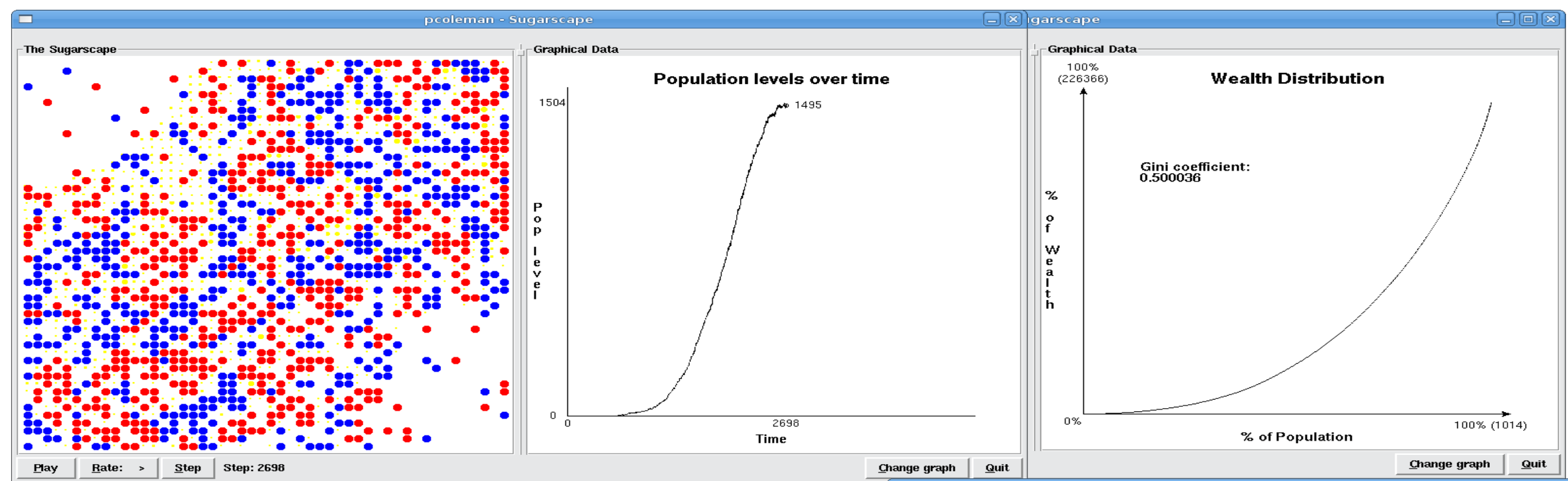


Programming a New Sugarscape

By Patrick Coleman



BACKGROUND

Growing Artificial Societies: Social Sciences from the Bottom Up written by Joshua M. Epstein and Robert Axtell and *Micromotives and Macrobehavior* by Thomas Schelling define Sugarscape and the Schelling segregation model. Tony Bigbee from George Mason University has written the Sugarscape in Java and his code will be used for reference along with the first book primarily. In the book by Axtell and Epstein Schelling's segregation model is mentioned and the Sugarscape is built with two separate groups (tribes) which combat against each other. The results should mirror those of the Sugarscape models in *Growing Artificial Societies*. However, once Schelling segregation is implemented with possibly more than two different colored populations the results will differ. In all likelihood only two groups will survive in the long run. The final results will be presented with screenshots of the running program along with graphs of relationships of variables. It will perform like previous Sugarscape models. *Growing Artificial Societies* and *Micromotives and Macrobehavior* are two books which are used as references to develop this project.

DEVELOPMENT

I. Theory. The algorithm driving the move method of the agents is at the core of the simulation. The agents look out in the four cardinal directions as far as their vision allows and move one square in the direction of the closest location with the most sugar. If more than one location is optimum, a random direction is chosen. To incorporate Schelling segregation, locations in which there would be more agents of the opposite color than of the same color are removed from the possible choices. See the section on agent movement in Appendix A. Agents are added to the environment according to an exponential function which models real life. The section on population growth in Appendix A shows how the population growth is raphed. The inequality of the population is found using the Gini coefficient. The Gini coefficient is calculated according to the formula: $1-2*L$ where L is the area under the Lorenz curve, which is calculated using trapezoidal Reiman sums. The section on wealth distribution in Appendix A shows how the Gini coefficient is calculated and how the Lorenz curve is graphed. Lastly, hemispherical winters cause agent migration to other high density locations and cause a drop in population.

