# Integration of Agent-Based Simulation and GIS: Applied to Segregation

Andrew Crooks

# Outline

- Initial Aims
- Model detail
- Preliminary Outputs
- Summary

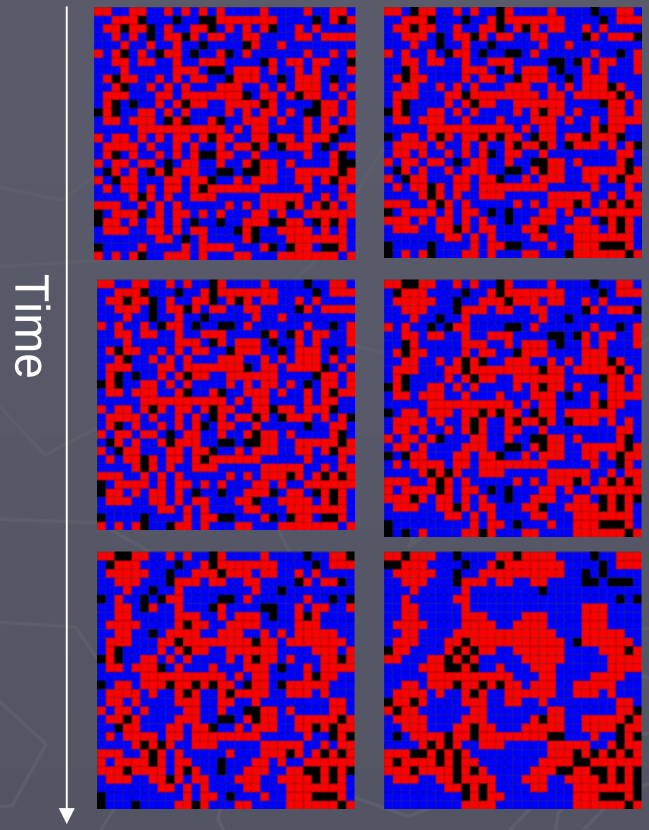
# Initial Model Aims

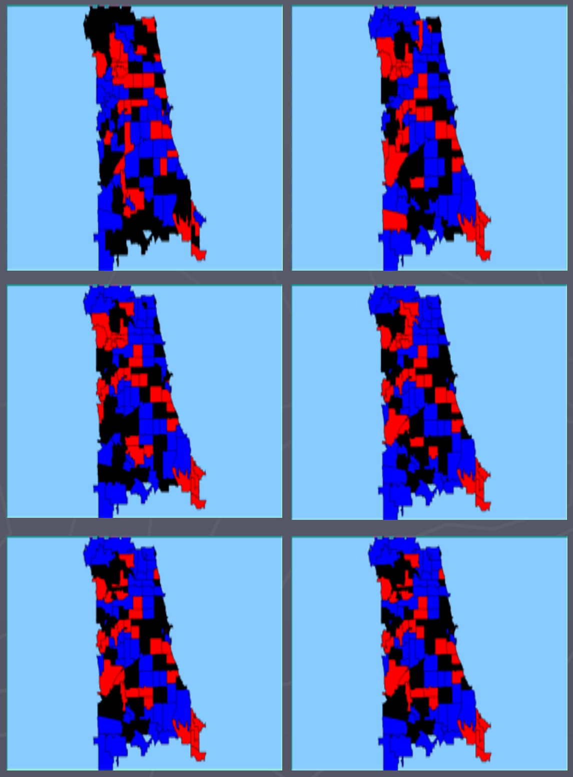
- Incorporate scenario testing, a tool to think with.
- Complexity Theory (micro interaction to macro policy=> emergent patterns)
- Make the models geographic, as past ABM have been criticised for not being spatial.
- The ability to use actual data and areas for initial starting conditions.
- Develop a simple model that can easily be extended into examining different types of segregation.
  - Segregation is a good example of emergent phenomena.

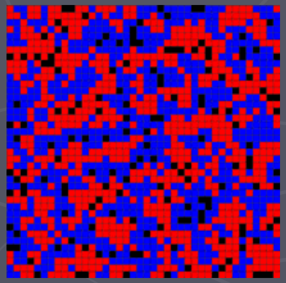
# Examples of Segregation Models

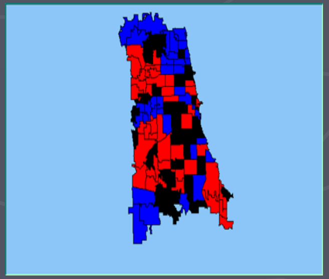
- Agents move if the number of surrounding cells is greater than its preference

- Agents move to their nearest empty cell.





