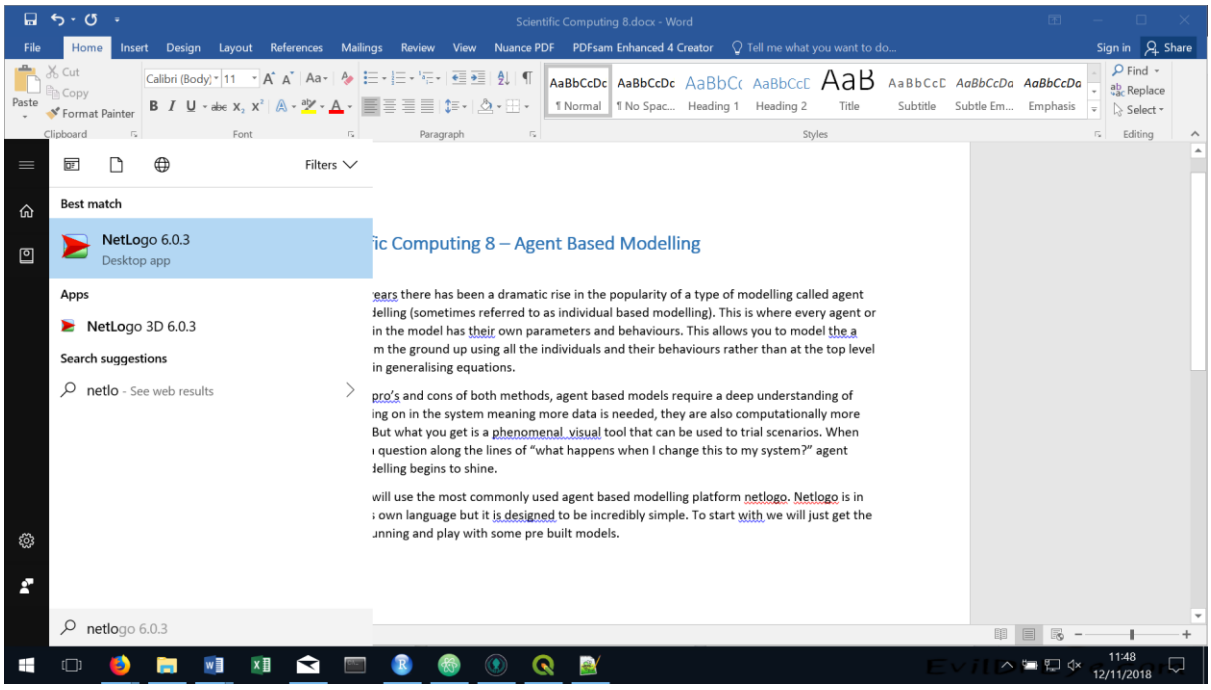
Scientific Computing 8 – Agent Based Modelling

In recent years there has been a dramatic rise in the popularity of a type of modelling called agent based modelling (sometimes referred to as individual based modelling). This is where every agent or individual in the model has their own parameters and behaviours. This allows you to model the a system from the ground up using all the individuals and their behaviours rather than at the top level described in generalising equations.

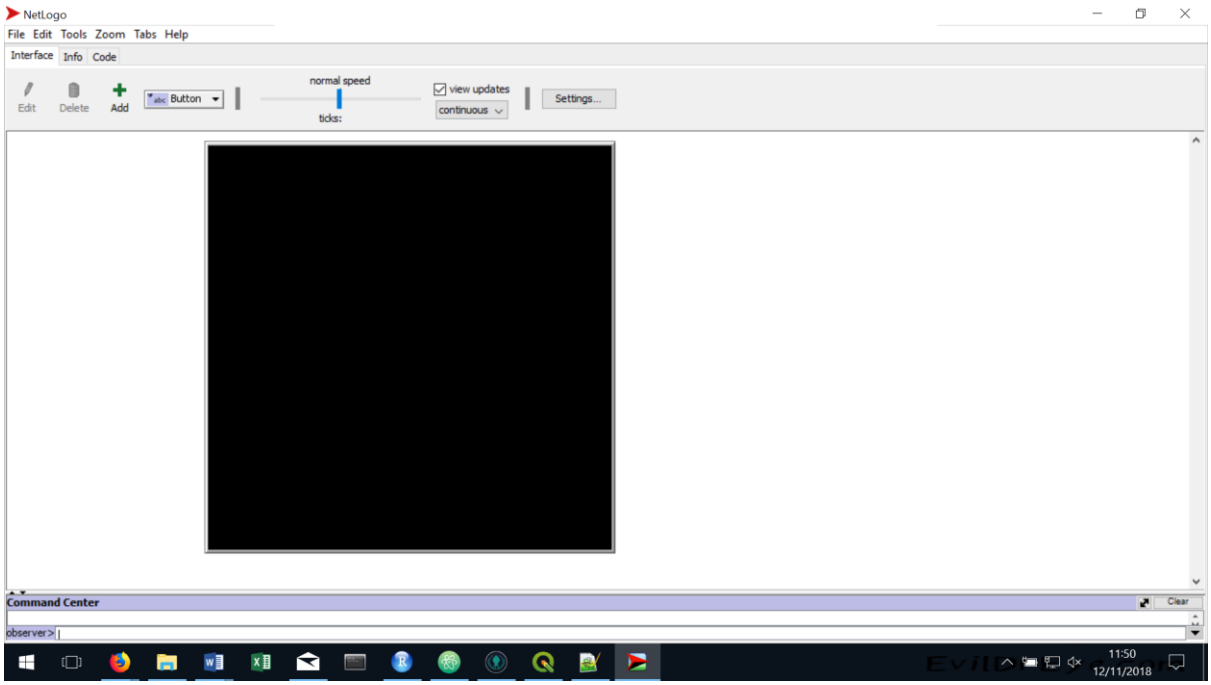
There are pro's and cons of both methods, agent based models require a deep understanding of what is going on in the system meaning more data is needed, they are also computationally more intensive. But what you get is a phenomenal visual tool that can be used to trial scenarios. When you have a question along the lines of "what happens when I change this to my system?" agent based modelling begins to shine.