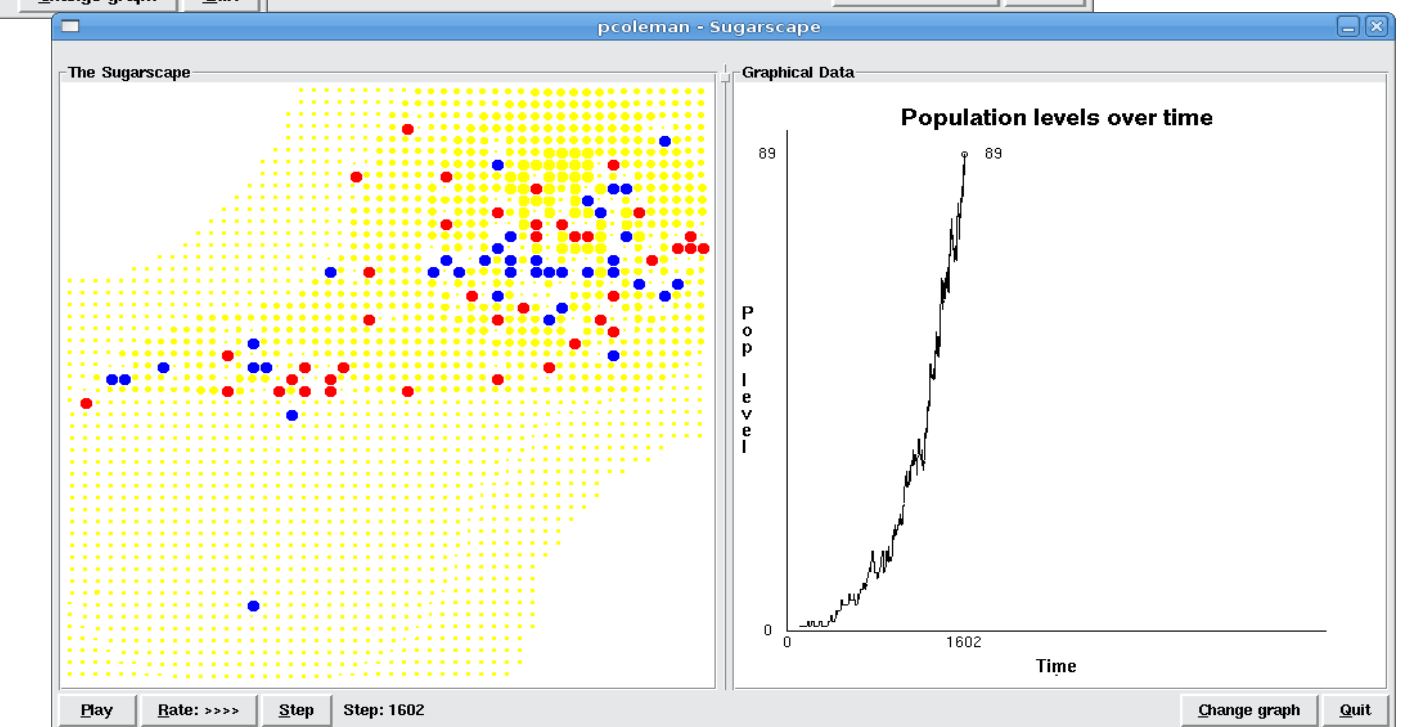
II. Design Criteria. The goal of the project is to accurately represent the models it is implementing. It follows the Sugarscape design from *Growing Artificial Societies* by Axtell and Epstein and the Schelling segregation design from Schelling's book *Micromotives and Macrobehavior*. The agents and the environment behave as they should with respect to the aspect implemented so far. The Schelling segregation model will accurately represent Schelling's model as best as possible, but will not be perfect because concessions will need to be made to allow it to run in the Sugarscape. The information shown in the graphs and the display of the environment will be compared to the results found by the authors.

III. Materials. The program code was written in Ruby (see <http://ruby-lang.org/>). Tk toolkit is used for the GUI representation and graphics in the program. A text file which represents the maximum capacities of sugar in various locations in the environment was used from GMU's Tony Bigbee's files (he wrote a Java version).

