Human Traffic Flow

New Mexico

Super Computing Challenge

Final Report

April 2, 2008

Team 53

Los Alamos High School

Team Members:

Ben Batha

Gannon Nelson

Will Phillips

Jake Poston

Teacher:

Diane Medford

Project Mentor:

Stephen Guerin

& Others at the Red Fish Group of Santa Fe, NM

Table of Contents

Executive Summary 3

Problem Statement 4

The Solution 4

Results 5

Conclusion 6

What made this project original 7

Appendix A: Code 8

Appendix B: The Map 20

Appendix C: Acknowledgements 21

Appendix D: References 22

Executive Summary

In the modern world, people are no longer spread out sporadically across vast plains, but are concentrated in urban areas. People spend much of their time in crowded spaces, such as apartment buildings, stores like Wal-Mart, amusement parks, etc. Areas of dense populations are often subject to common problems from traffic jams to hallway scuffles. While these areas of concentrated population are unavoidable, it is worthwhile to try to optimize the interiors of these facilities to improve traffic flow. The purpose of our project was to study and model the behaviors of individuals within a crowded system such as a hallway, or even a store. The goal of the model is simple. To allow agents to simulate what would happen when people are evacuating a crowded area in the event of an emergency. We did this through Artificial Intelligence and grid maps for the agents to navigate, similar to what our brain does every day. – This data could then be used to optimize the way architects design stores, sidewalks, buildings, and anything else intended for human travel.

There were many ways to solve this problem. We used the premise of artificial intelligence and grid maps as the tools address this problem. As stated earlier, these data could then be used to optimize the footpaths of urban areas. In the end, these data proved to be quite useful and insightful.

Problem Statement

In recent years the problem of human traffic flow has arisen. Large crowded areas, like airports, and schools, are becoming a more frequent sight. These areas must be able to be safely evacuated in the event of an emergency. This problem will be modeled through net logo with map of Los Alamos High School (Appendix B). Being able to calculate the most efficient and safest way through a building, with a computer could save time, money, and lives. Clearly, modeling evacuations is a worthy task for our team, our world, and certainly a supercomputer.

The Solution

To solve this problem a model, written in Net Logo (Appendix A), was used that was based on Los Alamos High School (Appendix B). This model used the flood-fill algorithm which can be best described as; flood water entering the building through the exits and as the water passes over the floor the distance from the nearest exit is marked. If this were to happen in a real situation, an on looker would see water poor in from every exit and a number placed on every tile. This number signifies the distance to the exit. This algorithm allows the agents in the model to move to the exit with ease since all they must do is follow the numbers on the ground. This algorithm is remarkably efficient because the distances are calculated a single time for the entire map. The agents are O (n) efficient because they must only compare what is on the ground to their current position. This solution can be used to test trial maps so the architect can improve foot traffic efficiency. The architect of any new structure would be able to optimize the layout of their new building with this application.

Results

The results of our study were intriguing and met our goal. As the project went on, more questions arose but, we came closer to our main question, "How do different people navigate crowded areas?" As we completed our model we began to understand how they, and ultimately what they do in these scenarios.

As the project was in an unpolished state firm data gathering mechanisms were not in place at the time of the trials. The best the team could do was to observe the model and compare it to our experiences with navigating crowded areas. So further results are qualitative not quantitive.

The answer to our question, and the results of our studies were quite synonymous with our expectations of them. First things, the people reacted in the ways we programmed them to act. They all had problems with confined areas, and tended to bump into things or other people. We encountered many problems; our flood fill algorithm always had conflicts from filling from other exits of our map, this in turn created a lack of accurate distances. So to compensate we had to limit the size of our area, and in this area the AI's had problems. Even when we made the map smaller, they still had the problems brought on by the algorithm. These problems consist of: Having troubles leaving the rooms that they started in or never going to the closest exit.

Conclusion

In conclusion, the project was a minimal success compared to the predicted capabilities. The project's job was to explain, and map the situations when a problem happens and people need to evacuate. People use many levels of understanding to get out as fast as possible in an emergency. Some people may be handicapped and be blind, others may not be able to handle the stress and rush around, pushing other people, finally there is the person who has one last thread of sanity and calmly evacuates. Also the project would be able to take any building and perform our flood fill algorithm; the algorithm would be able to map this layout and give the agents the ability to navigate it.

