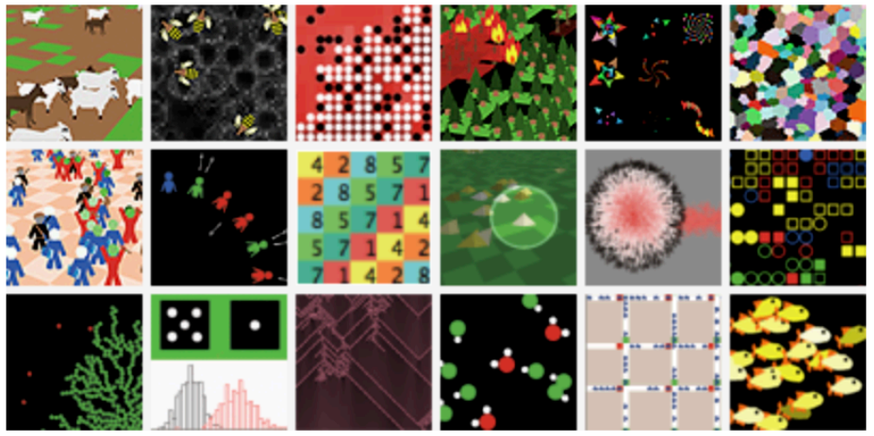
Create a spatial ABM

- Load data (image, vectors, csv) (tips to make it faster with speed slider, and pixel size and making smaller input file via smoothing)
- Make the map: Display, label features (draw features, apply color, label)
- Create agents based on map attributes
- Agent-environment interaction (e.g. movement constrained by map properties - road following, move within state,)
- Monitors, plots
- Behavior-space
- Saving/Sharing model

Where to find info about Netlogo Extensions

NetLogo

What is NetLogo?



NetLogo is a programmable modeling environment for simulating
continuous development ever since at the Center for Connected L

NetLogo is particularly well suited for modeling complex systems
“agents” all operating independently. This makes it possible to exp
patterns that emerge from their interaction.

NetLogo lets students open simulations and “play” with them, exp
enables students, teachers and curriculum developers to create th
enough to serve as a powerful tool for researchers in many fields.

NetLogo has extensive documentation and tutorials. It also come
and modified. These simulations address content areas in the nat
mathematics and computer science, and economics and social ps
are under development.

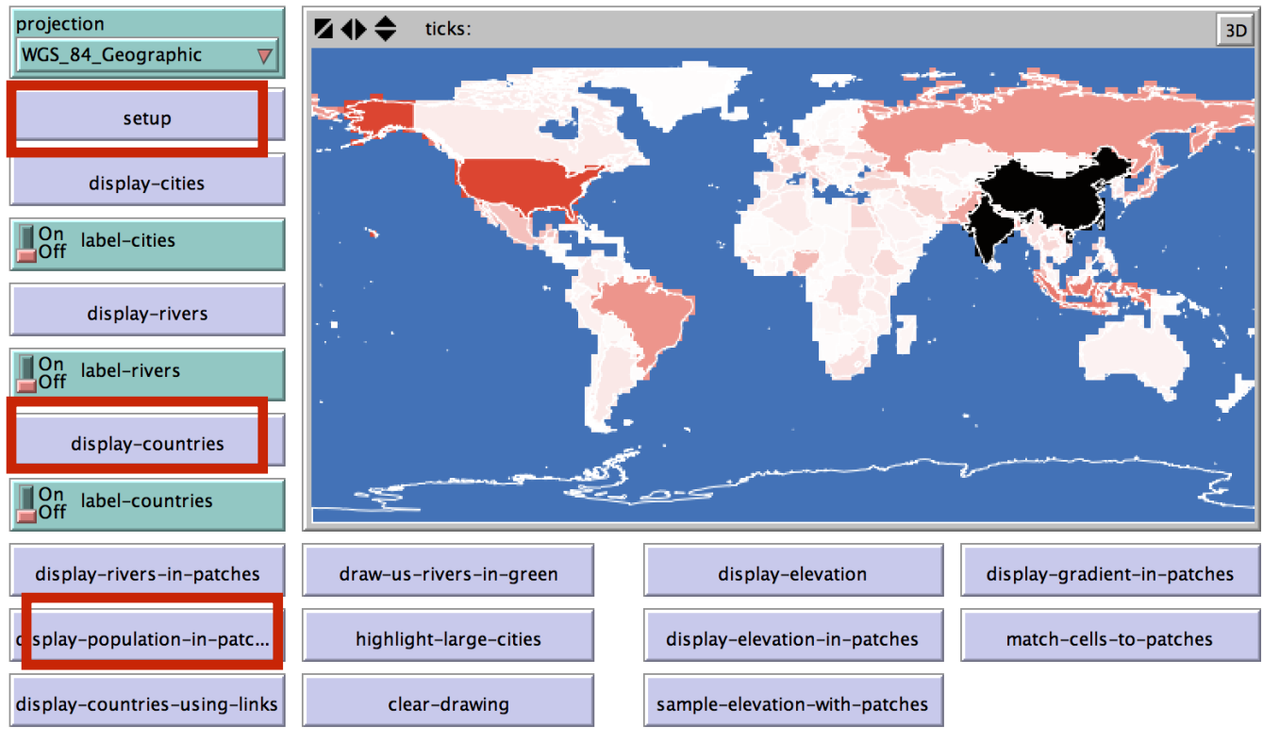
NetLogo is the next generation of the series of multi-agent modeli
Machine, so it works on all major platforms (Mac, Windows, Linux

Features

Netlogo and GIS

A great reference for working with GIS data in Netlogo is the model in the model library:

- **Open** Netlogo>Models Library>Code Examples>Extensions Examples>gis>**GIS General Examples**
  - Click **setup** (to load the GIS)
  - Click **display-countries**
  - **display-population-in-patches**



- **Go to code tab** (notice extensions at top, setup commands, good commenting throughout).

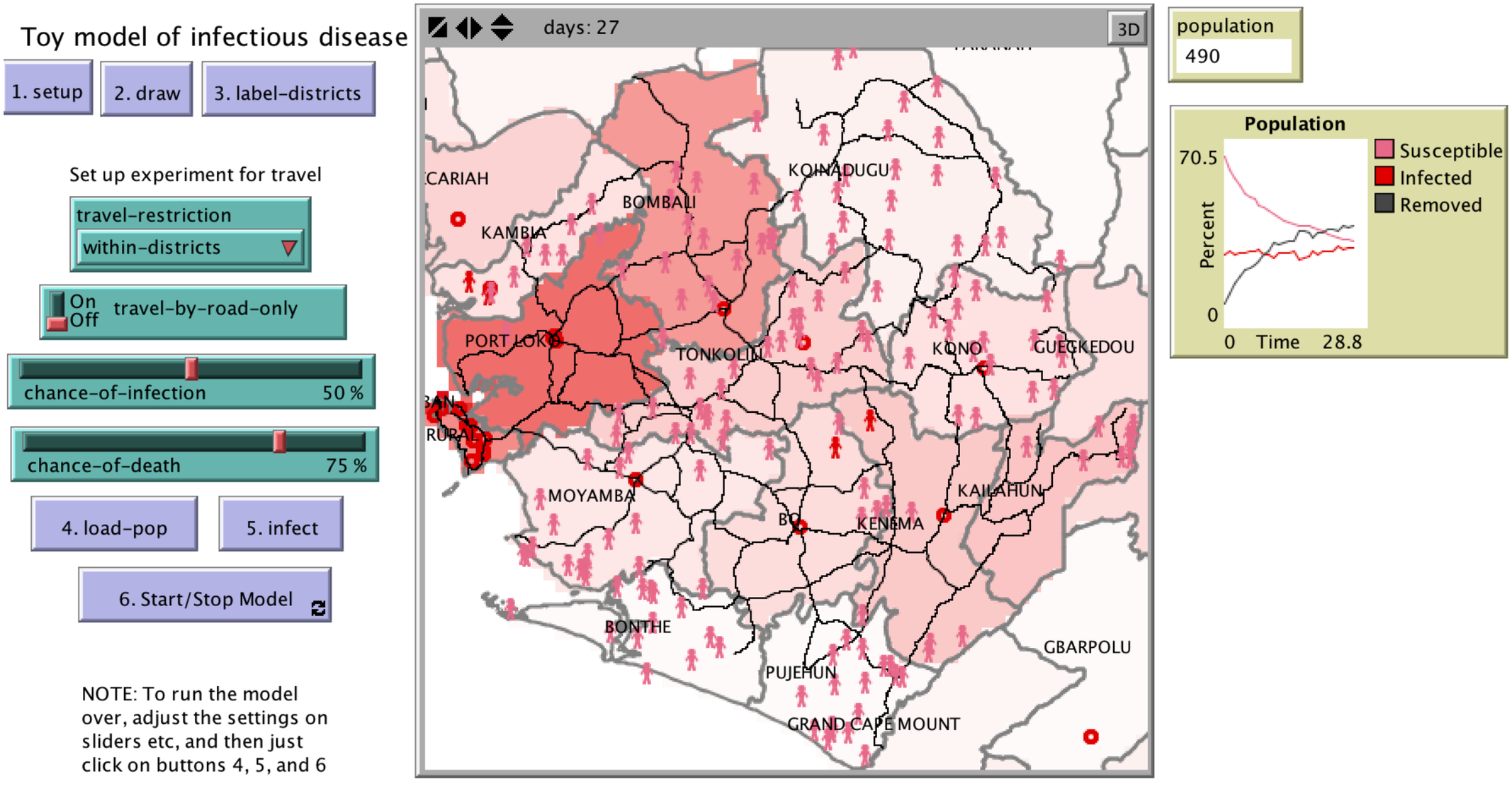
Note: I refer to this model and copy and adapt code sections for my own models all the time. I often have this model open in one instance of Netlogo while I work on my own model in another Netlogo instance.

Let’s make our own model

Coding goals and objectives

- Our model will **use the GIS and CSV extensions** to load shape files and csv data
- **Load GIS data** of country and district boundaries, roads, and treatment centers
- **Color patches based on attribute** in district boundary polygons shape file
- **Label** districts
- Create **agents in proportion** to population using a CSV
- Have infected **agents spread disease** upon contact
- Enable **movement of agents** along roads
- **Restrict movement** of agents to districts they were created in
- **Plot information** and metrics

An expanded version of the model with plots and ability to adjust input parameters and test scenarios related to travel restrictions



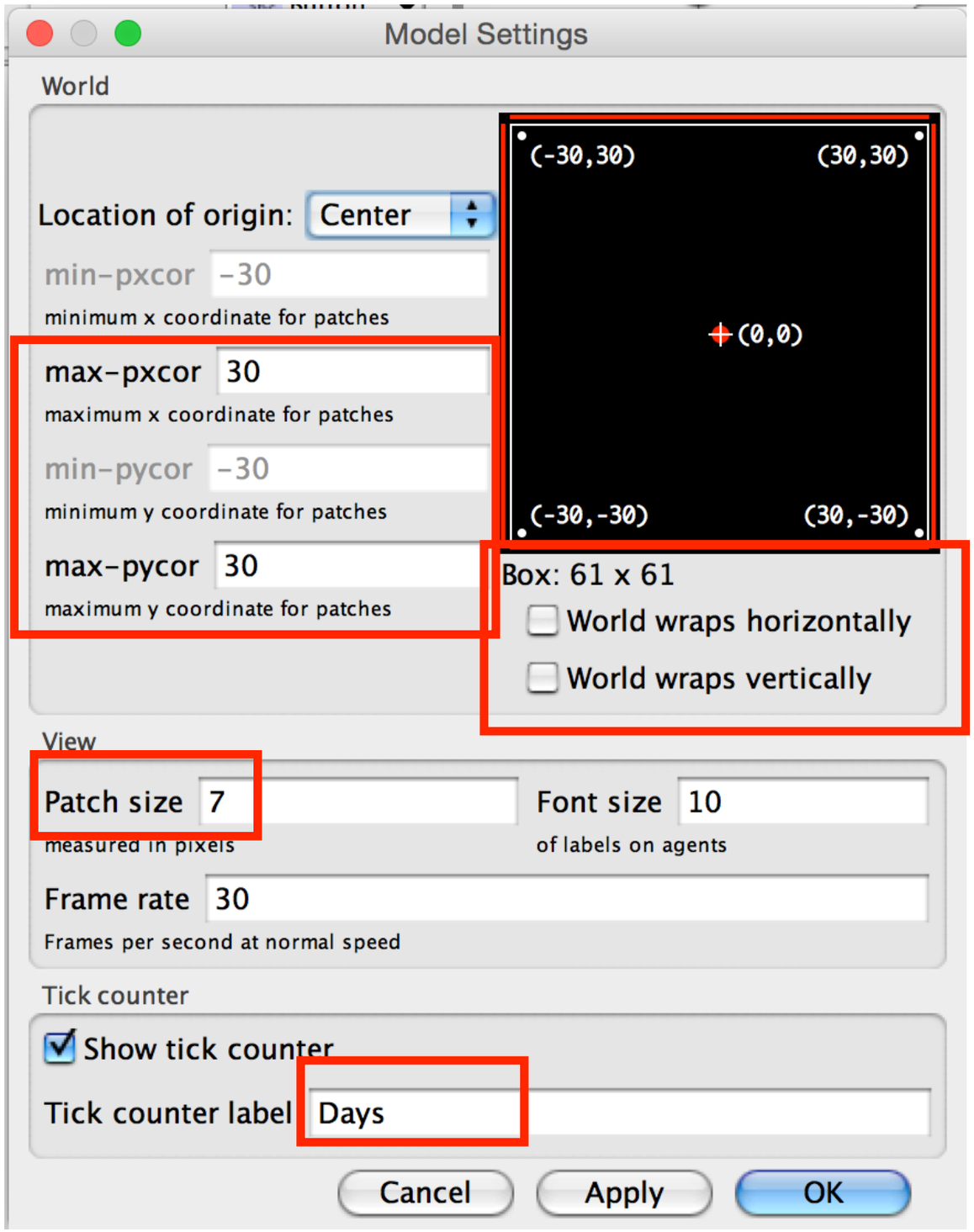
Setting up the model display

- Open Netlogo or if open **Create a new model** in Netlogo File>New

- On the Interface tab, right click on the View (the black box display area) > select Edit> go to **Model Settings**

  - Change the values to match image shown, click OK

- **Save model** in GIS_ebola folder but not inside any of the other folders like data.



On the code tab type the following code to define globals and attributes:

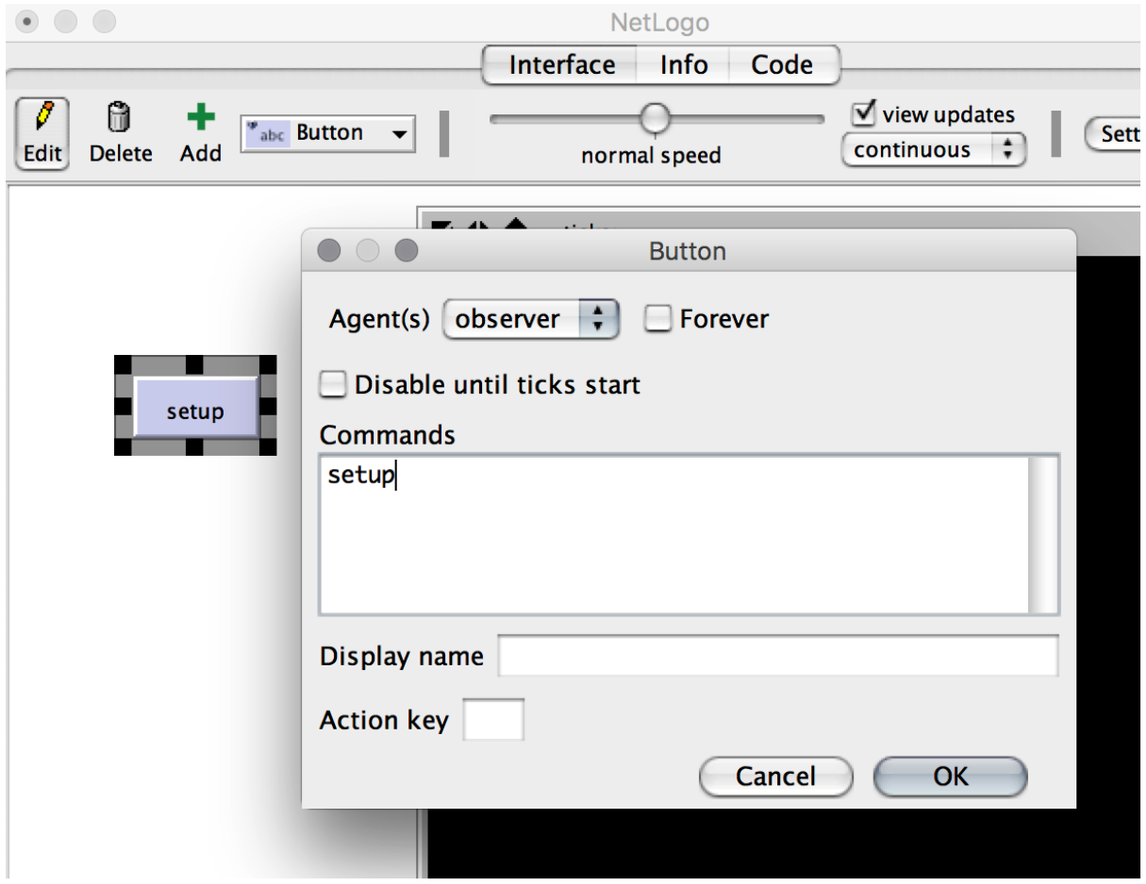
```
extensions [ gis csv ]
globals [ sites roads districts SL]

patches-own [district-name confirmed population road-here]
turtles-own []

breed [people person]
people-own [district status time-infected]

breed [admin-labels admin-label]
admin-labels-own [name]
```

Model Setup - to load GIS data

- On the interface **make button** called setup

- To add a button: Right click on white area on interface tab > choose button> in the Commands type in the word: setup and hit Ok

- Then go to code tab where we will enter the procedure for the model to execute when setup button is pressed

