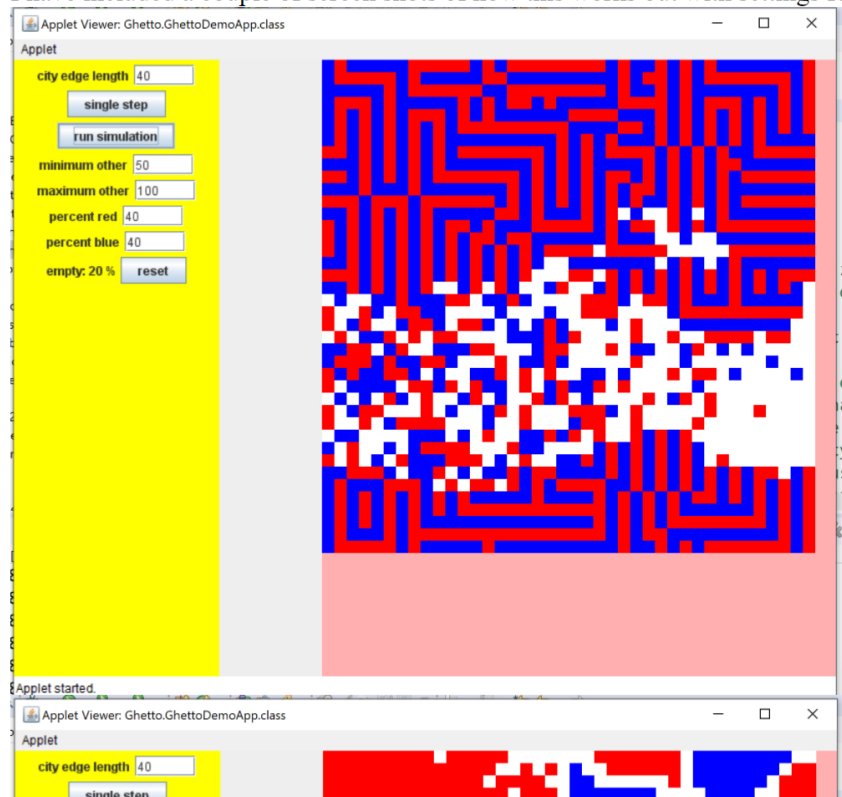
Write a program which simulates this situation. There should be controls for the following:

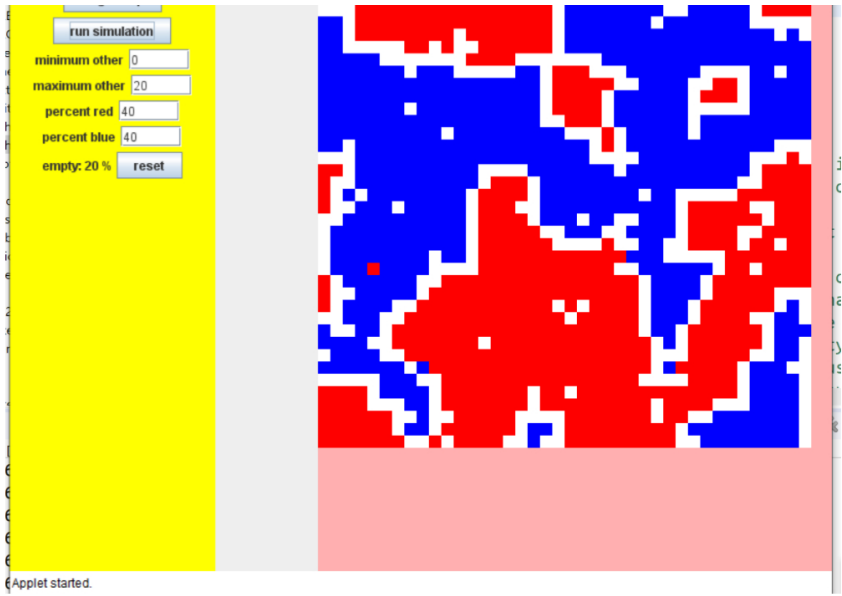
- Reset button to restart with current parameters.
- Setting (textfield or slider or ??) for the size of the world (number of squares on an edge).
- Setting for the maximum percentage of neighbors not-like-you that a household will tolerate (tolerate = no move). For example, if the number is set to 70, a red person with 6 blue neighbors will move (because $6/8 = 75\% > 70\%$); a red person with 5 blue neighbors will stay where they are (because $5/8 = 62.5\% < 70\%$). Note that empty houses do not count as different than you.
- Setting for a minimum percentage of neighbors not-like-you that a person will tolerate (no move). Make this greater than 0 for a person who LIKES diversity. For example, if you set it to 20, a red person with zero or one blue neighbor will move in hopes of being near MORE blue neighbors.
- Setting for the percentage of red persons. Note: you can do this with probability, e.g., if you say to fill 40% red, you can fill each house randomly with a probability of 40% red, and the numbers should come out close enough.
- Setting for the percentage of blue persons. White will be what is left over. 40% or 45% is typical for each of red and blue, to make the whole thing work. So start with that but let the user change it.
- Single step button gives every person one chance to move. Note that if you check houses for moving from top left to lower right, persons who move down or right might get two chances, not good. You may need to make a separate list of persons and randomize it, then go through it to let each one move just once.
- "Go" or "run" button, to make the program do repeated steps, until you say "stop", which can be via the same button or a "stop" button.
-

Then play with it. See how low the tolerance of others has to get to make segregation happen. Does 50% do it? See how high you have to make the desire for diversity to make segregation stop. You do not need to deliver any of these results. I will test your program myself.

Torus: a rectangular grid where the world wraps -- connecting left and right edges -- like a world map East to West -- and also top and bottom. This forms a torus. This may sound complicated, but it is actually simpler -- everyone has 8 neighbors, no edge effects.

I have included a couple of screen shots of how this works out with settings for integration and for segregation.





To submit, name your package "NeibLastname" and then zip it up and put on Bb.

rubric

what	points
control fields, places to put numbers or sliders, layout	20
town shows up, approx right percent red/blue	20
simulation runs, start and stop	20
low max causes segregation	10
high min (30%?) causes integration	10
clean code, comments	20
total	100