As work on the project continued many problems were faced by the team. As the year went on the team had some issues themselves, other activities conflicted. Another major problem that the team hit was that they could not find a mentor; this slowed the speed at which we could code. Towards the end those were fixed, but the biggest issues were within the code itself. The team started in C++ but as the project grew the code got progressively harder, so the team switched to Net Logo. The project hit many problems with this, but succeeded in the end.

In this end of this project the goals were to make it very versatile, and be able to perform it on any building, street, or other areas of foot traffic. The ability for the project to be so versatile was the ultimate point. This ability would allow any architect, or construction worker to optimize any were that humans would lurk about; and in the unusual case of accidents, be able to evacuate.

What made this project original

On this project significant original strides were planned from the beginning. Some of these attempts were discussed earlier. Of these attempts the most significant was the attempt to model the world before the emergency situations. Many areas of the world have far too many people for the design of the space. These areas present a problem for foot traffic not only in emergency situations but in everyday life as well. The project, because of time constraints and team issues, did not have any significant strides in originality. The project provides a solid base for these original strides to be made and implemented.

Appendix A: Code

```
;;initializes the turtles with variables
  turtles-own[
             Blind?
             Normal?
             Rush?
            ]
;;initializes the patches with variables
  patches-own[
             exit?
             floor?
             door?
             exit-distance
             sexit?
             sfloor?
             sdoor?
             sexit-distance
            ]
;;initializes the globals
  globals [
          safe-count
          exitcolor
          wallcolor
          floorcolor
          doorcolor
          classdoorcolor
          roomfloorcolor
         ]
;;sets u the model
  to Setup
    ca
    set safe-count 0
    set exitcolor 14.9
    set wallcolor 1.9
    set floorcolor 9.9
    set doorcolor 65.1
    set classdoorcolor 54.9
    set roomfloorcolor 6.7
    crt 100
    ask turtles[set size 10 fd 5]
    ImportMap
  end
;;Old verision of flood-fill that responds to a single floor color
  to Setup-Fill
```

```
    ask patches[
      set exit?(pcolor = exitcolor or pcolor = 18.3 or pcolor = 16.6 or pcolor = 13.3 or pcolor =
15.7 or pcolor = 17.5 )
      set door?(pcolor = doorcolor)
      set floor?(pcolor = floorcolor)
    ]
    Fill patches with [exit? = true]
    print "Im done"
  end

;;wrapper for flood-fill that passes in the patch and initializes exit-distance
  to Fill[pset]

   if is-patch? pset [set pset patches with[self = pset]]
   ask patches [set exit-distance 9999]
   let n pset
   ask n [set exit-distance 0]
   while [count n > 0][
     let nnext patch-set  [neighbors with [exit-distance = 9999 and floor? or door?]] of n
     ask nnext [set exit-distance min [exit-distance + distance myself] of n]
     ask nnext [set pcolor scale-color red exit-distance 700 0]
     set n nnext
   ]
  end
;;New version of flood-fill to allow the classrooms to be different colors to allow the agents to
;;escape the classrooms properly
    to newSetup-Fill
    ask patches[
      set sexit?(pcolor = classdoorcolor or pcolor = 55.7 or pcolor = 55.1 or pcolor = 55.4 or pcolor
= 57.1 or pcolor = 55.0 )
      set sdoor?(pcolor = doorcolor)
      set sfloor?(pcolor = roomfloorcolor)
    ]
    Fill patches with [exit? = true]
    print "Im done"
  end
;;wrapper for the new flood-fill passes in the patch
  to newFill[pset]

   if is-patch? pset [set pset patches with[self = pset]]
   ask patches [set sexit-distance 9999]
   let n pset
   ask n [set sexit-distance 0]
   while [count n > 0][
     let nnext patch-set  [neighbors with [sexit-distance = 9999 and sfloor? or sdoor?]] of n
     ask nnext [set sexit-distance min [sexit-distance + distance myself] of n]
```

```
     ask nnext [set pcolor scale-color red sexit-distance 700 0]
     set n nnext
   ]
  end
;;imports the map
  to ImportMap
    ;;import-pcolors "school-map42.png"
    import-pcolors "mapfinal.png"
  end
;; Runs the model
  to Go
    Navigate
  end
;;Asks the agents to navigate the map and keeps track of the saved agents
  to Navigate
    ask turtles with [pcolor = exitcolor] [set safe-count safe-count + 1 die]
    ask turtles with [pcolor = wallcolor] [ set heading right 180 fd .3 set heading right 180]
    ask turtles with [pcolor = floorcolor] [ look Move ]

  end


;;This moves the turtles at different rates depending on what they are
  to Move
      ask turtles with [Blind? = true][fd .01]
      ask turtles with [Normal? = true][fd .02]
      ask turtles with [Rush? = true][fd .03]
  end




@#$#@#$#@
;;All of this sets up GUI and is generated by Net Logo
GRAPHICS-WINDOW
205
10
1319
1540
-1
-1
0.43
1
10
1
```