Today we will use the most commonly used agent based modelling platform netlogo. Netlogo is in essence its own language but it is designed to be incredibly simple. To start with we will just get the program running and play with some pre built models.

Netlogo can be launched from the start bar easily

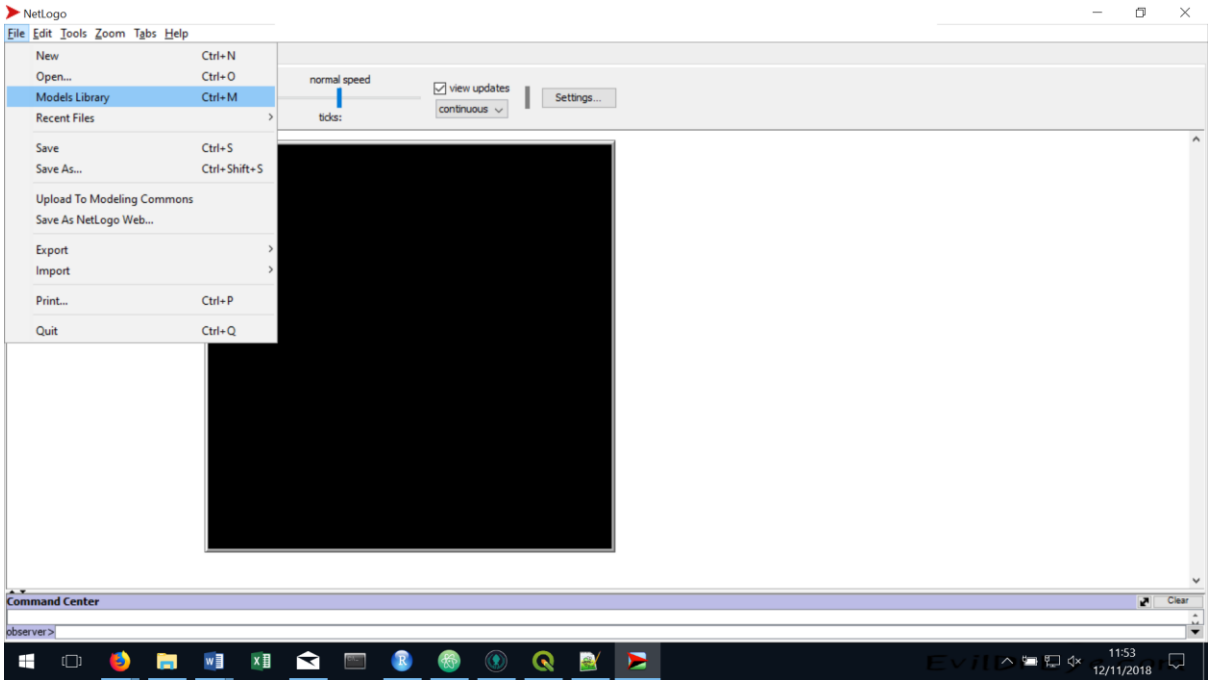


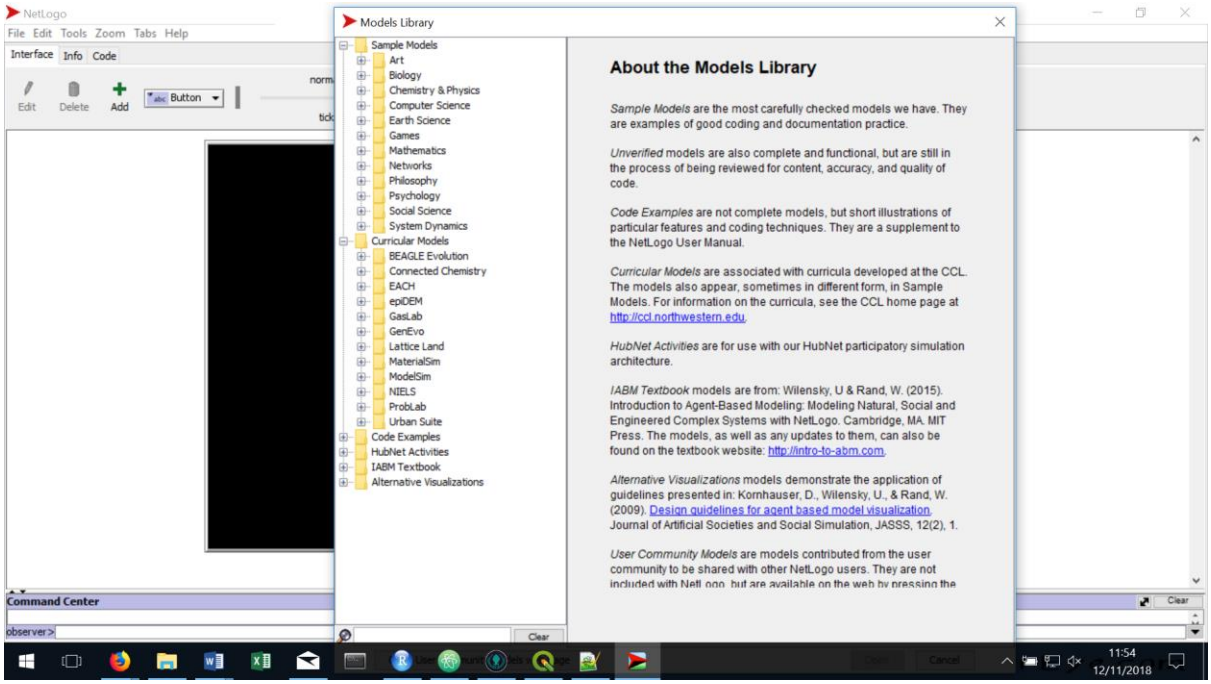
It will look like this when it is launched



The first thing we are going do to is to look at a previously created model.

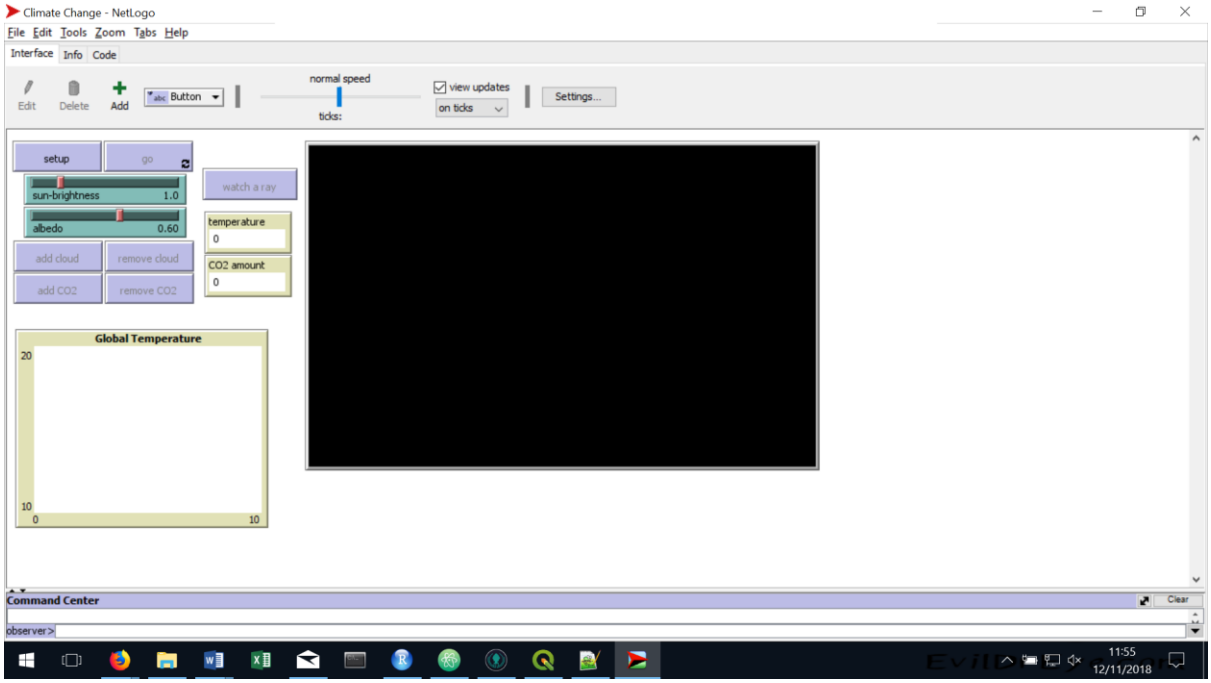
There are hundreds of models available in the model library.



