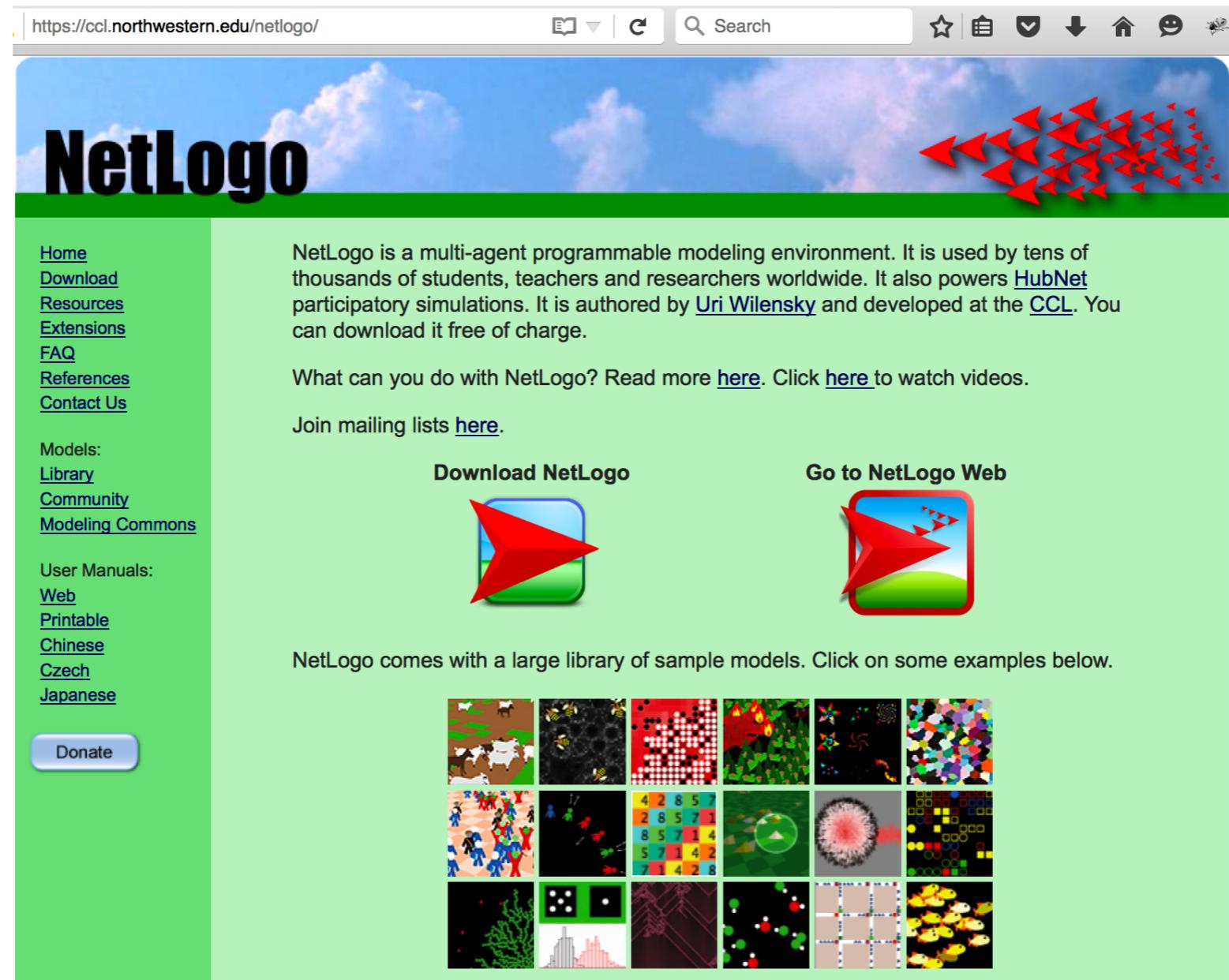


Intro to NetLogo: Let's explore



The screenshot shows the official NetLogo website at <https://ccl.northwestern.edu/netlogo/>. The page features a green sidebar on the left with links to Home, Download, Resources, Extensions, FAQ, References, and Contact Us. It also lists Models (Library, Community, Modeling Commons) and User Manuals (Web, Printable, Chinese, Czech, Japanese). A 'Donate' button is located at the bottom of the sidebar. The main content area has a blue header with the NetLogo logo and a flock of red arrows. The text explains that NetLogo is a multi-agent programmable modeling environment used worldwide, authored by Uri Wilensky and developed at the CCL. It links to more information and videos. Below this, there are 'Download NetLogo' and 'Go to NetLogo Web' buttons with corresponding icons. A grid of thumbnail images shows various sample models, including a landscape with sheep, a bee colony, a grid pattern, a forest scene, a star pattern, a complex collage, a grid of people, a grid of numbers, a circular pattern, a grid of squares, a grid of dots, a grid of trees, a grid of mountains, a grid of dots, a grid of squares, and a grid of colorful shapes.

Let's explore NetLogo

What is NetLogo?

Who already has downloaded NetLogo?

What version do you have?

Let's explore NetLogo

What is NetLogo?