1
1
0
1
1
1
0
1103
0
1498
0
0
1
ticks

CC-WINDOW
5
1554
1328
1649
Command Center
0

BUTTON
121
58
185
91
Setup
Setup
NIL
1
T
OBSERVER
NIL
S
NIL
NIL

BUTTON
29
56
92
89
Go
Go

```
T
1
T
OBSERVER
NIL
G
NIL
NIL

BUTTON
31
140
94
173
Move
Move
NIL
1
T
OBSERVER
NIL
NIL
NIL
NIL

BUTTON
115
135
178
168
Fill
Setup-Fill\nnewSetup-Fill\n\n
NIL
1
T
OBSERVER
NIL
F
NIL
NIL

;;Note the rest of the code is automatically generating by Net Logo to load in libraries, etc.
default
true
0
Polygon -7500403 true true 150 5 40 250 150 205 260 250
```

airplane
true
0
Polygon -7500403 true true 150 0 135 15 120 60 120 105 15 165 15 195 120 180 135 240 105 270 120 285 150 270 180 285 210 270 165 240 180 180 285 195 285 165 180 105 180 60 165 15

arrow
true
0
Polygon -7500403 true true 150 0 0 150 105 150 105 293 195 293 195 150 300 150

box
false
0
Polygon -7500403 true true 150 285 285 225 285 75 150 135
Polygon -7500403 true true 150 135 15 75 150 15 285 75
Polygon -7500403 true true 15 75 15 225 150 285 150 135
Line -16777216 false 150 285 150 135
Line -16777216 false 150 135 15 75
Line -16777216 false 150 135 285 75

bug
true
0
Circle -7500403 true true 96 182 108
Circle -7500403 true true 110 127 80
Circle -7500403 true true 110 75 80
Line -7500403 true 150 100 80 30
Line -7500403 true 150 100 220 30

butterfly
true
0
Polygon -7500403 true true 150 165 209 199 225 225 225 255 195 270 165 255 150 240
Polygon -7500403 true true 150 165 89 198 75 225 75 255 105 270 135 255 150 240
Polygon -7500403 true true 139 148 100 105 55 90 25 90 10 105 10 135 25 180 40 195 85 194 139 163
Polygon -7500403 true true 162 150 200 105 245 90 275 90 290 105 290 135 275 180 260 195 215 195 162 165
Polygon -16777216 true false 150 255 135 225 120 150 135 120 150 105 165 120 180 150 165 225
Circle -16777216 true false 135 90 30
Line -16777216 false 150 105 195 60
Line -16777216 false 150 105 105 60

car
false
0
Polygon -7500403 true true 300 180 279 164 261 144 240 135 226 132 213 106 203 84 185 63 159 50 135 50 75 60 0 150 0 165 0 225 300 225 300 180
Circle -16777216 true false 180 180 90
Circle -16777216 true false 30 180 90
Polygon -16777216 true false 162 80 132 78 134 135 209 135 194 105 189 96 180 89
Circle -7500403 true true 47 195 58
Circle -7500403 true true 195 195 58

circle
false
0
Circle -7500403 true true 0 0 300

circle 2
false
0
Circle -7500403 true true 0 0 300
Circle -16777216 true false 30 30 240

cow
false
0
Polygon -7500403 true true 200 193 197 249 179 249 177 196 166 187 140 189 93 191 78 179 72 211 49 209 48 181 37 149 25 120 25 89 45 72 103 84 179 75 198 76 252 64 272 81 293 103 285 121 255 121 242 118 224 167
Polygon -7500403 true true 73 210 86 251 62 249 48 208
Polygon -7500403 true true 25 114 16 195 9 204 23 213 25 200 39 123

cylinder
false
0
Circle -7500403 true true 0 0 300

dot
false
0
Circle -7500403 true true 90 90 120

face happy
false
0
Circle -7500403 true true 8 8 285
Circle -16777216 true false 60 75 60