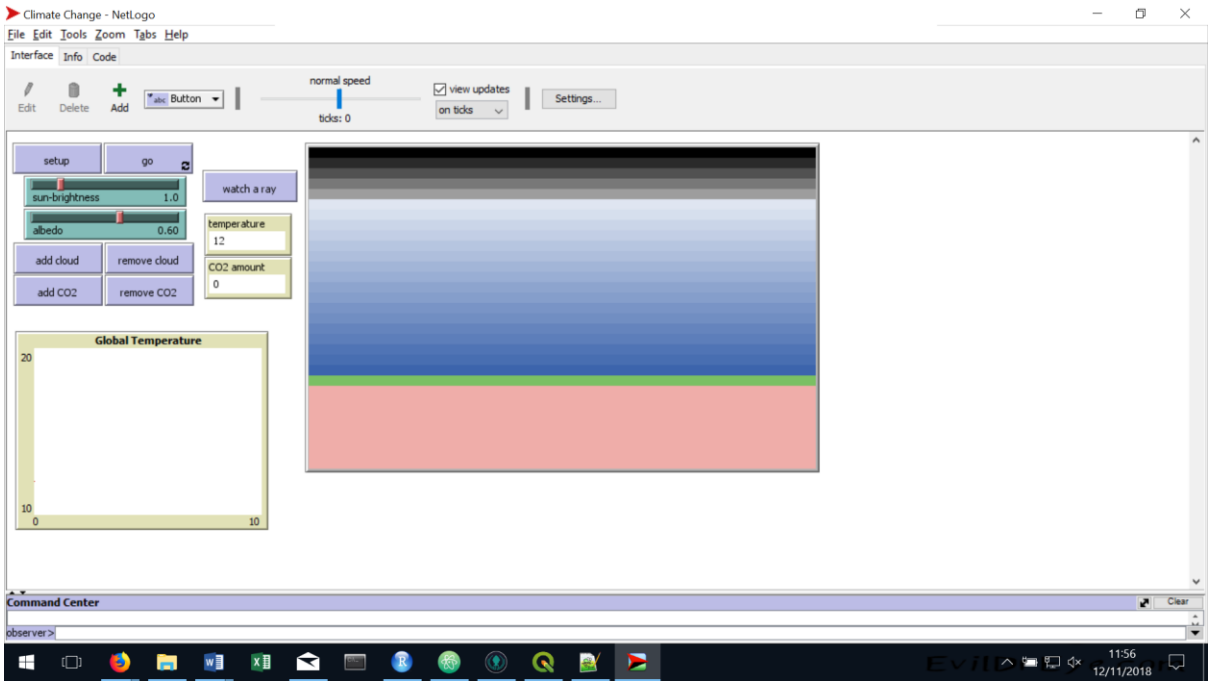


Open up the Climate change model inside the earth science folder.

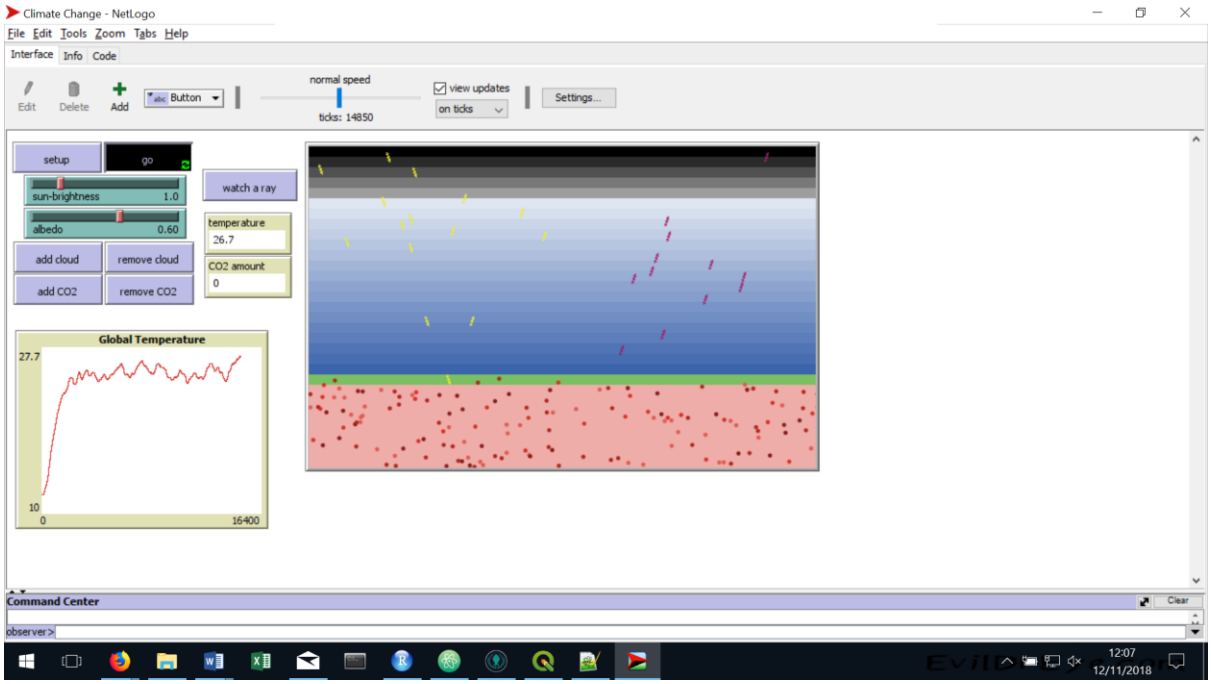
The model will initially look like this until we press the setup button. All netlogo models use a setup command that sets up the environment, the agents and the initial parameters.