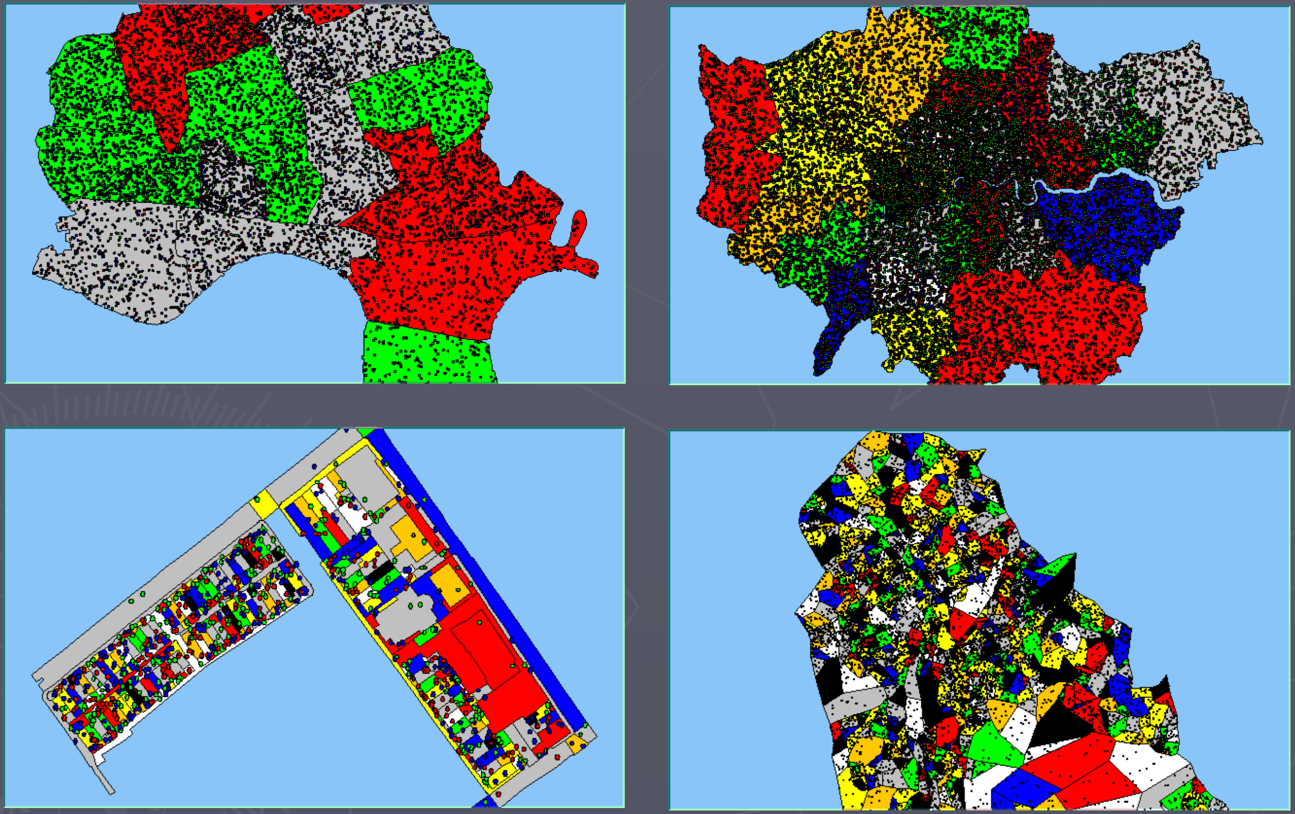


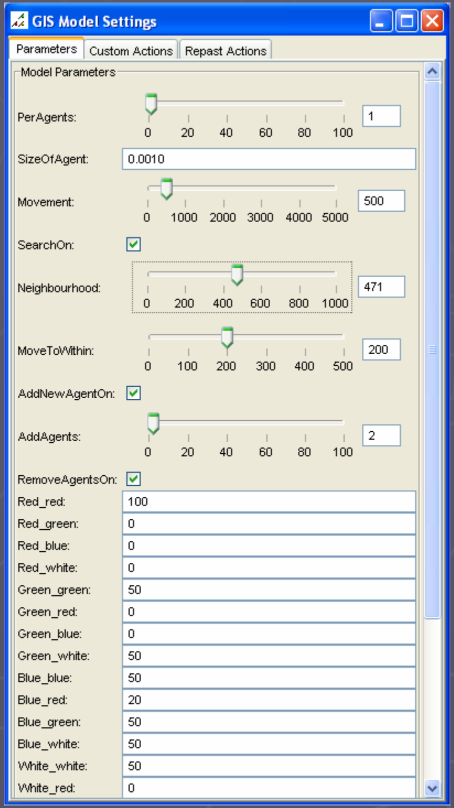


# Basic Model

- Built a very generic model structure (spatial)
  - Cells and agents can interact.
  - Applied at different scales (Boroughs, wards, OA or OS Mastermap).
  - Easy to adapt to other situations.
  - Different .shp files can be loaded (via GUI).
    - To see if the same interactions/rules apply in different areas.
- Allows user interaction via GUI e.g. parameter setting and .shp choosing.