Circle -16777216 true false 180 75 60
Polygon -16777216 true false 150 255 90 239 62 213 47 191 67 179 90 203 109 218 150 225 192 218 210 203 227 181 251 194 236 217 212 240

face neutral
false
0
Circle -7500403 true true 8 7 285
Circle -16777216 true false 60 75 60
Circle -16777216 true false 180 75 60
Rectangle -16777216 true false 60 195 240 225

face sad
false
0
Circle -7500403 true true 8 8 285
Circle -16777216 true false 60 75 60
Circle -16777216 true false 180 75 60
Polygon -16777216 true false 150 168 90 184 62 210 47 232 67 244 90 220 109 205 150 198 192 205 210 220 227 242 251 229 236 206 212 183

fish
false
0
Polygon -1 true false 44 131 21 87 15 86 0 120 15 150 0 180 13 214 20 212 45 166
Polygon -1 true false 135 195 119 235 95 218 76 210 46 204 60 165
Polygon -1 true false 75 45 83 77 71 103 86 114 166 78 135 60
Polygon -7500403 true true 30 136 151 77 226 81 280 119 292 146 292 160 287 170 270 195 195 210 151 212 30 166
Circle -16777216 true false 215 106 30

flag
false
0
Rectangle -7500403 true true 60 15 75 300
Polygon -7500403 true true 90 150 270 90 90 30
Line -7500403 true 75 135 90 135
Line -7500403 true 75 45 90 45

flower
false
0
Polygon -10899396 true false 135 120 165 165 180 210 180 240 150 300 165 300 195 240 195 195 165 135
Circle -7500403 true true 85 132 38
Circle -7500403 true true 130 147 38

Circle -7500403 true true 192 85 38
Circle -7500403 true true 85 40 38
Circle -7500403 true true 177 40 38
Circle -7500403 true true 177 132 38
Circle -7500403 true true 70 85 38
Circle -7500403 true true 130 25 38
Circle -7500403 true true 96 51 108
Circle -16777216 true false 113 68 74
Polygon -10899396 true false 189 233 219 188 249 173 279 188 234 218
Polygon -10899396 true false 180 255 150 210 105 210 75 240 135 240

house
false
0
Rectangle -7500403 true true 45 120 255 285
Rectangle -16777216 true false 120 210 180 285
Polygon -7500403 true true 15 120 150 15 285 120
Line -16777216 false 30 120 270 120

leaf
false
0
Polygon -7500403 true true 150 210 135 195 120 210 60 210 30 195 60 180 60 165 15 135 30 120 15 105 40 104 45 90 60 90 90 105 105 120 120 120 105 60 120 60 135 30 150 15 165 30 180 60 195 60 180 120 195 120 210 105 240 90 255 90 263 104 285 105 270 120 285 135 240 165 240 180 270 195 240 210 180 210 165 195
Polygon -7500403 true true 135 195 135 240 120 255 105 255 105 285 135 285 165 240 165 195

line
true
0
Line -7500403 true 150 0 150 300

line half
true
0
Line -7500403 true 150 0 150 150

pentagon
false
0
Polygon -7500403 true true 150 15 15 120 60 285 240 285 285 120

person
false

0
Circle -7500403 true true 110 5 80
Polygon -7500403 true true 105 90 120 195 90 285 105 300 135 300 150 225 165 300 195 300 210 285 180 195 195 90
Rectangle -7500403 true true 127 79 172 94
Polygon -7500403 true true 195 90 240 150 225 180 165 105
Polygon -7500403 true true 105 90 60 150 75 180 135 105

plant
false
0
Rectangle -7500403 true true 135 90 165 300
Polygon -7500403 true true 135 255 90 210 45 195 75 255 135 285
Polygon -7500403 true true 165 255 210 210 255 195 225 255 165 285
Polygon -7500403 true true 135 180 90 135 45 120 75 180 135 210
Polygon -7500403 true true 165 180 165 210 225 180 255 120 210 135
Polygon -7500403 true true 135 105 90 60 45 45 75 105 135 135
Polygon -7500403 true true 165 105 165 135 225 105 255 45 210 60
Polygon -7500403 true true 135 90 120 45 150 15 180 45 165 90

square
false
0
Rectangle -7500403 true true 30 30 270 270

square 2
false
0
Rectangle -7500403 true true 30 30 270 270
Rectangle -16777216 true false 60 60 240 240

star
false
0
Polygon -7500403 true true 151 1 185 108 298 108 207 175 242 282 151 216 59 282 94 175 3 108 116 108