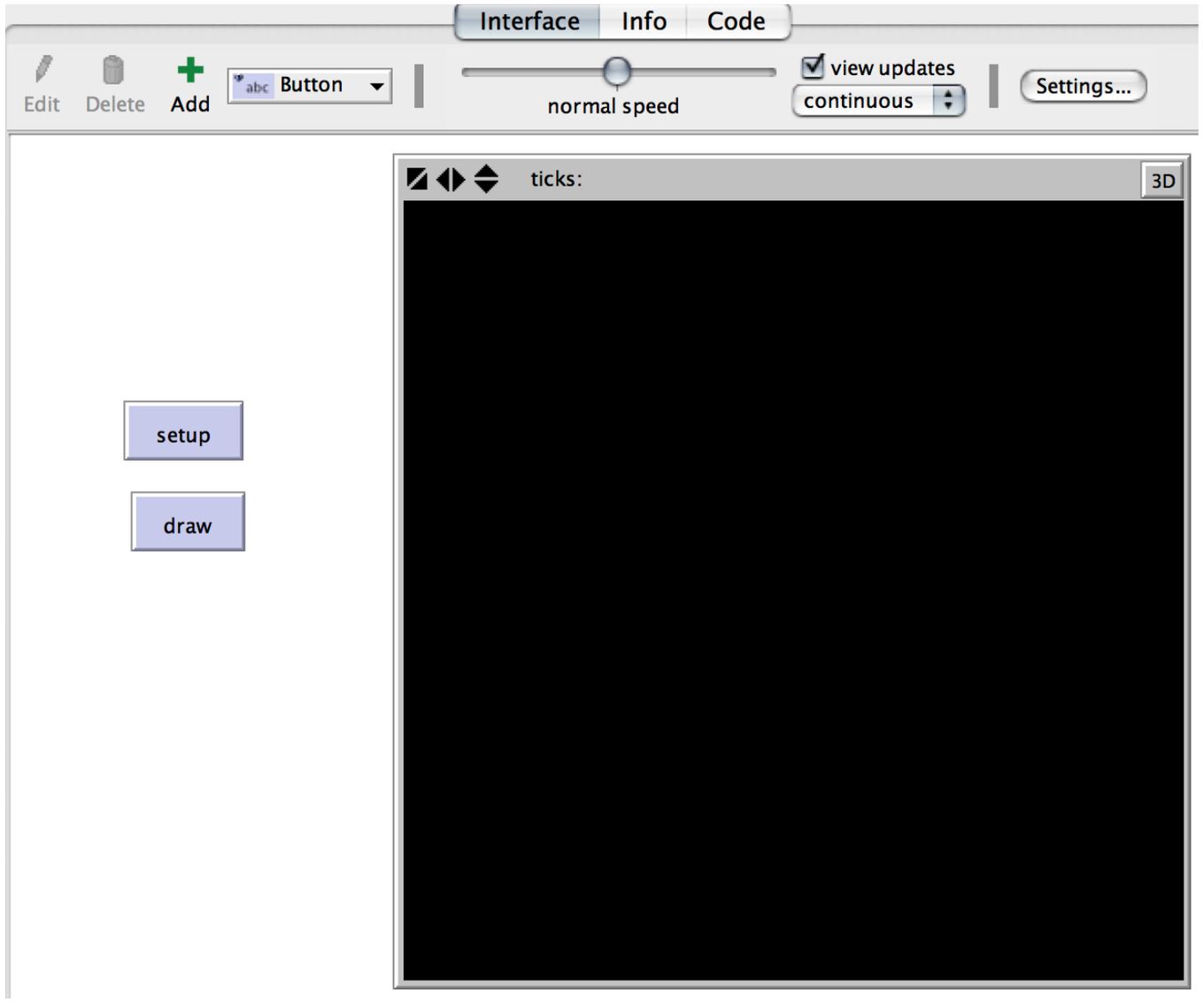


Type the following procedure for the setup button to load GIS data when setup button is pressed

```
to setup
  clear-all
  reset-ticks
  gis:load-coordinate-system (word "data/projection.prj")
  set sites gis:load-dataset "data/Ebola_Treatment_Centers.shp"
  set roads gis:load-dataset "data/Sierra Leone_Roads.shp"
  set districts gis:load-dataset "data/Cases_at_Admin2_Level.shp"
  set SL gis:load-dataset "data/SL_Admin01.shp"
end
```

Draw button to show GIS

- On interface **make button** called draw, then go to code tab

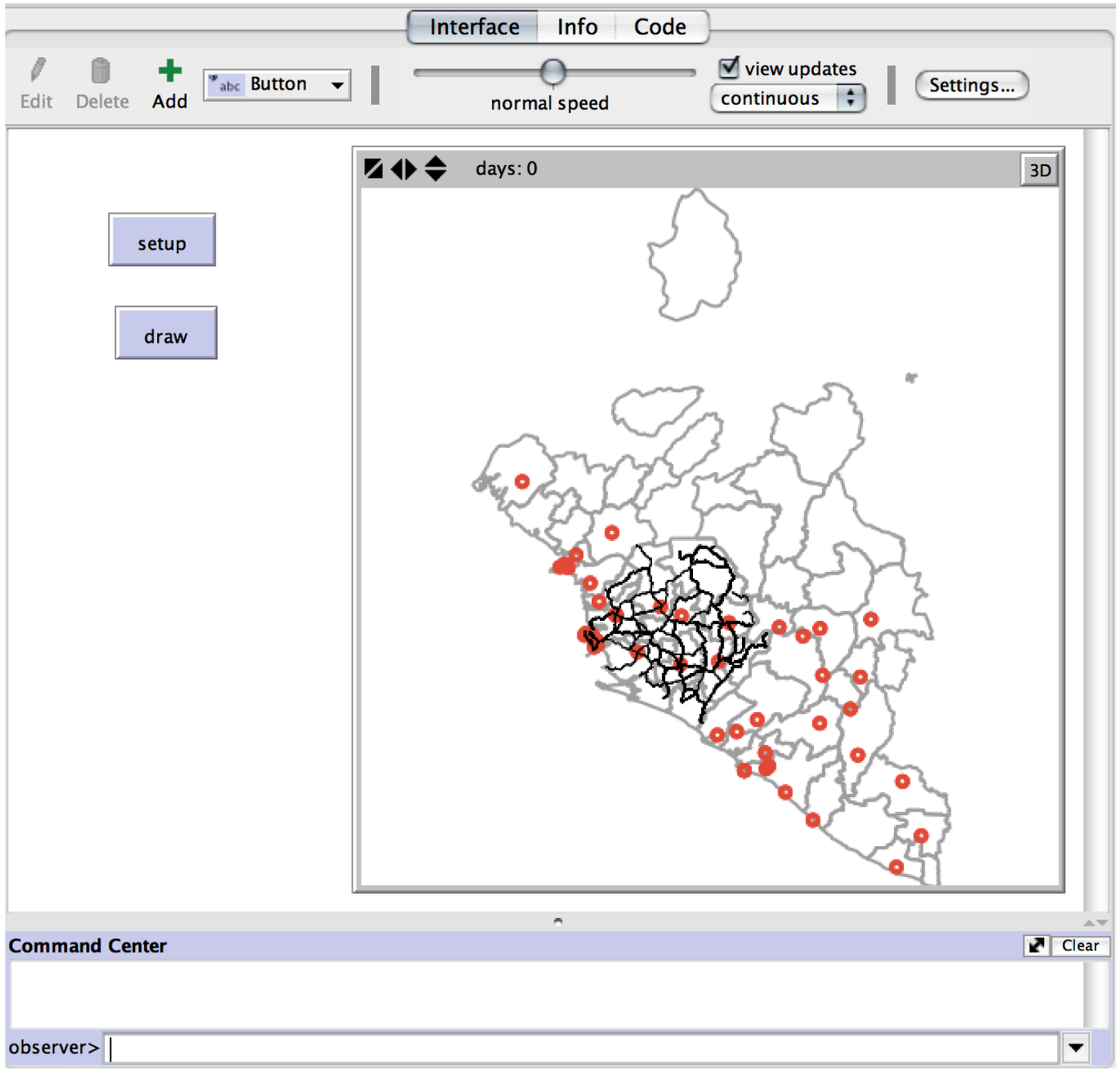


Code tab: This draw procedure will display the map data

```
to draw
clear-drawing
reset-ticks
gis:set-world-envelope (gis:envelope-union-of ;(gis:envelope-of sites)
                                              ;(gis:envelope-of roads)
                                              ;(gis:envelope-of districts)
                                              (gis:envelope-of SL)
                                              )
ask patches [set pcolor white]

gis:apply-coverage districts "GLOBAL_A_1" district-name
gis:apply-coverage districts "V_ADM2_C_3" confirmed

ask patches
  [ifelse (confirmed > 0)
    [set pcolor scale-color red confirmed 5000 0]
    [set pcolor white]
  ]

gis:set-drawing-color gray
gis:draw districts 2

gis:set-drawing-color red
gis:draw sites 3

gis:set-drawing-color black
gis:draw roads 1

ask patches
    [if gis:intersects? roads self
        [set road-here 1 ] ]
end
```

Load and display map data

- Press the setup button
- Press the draw button
- Save the model



Let’s zoom in

Our model is about Sierra Leone, so let’s have the model focus on that area. Note: The “envelope” controls the spatial extent of the display.

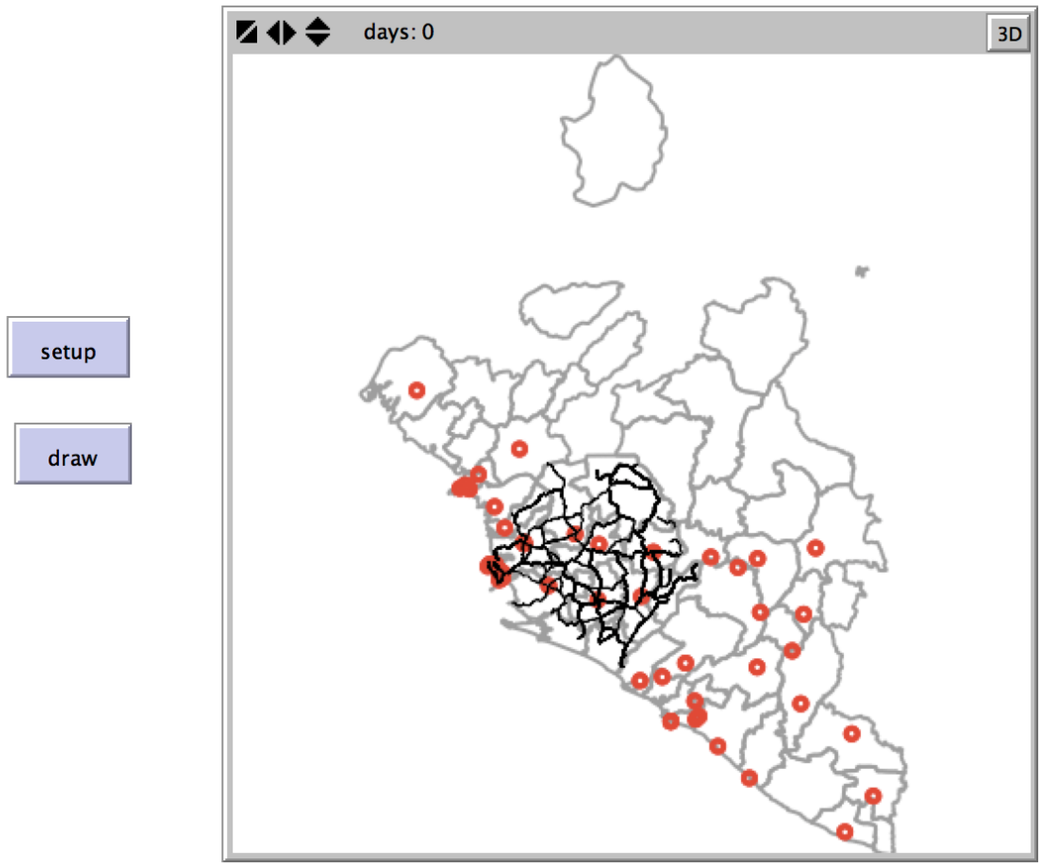
- In the code tab, in the draw procedure, **comment out** the code as shown below by inserting a semi-colon as shown. Then **save** the model.

```
gis:set-world-envelope (gis:envelope-union-of ;(gis:envelope-of sites)
;(gis:envelope-of roads)
;(gis:envelope-of districts)
(gis:envelope-of SL)
)
```

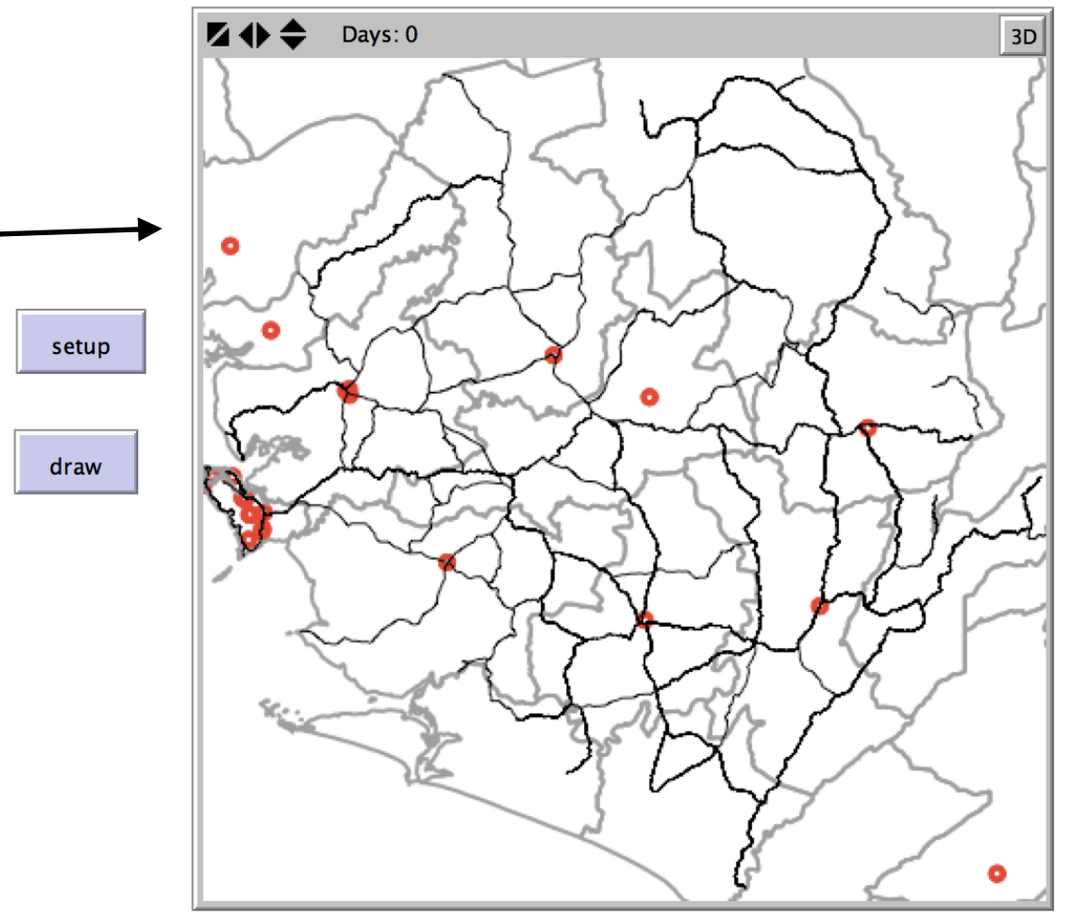
- Then **press draw** again.

Let’s zoom in

Before

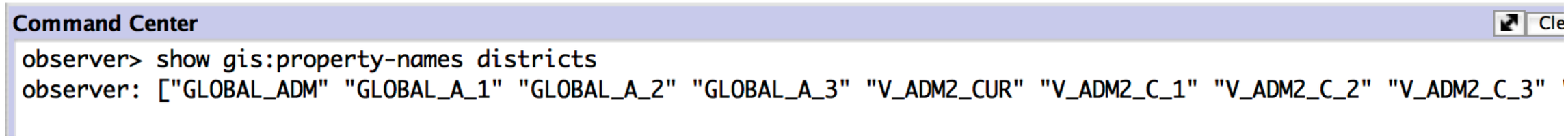


After



To get name of attributes of GIS data

- To find out attribute names of GIS data loaded, **in the Command Center** at the bottom of the interface tab, **type:** show gis:property-names districts



Apply attributes of GIS data to patch

For shapefiles with polygons, we can apply values for GIS properties to a patch attribute.

Let's apply values for district name stored in the districts layer as property "GLOBAL_A_1" to the patch attribute district-name that we defined earlier in patches-own. Let's do the same for values stored in "V_ADM2_C_3" in districts layer to patch attribute of *confirmed*.

- **Add code** below in to draw section in the same place shown.

```
to draw
clear-drawing
reset-ticks
gis:set-world-envelope (gis:envelope-union-of ;(gis:envelope-of sites)
                                              ;(gis:envelope-of roads)
                                              ;(gis:envelope-of districts)
                                              (gis:envelope-of SL)
                                              )
ask patches [set pcolor white]

gis:apply-coverage districts "GLOBAL_A_1" district-name
gis:apply-coverage districts "V_ADM2_C_3" confirmed

gis:set-drawing-color gray
gis:draw districts 2

gis:set-drawing-color red
gis:draw sites 3

gis:set-drawing-color black
gis:draw roads 1
end
```

gis:apply-coverage districts "GLOBAL_A_1" district-name
gis:apply-coverage districts "V_ADM2_C_3" confirmed

Apply color gradient to patches based on attribute

- **Add code** below in to draw section in the same place shown.
- **Check** code, **save**
- **Press draw** button