NetLogo is a free open source agent based modeling and simulation application. It comes with an extensive library of models for various domains and topics. NetLogo supports the creation of agent based models (ABM) with agents in the form of turtles, patches, links, and the observer. Users can interact with and explore scenarios and emergent phenomena by modifying model parameters and observing how the model's display and results change over time as a result of user or agent interactions.

Who already has downloaded NetLogo?

What version do you have?

Module Objectives

Explore the NetLogo application

- Interface (IDE), Info, Code
- User controls and parameters
- Environment/World
- Agents, Turtles, Patches
- Ticks/Time

Review models in the Model Library

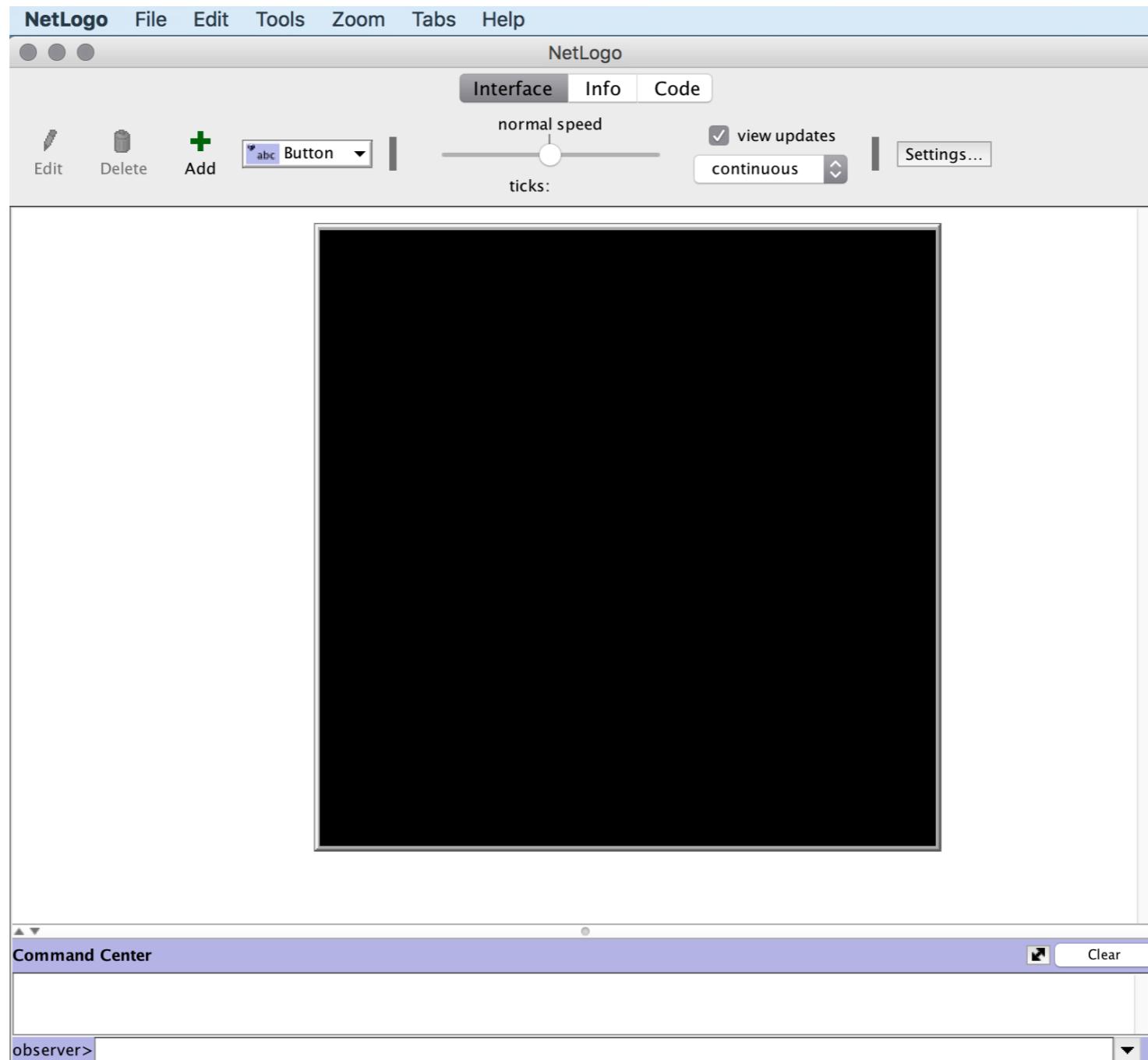
Get NetLogo

Go to <https://ccl.northwestern.edu/netlogo/>

Install NetLogo

- Click on Download NetLogo
- Enter info and click Download
- Click on the Download link for your computer
(most are 64-bit now)
- Save the download package
- Install NetLogo by double clicking on the package