Once set up the model will look like this.



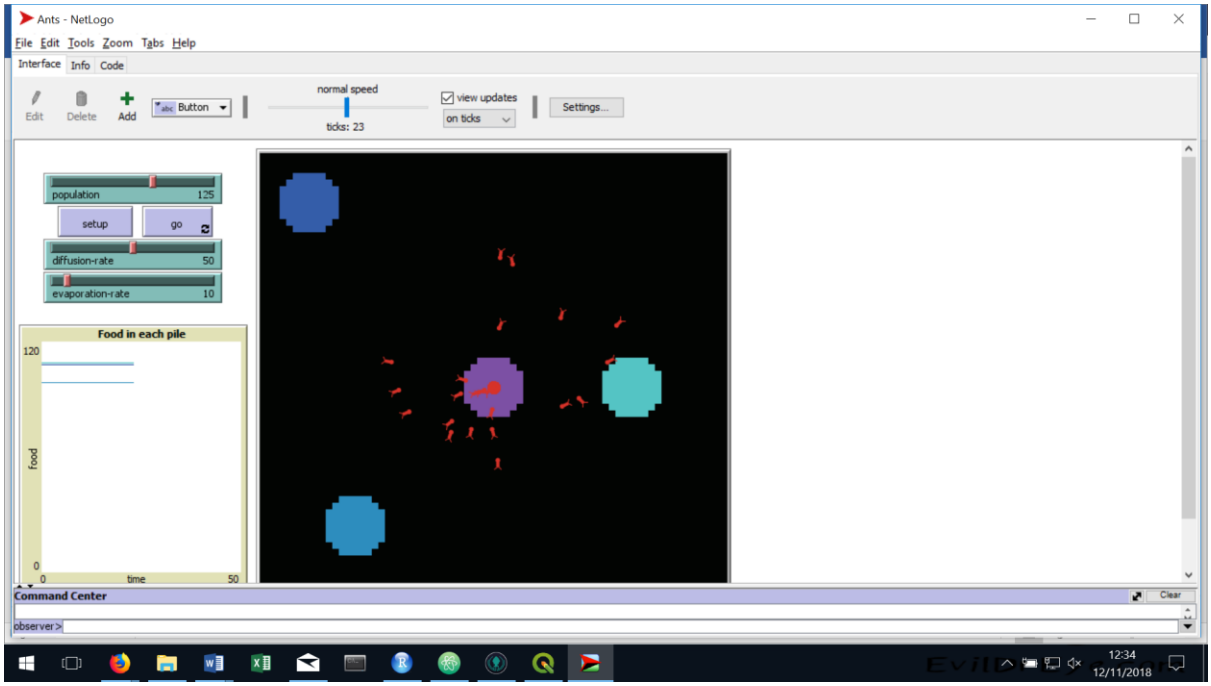
Press go and watch the model run. It will take a while to reach its fixation point somewhere around 25 degrees.



You can then begin to trial different things like adding clouds or CO2, changing the suns brightness etc.

Here we can see the benefits of agent based modelling when it comes to trialling things.

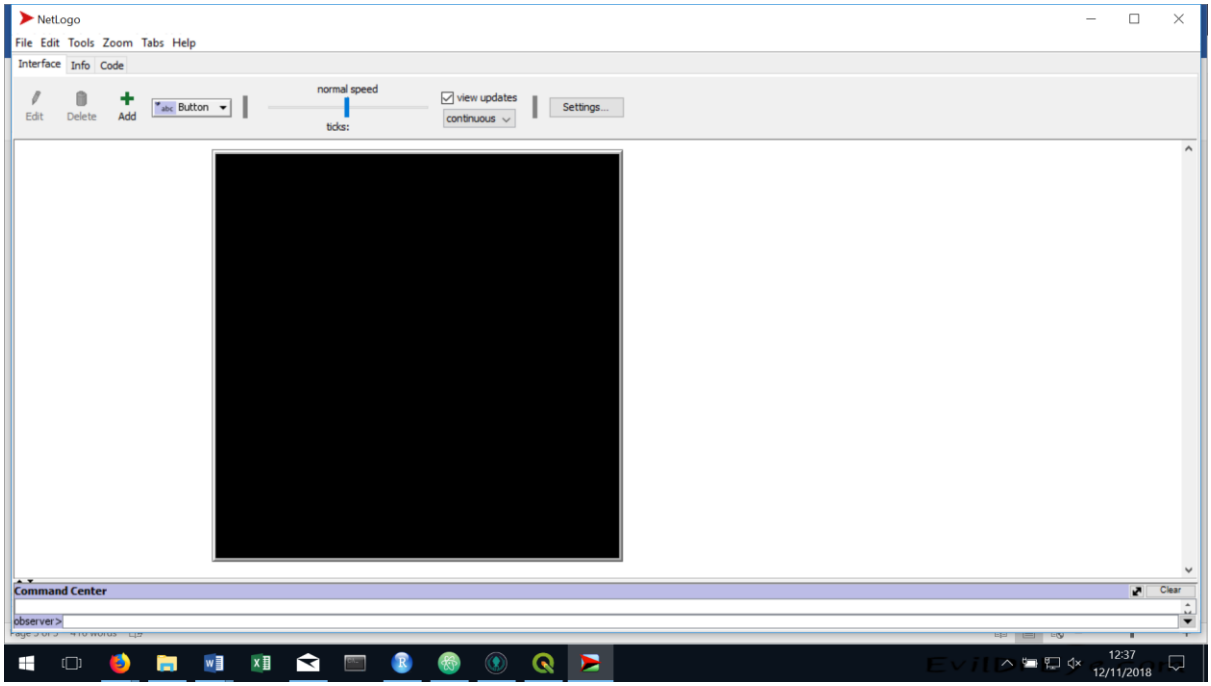
Let’s try a different model. Open up the Ants model inside of the biology folder.



