Modeling Evacuation of Population Centers Using NetLogo

Kurt Andrew DeSoto
Thomas Jefferson High School
for Science and Technology
Computer Systems Research Laboratory
Alexandria, Virginia 22312
Email: kdesoto@tjhsst.edu

***Abstract*— The actions of citizens during terror and evacuation events are oftentimes hard to predict. Using NetLogo, a 'cross-platform multi-agent programmable modeling environment' from The Center for Connected Learning and Computer-Based Modeling (CCL), the socio- and psychological factors affecting decision-making in these situations can be effectively simulated. Through appropriate research in categories of modeling and sociology, human behavior can be studied to help urban developers and social engineers protect the nation's interest: its citizens. Citizens that follow certain algorithms, or behave in certain ways, have a much greater chance of survival.**

I. Introduction to NetLogo

NetLogo is a cross platform multi-agent programmable modeling environment created by Northwestern University's Center for Connected Learning and Computer-Based Modeling (hereafter CCL). It was designed specifically for simulation of social and natural phenomenon. Originally StarLogoT, this language and graphical user interface hybrid offers a multitude of possibilities for researchers and students, allowing users to examine interactions and behaviors on both micro and macro scales. NetLogo is designed with accessibility in mind; the program allows for quick Java applet exportation and interactivity with CCL's HubNet interface, which lets the user run participatory simulations in which a class or group of testers takes part in enacting the behavior of a system as each user controls a part of the system by using an individual device (such as a TI-83+ calculator or a networked computer). NetLogo comes packaged with an expansive models library and an assortment of code samples for easy access.

A. NetLogo Graphic User Interface

NetLogo's main advantage is that it allows a quick and relatively simple way for the programmer to implement a convenient graphical user interface (hereafter GUI). A NetLogo GUI can contain the following components: buttons, sliders, switches, choices, monitors, plots, texts, and outputs.

1) Buttons: Buttons activate on the user's command. There are two types of buttons, standard buttons and forever buttons. Standard buttons activate the contained code on click, but forever buttons execute the contained commands repeatedly until the button is clicked a second time, mimicking a while-loop.

2) Sliders: Sliders allow the user to manually select the value of a certain variable. Sliders control the minimum and maximum values, as well as the increment of increase/decrease, while the user chooses the value.

3) Switches: Switches allow for a user to directly affect a boolean value.

4) Choices: Through the use of choices, the user can select certain values, Strings, and more that the programmer allows.

5) Monitors: Monitors allow the user to examine the value of a variable as the simulation runs.

6) Texts and Outputs: Texts and Outputs display certain messages the programmer encodes. These can be displayed at any time during the simulation.

B. HubNet and Other Extensions

NetLogo's versatility allows programs to be accessed and used in environments outside the computer laboratory setting. HubNet, one instance, allows a multitude of users to interact with a single model through the use of Texas Instruments TI-83+ calculators or networked computers. This is ideal for classroom settings because it allows each tester involved to affect the model in some way. Also, NetLogo offers easy exportation to Java Applet format, and with several keystrokes, an applet embedded in an HTML page can be created.

II. Introduction to Modeling

Modeling is a powerful tool that allows a programmer or social engineer to observe cause-and-effect relationships in occurences that a) happen too slowly or quickly to see, b) involve danger or safety concerns, c) occur on a scale too large or too small for study or d) is not a common occurrence. Armed with this knowledge, a scientist can use the material learned to help a community, understand a cause, or solve a problem. A perfect use for the modeling technology of today is a subject that meets all of these above criteria. The modeling of population centers during evacuation situations is perfect for such a study, because evacuation frequently occurs too quickly, is quite dangerous, happens rarely, and happens on such a wide scale that observing patterns is nearly impossible. In order to accurately code such an event, certain features must be researched.

A. The Environment

In general, areas to be evacuated generally contain a large number of citizens in a small area; the population density in such regions tends to be quite high. In the metropolises and cities of today, high-rise apartments, parking garages, subway stations all provide unique, but crowded, environments. These situations provide the greatest trouble to researchers and social engineers.