Open Netlogo

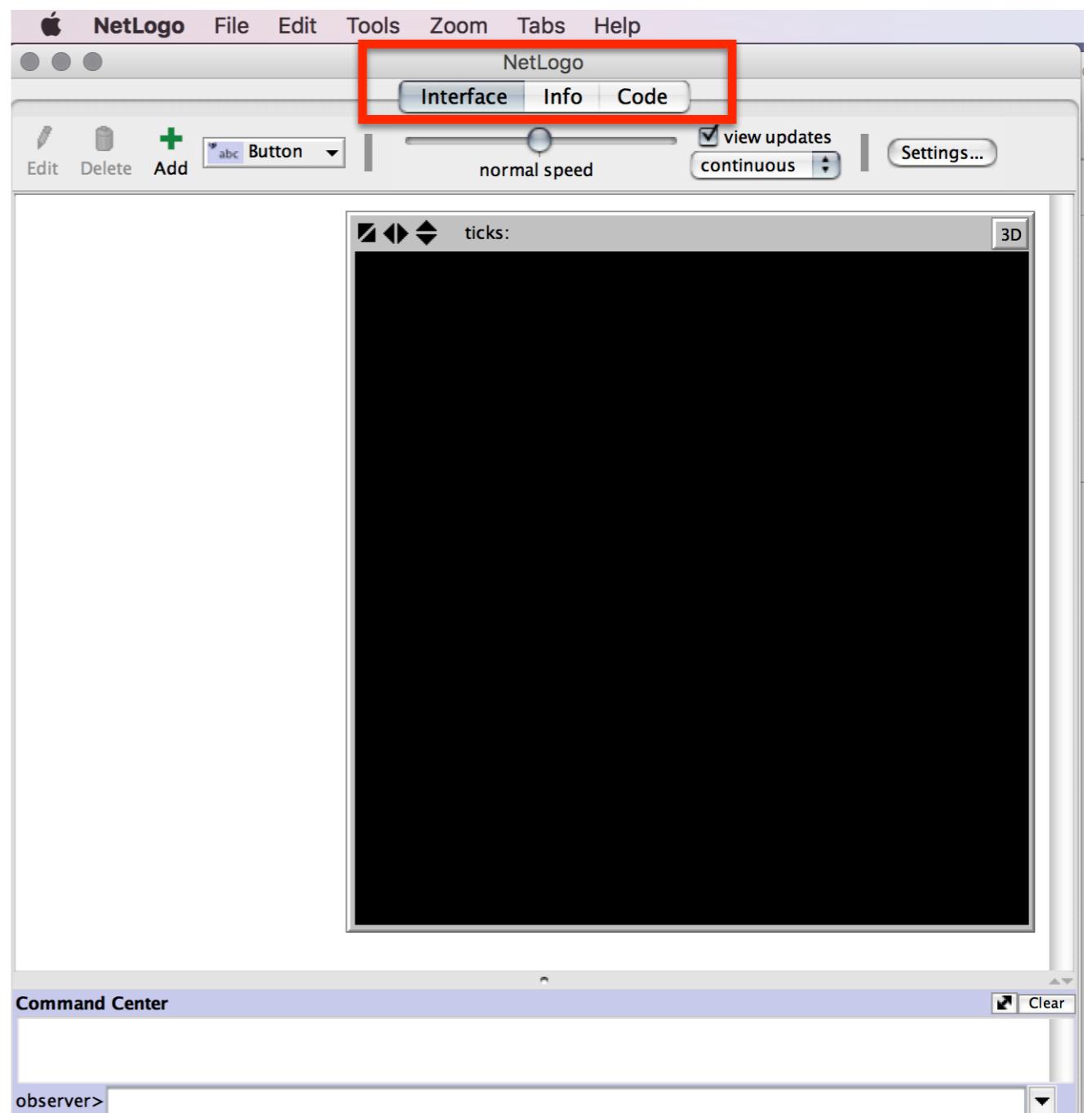


Pro Tip: You can download multiple NetLogos and versions. You can have multiple NetLogos open to view and run models at same time.

NetLogo IDE

Integrated Development Environment (IDE)

- Interface (user controls and display)
- Info (meta about the model)
- Code (model syntax)