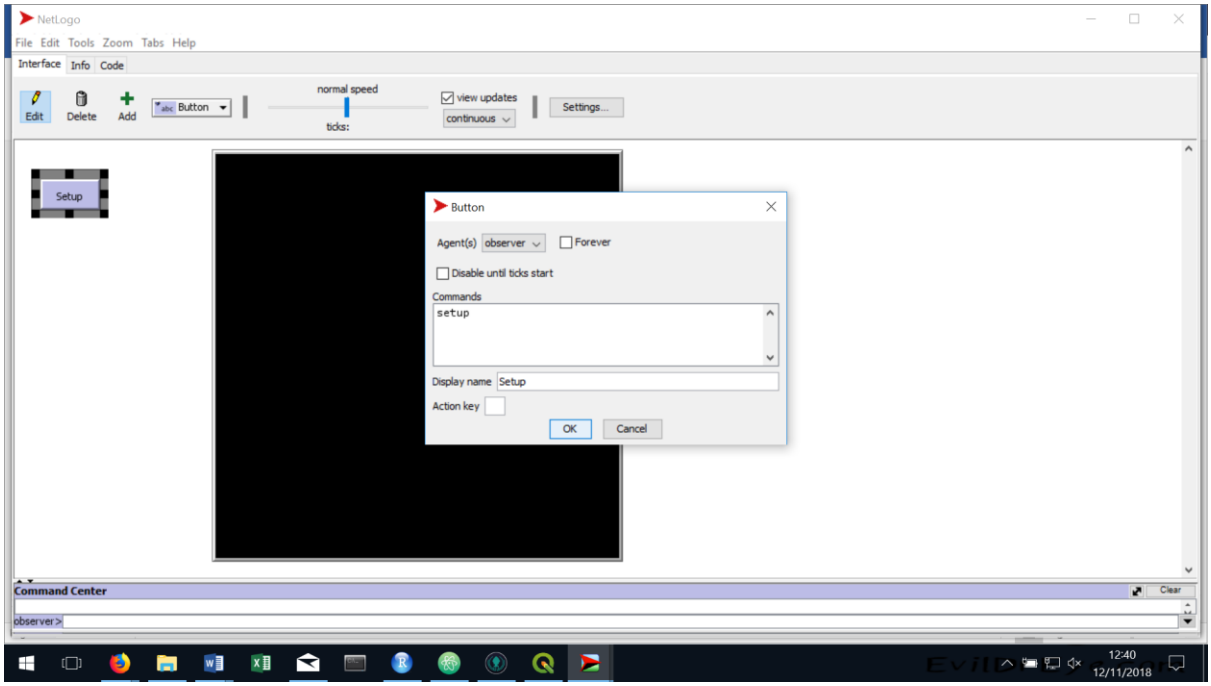
In this model there are ants and food sources, the ants then leave a pheromone trail once they have food which other ants can follow. Play around with the settings to see what you can do.

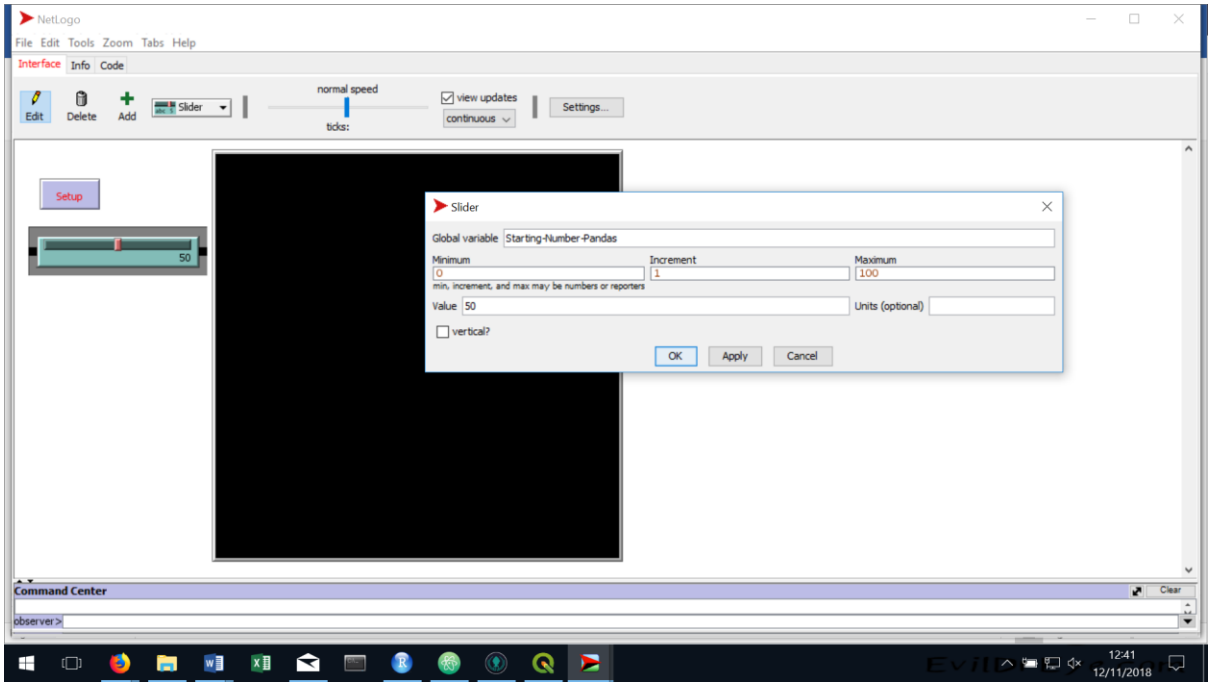
What we are going to do is build our red panda cabage model.