# Different Scales





Model Parameters

PerAgents: 1

SizeOfAgent: 0.0010

Movement: 500

SearchOn: ☑

Neighbourhood: 471

MoveToWithin: 200

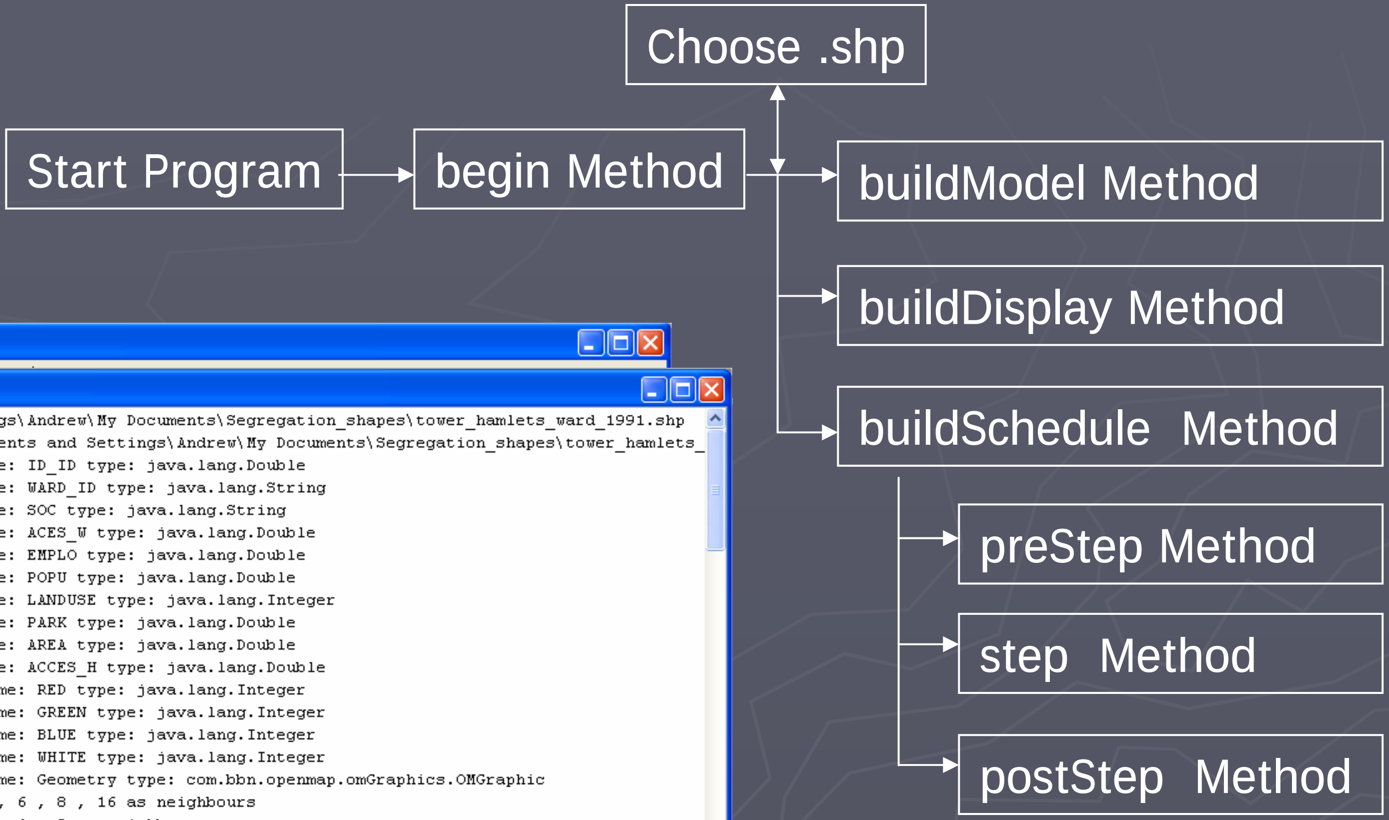
AddNewAgentOn: ☑

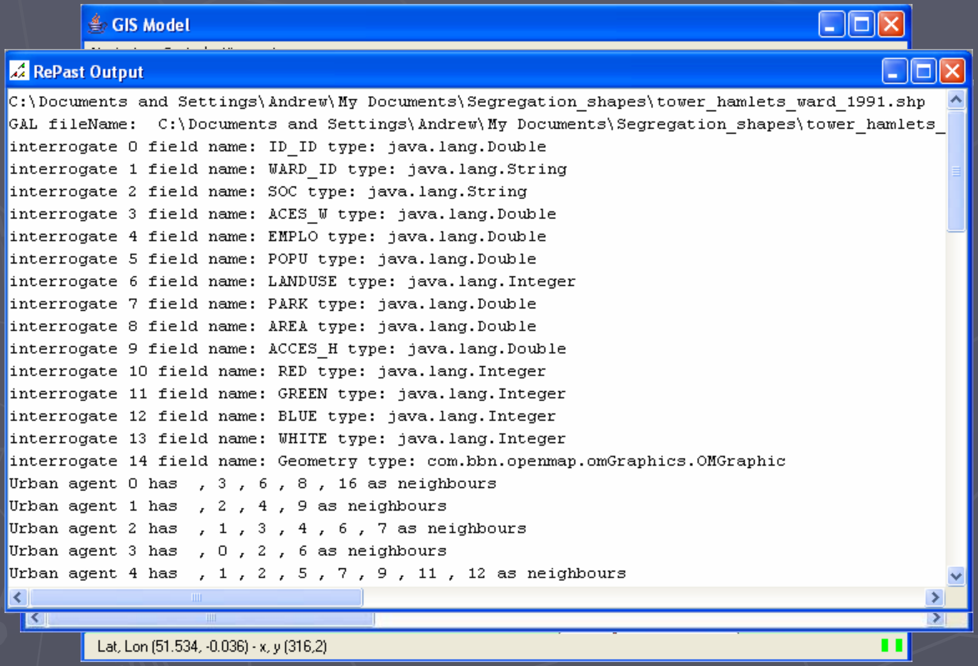
AddAgents: 2

RemoveAgentsOn: ☑

| | |
|---|---|
| Red_red: | 100 |
| Red_green: | 0 |
| Red_blue: | 0 |
| Red_white: | 0 |
| Green_green: | 50 |
| Green_red: | 0 |
| Green_blue: | 0 |
| Green_white: | 50 |
| Blue_blue: | 50 |
| Blue_red: | 20 |
| Blue_green: | 50 |
| Blue_white: | 50 |
| White_white: | 50 |
| White_red: | 0 |

# SegGIS Basic Model methods



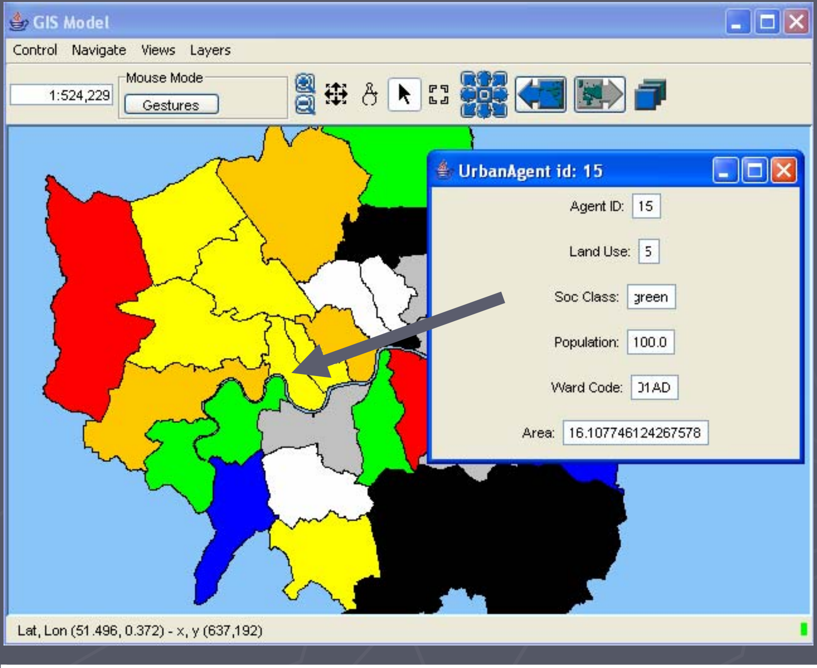


```
C:\Documents and Settings\Andrew\My Documents\Segregation_shapes\tower_hamlets_ward_1991.shp
GAL fileName:  C:\Documents and Settings\Andrew\My Documents\Segregation_shapes\tower_hamlets_
interrogate 0 field name: ID_ID type: java.lang.Double
interrogate 1 field name: WARD_ID type: java.lang.String
interrogate 2 field name: SOC type: java.lang.String
interrogate 3 field name: ACES_W type: java.lang.Double
interrogate 4 field name: EMPLO type: java.lang.Double
interrogate 5 field name: POPU type: java.lang.Double
interrogate 6 field name: LANDUSE type: java.lang.Integer
interrogate 7 field name: PARK type: java.lang.Double
interrogate 8 field name: AREA type: java.lang.Double
interrogate 9 field name: ACCES_H type: java.lang.Double
interrogate 10 field name: RED type: java.lang.Integer
interrogate 11 field name: GREEN type: java.lang.Integer
interrogate 12 field name: BLUE type: java.lang.Integer
interrogate 13 field name: WHITE type: java.lang.Integer
interrogate 14 field name: Geometry type: com.bbn.openmap.omGraphics.OMGraphic
Urban agent 0 has  , 3 , 6 , 8 , 16 as neighbours
Urban agent 1 has  , 2 , 4 , 9 as neighbours
Urban agent 2 has  , 1 , 3 , 4 , 6 , 7 as neighbours
Urban agent 3 has  , 0 , 2 , 6 as neighbours
Urban agent 4 has  , 1 , 2 , 5 , 7 , 9 , 11 , 12 as neighbours
```

Lat, Lon (51.534, -0.036) - x, y (316,2)

# Urban Environments

- The GIS layer. Contains information from the .shp.
- Used to calculate population density, contains the residents.
- Attributes are changeable: social class and landuse depending on the type of residents within.



