

# Canada's Provinces Geospatial Data Visualization Using NetLogo and GIS

Leogena Zhaka (Corresponding author)  
Faculty of Computer Science, University of Tirana  
Bulevardi Zogu I, Tirane, Albania  
E-mail: leogena.zhaka@fshn.edu.al

Endrit Xhina  
Faculty of Computer Science, University of Tirana  
Bulevardi Zogu I, Tirane, Albania  
E-mail: endri.xhina@fshn.edu.al

## Abstract

NetLogo is a free of charge multi-agent programmable modeling environment authored by Dr. Uri Wilensky and developed at Center for Connected Learning and Computer-Based Modeling (CCL) [1]. It is a relatively easy to learn-and-use software that allows researchers to set up and test their agent-based models [2]. One of the most powerful extensions that the software has is the NetLogo GIS Extension [3]. It provides the ability to load vector, and raster GIS data into the models [3].

The focus of this study is to research the steps to visualize the geospatial data of Canada's Provinces in NetLogo. Each Province will be assigned a color and the Province name will be shown as well. The GIS Extension will be used, and the dataset is a real-world dataset found on Statistics Canada [4]. This project has the potential to be used as a base for future ABM (Agent-Based Models) that will be developed in the NetLogo software related to Canada's Provinces' urban development.

**Keywords:** Geospatial, GIS, NetLogo, Canada, Provinces.

**DOI:** 10.7176/JSTR/7-07-02

## 1. Introduction

Agent-based modelling (ABM) is a technique used in many scientific domains such as social sciences and natural sciences. It requires to build a computational model for simulating actions and interactions of "agents," (both individual or collective entities) in a "world" in which those actions and interactions occur. These agents are programmed to be autonomous and be able to perceive the world. They help with solving problems that are difficult to be solved by individual agents. Another major benefit is their potential to help with understanding the behavior of a system and its outcomes [5].

### 1.1 NetLogo

NetLogo is a free-of-charge, open-source, multi-agent, programmable modelling environment authored by Dr. Uri Wilensky in 1990 and developed at Center for Connected Learning and Computer-Based Modeling (CCL) [1]. NetLogo has been under continuous development and continuous improvement since its first release. The latest version is 6.2.0 released in December 2020. It runs on the Java Virtual Machine, so it works on all major platforms. It is a relatively easy to learn-and-use [6] software that allows researchers to set up and test their agent-based models [2].

Figure 1 shows a simple example of a function implementation in NetLogo. The function is named "setup" and it executes two lines of code. The first is a NetLogo command that clears the environment from any previous setup and data. The second is a call to another function called "setup-maps".

From the interface (Figure 2) is easy to add and edit a new button by clicking on the "Add" menu item

and choosing the button type from the dropdown. Afterwards on the button edit interface (Figure 2) the command that the button executes is set to be the “setup” function that was implemented previously.

```
to setup
  clear-all
  setup-maps ;; load in the GIS data
end
```

Figure 1: Implementation of a simple function in NetLogo.

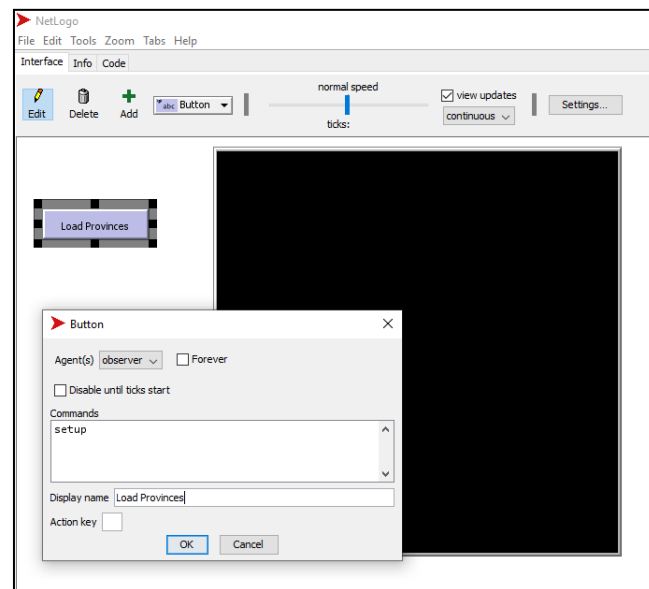


Figure 2: NetLogo interface

### 1.2 GIS extension

One of the most powerful extensions that this software offers, is the NetLogo GIS Extension [3]. It provides the ability to load vector, and raster GIS (Geographic Information Systems) data into the models. These models can use accurate maps as well as real-world datasets [3].