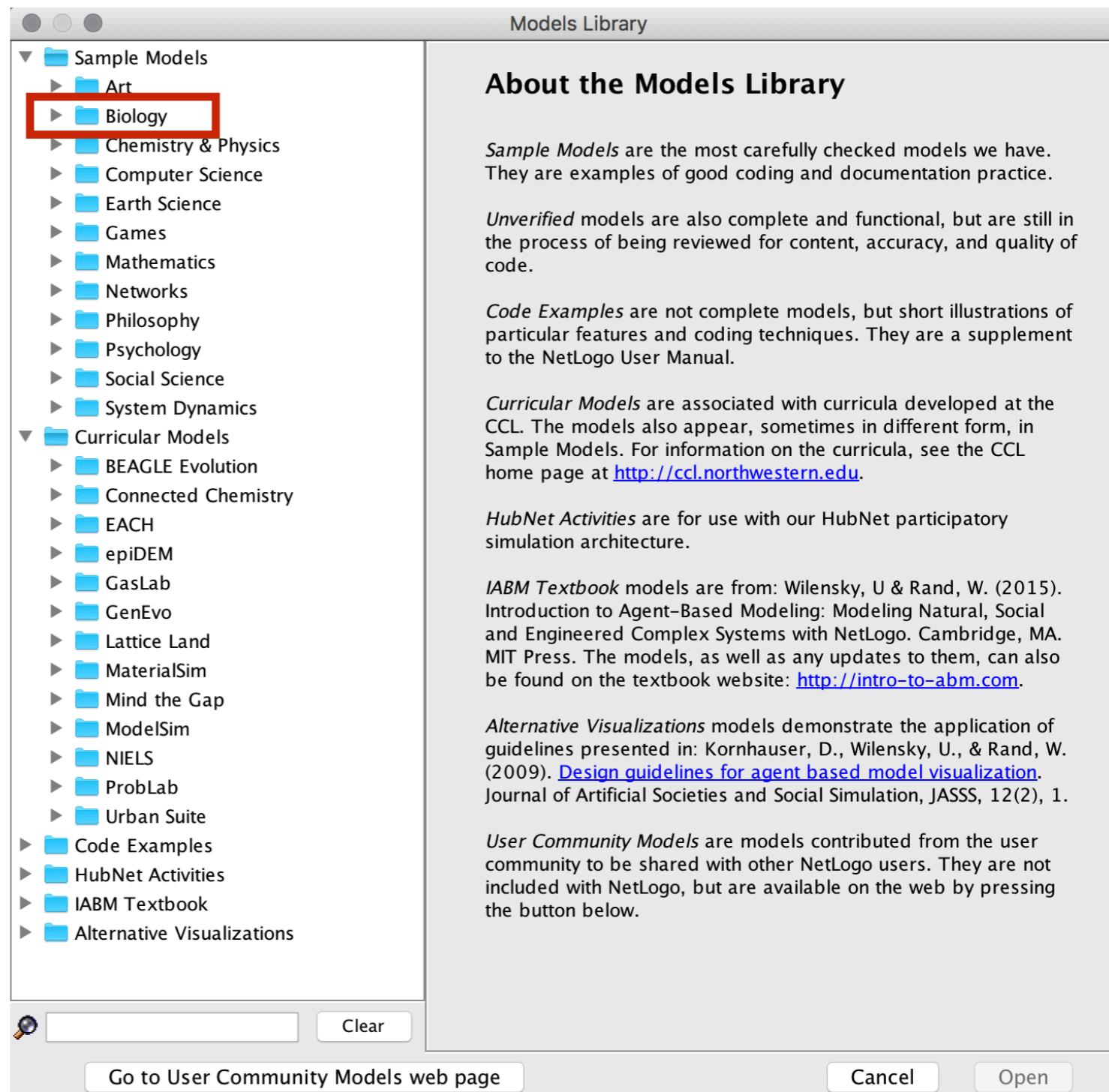


Models Library

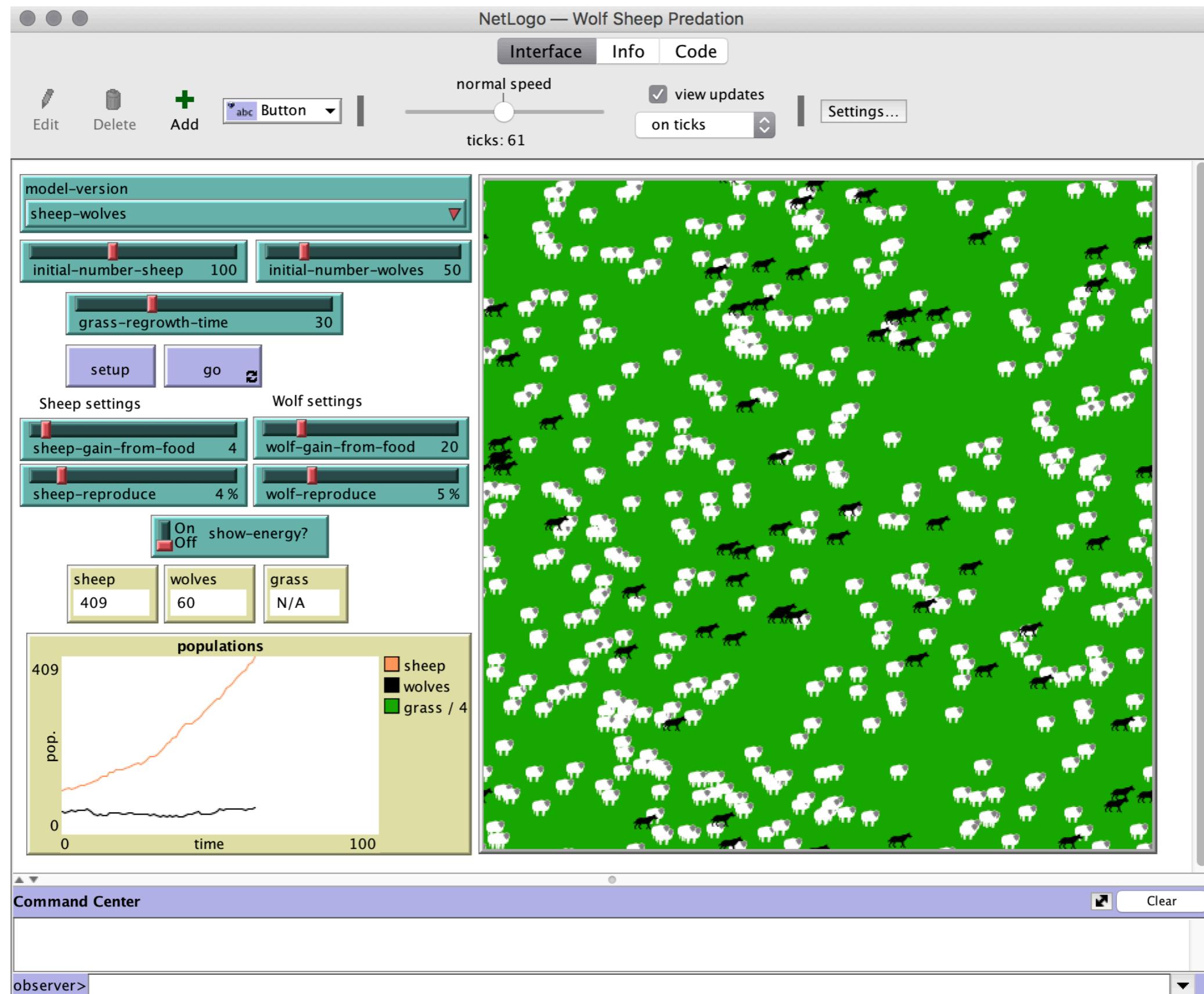
Let's explore a model:

Open the Wolf Sheep
Predation model

- File>Models Library
- Sample Models>
Biology> Wolf Sheep
Predation

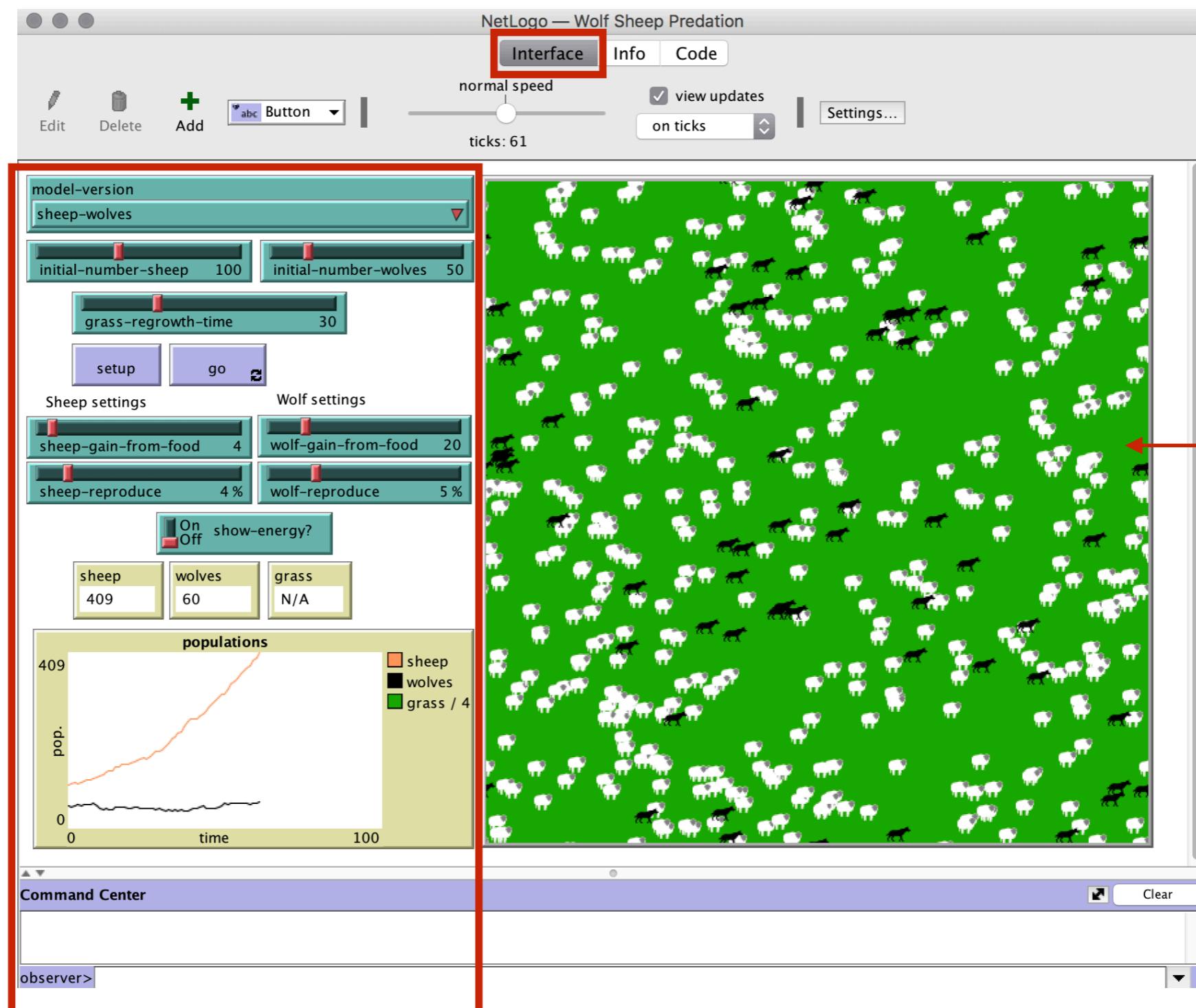


Wolf Sheep Predation Model



Wolf Sheep Predation Model

View and Control the model on the Interface tab



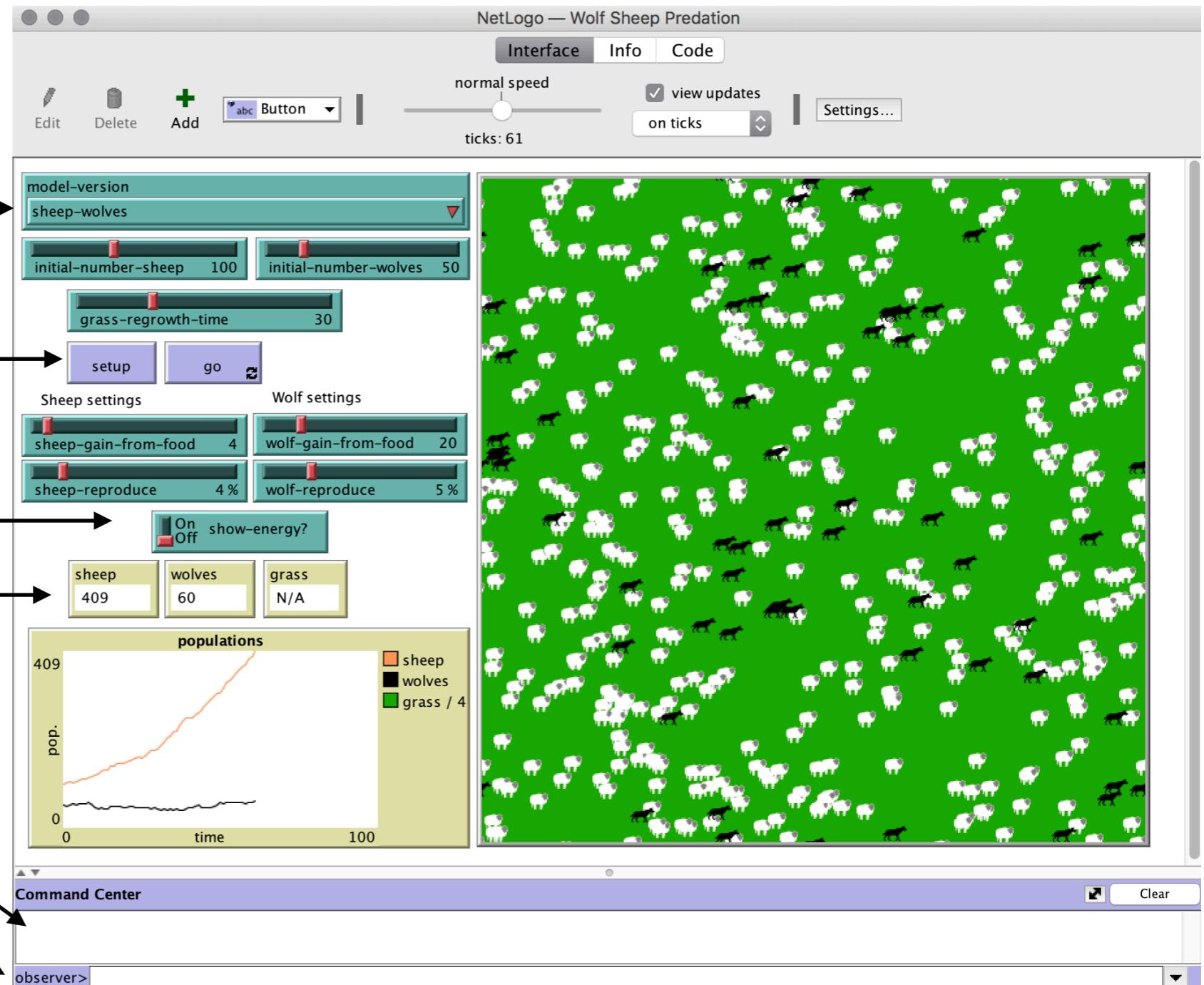
*the World/
View/display*

Model Interface

User controls / parameters

User controls/ parameters

- choosers →
- sliders →
- buttons →
(often a button to initialize model,
and another button to run model)
- switches →
- monitors →
- plots →
- print output
- command line