Agent ID: 15

Land Use: 5

Soc Class: green

Population: 100.0

Ward Code: 01AD

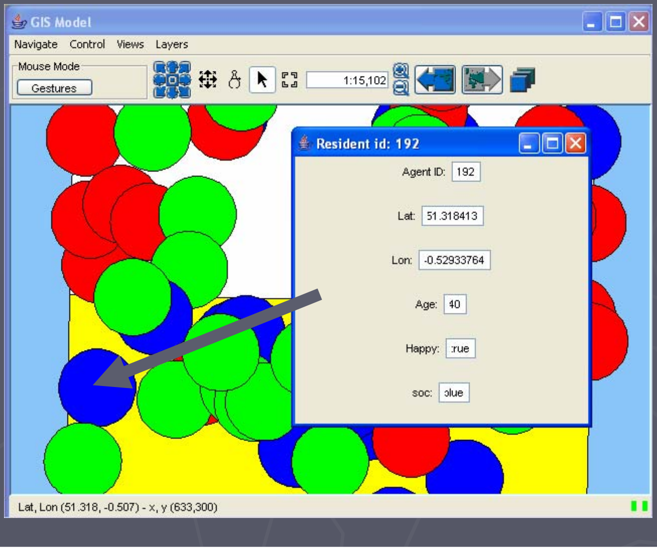
Area: 16.107746124267578

Lat, Lon (51.496, 0.372) - x, y (637,192)

| | Resident Agent social class | Urban agent | Colour |
|---|---|---|---|
| 1 | Green % highest | Green | Green |
| 2 | Red % highest | Red | Red |
| 3 | Blue % highest | Blue | Gray |
| 4 | White % highest | White | Blue |
| 5 | Green & Red % equal | Mixed | Yellow |
| 6 | Green & Blue equal | Mixed | Orange |
| 7 | R & B \|\| W & B \|\| R & W \|\| G & W | Mixed | White |
| 8 | All other combinations | Mixed | L Blue |
| 9 | Empty (no residents) | Empty | Black |

# Residential Agents

- Residential agents attributes:
  - Age.
  - Social class (4 types).
  - Happiness (Utility Function).
- Happiness: determined by Neighbourhood size and preferences to different types.



Agent ID: 192

Lat: 51.318413

Lon: -0.52933764

Age: 40

Happy: :rue

soc: olue

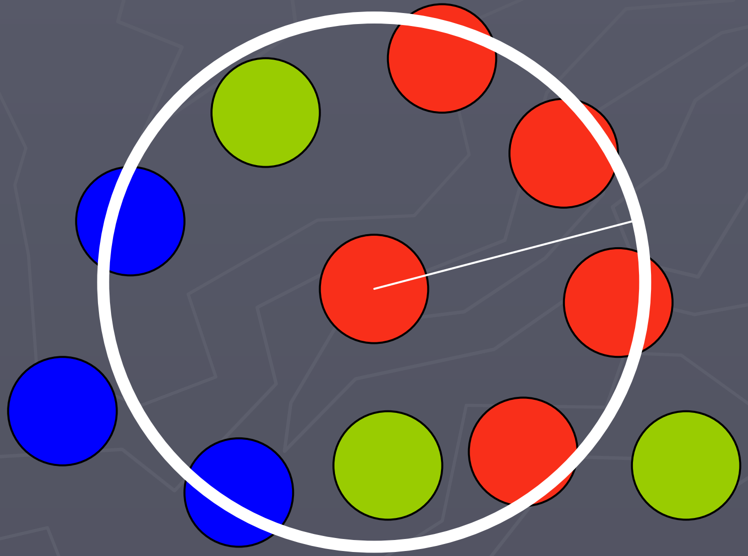
Lat, Lon (51.318, -0.507) - x, y (633,300)

| For Blue agents | Blue agent preferences |
|---|---|
| Blue with Blue | More than 50% Blue |
| Blue with Red | Less than 20% |
| Blue with Green | Less than 50% |
| Blue with White | Less than 50% |

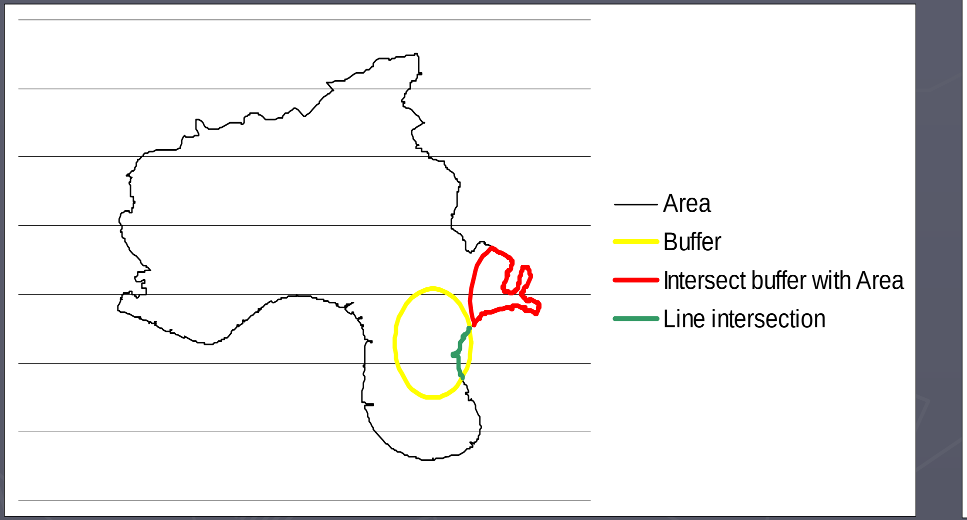
# Neighbourhoods

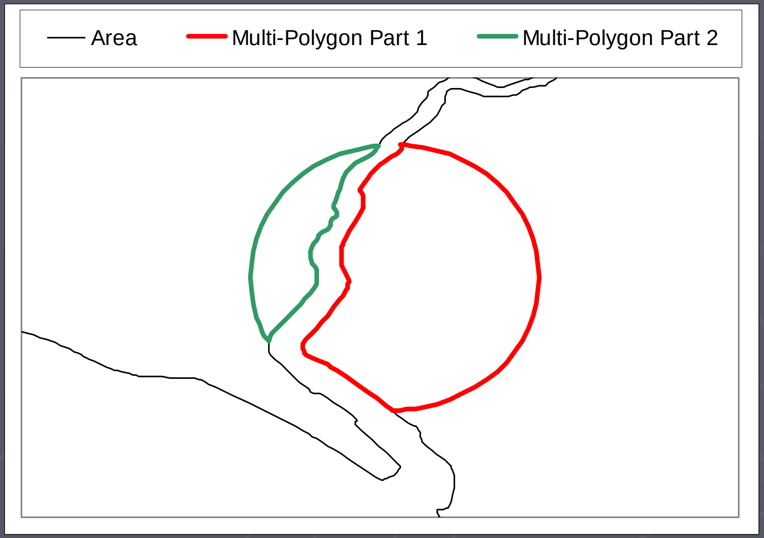
- Residents calculate which neighbours are within a specific distance.