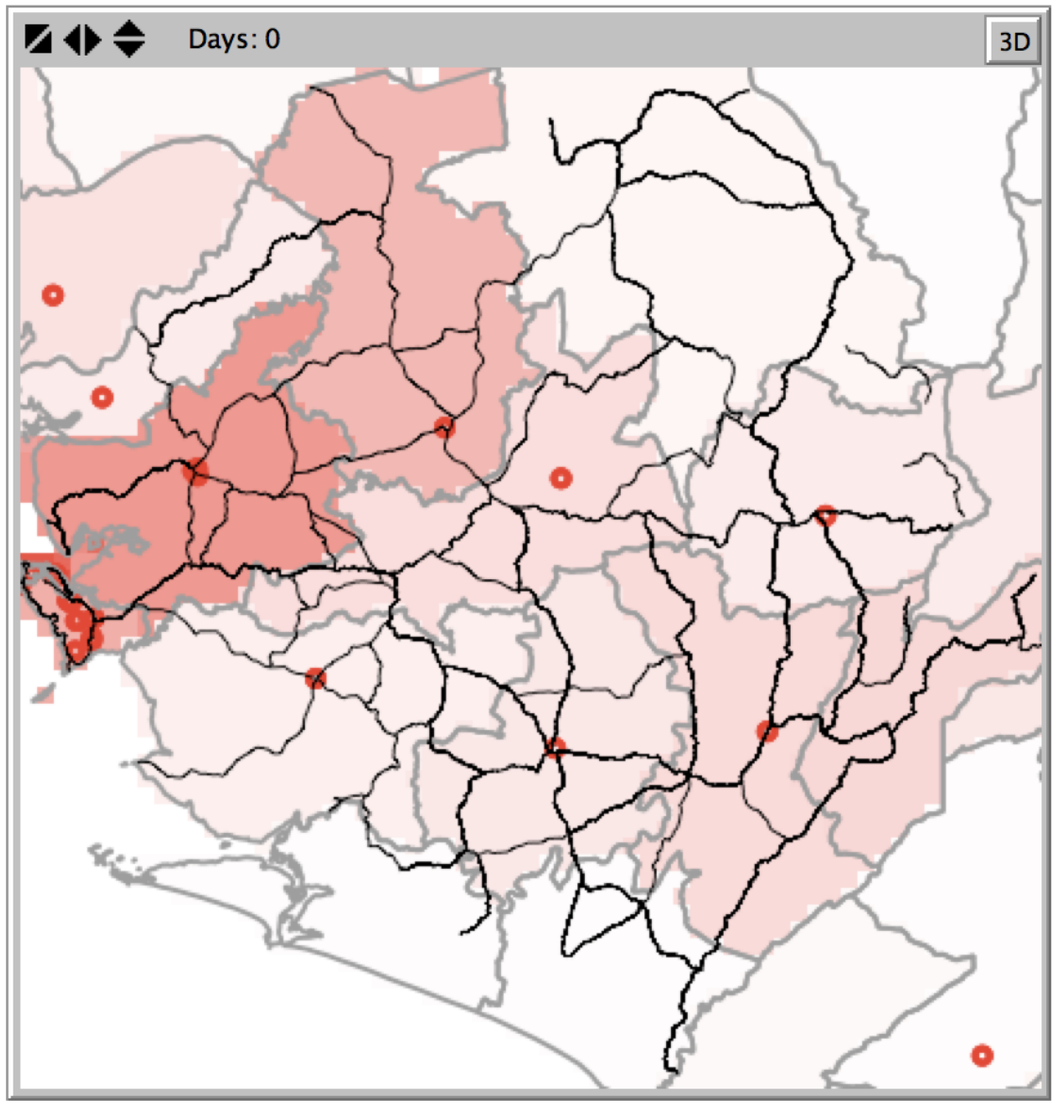
```
to draw
clear-drawing
reset-ticks
gis:set-world-envelope (gis:envelope-union-of ;(gis:envelope-of sites)
                                              ;(gis:envelope-of roads)
                                              ;(gis:envelope-of districts)
                                              (gis:envelope-of SL)
                                              )
ask patches [set pcolor white]

gis:apply-coverage districts "GLOBAL_A_1" district-name
gis:apply-coverage districts "V_ADM2_C_3" confirmed

ask patches
   [ifelse (confirmed > 0)
      [set pcolor scale-color red confirmed 5000 0]
      [set pcolor white]
   ]

gis:set-drawing-color gray
gis:draw districts 2

gis:set-drawing-color red
gis:draw sites 3

gis:set-drawing-color black
gis:draw roads 1
end
```

Note: Always color patches first. Netlogo draws layers in the order you specify.

```
ask patches
   [ ifelse (confirmed > 0)
      [set pcolor scale-color red confirmed 5000 0]
      [set pcolor white]
   ]
```

Districts are colored based on confirmed cases

The results make it appear that the districts are colored based on the number of confirmed cases. Darker the red means more number of confirmed cases in that district.

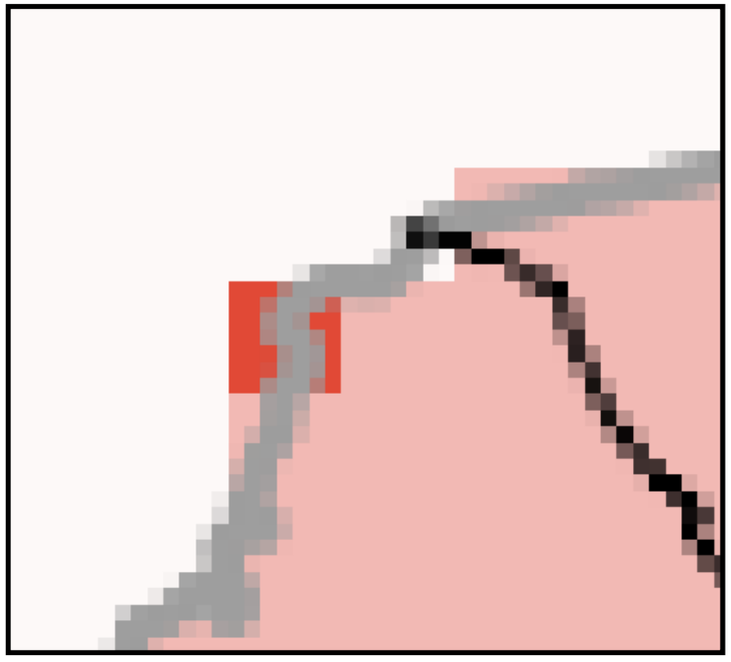


A quick note about gis:draw

Note that the code gis:draw only adds the GIS data to the display for visual effect. This does not actually apply any attributes to patches or turtles. And turtles and patches can not interact with the features that are drawn with gis:draw command.

Also note that the red square in the image represents the size of one patch. You can see that the gray line of the boundary and the skinny black line of the road are blocky when inspecting at the patch level - this is fine and just the way Netlogo converts vector data for display. Also note that the detail of these vector GIS features is more detailed than the size of the patch.

Screenshot showing zoom in with patch inspector to show size of one patch (shown in red) compared to level of detail of gis:draw command of lines shown in gray and black.



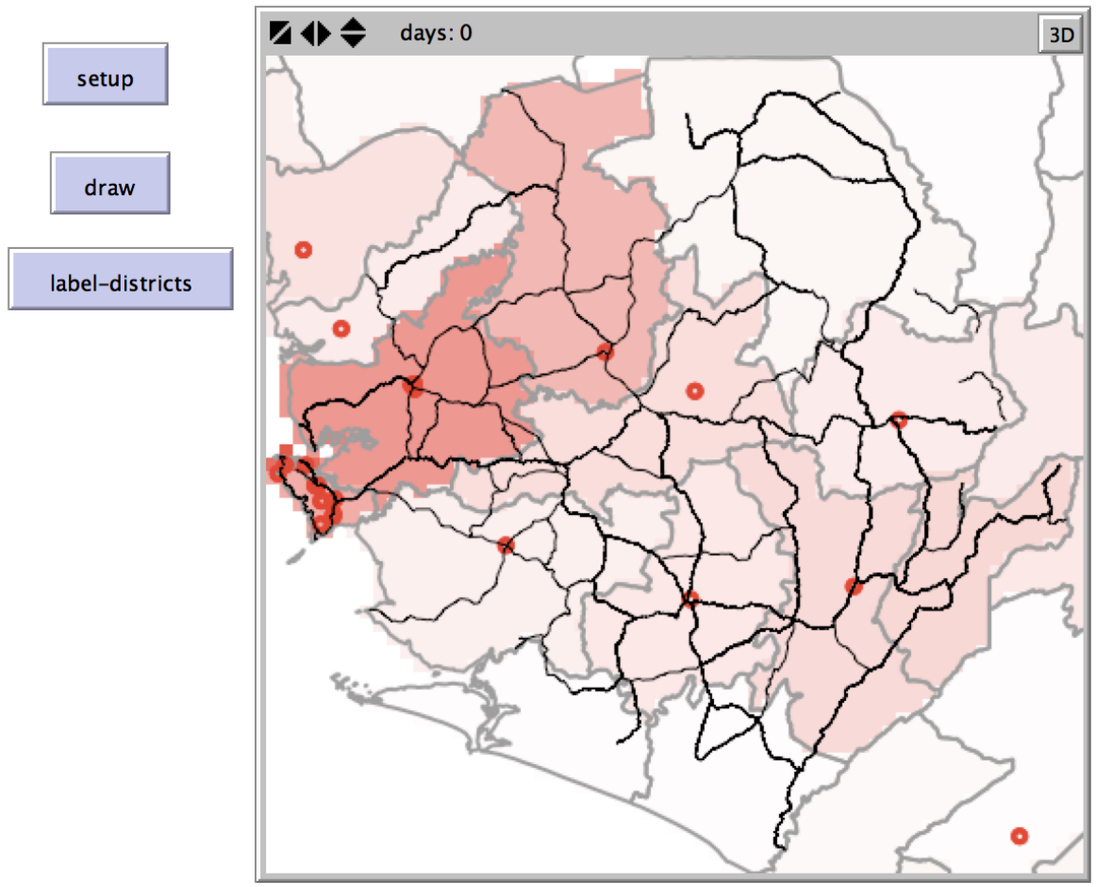
You can also see that the dark pink extends beyond the gray boundary line as discussed on the last slide.

Let’s label each district

It can be helpful to add labels to provide information as reference or context for the user. Our admin labels will be turtle agents, but of the breed admin-labels.

- **Add button** and name it label-districts

- Go to code tab



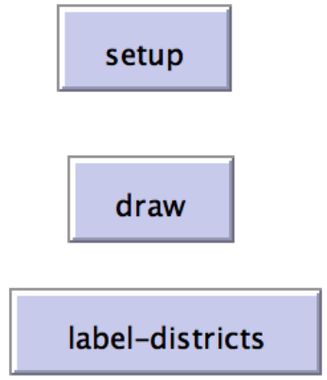
To label each district, we loop through each district in the districts GIS data. We find out the centroid of each district polygon and create an admin-label turtle at that location. Then we assign the admin-label object the name of the district, which is stored in the GIS property called GLOBAL_A_1.

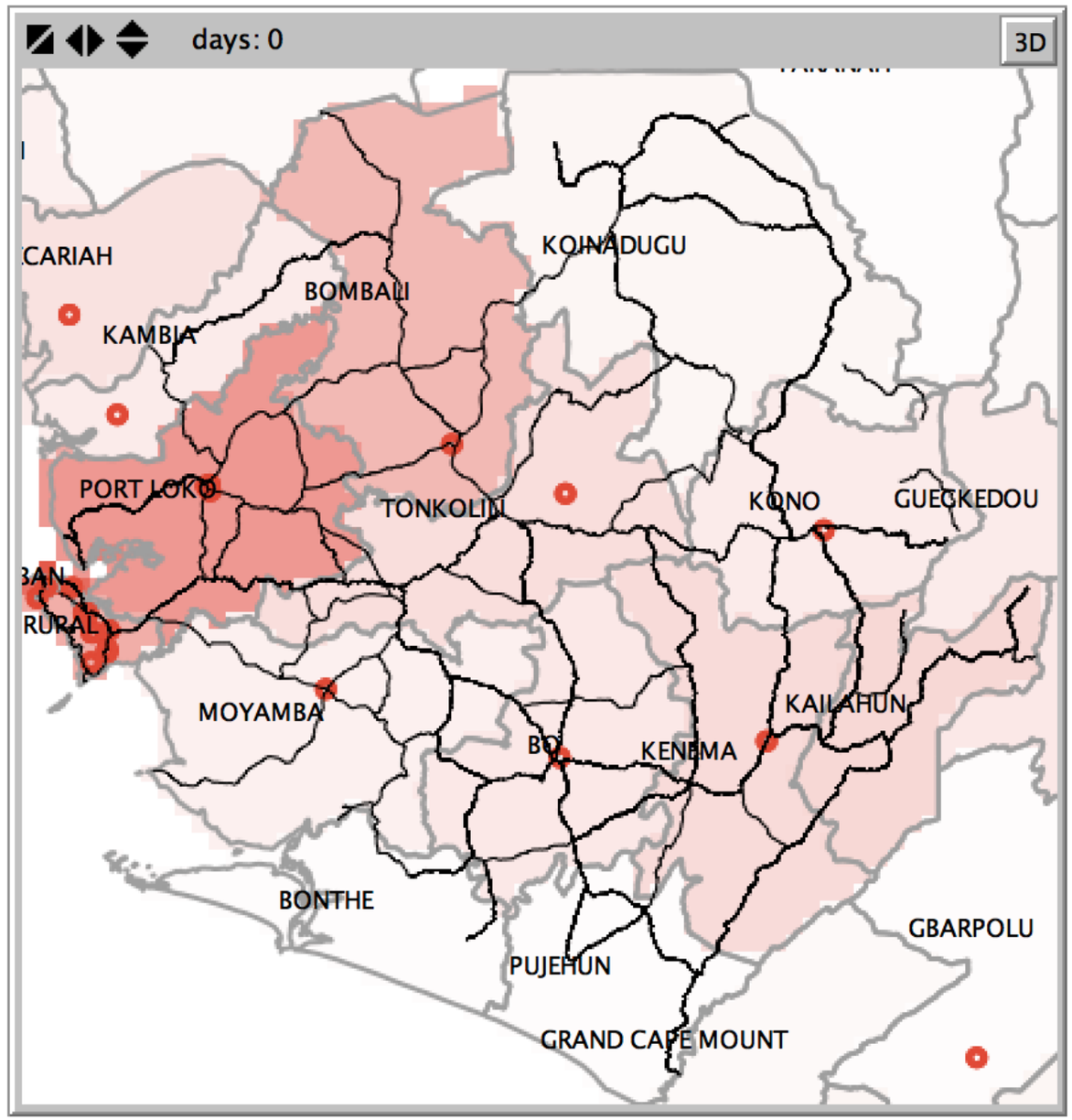
Type the following to label-districts:

```
to label-districts
  ask admin-labels [die]
  foreach gis:feature-list-of districts
    [ ?1 -> let centroid gis:location-of gis:centroid-of ?1
      if not empty? centroid
        [create-admin-labels 1
          [set xcor item 0 centroid
            set ycor item 1 centroid
            set size 0
            set shape "circle"
            set color gray
            set name gis:property-value ?1 "GLOBAL_A_1"
            set label name
          set label-color black ]]]
end
```

Districts are now labeled

- **Press** label-districts
- Save

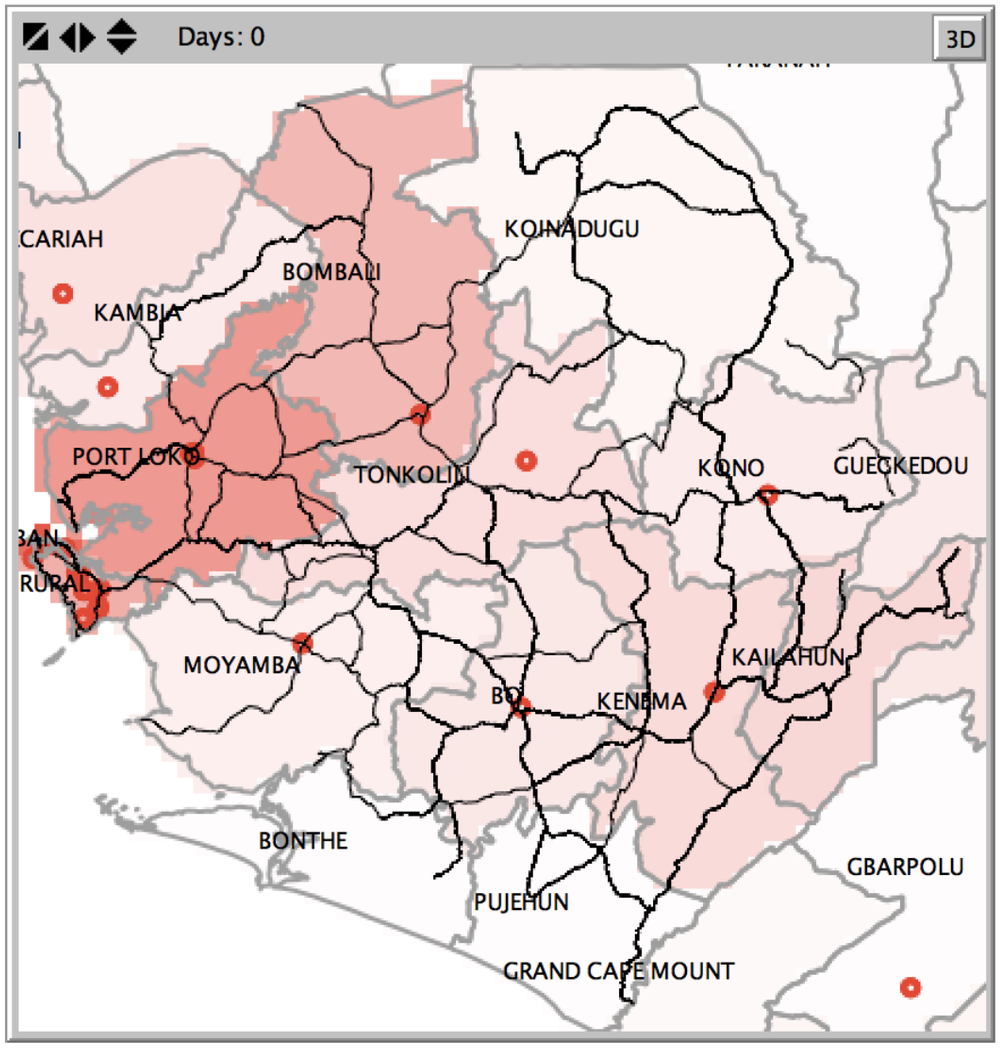




What does the model show us so far?

The model display is showing the **spatial extent** of the country of Sierra Leone, although the country is not outlined or labeled.

We can see which **districts** have more **confirmed ebola cases**. Those with more cases are shown in darker red. Those with fewer cases are shown in lighter shades of pinkish-white.



The districts are outlined with thick light gray lines. Each **district name** is labeled in black.

The **treatment centers** are shown as red circles. The **road network** is shown as thin black lines and they appear to connect most treatment centers.