The GIS extension requires a map and its associated data to load the information in NetLogo. This task can be accomplished by using a GIS software. Two of the most used software among researchers and engineers are ArcGIS, which is a paid service [7], and qGIS, which is open source and free to use [8]. The usage of a GIS software allows the modification of a geographical map file to include the information needed by the NetLogo model. These software also provide the option of conversion of the map to a supported extension. Currently, the GIS extension supports the use of ESRI shapefiles (.shp) for vector files, which is the most common type of file used in software, and ESRI ASCII Grid files (.asc, and .grd) for raster data [3].

When using a map in a model the extension [gis] must be included at the beginning of the model code as shown in Figure 3. The extension provides Dataset, RasterDataset, VectorDataset, Coordinate System, and Drawing Primitives [3].

```
extensions [ gis ] ;; the gis extension allows us to read standard GIS data
```

Figure 3: Including GIS in a project

## 2. Canada’s Provinces geospatial data visualization

To create a real-world base in NetLogo to be used by agent-based models focused on Canada’s Provinces first we will need to find a map of Canada including province borders. These types of maps are usually found in government websites and in this study, we used the map that includes information dataset found on Statistics Canada website [4].

As a first step we added the GIS extension to our model (Figure 3). Then we specified the dimensions of the window representing the real world (Figure 4). In our case was preferred a wide view given the proportions of the map of Canada.

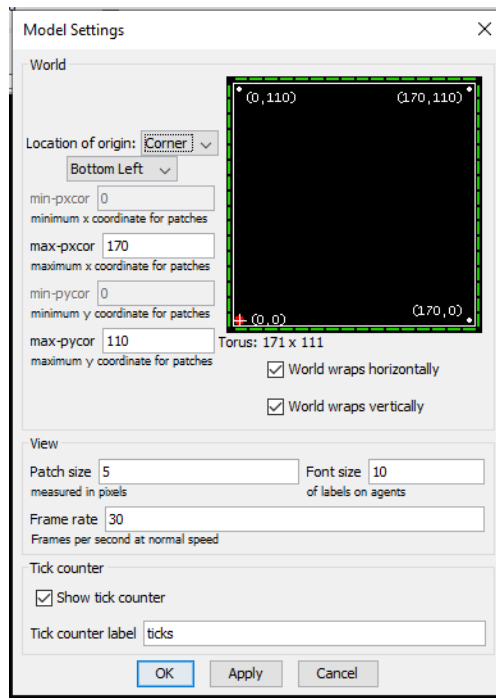


Figure 4: Setting the Coordinative System to visualize the map

```

breed [ province-labels province-label ]

patches-own [
  index
  province-id                ;; province/territory name
]

globals [
  provinces-dataset          ;; GIS data file
]
    
```

Figure 5: Setting variables and breed

We define one global variable *provinces-dataset* that will hold the imported GIS data (Figure 5). In the Interface part of NetLogo a button is added, and it is connected to a function named *setup* (Figure 1). The setup function calls another function, *setup-maps*, that includes the GIS primitives needed to accomplish the map visualization.

GIS primitives are called by using the prefix *gis* followed by a colon and the primitive's name. *gis:load-dataset* is used to assign the dataset to the global variable *provinces-dataset*. Command to get the file data to the dataset: *set provinces-dataset gis:load-dataset "data/canada.shp"* (Figure 6).