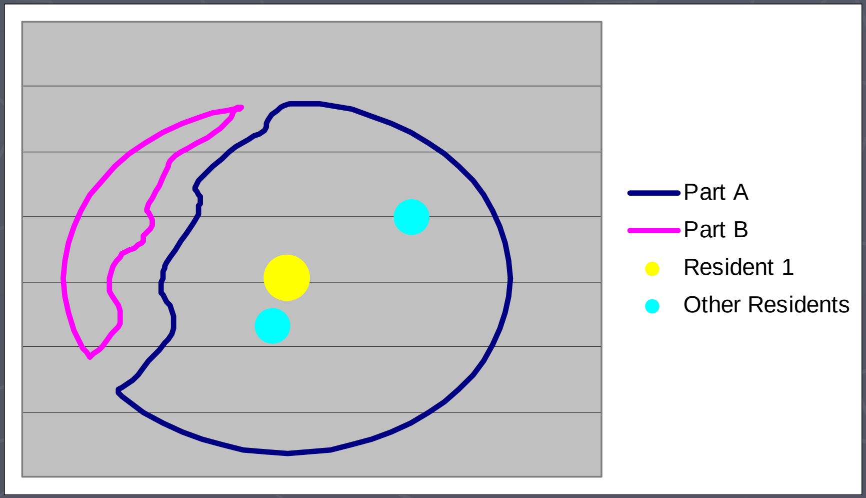
| | No. of agents |
|---|---|
| Red Agents | 4 |
| Blue Agents | 2 |
| Green Agents | 2 |



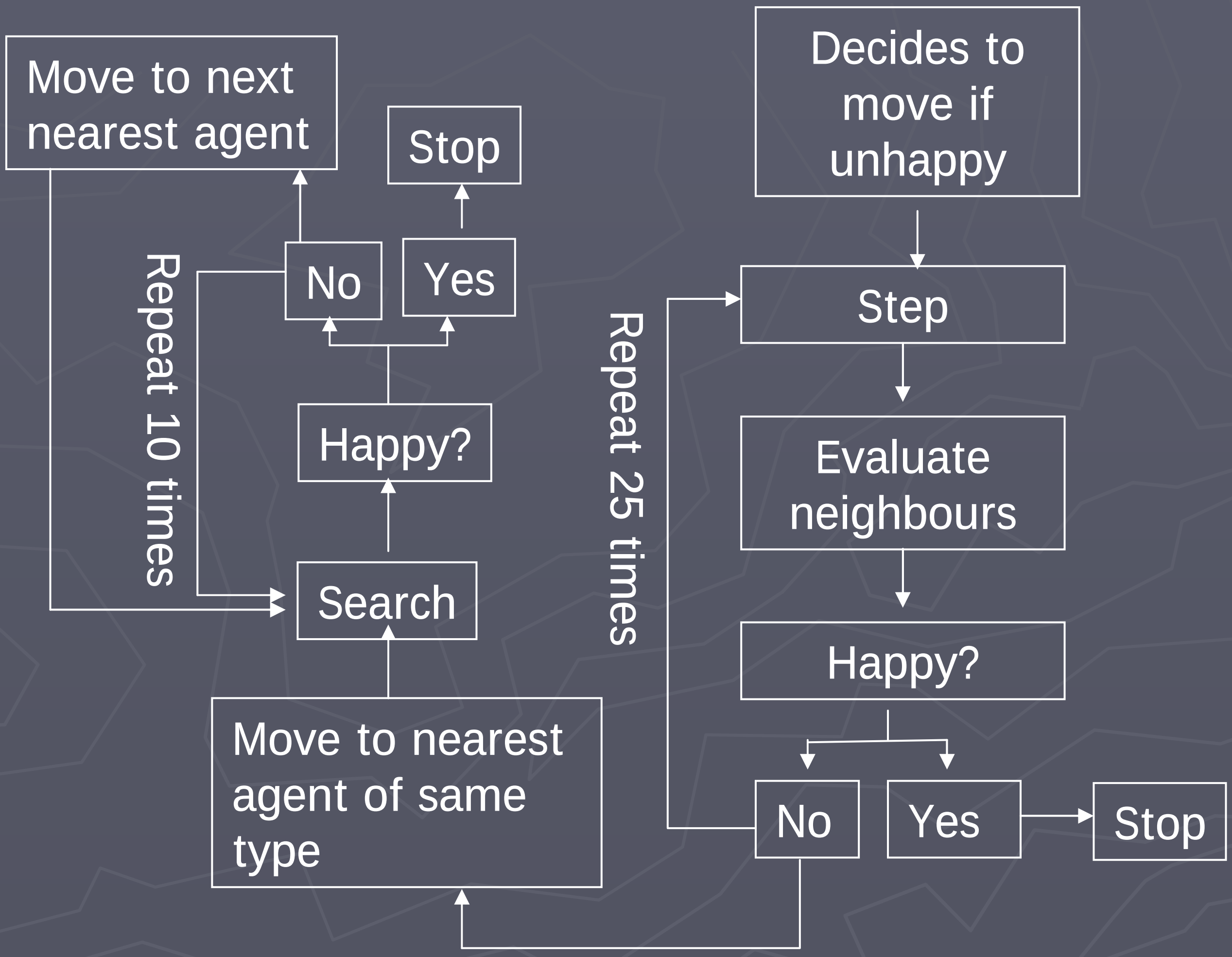
# Neighbourhoods Complications: Geographical Features

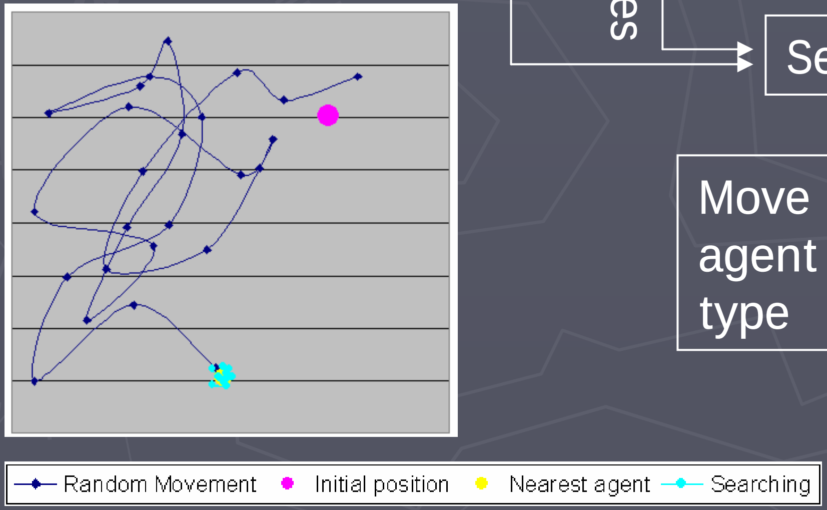






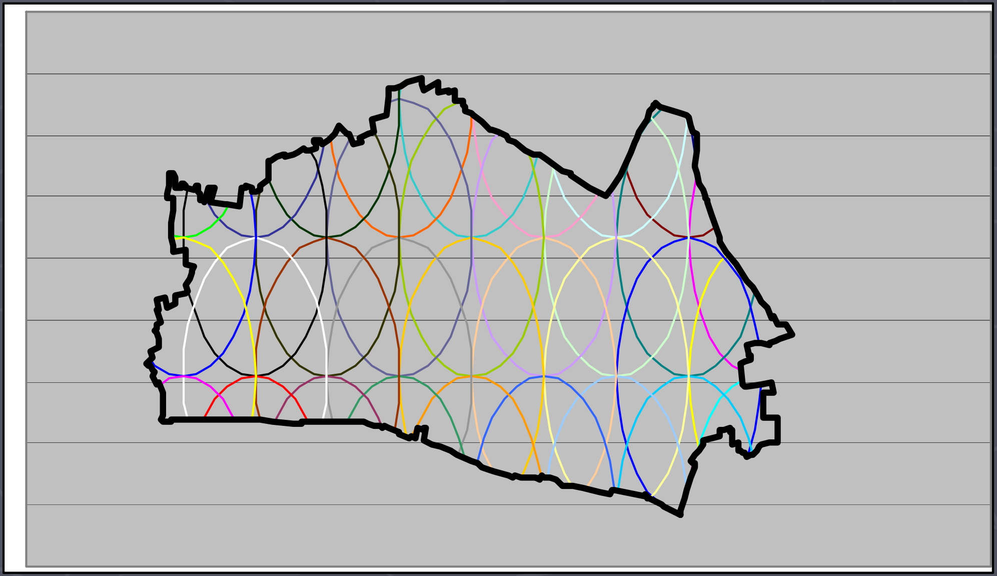
# Individuals Searching for a Suitable Location