IV. Procedures. Currently the program displays the environment, and has the agents move and harvest sugar. The display draws each location in the matrix using a circle whose radius increases based on the amount of sugar at that location. The display draws the agents as a red circle with the same radius as a location with the maximum amount of sugar. The display also shows the current time step. The GUI window has a frame containing the canvas and buttons to play/pause, step the simulation, increase the refresh rate, and to quit the program. The agents themselves choose the closest location with the greatest amount of sugar. If more than one location matches these requirements, one of them is randomly chosen. Locations with more agents of the opposite color are removed from the choices. Then the agent harvests the sugar and consumes from his own supply of sugar. At each time step the sugar in the environment grows back by one. The program begins with a small number of agents and adds to the population using an exponential function so that it reaches carrying capacity. Modifications in the individual agents include an improved move method, a random age limit, and a variable for red or blue color, to allow for segregation. The GUI window has been modified to include buttons to change the graph and change the refresh rate. There is a button to change the refresh rate in the display of the environment and of the graphs. The two graphs which are now displayed are the population growth over time, and the percent of total wealth over the percent of the population (Lorenz curve). To get the population graph, it keeps track of the length of the array of agents at each time step in the simulation file and cycles through the array of population values in the display file. To get the wealth graph, it cycles through the array of agents and stores the wealth of each individual agent. Then it sorts this array and cycles through it keeping a running total to determine percents.

ABSTRACT

This project studies artificial societies, especially the Sugarscape and the Schelling segregation model. To implement the Sugarscape, a display of the sugar-filled environment with agents is outputted. The simulation allows agents to harvest sugar, consume sugar, die of starvation, and die of old age and allows the environment to grow back at a given rate. To implement the Schelling segregation model, two distinct groups of agents are added to the environment with a preference for neighbors of their own kind to determine the effects of the individual preferences on the society at large. The reasons these two projects are being implemented is because while both are often compared, the two models in their original forms have not been combined and analyzed in a single simulation. In addition to displaying the environment, graphs showing the population growth and wealth distribution are displayed. These graphs analyze what is occurring in the simulation. Seasons are implemented to analyze agent migration. The program code is broken up into files: a main file, an environment file, an agent file, a location file, a display file, and a simulation file. The conclusions show that the the model conforms to Axtell and Epstein's models in the areas which were implemented. But more importantly, it shows that the simulation conforms to real world phenomena reasonably well.



RESULTS

It has been determined that the program meets the design criteria in the areas in which it was implemented. The graphs are what answer many of the experimental questions. Descriptions of the population growth graph refer to the section on population growth in Appendix B. In general it follows the shape of logistic graphs which are proven to be a fairly accurate representation of population growth. Growth is slow when the population is close to zero and close to the carrying capacity, and growth is highest at half of the carrying capacity. The few anomalies reveal certain aspects of the simulation. The initial portion of slow growth is smaller than the final portion because the population begins with three individuals instead of one (but starting with one agent would not completely remedy this). The oscillations near carrying capacity come from the age limit of agents. It takes longer for the population to decrease due to dead agents than it does for it to react to the added agents. The oscillations decrease over time and will eventually disappear. At about half of carrying capacity the line begins to become jagged instead of fairly straight like it was earlier in the simulation. This is a result of the heterogeneous population. In the beginning even agents with low vision and high metabolism (less fit agents) have room to survive in the regions of abundant sugar. As the environment fills up only better fit agents can survive on the fringes, areas with less sugar, so many added agents die quickly. The effects are even more pronounced as population approaches carrying capacity. The graph has even more information to offer when seasons are included. The spikes in the graph represent the winters. The spikes alternate in intensity because the northern winter is more sevr removing more high desnity sugar locations. Descriptions of population inequality refer to the graph in the wealth distribution section of Appendix B. At first a bar graph was used to represent wealth distribution, but it was replaced with the Lorenz curve. Both conform to the graphs in Axtell and Epstein's book. They show that there are very few wealthy agents (agents with a lot of harvested sugar stored) and many poor agents. In this sense the population is pretty unequal. The Gini coefficient is a numerical representation of this phenomenon. A coefficient of zero represents perfect equality and one represents perfect inequality (one agent has all the wealth). The number is just over .5 showing that the Sugarscape population is closer to perfect inequality than to perfect equality. See Appendix B for a display showing Schelling segregation. There is some segregation at this point. The environment did not split in half as was expected. The most probable explanation is that this is due to the method of adding agents. Random colored agents are added to random empty locations. Asexual reproduction of same-color agents would likely produce better results.