The world envelope is set to ensure the vector data is properly associated with the model's world (Figure 6). Command used: *gis:set-world-envelope (gis:envelope-of provinces-dataset)*

A first drawing is applied to visualize the provinces lined with a white border. Commands used: *gis:set-drawing-color white* and *gis:draw provinces-dataset 2*.

A *foreach* loop will assign data to the patches that intersect the vector features, in our case polygons, that represent the provinces. Before the foreach loop, a local index variable is defined and initialized to 1. It gets incremented by 1 at the end of each cycle and will be the *index* of the patches.

The *foreach* loops the *province-dataset* using the command *gis:feature-list-of* (Figure 6).

```

to setup-maps
  clear-drawing
  ; Load all of our datasets
  set provinces-dataset gis:load-dataset "data/canada.shp"
  ; Set the world envelope to our dataset's envelope
  gis:set-world-envelope (gis:envelope-of provinces-dataset)

  ; Drawing polygon data from a shapefile, and optionally loading some
  ask province-labels [ die ]
  gis:set-drawing-color white
  gis:draw provinces-dataset 2
  let i 1;
  foreach gis:feature-list-of provinces-dataset [ vector-feature ->
    let centroid-province gis:location-of gis:centroid-of vector-feature
    ; centroid will be an empty list if it lies outside the bounds
    ; of the current NetLogo world, as defined by our current GIS
    ; coordinate transformation
    if not empty? centroid-province
    [ create-province-labels 1
      [ set xcor item 0 centroid-province
        set ycor item 1 centroid-province
        set size 0
        set label-color black
        set label gis:property-value vector-feature "PRENAME"
        set province-id gis:property-value vector-feature "PRUID"
        set index i
      ]
    ]
    set i (i + 1)
  ]
]
    
```

Figure 6: Map setup

```

ask patches with [ index > 0 ] [
  let mod-value ( index mod 3 )
  ifelse mod-value = 0 [
    gis:set-drawing-color blue
    gis:fill item (index - 1) gis:feature-list-of provinces-dataset 2.0
  ]
  [
    ifelse mod-value = 1 [
      gis:set-drawing-color red
      gis:fill item (index - 1) gis:feature-list-of provinces-dataset 2.0
    ]
    [
      gis:set-drawing-color yellow
      gis:fill item (index - 1) gis:feature-list-of provinces-dataset 2.0
    ]
  ]
]
]
    
```

Figure 7: Filling color specification

Each centroid will have a label that is the name of the province that gets the value from “PRENAME” (Province English Name) and the id from “PRUID” (Province Unique ID), both values found in the shape file associated with the provinces. The color of the label is set to be black. Commands: *set label-color black*, *set label gis:property-value vector-feature "PRENAME"*, and *set province-id gis:property-value vector-feature "PRUID"*.