If no more agents, remove from area
Move to next nearest agent
Stop
No
Yes
Repeat 10 times
Happy?
Search
Move to nearest agent of same type
Decides to move if unhappy
Step
Evaluate neighbours
Happy?
Repeat 25 times
No
Yes
Stop
After 25 searches

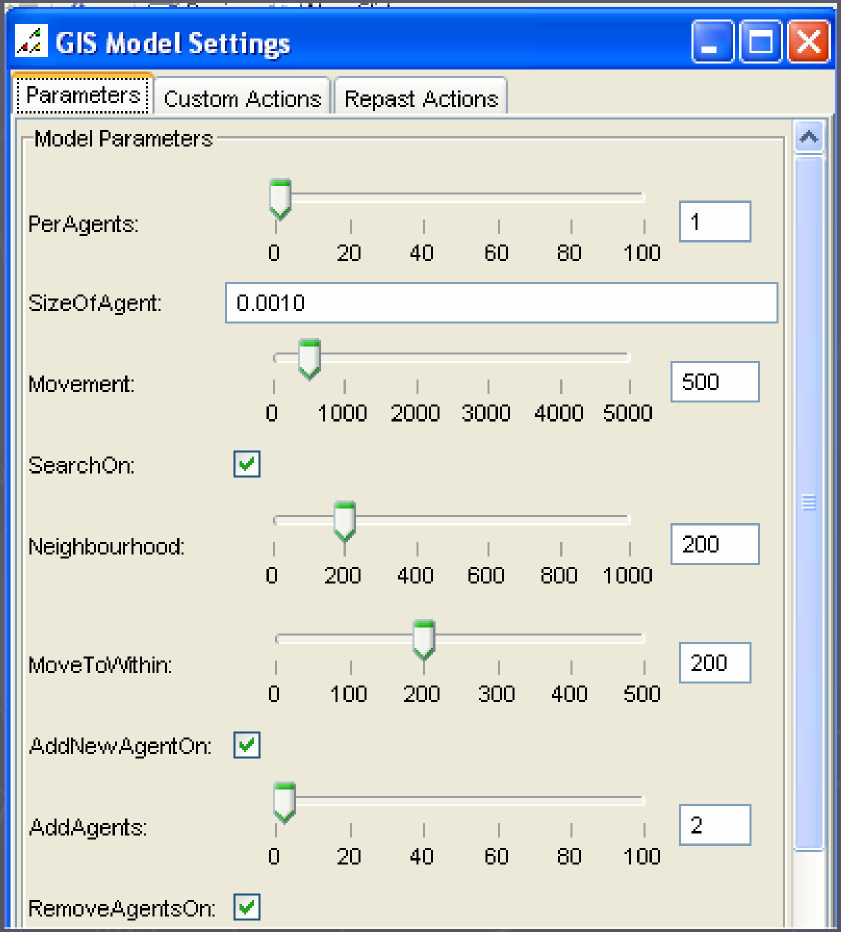


# Neighbourhood searching?

- Series of overlapping polygons –acting as neighbourhoods.
- Each contains summary attributes of agents within.
- Agents choose suitable areas and search until satisfied rather than searching around individual agents.