target
false
0
Circle -7500403 true true 0 0 300
Circle -16777216 true false 30 30 240
Circle -7500403 true true 60 60 180
Circle -16777216 true false 90 90 120
Circle -7500403 true true 120 120 60

tree
false
0
Circle -7500403 true true 118 3 94
Rectangle -6459832 true false 120 195 180 300
Circle -7500403 true true 65 21 108
Circle -7500403 true true 116 41 127
Circle -7500403 true true 45 90 120
Circle -7500403 true true 104 74 152

triangle
false
0
Polygon -7500403 true true 150 30 15 255 285 255

triangle 2
false
0
Polygon -7500403 true true 150 30 15 255 285 255
Polygon -16777216 true false 151 99 225 223 75 224

truck
false
0
Rectangle -7500403 true true 4 45 195 187
Polygon -7500403 true true 296 193 296 150 259 134 244 104 208 104 207 194
Rectangle -1 true false 195 60 195 105
Polygon -16777216 true false 238 112 252 141 219 141 218 112
Circle -16777216 true false 234 174 42
Rectangle -7500403 true true 181 185 214 194
Circle -16777216 true false 144 174 42
Circle -16777216 true false 24 174 42
Circle -7500403 false true 24 174 42
Circle -7500403 false true 144 174 42
Circle -7500403 false true 234 174 42

turtle
true
0
Polygon -10899396 true false 215 204 240 233 246 254 228 266 215 252 193 210
Polygon -10899396 true false 195 90 225 75 245 75 260 89 269 108 261 124 240 105 225 105 210 105
Polygon -10899396 true false 105 90 75 75 55 75 40 89 31 108 39 124 60 105 75 105 90 105
Polygon -10899396 true false 132 85 134 64 107 51 108 17 150 2 192 18 192 52 169 65 172 87
Polygon -10899396 true false 85 204 60 233 54 254 72 266 85 252 107 210
Polygon -7500403 true true 119 75 179 75 209 101 224 135 220 225 175 261 128 261 81 224 74