B. The Citizens

In times of terror or enormous stress, citizens often behave irrationally; different than they would normally. Thus, citizens' behavior in the simulation must reflect this behavior. The intelligences programmed into version 1.2 follow.

1) Speedy, Normal, Slow: The Speedy, Normal, and Slow citizens differ in the number of times they move per timestep. Speedy, Normal, and Slow citizens more 3, 2, and 1 squares, respectively. When they collide with a wall (or another citizen, if Collision? is turned on), they turn in a random direction and continue their movement.

2) Righty: Righty citizens react to collisions by turning ninety degrees. They move two squares per timestep.

3) Retreater: Retreaters react to collisions by turning 180 degrees, taking a step forward, and choosing a direction in their 90 degree front arc.

III. Running the Simulation

Utilizing the Simulation consists of three steps: initializing the Basic Environment Setup, or BES; running the model; analyzing the model with the Graphical (Statistical) Interface, or GSI.

A. Basic Environment Setup (BES)

The user controls the following sliders, choices, and etc. to set the parameters for the environment.

1) 'walldistancey' Slider: sets the distance between the horizontal walls. When a citizen's movement would take it into a wall, it runs its collision program detailed above. The two horizontal walls each contain a passable gap separating the danger from the safety area.

2) 'walldistancex' Slider: sets the distance between the vertical walls. When a citizen's movement would take it into a wall, it runs its collision program detailed above.

3) 'evacuationtime' Slider: sets how many timesteps the model runs for.

4) 'collide?' Switch: selects whether or not citizens collide with one another, activating collision code upon impact.

5) 'areasize' Selector: selects how many citizens begin in your model. The selector has five options: a) Metropolis (225 citizens) b) City (150 citizens) c) Village d) Town (90 citizens) e) Thorpe (15 citizens) f) Abode (3 citizens). There are (citizens / 3) citizens of each color (red, white, blue) in the model.

6) 'explosiontime' Slider: selects at what time the explosion occurs. The explosion is the radius selected below.

7) 'explosionrad' Slider: chooses the radius of the explosion. Explosions kill citizens contained within them and creates an impenetrable wall.

8) 'Draw Walls' Button: while depressed, lets the user draw black walls 1 pixel in diameter.

IV. The Project

This project was set up to determine the most effective ways to get large numbers of citizens to safety in times of danger. In order to determine how to do this, certain things must be determined. First, the parameters that add to the dangerousness of the environment should be measured. Secondly, once the situations of greatest danger are found, the most effective intelligences to escape this danger must be poinpointed as well. Using the environment instantiated in the setup (see subsection "The Environment"), different situations are tested.

The tests were run in environments of four different dimensions: large x large, short x large, large x short, and short x short.

V. The Test

As described above (see "Project" section), the first part of the test is to determine the dangerous situation. The results of this test follow. These tests were ran with build version 1.2, located on the web: www.tjhsst.edu/ kdesoto/techlab.

Large x large = 15m x 5m
Short x large = 5m x 15m
Large x short = 15m x 5m
Short x short = 5m x 5m

A set of 225 "control" citizens was given a period of 120 minutes to escape each danger situation. This was performed ten times, and escaped citizens were graphed. This can be found in the appendix.

The l x l area, on average, trapped almost 111 citizens per 120 minutes. This is almost 50 percent.

The s x l area, on average, trapped almost 22 citizens per 120 minutes. This is almost 10 percent.

The l x s area, on average, trapped almost 27 citizens per 120 minutes. This is almost 12 percent.

The s x s area, on average, trapped 0 citizens per 120 minutes, 0 percent.

VI. Conclusion

The conclusion goes here.

VII. Acknowledgment

The authors would like to thank...
Mr. Randy Latimer, Computer Systems Laboratory
Mr. Uri Wilenski, CCL

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.