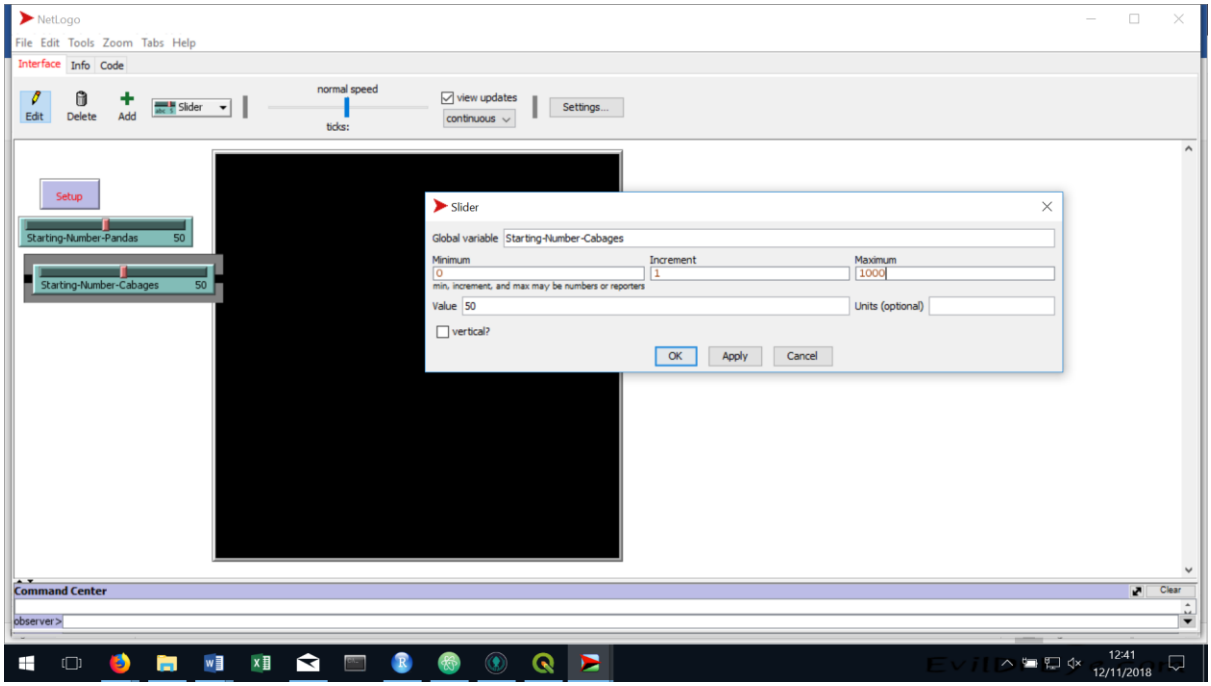
Open a new model



Lets start by throwing some starting interface tools in there. Click the Add button and add in a button to setup and two sliders to decide the starting number of pandas and cabages


NetLogo
File Edit Tools Zoom Tabs Help
Interface Info Code
Edit
Delete
Add
Button
normal speed
ticks:
view updates
continuous
Settings...
Setup
Button
Agent(s) observer
Forever
Disable until ticks start
Commands
setup
Display name Setup
Action key
OK
Cancel
Command Center
Clear
observer>
12:40
12/11/2018


NetLogo
File Edit Tools Zoom Tabs Help
Interface Info Code
Edit
Delete
Add
Slider
normal speed
ticks:
view updates
continuous
Settings...
Setup
50
Slider
Global variable Starting-Number-Pandas
Minimum
0
Increment
1
Maximum
100
min, increment, and max may be numbers or reporters
Value 50
Units (optional)
vertical?
OK
Apply
Cancel
Command Center
Clear
observer>
12:41
12/11/2018



These will give errors initially because we haven't written any code.