# Parameter Settings



Model Parameters

PerAgents: 1

SizeOfAgent: 0.0010

Movement: 500

SearchOn: ☑

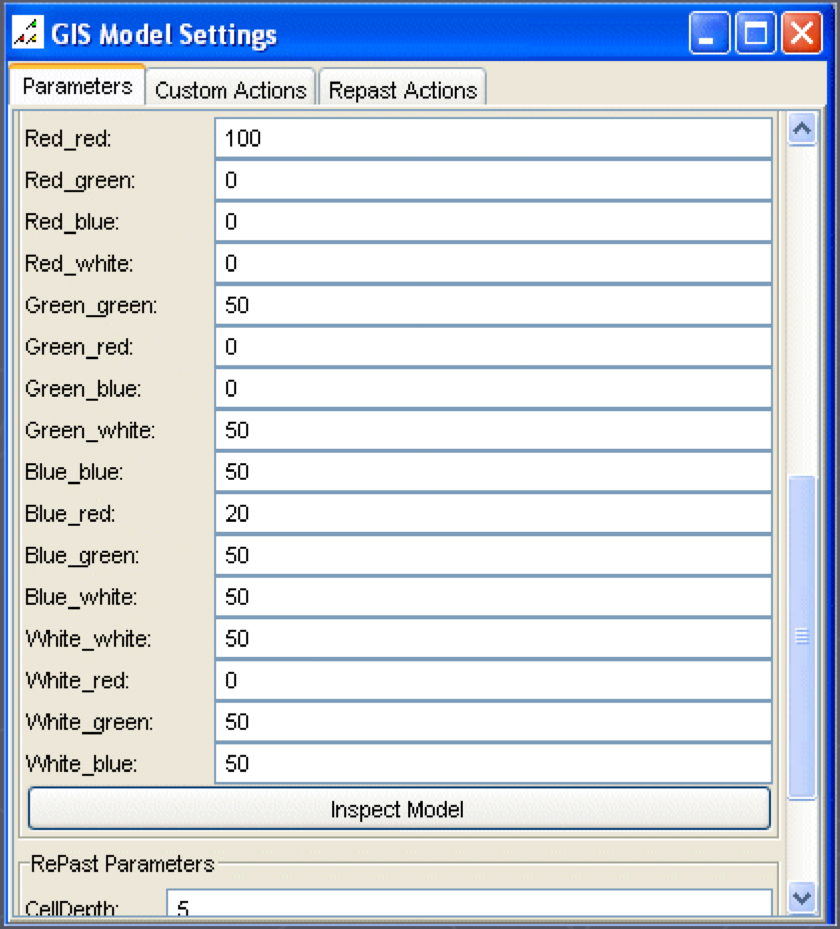
Neighbourhood: 200

MoveToWithin: 200

AddNewAgentOn: ☑

AddAgents: 2

RemoveAgentsOn: ☑

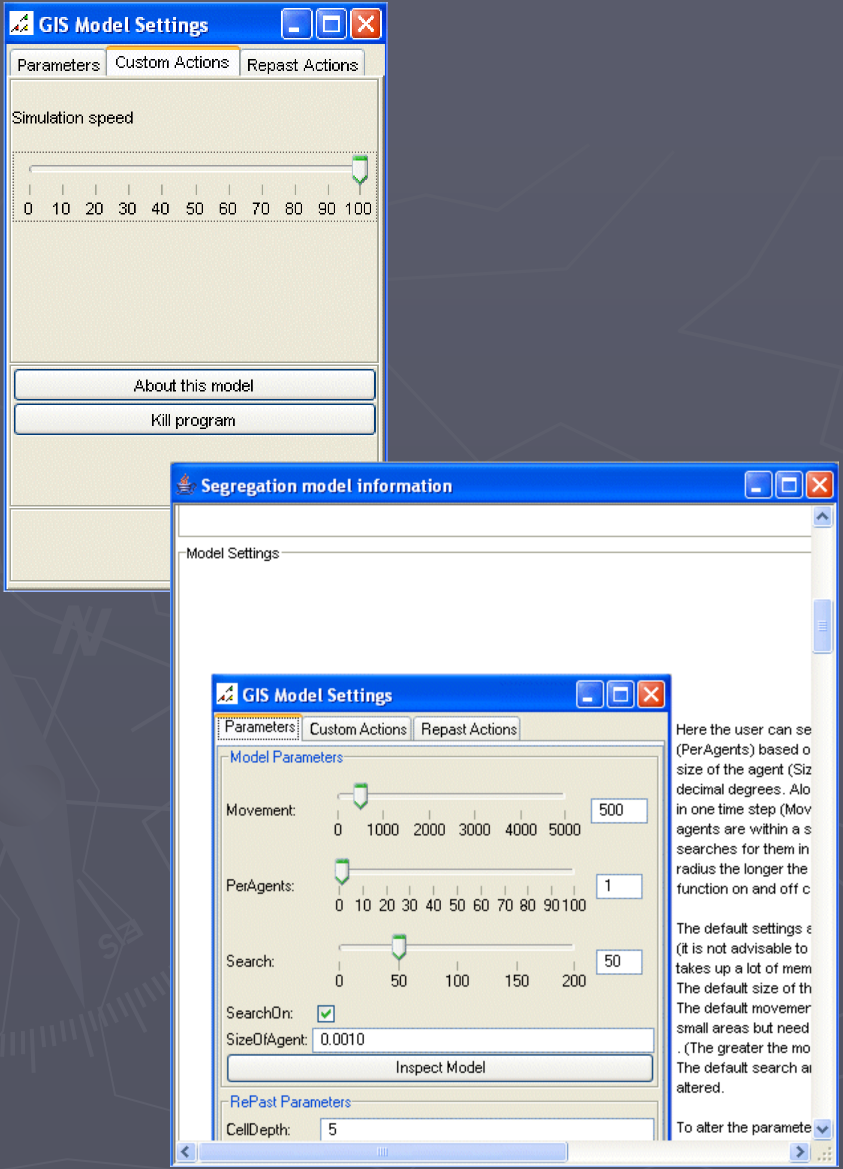


| | |
|---|---|
| Red_red: | 100 |
| Red_green: | 0 |
| Red_blue: | 0 |
| Red_white: | 0 |
| Green_green: | 50 |
| Green_red: | 0 |
| Green_blue: | 0 |
| Green_white: | 50 |
| Blue_blue: | 50 |
| Blue_red: | 20 |
| Blue_green: | 50 |
| Blue_white: | 50 |
| White_white: | 50 |
| White_red: | 0 |
| White_green: | 50 |
| White_blue: | 50 |

RePast Parameters

CellDepth: 5

# Custom Actions



Simulation speed

0 10 20 30 40 50 60 70 80 90 100

Model Settings

Model Parameters

Movement: 500 (0 1000 2000 3000 4000 5000)

PerAgents: 1 (0 10 20 30 40 50 60 70 80 90 100)

Search: 50 (0 50 100 150 200)

SearchOn: ☑

SizeOfAgent: 0.0010

RePast Parameters

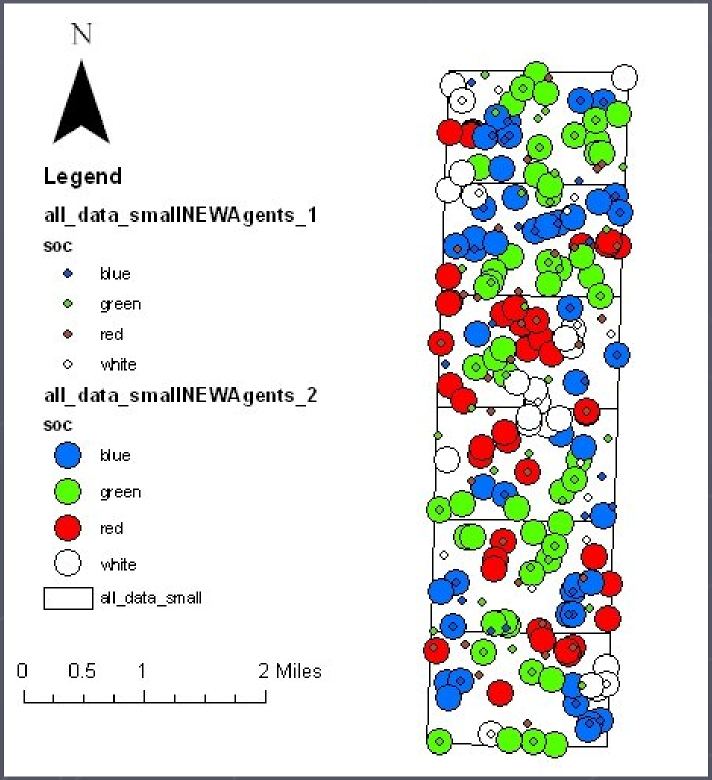
CellDepth: 5

Here the user can se
(PerAgents) based o
size of the agent (Siz
decimal degrees. Alo
in one time step (Mov
agents are within a s
searches for them in
radius the longer the
function on and off c

The default settings a
(it is not advisable to
takes up a lot of mem
The default size of th
The default movemer
small areas but need
. (The greater the mo
The default search ai
altered.