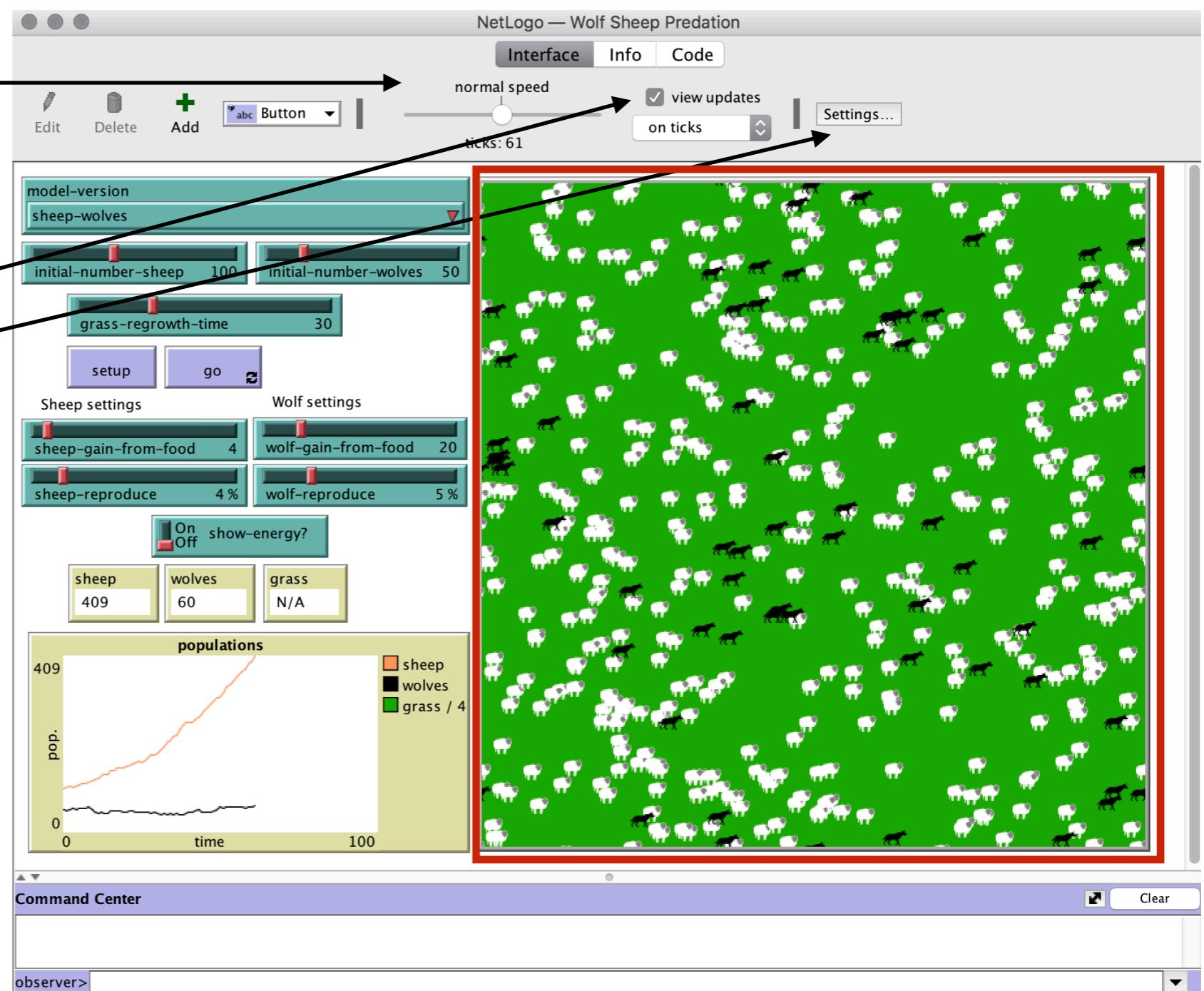


Model Interface

Control the World view display

Display/ View / World

- speed —
(how fast model runs)
- view updates
(enable view display to show changes)
- Settings
(World display:
size,
center point,
resolution
how to handle edges:
 - torus with wrapping
 - bounded to box)