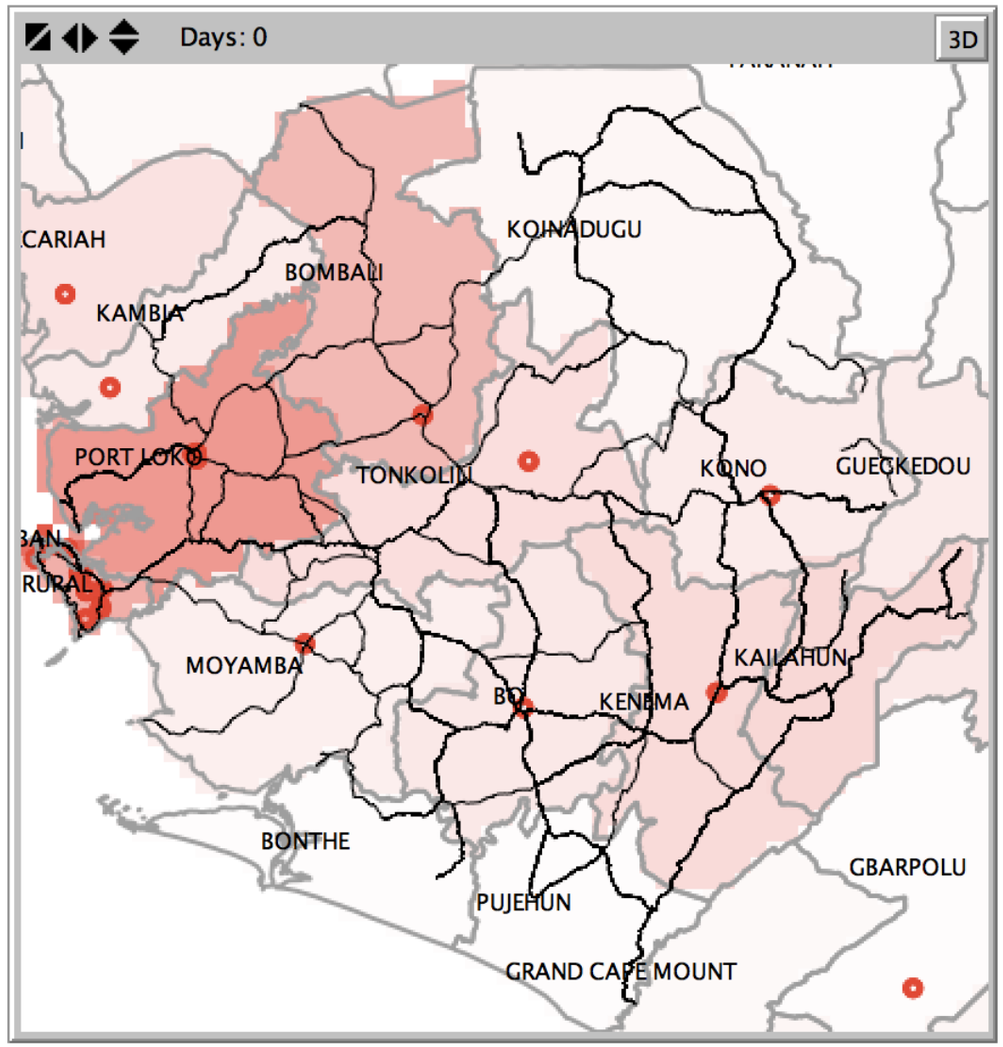
You did all this with GIS data you downloaded and by using the Netlogo GIS extension. Although this model is not complete, you have made a nice visual base model upon which to add more data and behavior. **Let's add some more data.**

We need more data to support the story the model is trying to portray

We can see which districts had more confirmed ebola cases, but we don't know how much of the population in each district was impacted.

The GIS data we downloaded only had information about number of people infected with ebola and not anything about total population numbers.

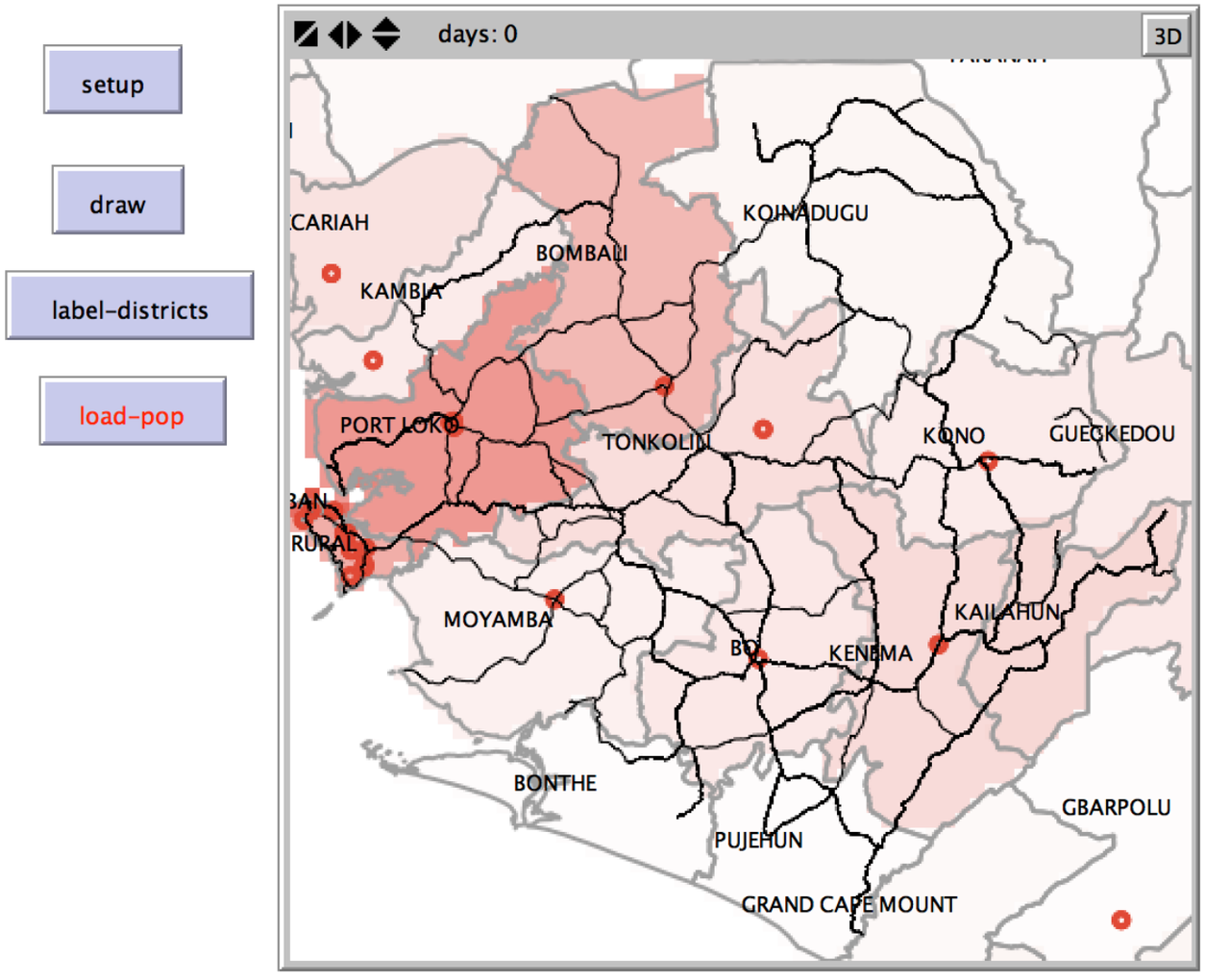
We need to get more data about the population in each district of Sierra Leone



Let's use the CSV to make a population of agents

For each district, let's create a representative population of agents where 1 agent = 10000 people using the Population value stored in the .csv. We will initialize the population as being susceptible to ebola by setting the attribute of status as "S" and color pink and assign them a district based on the District name from the csv. We will then move the agents to a random patch in their district based on the patch attribute of district-name.

- **Add button** load-pop
- Go to code tab



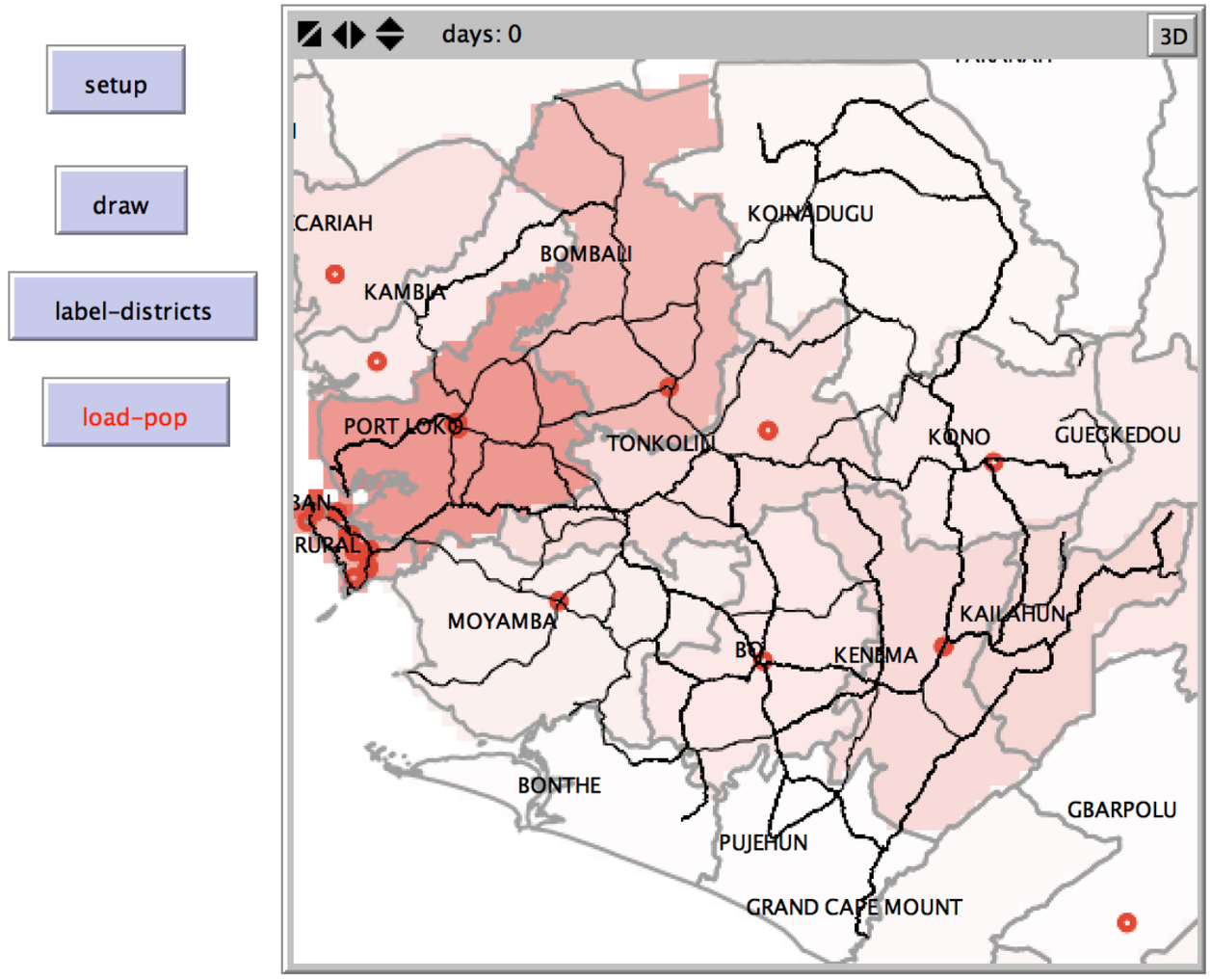
Add the following code to load CSV and create agents based on value in CSV.

```
to load-pop
  ask turtles with [breed = people][die]
  reset-ticks
  file-open "data/SL_pop.csv"
    if not file-at-end? [let header csv:from-row file-read-line]
   while [not file-at-end?]
    [let row csv:from-row file-read-line
      let district_name item 0 row
      let district_pop item 4 row
      let small_pop (district_pop / 10000)
     create-people small_pop
        [set district district_name
         set shape "person"
         set size 2
         set label ""
         set status "S"
         set color pink
         move-to one-of patches with [district-name = [district] of myself]
        ]
    ]
    file-close
end
```

It didn't work

- **Save** the model.
- **Press** load-pop. You get an error!

Your agents are created at the center of the display, but you get an error when they try to move to their district. This is because 1) The shapefile names for the districts are all uppercase, but the csv district names are mixed case, and 2) two of the names for the districts are slightly different spelling than the csv.

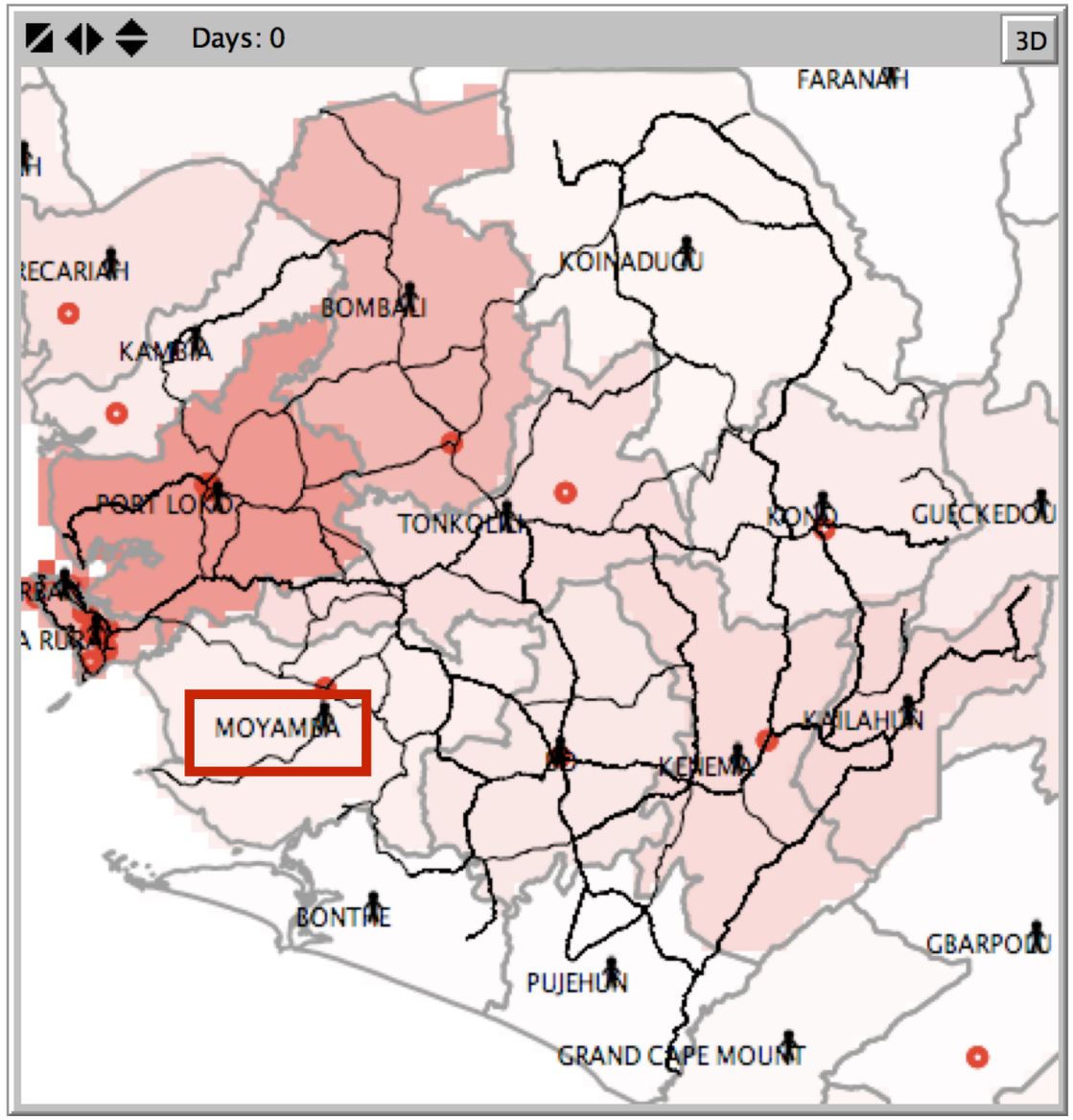


- Go to code tab

Troubleshooting

Note the name of the districts in the GIS data and shown on the display are all upper case, but the name of districts in the csv data is mixed case. Netlogo does not do fuzzy matching. It must be exactly the same, so let's add code to fix this.

	A	B	C	D	E
	District	Province	Capital	Area	Population
	Kailahun	Eastern	Kailahun	3859	358190
	Kenema	Eastern	Kenema	6053	497948
	Kono	Eastern	Sefadu	5641	335401
	Bombali	Northern	Makeni	7895	408390
	Kambia	Northern	Kambia	3108	270462
	Koinadugu	Northern	Kabala	12121	265758
	Port Loko	Northern	Port Loko	5719	453746
	Tonkolili	Northern	Magburaka	7003	347197
	Bo	Southern	Bo	5219	463668
	Bonthe	Southern	Bonthe	3468	139687
	Moyamba	Southern	Moyamba	6902	260910
	Pujehun	Southern	Pujehun	4105	228392
	Western Rural	Western	Waterloo	544	174249
	Western Urban	Western	Freetown	13	772873



Modify code to change to uppercase and fix name differences.

- **Insert semicolon to comment out the following:**

let district_name item 0 row

- **Insert code below, as shown in red box**

```
to load-pop
  ask turtles with [breed = people][die]

  file-open "data/SL_pop.csv"
    if not file-at-end? [let header csv:from-row file-read-line]
   while [not file-at-end?]
    [let row csv:from-row file-read-line
     ;let district_name item 0 row
     let d_name item 0 row
     let district_name upper-case-string d_name ;convert text to uppercase
     if district_name = "WESTERN RURAL" [set district_name "WESTERN AREA RURAL"]
     if district_name = "WESTERN URBAN" [set district_name "WESTERN AREA URBAN"]
     let district_pop item 4 row
     let small_pop (district_pop / 10000)
     create-people small_pop
        [set district district_name
         set shape "person"
         set size 2
         set label ""
         set status "S"
         set color pink
         move-to one-of patches with [district-name = [district] of myself]
        ]
    ]
    file-close
end
```

let d_name item 0 row
let district_name upper-case-string d_name ;convert text to uppercase
if district_name = "WESTERN RURAL" [set district_name "WESTERN AREA RURAL"]
if district_name = "WESTERN URBAN" [set district_name "WESTERN AREA URBAN"]

