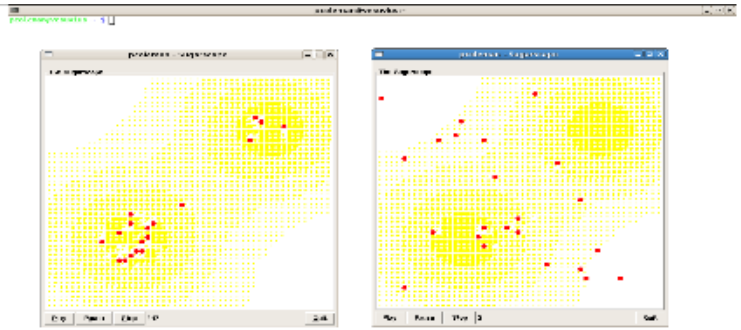
THE SUGARSCAPE

By Patrick Coleman



ABSTRACT

This project will study artificial societies, especially the Sugarscape and the Schelling segregation model. To implement the Sugarscape, a display of the sugar-filled environment with agents will be outputted, the simulation will allow agents to harvest sugar, consume sugar, die of starvation, migrate during plagues, reproduce, and combat each other and will allow the environment to grow back at a given rate and undergo plagues. To implement the Schelling segregation model, two or three distinct groups of agents will be 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. The program code will be broken up into files: a main file, an environment file, an agent file, a location file, a display file, and a simulation file. When concluded the product will accurately implement the models as described by Schelling and Axtell and Epstein.

RESULTS

The program will be checked to see if it corresponds to the results obtained by Axtell and Epstein. Mathematical formulas are listed in the back of the book, and displays of charts and graphs showing the relationships between various variables are shown throughout the book. These can be used in conjunction with the version implemented in Ruby which will eventually display graphs that can be compared to said graphs and mathematical formulas. The program will meet as many of the specifications of Sugarscape as defined by Axtell and Epstein as possible. As there is a large amount of data on the results of certain variations of the Sugarscape in the book by Axtell and Epstein, versions of the project can be compared to the results in the book to see if it is running as it should. The testing I have done so far has been verifying whether or not my running program matches the one described by Axtell and Epstein in Growing Artificial Societies. I have done various tests displaying to the command line different information such as the possible choices of an agent or the amount of sugar at each location. This information is used to track down the problem in the code so that the program will run correctly.

INTRODUCTION

As of yet the Sugarscape society has not been implemented in Ruby and it would be valuable for this code to be available because of the scope of the Sugarscape research. Sugarscape has inspired further research concerning agent-based modeling and artificial societies. The Schelling segregation model was one of the first artificial societies to be implemented on a computer and has defined the area of study. The combination of these two models can provide valuable insight into human culture. Perhaps 3 different groups could be put into the Sugarscape instead of the usual two different groups. Lastly, combat between different groups will be implemented, as this has not yet been done by Tony Bigbee at George Mason.

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.

DEVELOPMENT

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, and step the simulation 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. 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.