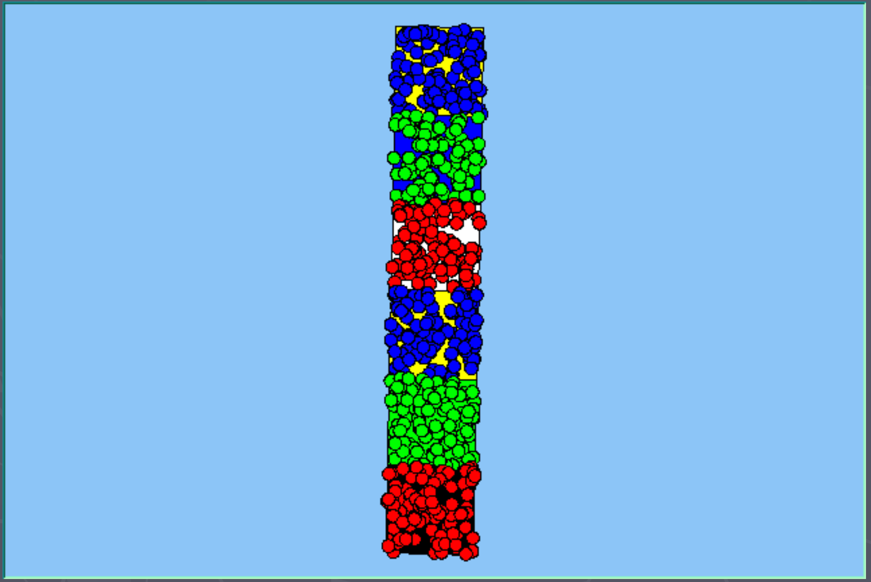
To alter the paramete

# Model Outputs

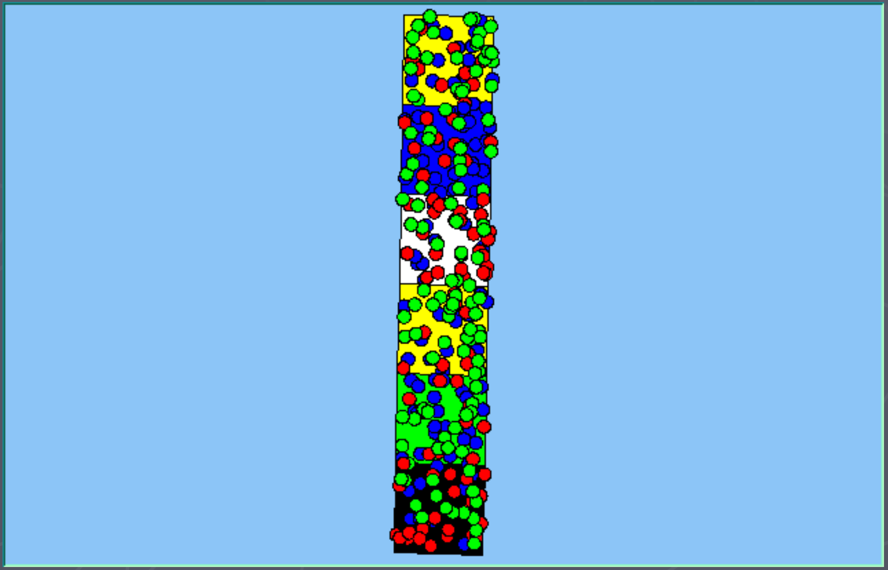


- Changes to .shp
- New Agent .shp (time stamped)
- Aggregate data to .txt

# Examples: Test Case, Random Placement of Agents

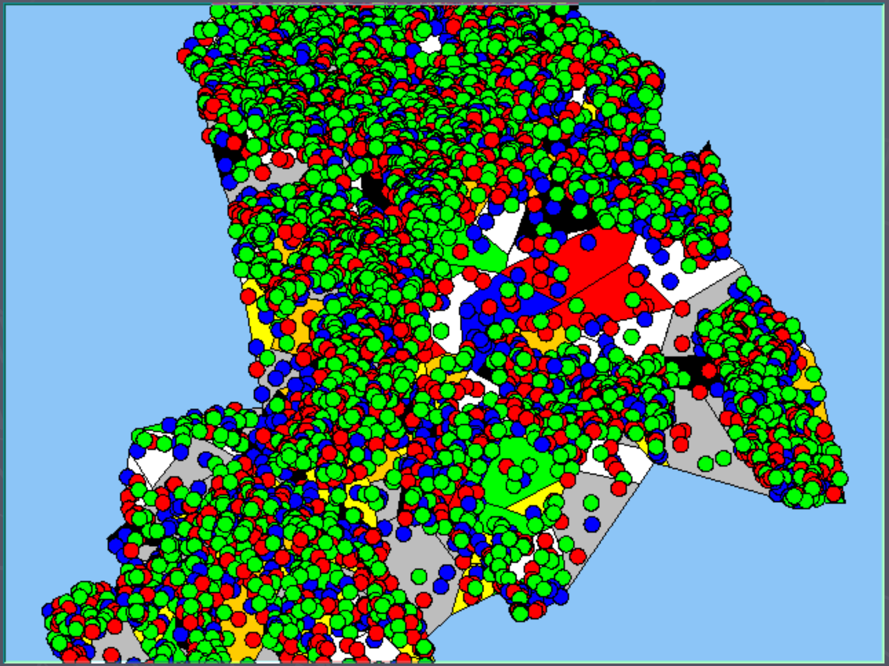


Area at start is of same type.

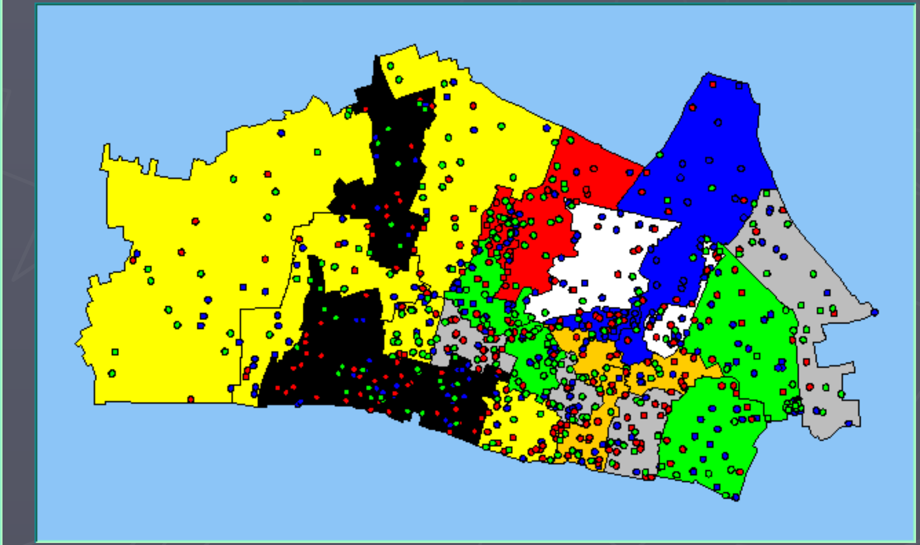


Area at start is of mixed  type.

# Examples: OA and Ward



Croydon OA



City of London Wards

# Conclusion/ What next?

- Presented a simple model integrating GIS & ABM using certain functions from Repast.
- Model Rules are easily altered and applied to different areas.
  - Patterns emerge based on individual interaction.
- To Do:
  - See if the basic rules can be applied to Residential segregation.
  - Compare searching mechanisms.
  - Carry out a series of batch runs.