Add code to allow change to uppercase

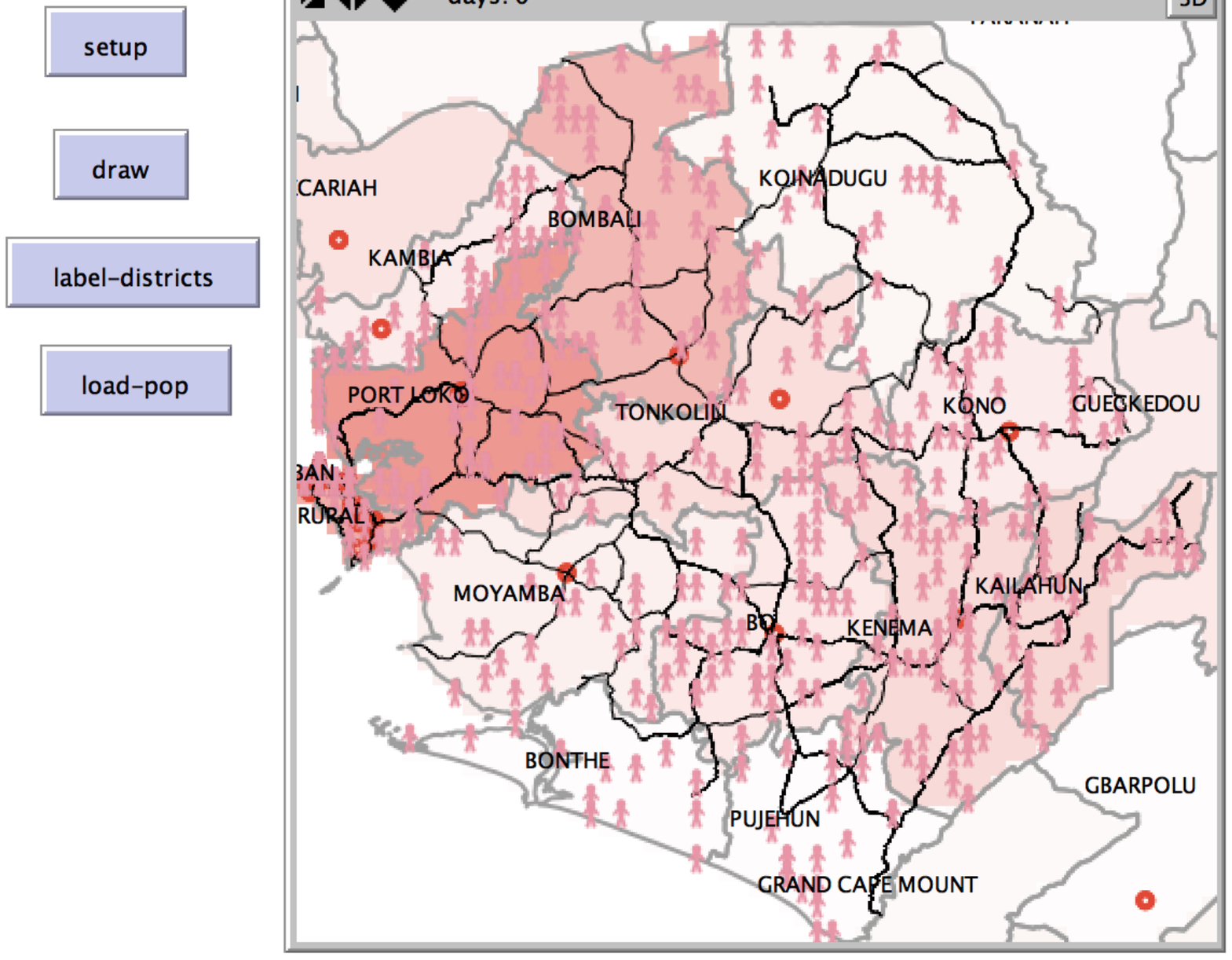
```
to-report upper-case-string [s]
   ifelse empty? s
   [report ""]
    [ report word (upper-case-char first s)
         (upper-case-string butfirst s) ]
end

to-report upper-case-char [c]
   let pos position c "abcdefghijklmnopqrstuvwxyz"
   ifelse pos = false
     [ report c ]
     [ report item pos "ABCDEFGHIJKLMNOPQRSTUVWXYZ" ]
end
```

New code for load pop and uppercase

```
to load-pop
  clear-plot
  reset-ticks

  ask turtles with [breed = people][die]
  reset-ticks
  file-open "data/SL_pop.csv"
    if not file-at-end? [let header csv:from-row file-read-line]
   while [not file-at-end?]
    [let row csv:from-row file-read-line
      ;let district_name item 0 row    let d_name item 0 row
      let district_name upper-case-string d_name ;convert text to uppercase
      if district_name = "WESTERN RURAL" [set district_name "WESTERN AREA RURAL"]
      if district_name = "WESTERN URBAN" [set district_name "WESTERN AREA URBAN"]
      let district_pop item 4 row
      let small_pop (district_pop / 10000)
      create-people small_pop
        [set district district_name  set shape "person" set size 2  set label ""  set status "S"  set color pink
         move-to one-of patches with [district-name = [district] of myself]]]
    file-close
end

to-report upper-case-string [s]
   ifelse empty? s
   [report ""]
    [ report word (upper-case-char first s)
                  (upper-case-string butfirst s) ]
end

to-report upper-case-char [c]
   let pos position c "abcdefghijklmnopqrstuvwxyz"
   ifelse pos = false
     [ report c ]
     [ report item pos "ABCDEFGHIJKLMNOPQRSTUVWXYZ" ]
end
```

Now Load people

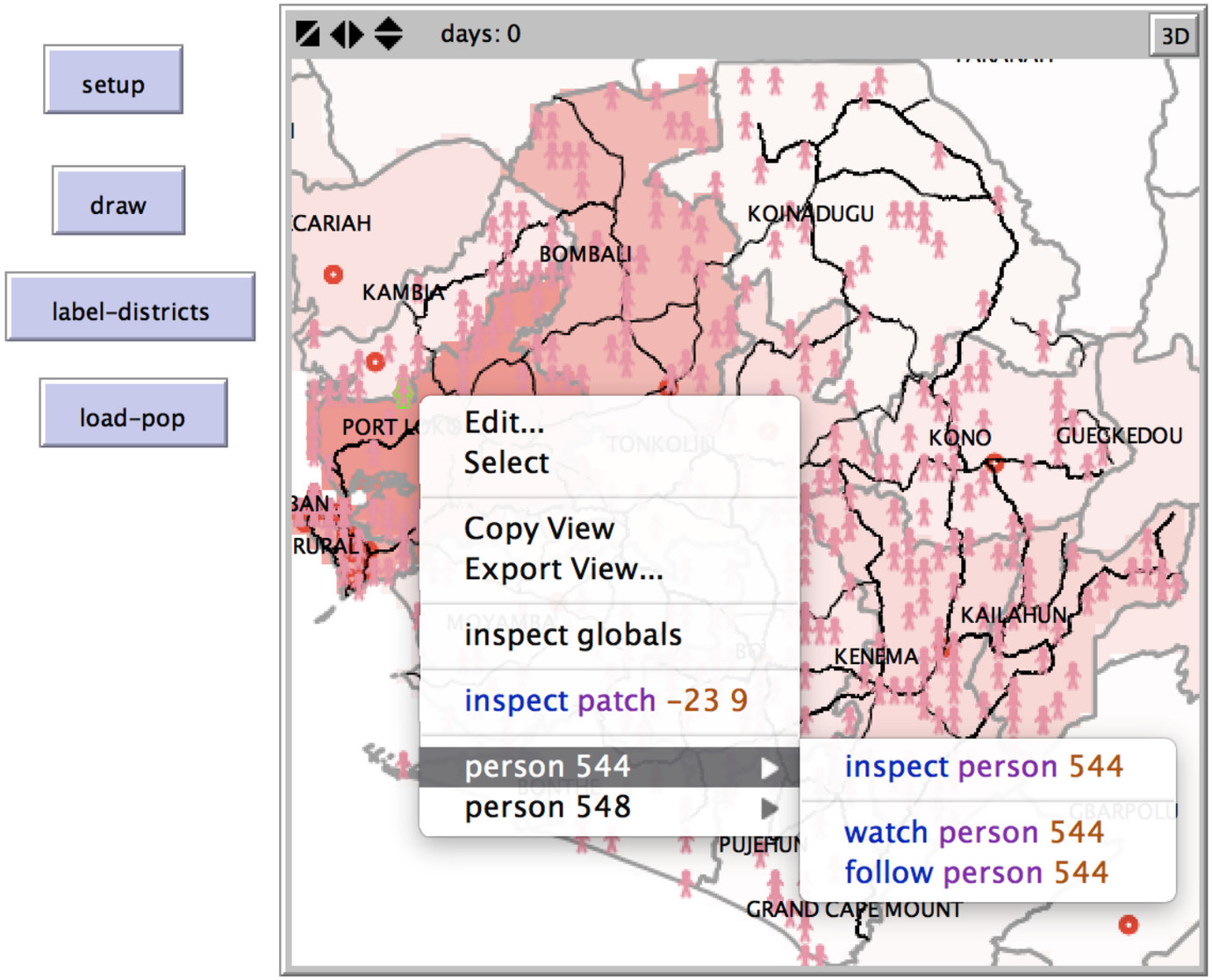
- **Save** the model.
- **Press** load-pop.



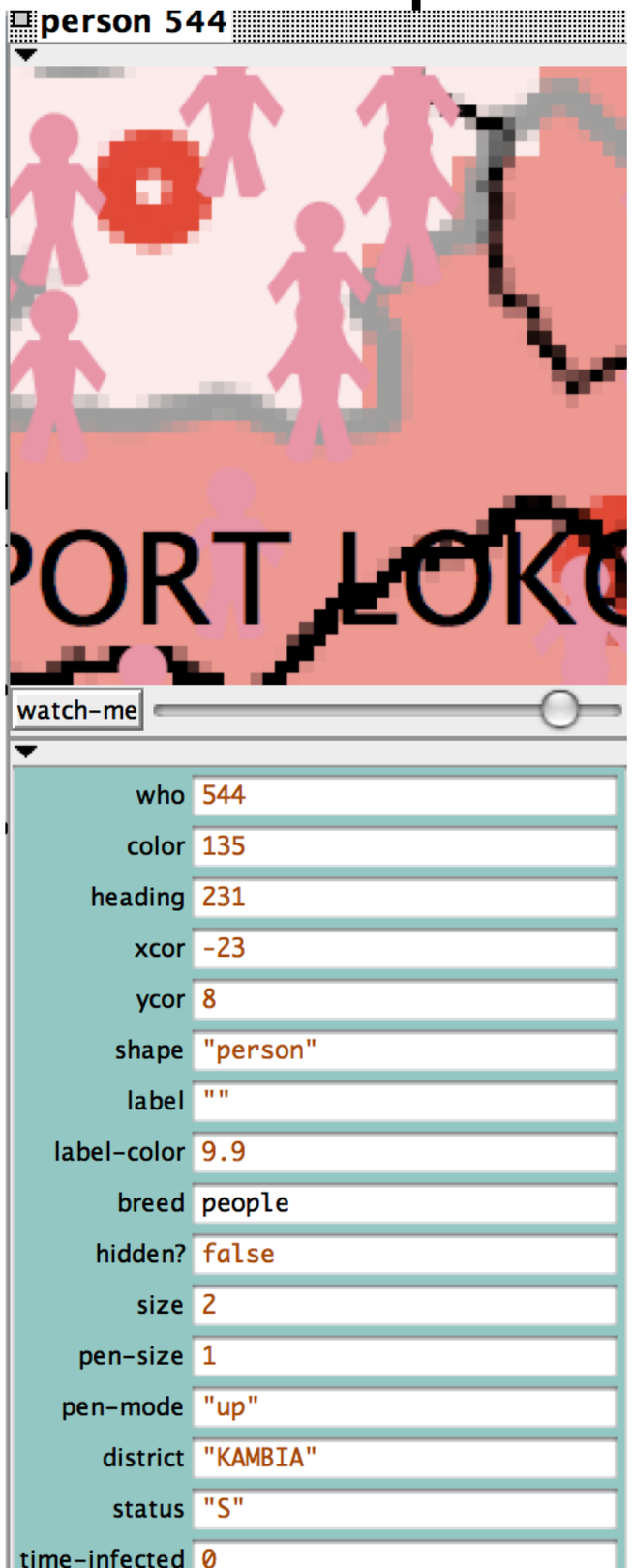
Inspect model

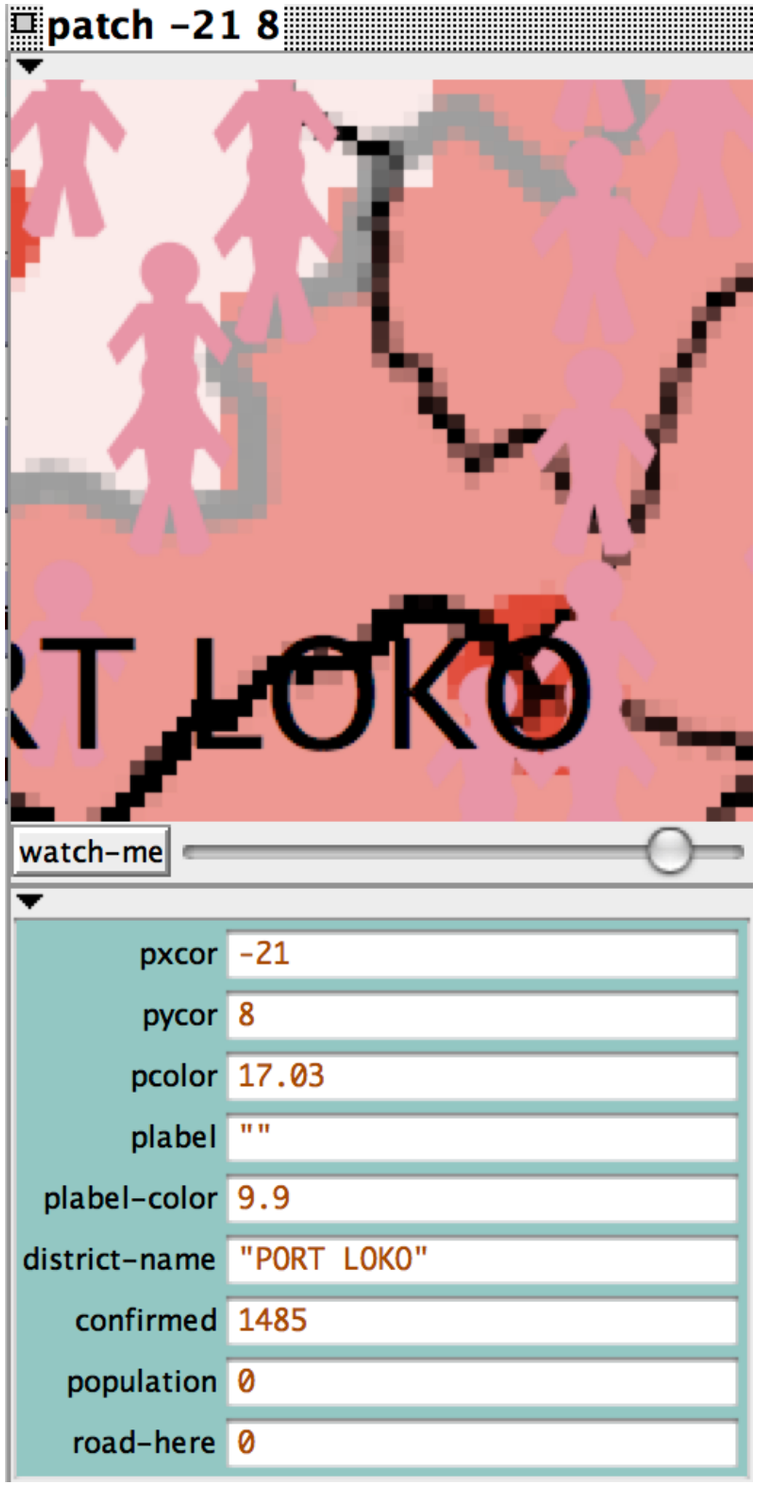
To see details about the agents and patches, right click on one, hover over arrow next to it, select inspect the person or patch

- **Inspect agents**
- **Inspect patches**



Inspect model

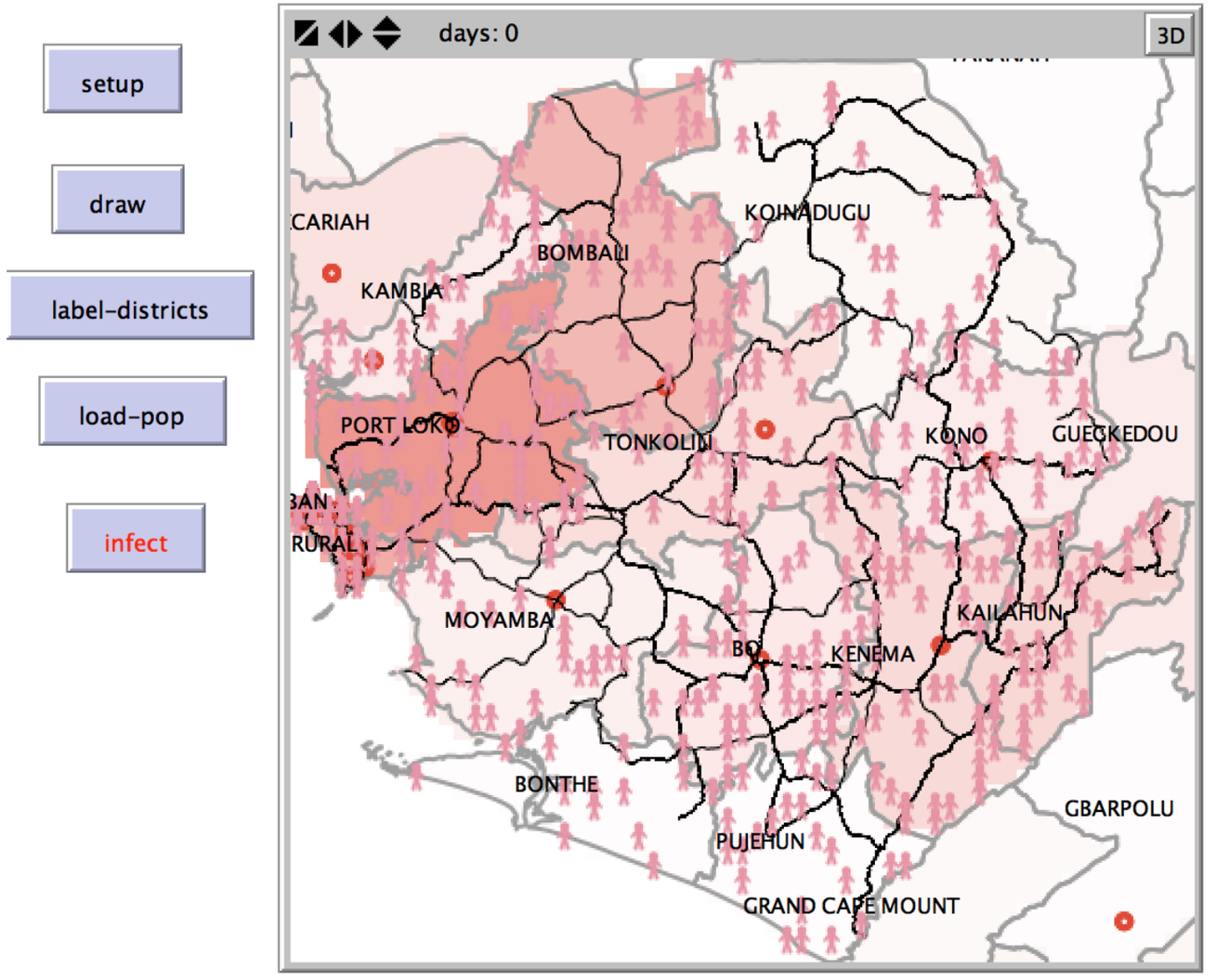




Infect people

- **Save** the model.
- **Create button** infect

Create initial infected people. The districts have the number of confirmed cases. We will use that value to make the agents infected and change their status to “I” and color to red.