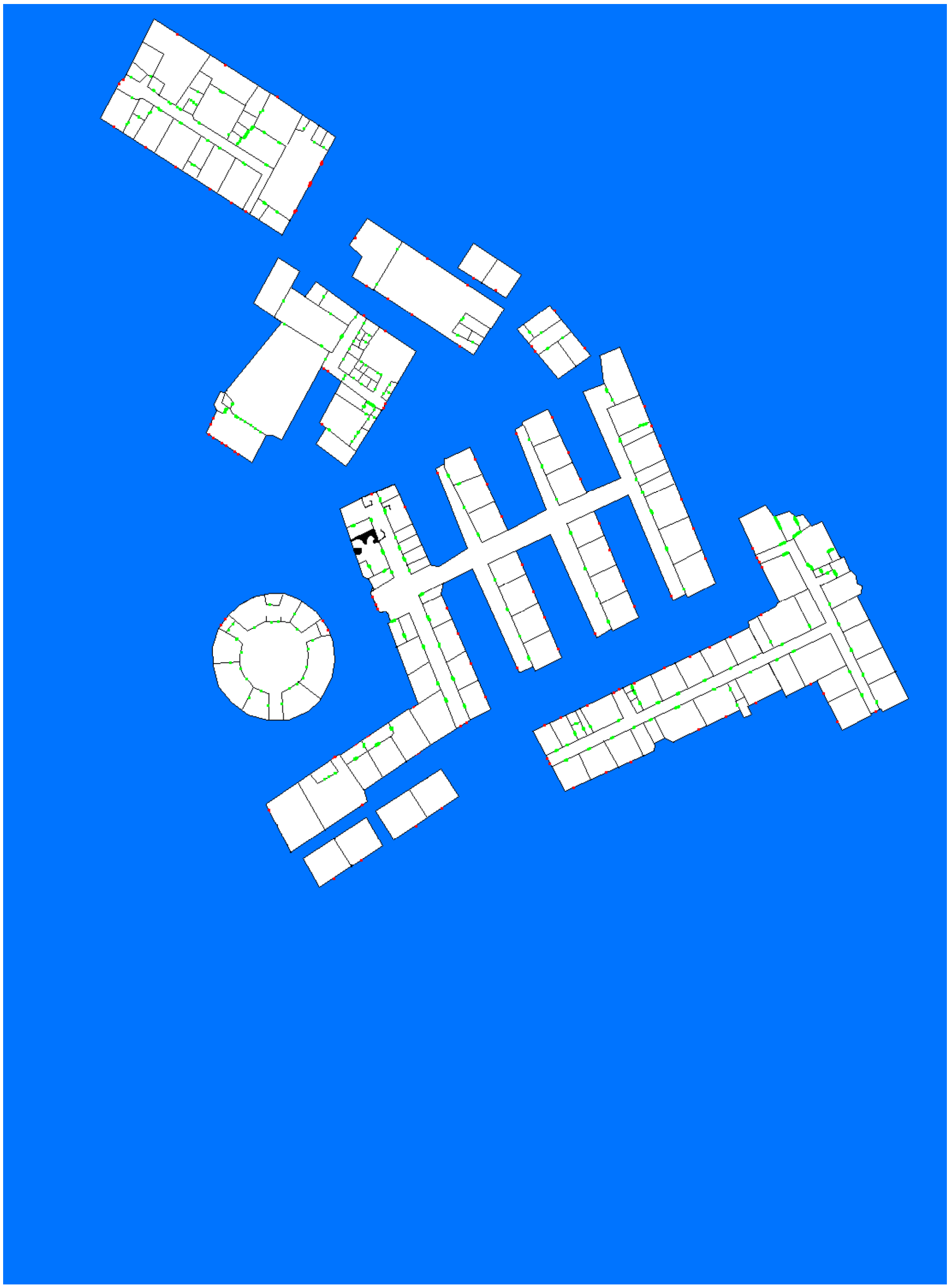
135 88 99

wheel
false
0
Circle -7500403 true true 3 3 294
Circle -16777216 true false 30 30 240
Line -7500403 true 150 285 150 15
Line -7500403 true 15 150 285 150
Circle -7500403 true true 120 120 60
Line -7500403 true 216 40 79 269
Line -7500403 true 40 84 269 221
Line -7500403 true 40 216 269 79
Line -7500403 true 84 40 221 269

x
false
0
Polygon -7500403 true true 270 75 225 30 30 225 75 270
Polygon -7500403 true true 30 75 75 30 270 225 225 270

@#$#@#$#@
NetLogo 4.0.2
@#$#@#$#@
@#$#@#$#@
@#$#@#$#@
@#$#@#$#@
@#$#@#$#@
default
0.0
-0.2 0 0.0 1.0
0.0 1 1.0 0.0
0.2 0 0.0 1.0
link direction
true
0
Line -7500403 true 150 150 90 180
Line -7500403 true 150 150 210 180

Appendix B: The Map



Appendix C: Acknowledgements

On the behalf of Will, Jake, Ben and Gannon we would like to thank Diane Medford and Bob Robey for providing guidance and making sure that we get to Super Computing Challenge events in a prepared timely fashion. We would also like to thank our parents for providing the nagging force that let actually complete the project. Finally we would like to thank the Redfish Group of Santa Fe for their expert knowledge of Net Logo and evacuation models. Without the Redfish Group this project would never be at the stage it is today.

Appendix D: References

Guerin, S. (2004). Peeking into the black-box: Some art and science to visualizing agent-based models. *Proceedings of the 2004 Winter Simulation Conference R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.*

Gambhir, M., Guerin, S., Kauffman, S., Kunkle, D. (2004) Steps toward a possible theory of organization. In: *Proceedings of International Conference on Complex Systems 2004. Boston, MA.*

Guerin, S. and Kunkle, D. (2004) Emergence of constraint in self-organizing systems. *Journal of Nonlinear Dynamics, Psychology, and Life Sciences, Vol. 8, No. 2, April, 2004.*

Gambhir, M., Guerin, S., Kunkle, D., and Harris, R. (2004) Measures Of Work in Artificial Life.

Thorp, J., Guerin, S., Wimberly, F., Rossbach, M., Densmore, O., Agar, M., Roberts, D. (2006) Santa Fe On Fire: agent-based modeling of wildfire evacuation dynamcis . in Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects, ANL/DIS-06-7, ISBN 0-9679168-7-9, Sallach, D.L., C.M. Macal, and M.J. North (editors), co-sponsored by Argonne National Laboratory and The University of Chicago, September 21-23.

"What is Net Logo?" 5 December 2007. Net Logo. 2 April 2008 <http://ccl.northwestern.edu/netlogo/docs/>.

Kuligowski, E. D.; Peacock, R. D. Review of Building Evacuation Models. July 2005. 2 April 2008 <http://www.fire.nist.gov/bfrlpubs/fire05/art107.html>.