Code in netlogo goes in the code tab and is generally done inside functions.

The first thing we will need to do is define our globals, breeds and agent parameters

```
breed [Pandas Panda]

turtles-own [energy]

patches-own[Cabbage]
```

Netlogo works on the basis of turtles and patches, turtles are agents that are active in the environment and patches are the agents of the environment. In this case we are making Pandas be a turtle and the cabbages be part of the environment. We then say that the pandas will have a variable called energy and the patches will have a variable called cabbages.

Next we will write the setup function.

```
to setup
  clear-all ; this cleans up any old models
  ask patches [set pcolor brown
  set Cabbage False] ;here we are setting all the patches to be brown and that they are not cabbages


  create-Pandas Starting-Number-Pandas[
    set shape "squirrel"
    set color red
```

```
   set size 1.5
   set energy 100
   setxy random-xcor random-ycor]; here we create our pandas, we specify their size shape colour
and location as well as giving them a starting energy of 100


  ask n-of Starting-Number-Cabbages patches[
   set pcolor green
   set Cabbage True];here we are making our starting number of Cabbages
  reset-ticks
end
```

Press setup and see what happens

next we need to write a go function. The go function is all the commands that are executed each time step or tick as it is called. First of all we need to write the functions that are going to be executed by go. i.e what the pandas and cabbages are going to do.

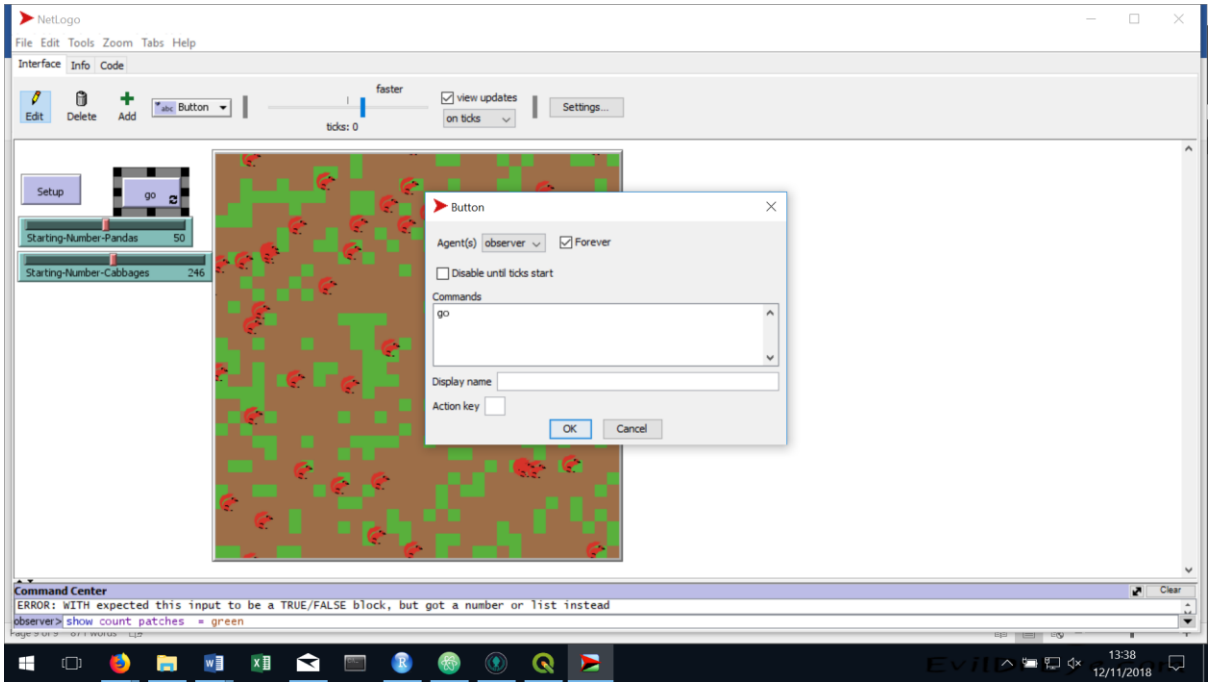
```
to move
  rt random 50
  lt random 50
  fd 1
  set energy energy - 1
end ; this just changes direction and then tells them to move forward 1 using up energy


to eat-cabbage
  if pcolor = green [
   set pcolor brown
   set energy energy + 2 ; sheep gain energy by eating
  ]
end ; if a panda is on a cabbage eat it and gain energy


to death
  if energy < 0 [ die ]
end ; if they run out of food die
```

```
to reproduce
 if random-float 100 < 1
 [hatch 1 [ rt random-float 360 fd 1 ]]
end ; if they roll a random number create a new panda.
```

Then we need to put all this in our go command

```
to go
 ask turtles[
  move
  eat-cabbage
  death
   reproduce]
 ask n-of ((count patches with [pcolor = green]) * 0.1) patches[
   set pcolor green]
 tick
end
```

now just add a new button called go make sure forever is ticked and then when you press go it should go.



Well done you have created a simple agent based model. Play around with the numbers add more sliders to control more things like reproductive rate etc. If you are feeling brave add a predator that eats red pandas.