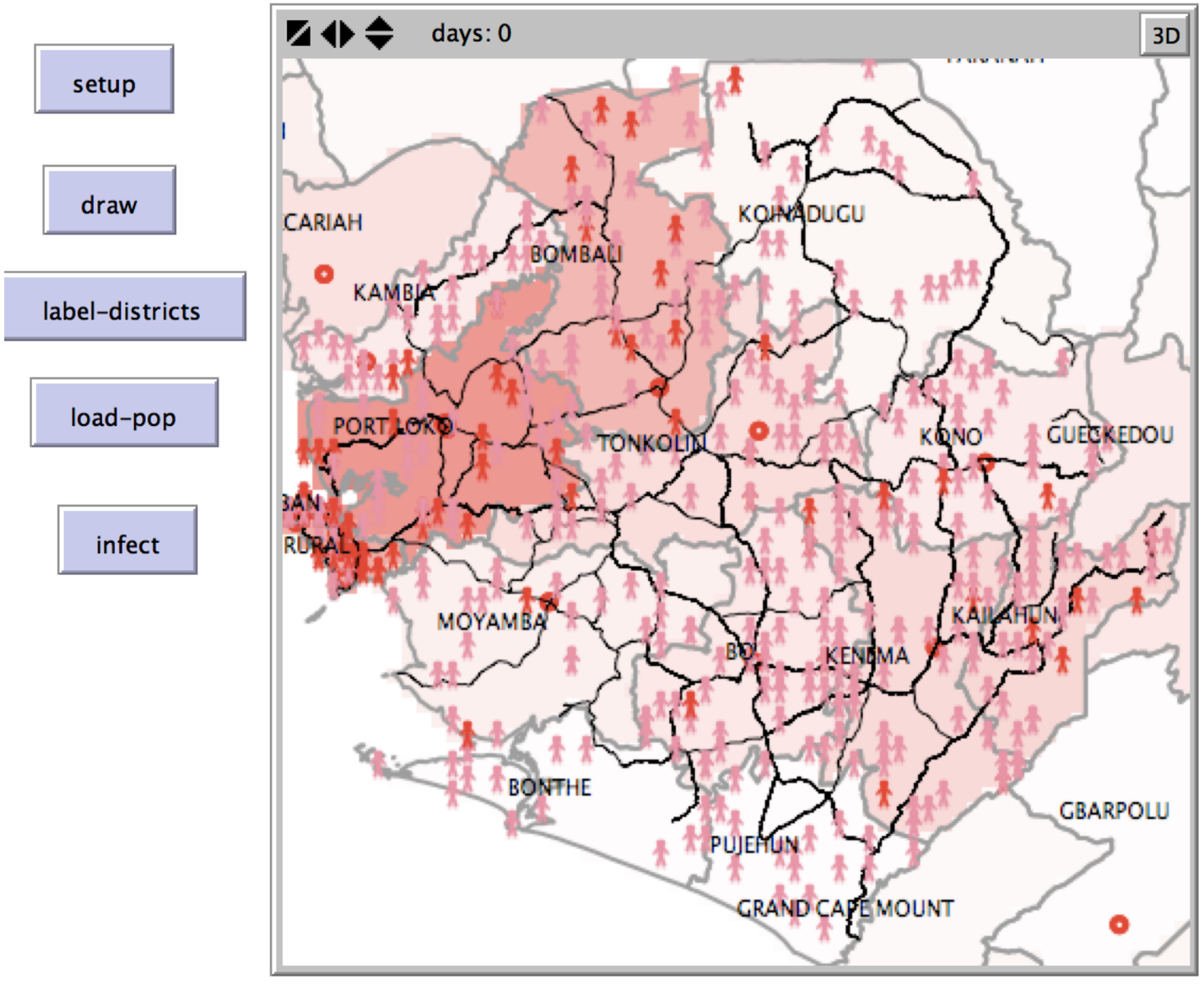
Add code to ask part of population to be infected at start:

```
to infect
  foreach gis:feature-list-of districts
    [ ?1 -> let d_name gis:property-value ?1 "GLOBAL_A_1"
      let infected gis:property-value ?1 "V_ADM2_C_3"
      let sm_infected round infected / 100 ;for visualization
      if sm_infected < 0 [set sm_infected 1]
      if any? people with [district = d_name]
         [ask n-of sm_infected people with [district = d_name]
            [set status "I" set color red ]]]
end
```

Infect people

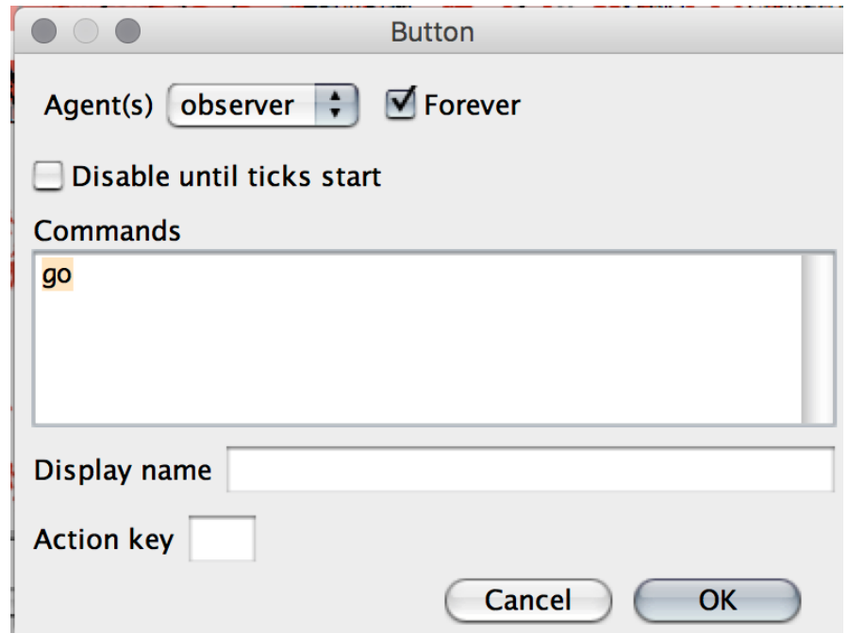
- **Press** infect
- **Save** the model.

Note that the proportion of infected to population is incorrect and we used 100 instead of 10000. The reason for this is solely for demonstration purposes.



Add go button and code for movement

- **Add button** go and check the Forever box (like shown)



to go
  ask people [travel];[rt random 360 fd 1]

  tick
end

It’s a basic model, but what does it tell us

Make them go on roads

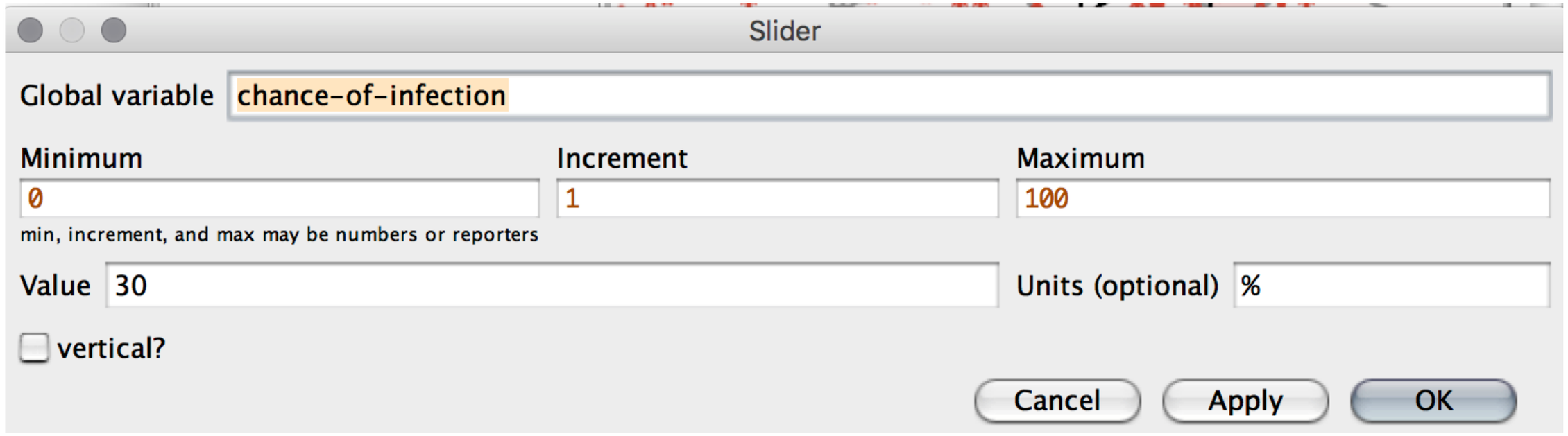
Add button that says go

```
to go
  ask turtles with [status = "S"]
    [rt random 360
     if [district-name] of patch-ahead 1 = name
      [if gis:intersects? roads patch-ahead 1
        [fd 1]
      ]
    ]
tick
end
```

Add the spread of infection

Ask the agents to move around at random. Infected agents spread infection to agents on the same patch with random chance of 30%.

- **Add slider** chance-of-infection (as shown)



code to infect other people

```
to infect-others
  let victims (turtle-set people-here people-on neighbors)

  ask victims
          [let chance random 101
            if chance > (100 - chance-of-infection)
              [set status "I"
                 set color red]]
end
```

Update go code to spread infection and stopping conditions

```
to go
  ask people [travel];[rt random 360 fd 1]

  ask people with [status = "I"]
        [infect-others
          ; to die
          let chance random 101
          if chance > (100 - chance-of-death)
             [set status "R" set size 0] ]
    tick
    if all? people [color = red] or not any? people with [color = red] [stop]
end
```

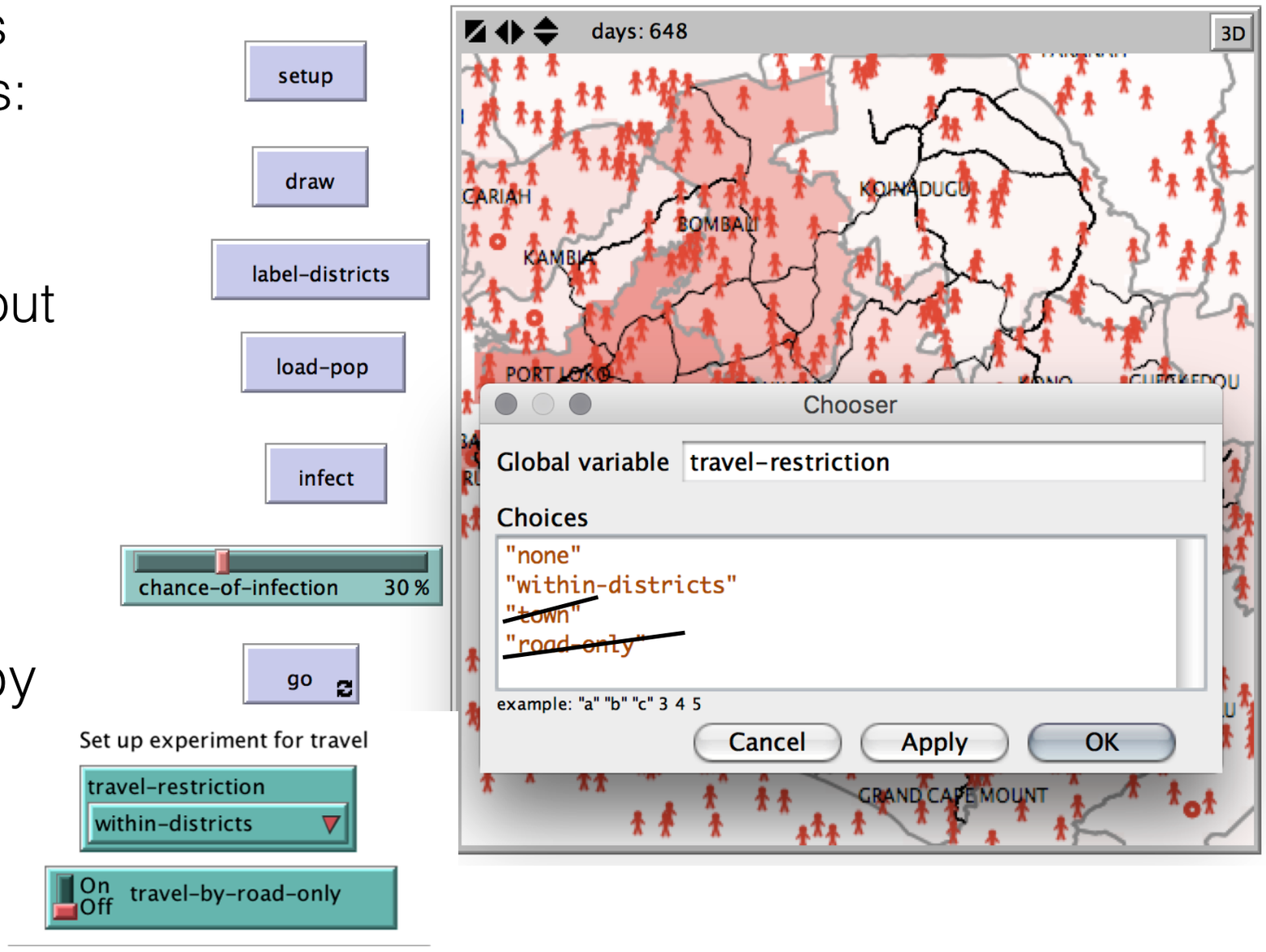
Add travel-restrictions

To change movement rules based on travel-restrictions:

- **Add chooser** call it travel-restriction and fill out as shown with choices:

  “none”
  “within-districts”

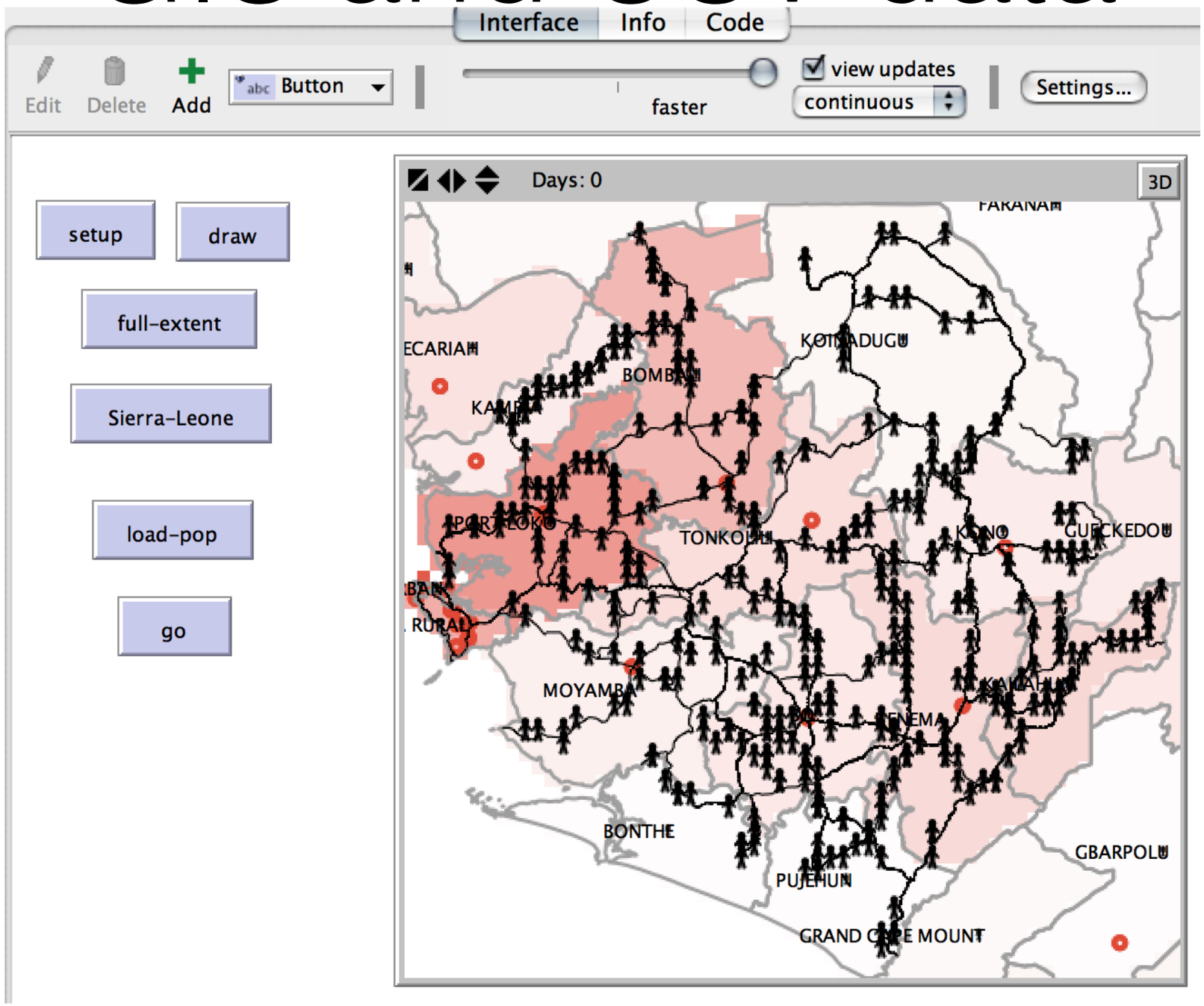
- **Add toggle** for travel by road



travel restrictions

```
to travel
 if travel-restriction = "none" and travel-by-road-only = true
   [rt random 360
     ifelse not any? neighbors with [road-here = 1]
         [move-to min-one-of patches with [road-here = 1] [distance myself]]
      [ move-to one-of neighbors with [road-here = 1]]]

 if travel-restriction = "none" and travel-by-road-only = false
   [rt random 360
    fd 1]

 if travel-restriction = "within-districts" and travel-by-road-only = false
   [rt random 360
      ifelse patch-ahead 1 = nobody
        []
        [if [district-name] of patch-ahead 1 = district
         [fd 1 ] ]]

 if travel-restriction = "within-districts" and travel-by-road-only = true
   [rt random 360
      ifelse patch-ahead 1 = nobody
        []
        [if [district-name] of patch-ahead 1 = district and [road-here] of patch-ahead 1 = 1
         [fd 1 ] ]]
end
```

A base model with GIS and CSV data

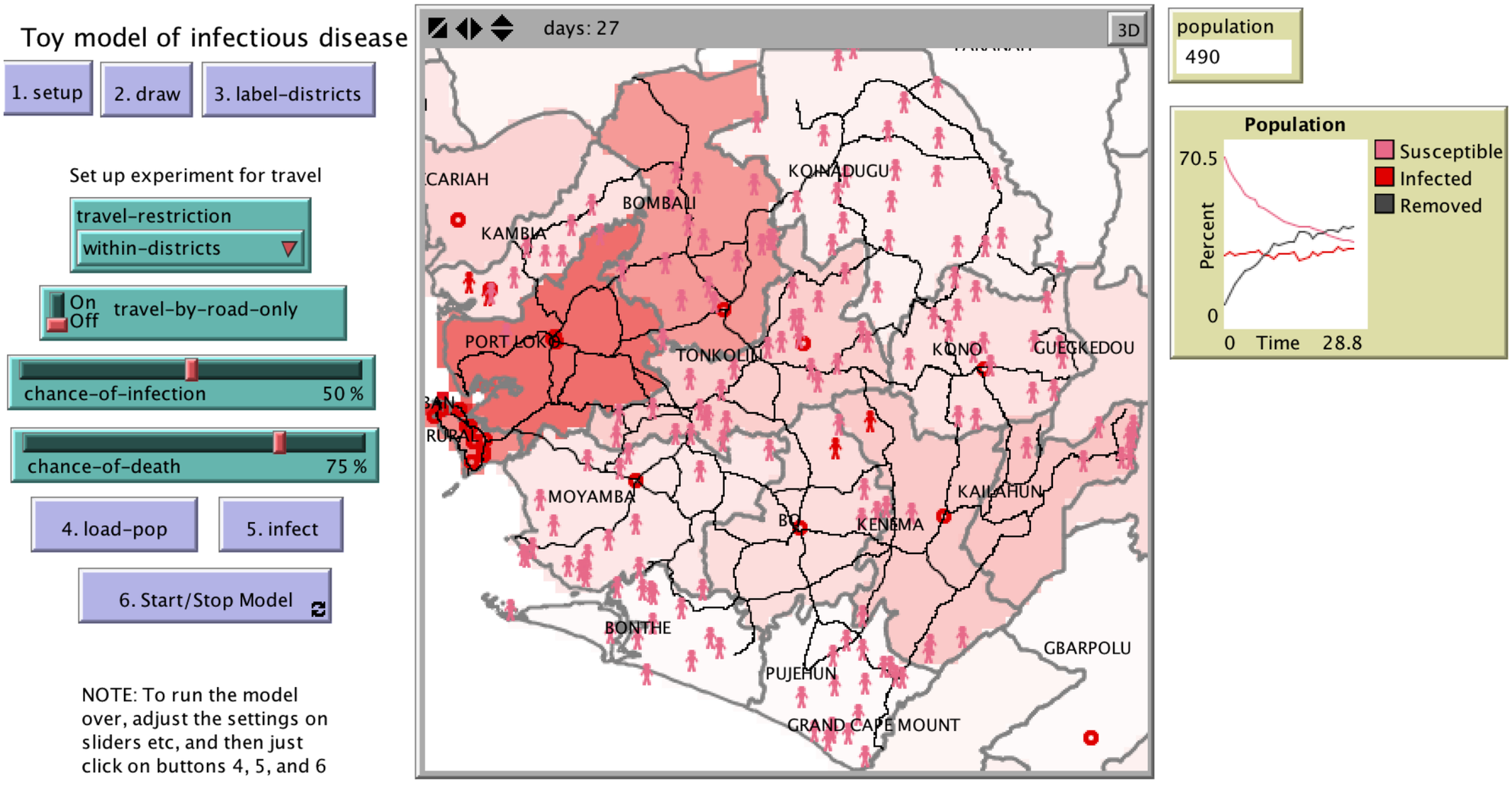


A model should enable us to test for varying conditions and experiments and measure the results.

