

Project 3-C Implementation Summary

By Kartikeya Kumar and Karan Shah

Our project, as already outlined in the Conceptual Model for 3-B, was to model migration phenomena between countries over different time steps. This week, our implementation was focused on initializing the world correctly. Currently, we are modeling our world as a grid with different number of countries fitted onto it. In the 2nd Code Block, global variables **M** and **N** represent the dimensions of the world with another variable **num_countries** representing the number of countries on the map. Another list variable called **attributes** lists the attributes that each country will have. We may add more along the way.

Currently, each of the country's attributes are initialized in a method called **initialize_country_attributes()** which creates a Dictionary mapping each country to an array which contains the values of their different attributes such as Wealth Score or Birth Rate. All the values of these attributes for each country are drawn from a uniform random distribution with fixed limits set for the possible values. Based on the values initialized for each of the countries, the values for each of the cells or regions within that country are initialized. This is done by passing the value of a country's attribute as a mean into a method called **create_distribution()** which is a self-implemented Gaussian generator. This outputs a list of evenly spaced Gaussian values which are then distributed across that country's cells. The attributes serve as means for the generator. So each country has different means and each cell in that country has some variation.

Passing the country's attribute value as a mean into a Gaussian generator ensures that each of the states within a country have attribute values closer to the average values that their country's attributes have. So, a rich country, has richer states whereas a poorer country has poorer states.

For the convenience of the reader, the population for each of the countries (sum of the population of all states within a country) is printed when the program is run. Along with that, a visualization for each of the attributes for all the countries is printed out. So, the reader can see, for example, how wealth is distributed all over the world.

Implementation Problems

We found it infeasible to implement countries with irregular shapes. It required a lot of bookkeeping. So, all the countries are rectangular shaped. We felt that implementing oceans, as said in our conceptual model, is not of any significance right now. We may, however, reconsider this later on.

Also, as of now, our dynamic formula for fitting countries onto the maps works only with an even number of countries. If the number of countries is odd, the map does not come out properly. It does display odd numbers of countries but 1 country gets wrapped around the map.

You can see this by changing the global properties (M,N,num_countries etc).