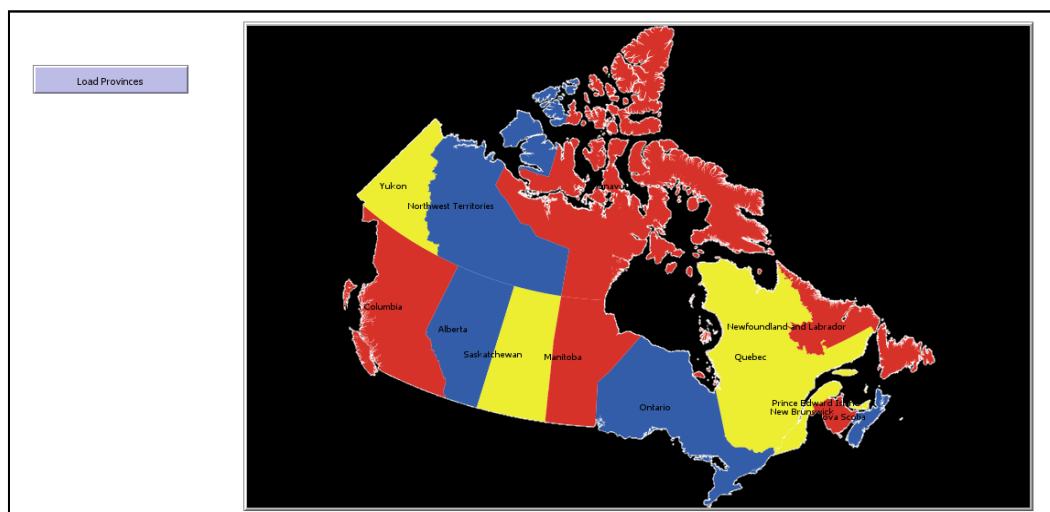


Figure 8: Canada's Provinces

The patches intersecting each feature in the list have a variable *centroid-province* that refers to the location of the centroid of the polygon that overlays them. Command: *let centroid-province gis:location-of gis:centroid-of vector-feature*.

The next step is filling each province with a color. The syntax differs slightly since we are accessing each polygon individually instead of the entire list of features. Each centroid patch will fill the polygon it controls to one of the colors, yellow, blue, or red, depending on the *index-id* value that the patch has assigned. The color is assigned by using the *gis:set-drawing-color*. To fill the polygon *gis:fill* primitive is used. Each polygon is contained in the feature list of the province-dataset, the *index-id* must be subtracted by 1 when accessing it in this procedure due to NetLogo lists beginning at element 0 (Figure 7).

The result of clicking the Load Provinces button will be the map of Canada's Provinces (Figure 8) outlined with a white border, filled with red, blue, or yellow, specified by the module of three of the Province's assigned index (Figure 7), and black Province's name label.

### 3. Conclusion

NetLogo is an easy-to-use agent-based modelling software. It allows a developer to implement a model of interacting agents in a simulated world/environment. One of the libraries that can be used to increase the capabilities of a model is the NetLogo GIS extension. This extension provides primitives that allow a representation of a real-world map and dataset in a NetLogo model to be used as an environment for the agents to exist and interact. In this paper we used a shape file of Canada's Provinces found in Statistics Canada to create the world base with color filled provinces and labels showing the province name. The map visualization in NetLogo increases the look and the potential for geospatial related model simulations.

### Acknowledgements

The authors would like to thank Mr Erni PAPAKRONI for their technical assistance in the software research phase of this article.

### References

- [1] NetLogo <https://ccl.northwestern.edu/netlogo/>
- [2] Broday Walker and Tina V. Johnson (2019). NetLogo and GIS: A Powerful Combination. *Proceedings of 34th International Conference on Computers and Their Applications*, Volume 58, Pages 257 - 264.
- [3] NetLogo GIS Extension <https://ccl.northwestern.edu/netlogo/6.0.1/docs/gis.html>
- [4] Statistics Canada: Census Program, Alternative format, [https://www12.statcan.gc.ca/census-recensement/alternative\\_alternatif.cfm?l=eng&dispext=zip&teng=lpr\\_000b16a\\_e.zip&k=%20%20%20%2027960&loc=http://www12.statcan.gc.ca/census-recensement/2011/geo/bound-limit/files-fichiers/2016/lpr\\_000b16a\\_e.zip](https://www12.statcan.gc.ca/census-recensement/alternative_alternatif.cfm?l=eng&dispext=zip&teng=lpr_000b16a_e.zip&k=%20%20%20%2027960&loc=http://www12.statcan.gc.ca/census-recensement/2011/geo/bound-limit/files-fichiers/2016/lpr_000b16a_e.zip)
- [5] Shekhar S., Xiong H. (2008) Agent-Based Models. In: Shekhar S., Xiong H. (eds) *Encyclopedia of GIS*. Springer, Boston, MA., [https://doi.org/10.1007/978-0-387-35973-1\\_39](https://doi.org/10.1007/978-0-387-35973-1_39).
- [6] Seth Tisue and Uri Wilensky (2004). Netlogo: A simple environment for modeling complexity. *International Conference on Complex Systems*, Pages 16–21,
- [7] ArcGIS <https://www.esri.com/en-us/arcgis/products/arcgis-online/overview>
- [8] qGIS <https://qgis.org/en/site/>