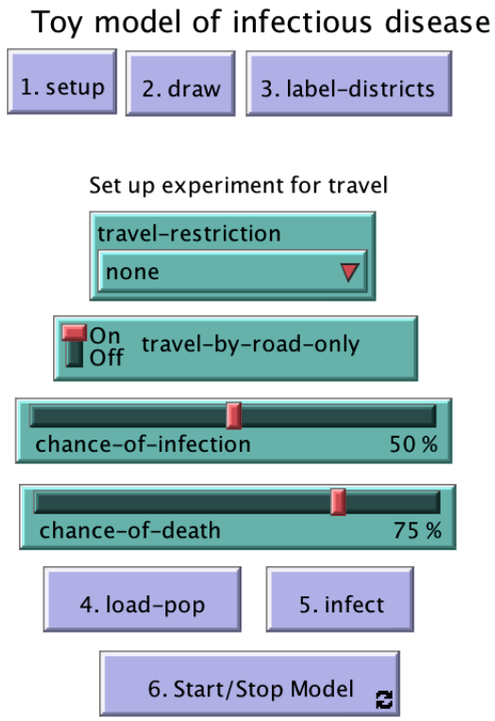
We can do this with plots and ability to adjust input parameters and test scenarios related to travel restrictions.

We can use behavior space to run many iterations of the model and analyze the results.

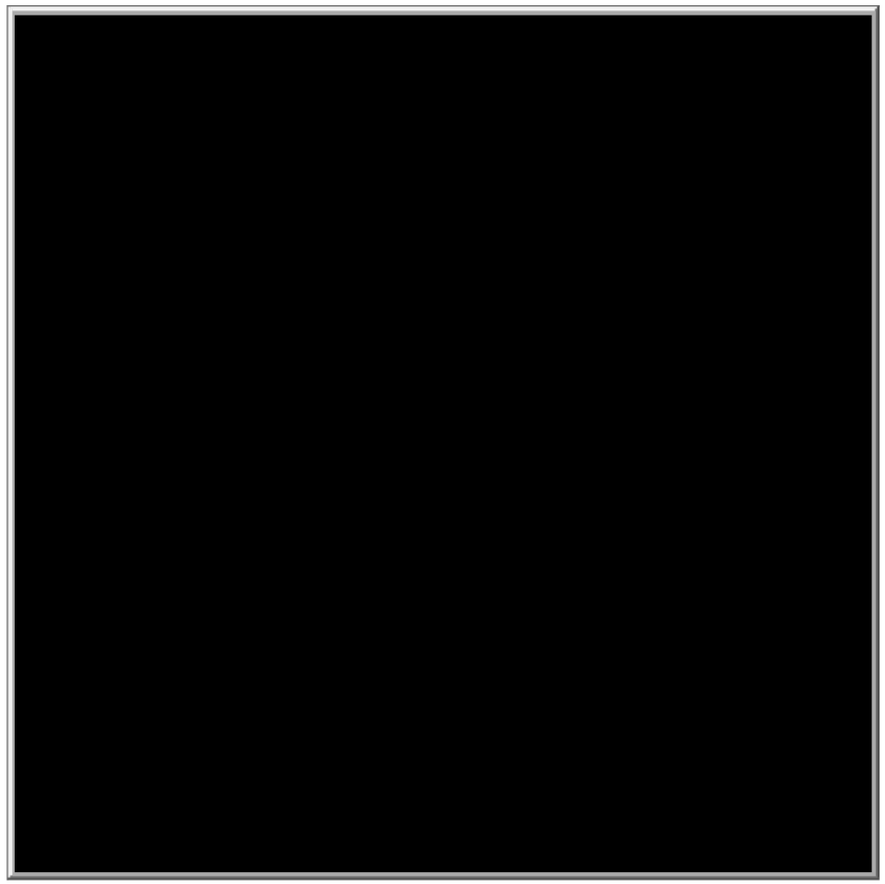
A model of disease spread

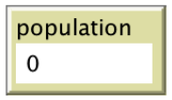


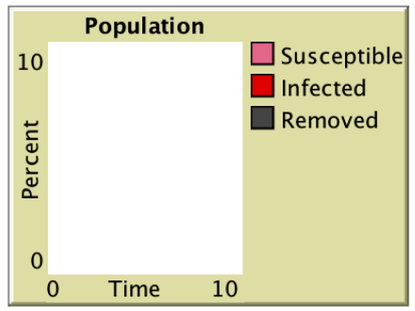
Model interface

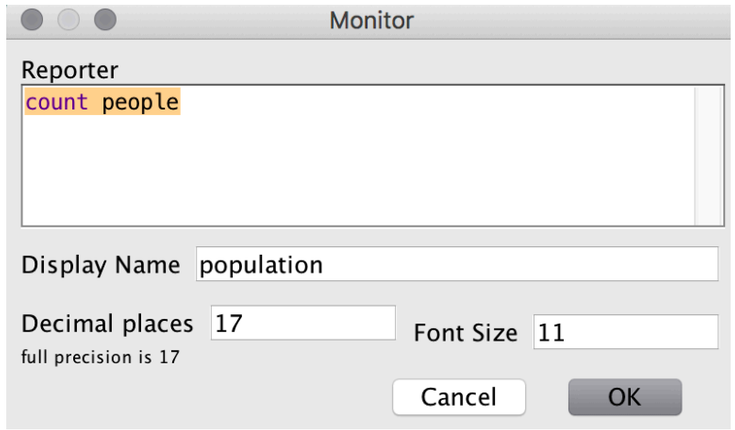


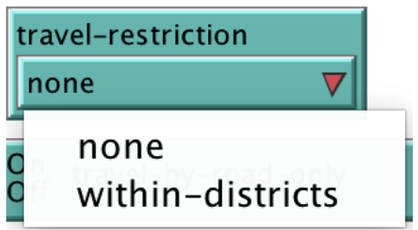
NOTE: To run the model over, adjust the settings on sliders etc, and then just click on buttons 4, 5, and 6

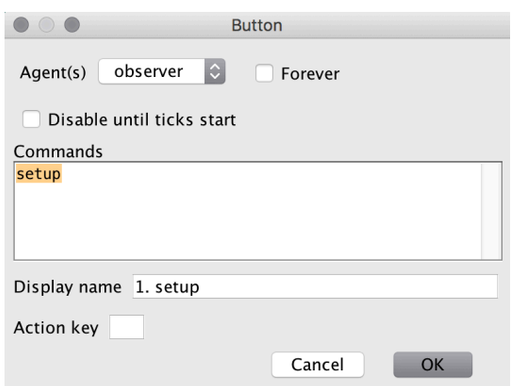


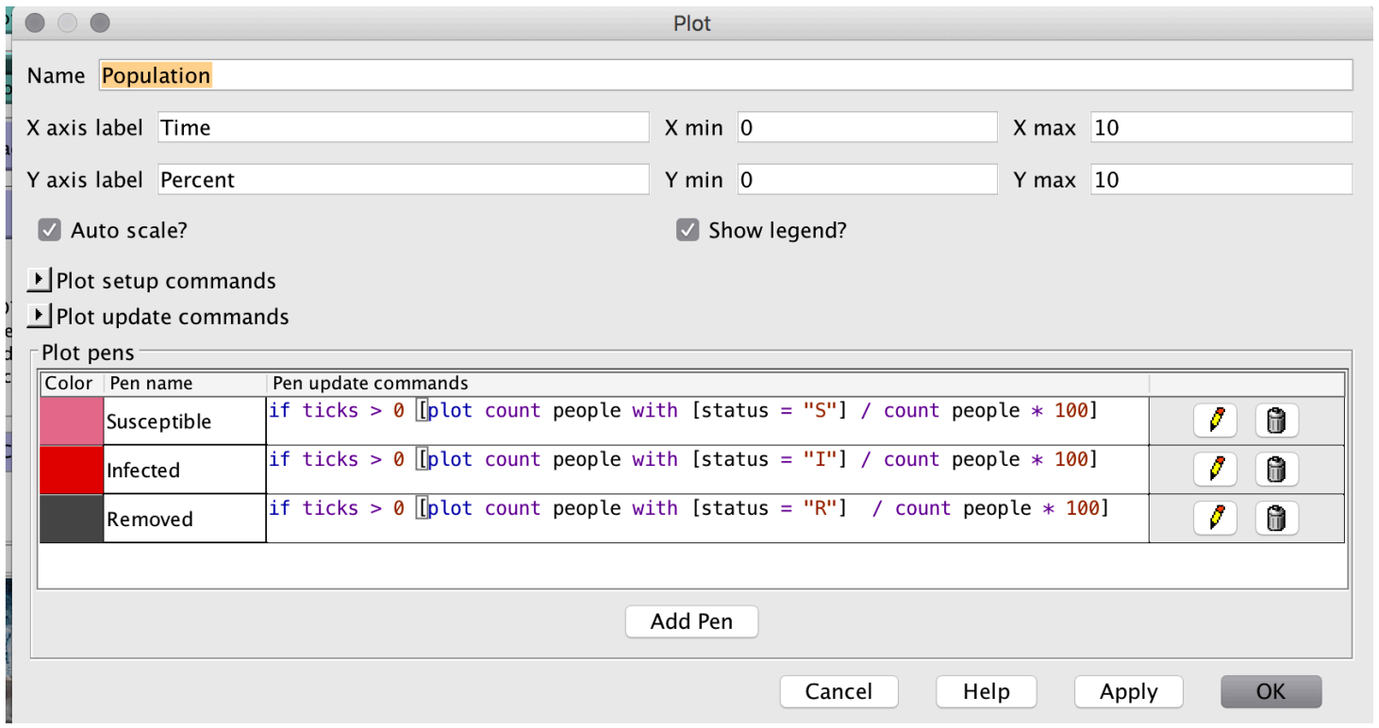












Let’s share the model

Update the info tab, add comments to code.

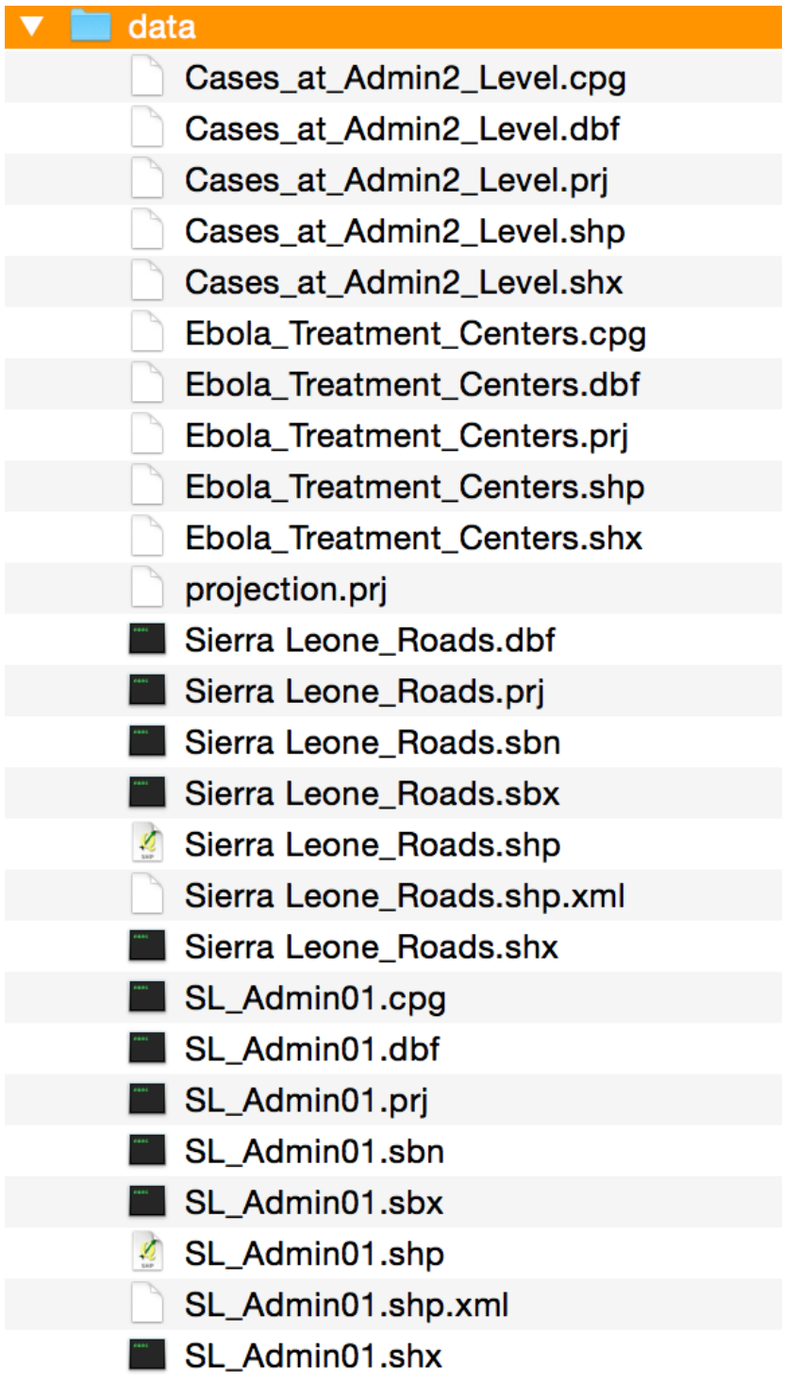
To send this model to someone, you need to send the Netlogo model, plus the data folder full of data. It’s best to **zip the model and data** folder together at the same time.

Final directory structure

File structure at the end

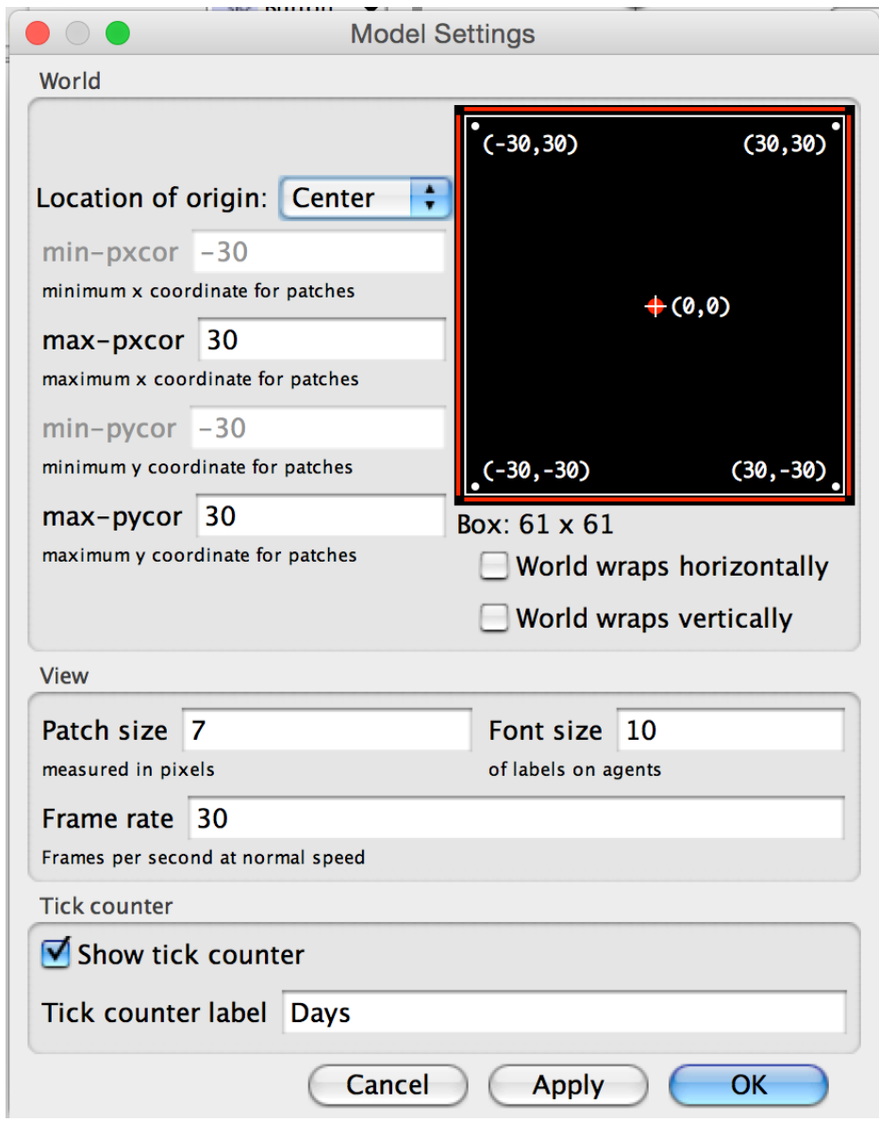
- folder: Ebola_GIS
  - your_model.nlogo
  - data (folder)
  - Model_and_data.zip
  - source_data (folder)

    You can move the original zips and folders from downloading data into this folder

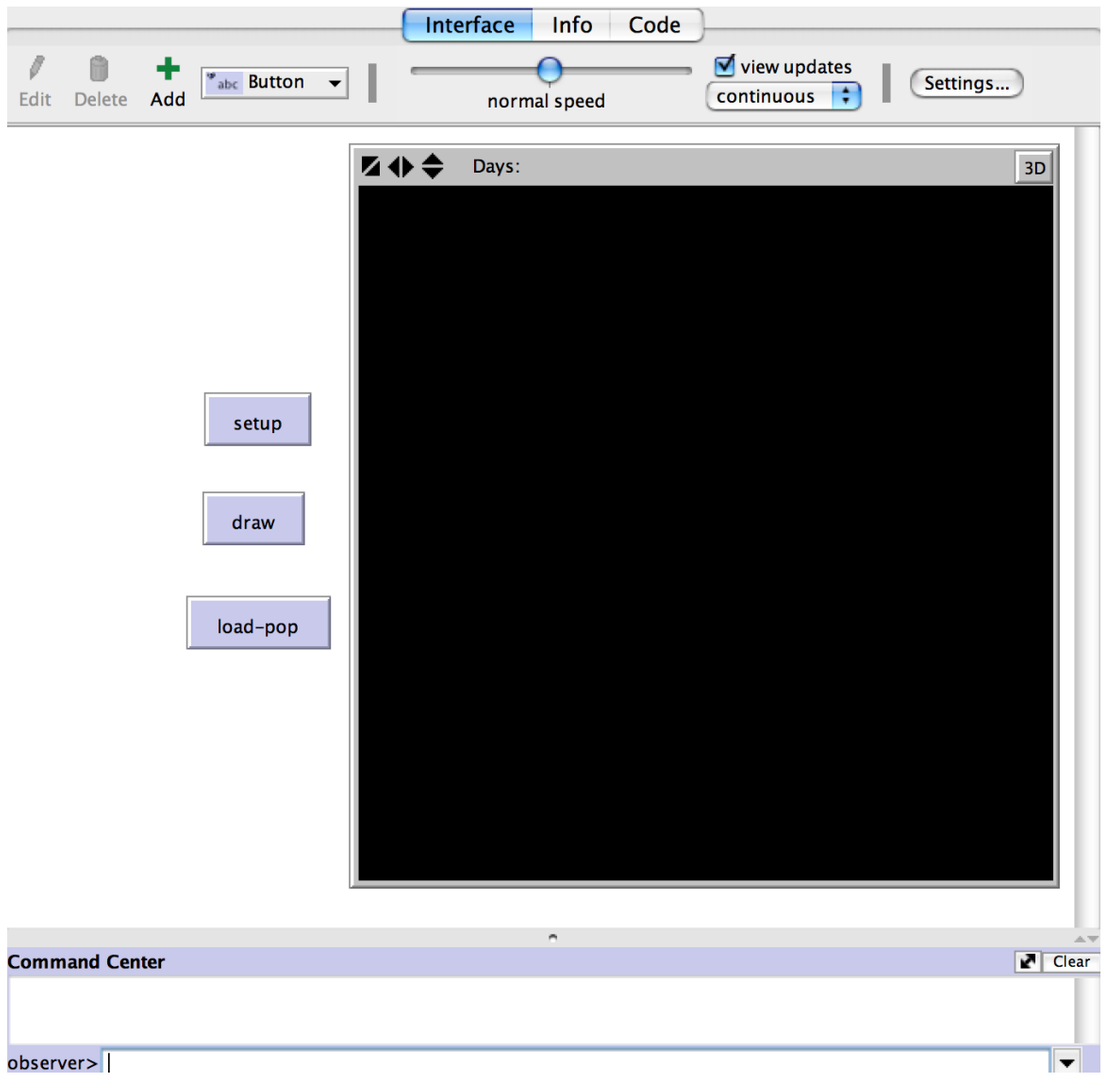


Here is the model

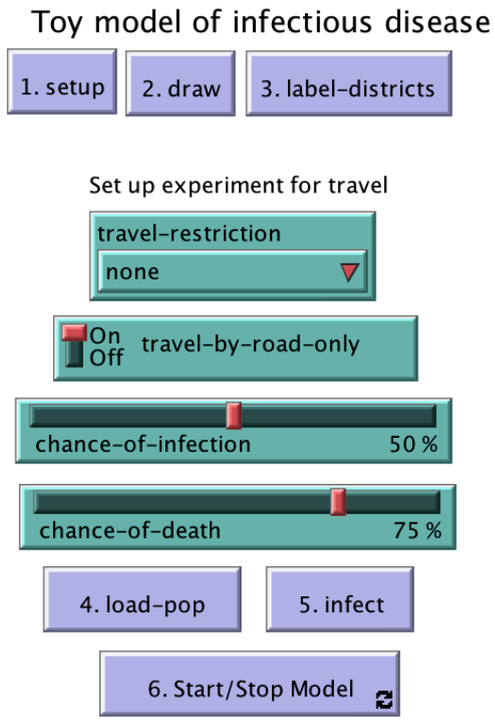
Model Settings



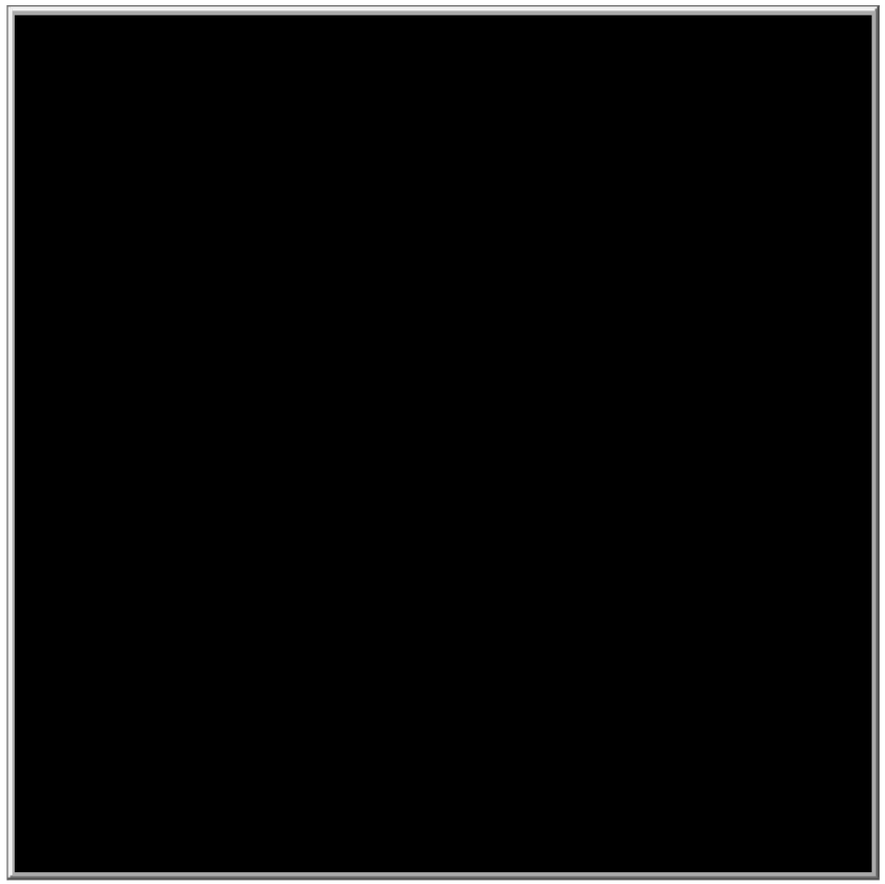
Blank Interface

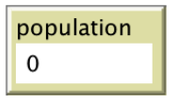


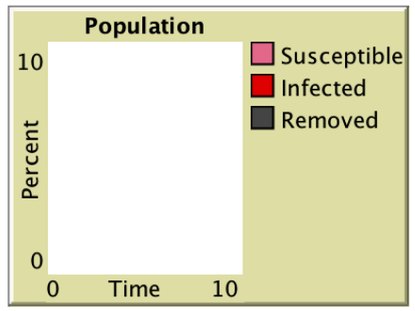
Model interface

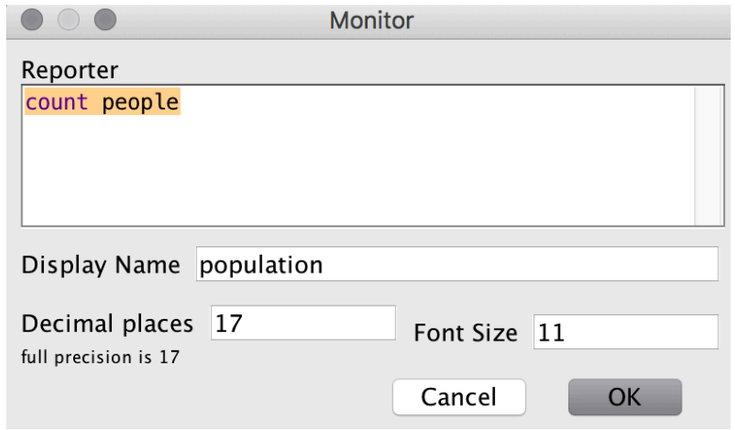


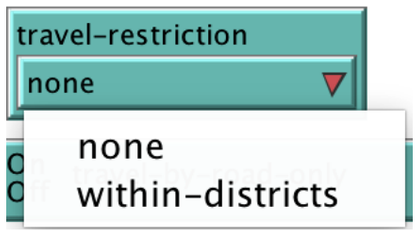
NOTE: To run the model over, adjust the settings on sliders etc, and then just click on buttons 4, 5, and 6

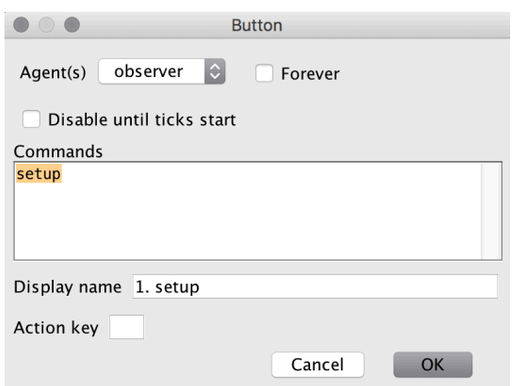


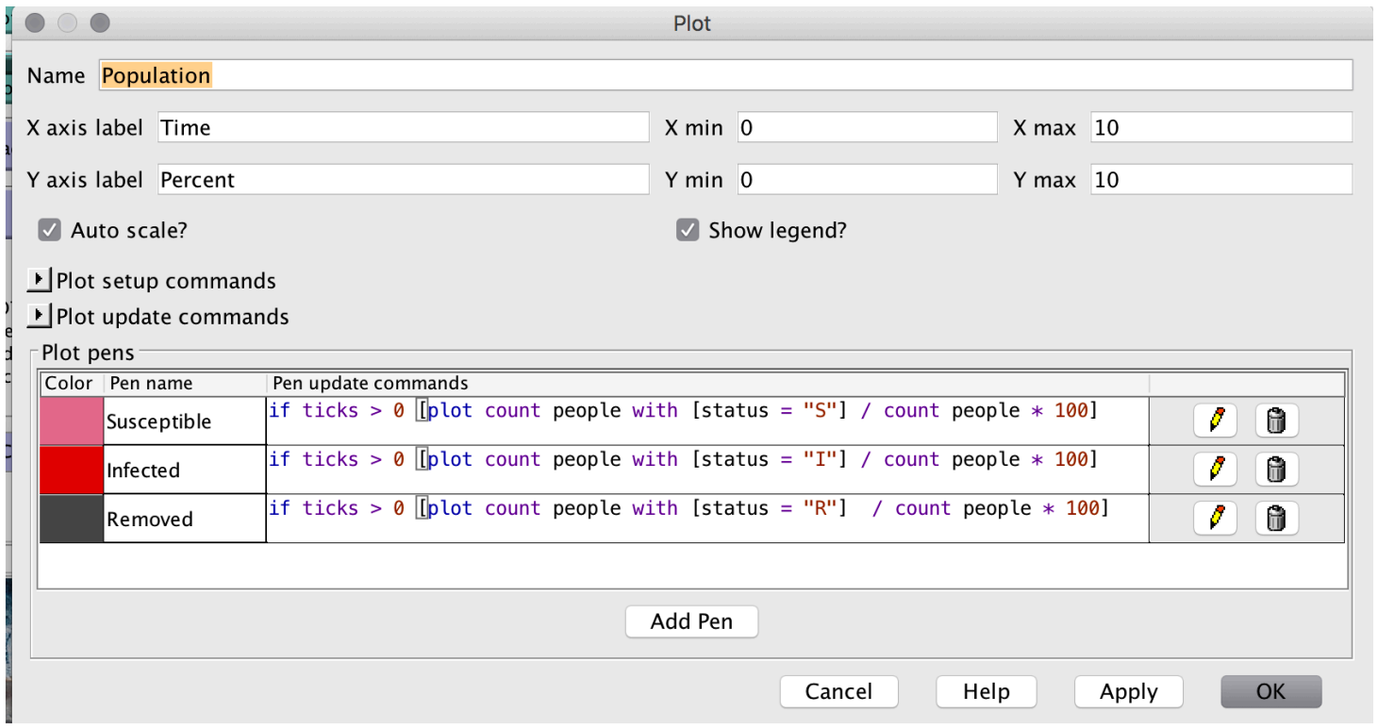












Here is all of the code for expanded (you can select/copy this)

```
extensions [ gis csv ]
globals [ sites roads districts SL]

patches-own [district-name confirmed population road-here]
turtles-own []

breed [people person]
people-own [district status time-infected]

breed [admin-labels admin-label]
admin-labels-own [name]

to setup
  clear-all
  reset-ticks
  gis:load-coordinate-system (word "data/projection.prj")
  set sites gis:load-dataset "data/Ebola_Treatment_Centers.shp"
  set roads gis:load-dataset "data/Sierra Leone_Roads.shp"
  set districts gis:load-dataset "data/Cases_at_Admin2_Level.shp"
  set SL gis:load-dataset "data/SL_Admin01.shp"
  end

to draw
clear-drawing
reset-ticks
gis:set-world-envelope (gis:envelope-union-of ;(gis:envelope-of sites)
                                ;(gis:envelope-of roads)
                                ;(gis:envelope-of districts)
                                (gis:envelope-of SL)
                                )
ask patches [set pcolor white]

gis:apply-coverage districts "GLOBAL_A_1" district-name
gis:apply-coverage districts "V_ADM2_C_3" confirmed

ask patches
  [ifelse (confirmed > 0)
    [set pcolor scale-color red confirmed 5000 0]
    [set pcolor white]
  ]

gis:set-drawing-color gray
gis:draw districts 2

gis:set-drawing-color red
gis:draw sites 3

gis:set-drawing-color black
gis:draw roads 1

ask patches
    [if gis:intersects? roads self
       [set road-here 1 ] ]
end

to label-districts
  ask admin-labels [die]
  foreach gis:feature-list-of districts
   [ ?1 -> let centroid gis:location-of gis:centroid-of ?1
     if not empty? centroid
      [create-admin-labels 1
        [set xcor item 0 centroid
          set ycor item 1 centroid
          set size 0
          set shape "circle"
          set color gray
          set name gis:property-value ?1 "GLOBAL_A_1"
          set label name
         set label-color black ]]]
end
```

```
to load-pop
  clear-plot
  reset-ticks

  ask turtles with [breed = people][die]
  reset-ticks
  file-open "data/SL_pop.csv"
    if not file-at-end? [let header csv:from-row file-read-line]
   while [not file-at-end?]
    [let row csv:from-row file-read-line
      ;let district_name item 0 row   let d_name item 0 row
      let district_name upper-case-string d_name ;convert text to uppercase
      if district_name = "WESTERN RURAL" [set district_name "WESTERN AREA RURAL"]
      if district_name = "WESTERN URBAN" [set district_name "WESTERN AREA URBAN"]
      let district_pop item 4 row
      let small_pop (district_pop / 10000)
      create-people small_pop
        [set district district_name  set shape "person" set size 2  set label ""  set status "S"  set
color pink
         move-to one-of patches with [district-name = [district] of myself]]]
    file-close
end

to-report upper-case-string [s]
   ifelse empty? s
   [report ""]
    [ report word (upper-case-char first s)
              (upper-case-string butfirst s) ]
end

to-report upper-case-char [c]
   let pos position c "abcdefghijklmnopqrstuvwxyz"
   ifelse pos = false
     [ report c ]
     [ report item pos "ABCDEFGHIJKLMNOPQRSTUVWXYZ" ]
end


to infect
  foreach gis:feature-list-of districts
    [ ?1 -> let d_name gis:property-value ?1 "GLOBAL_A_1"
      let infected gis:property-value ?1 "V_ADM2_C_3"
      let sm_infected round infected / 100 ;for visualization
      if sm_infected < 0 [set sm_infected 1]
      if any? people with [district = d_name]
        [ask n-of sm_infected people with [district = d_name]
          [set status "I" set color red ]]]
end

to go
  ask people [travel];[rt random 360 fd 1]

  ask people with [status = "I"]
       [infect-others
         ; to die
         let chance random 101
          if chance > (100 - chance-of-death)
            [set status "R" set size 0] ]
    tick
    if all? people [color = red] or not any? people with [color = red] [stop]
end
```

```
to infect-others
  let victims (turtle-set people-here people-on neighbors)

  ask victims
          [let chance random 101
            if chance > (100 - chance-of-infection)
              [set status "I"
                set color red]]
end
```

```
to travel
 if travel-restriction = "none" and travel-by-road-only = true
   [rt random 360
     ifelse not any? neighbors with [road-here = 1]
         [move-to min-one-of patches with [road-here = 1] [distance myself]]
      [ move-to one-of neighbors with [road-here = 1]]]

 if travel-restriction = "none" and travel-by-road-only = false
   [rt random 360
    fd 1]

 if travel-restriction = "within-districts" and travel-by-road-only = false
   [rt random 360
      ifelse patch-ahead 1 = nobody
        []
        [if [district-name] of patch-ahead 1 = district
         [fd 1 ] ]]

 if travel-restriction = "within-districts" and travel-by-road-only = true
   [rt random 360
      ifelse patch-ahead 1 = nobody
        []
        [if [district-name] of patch-ahead 1 = district and [road-here] of patch-ahead 1 = 1
         [fd 1 ] ]]
end
```