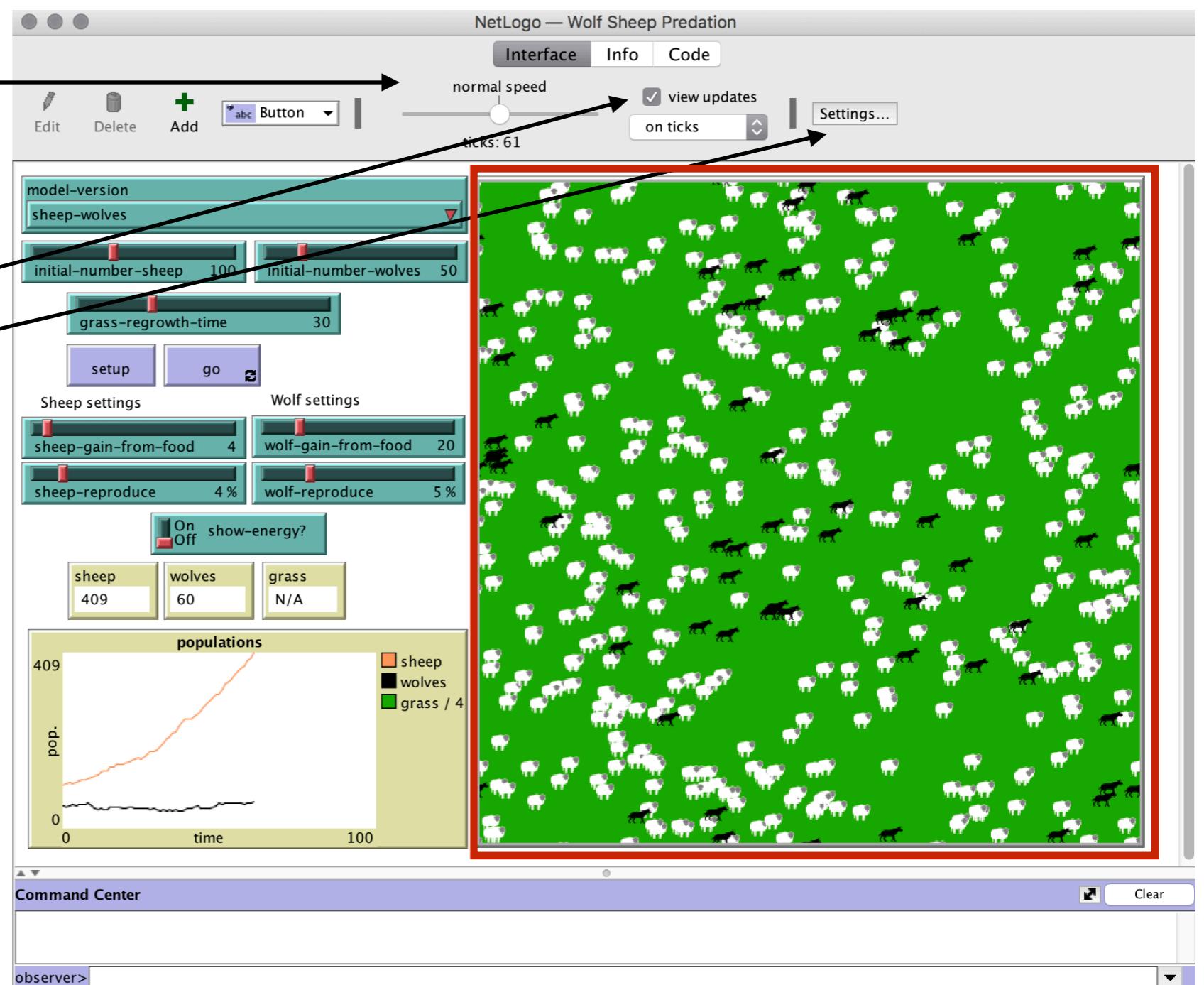
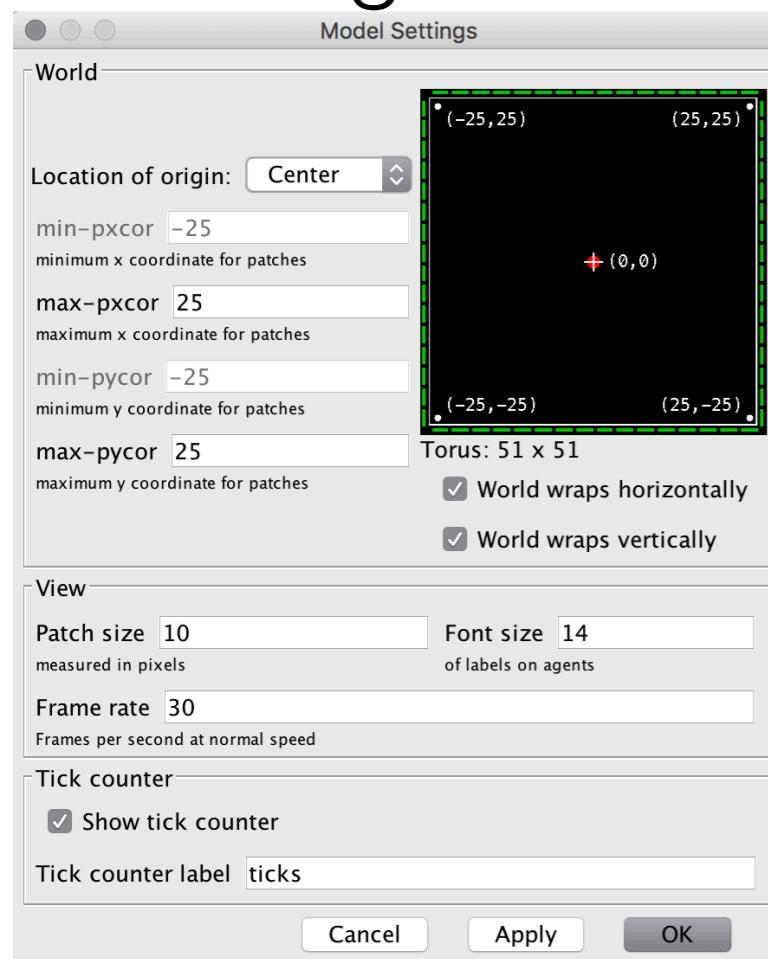
Pro tip: If you have a model with a lot of agents and movement, you can speed up model by unchecking view updates, slide speed to fastest, then when model is done running, check on View Updates to see final display.

Model Interface

Control the World view display

Display/ View / World

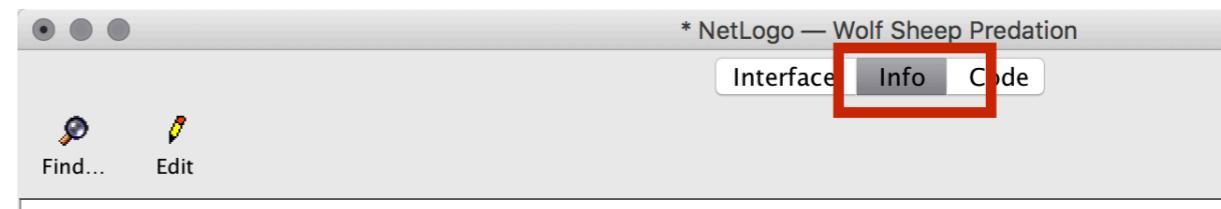
- speed —
(how fast model runs)
- view updates
(enable view display to show changes)
- Settings



Info about Model

Info Tab:

- **What the model is about**
Topic and background
- **How to use the model**
Describes the buttons, sliders
- **Things to notice**
What you see in plots and display when running model
- **Things to try**
Experiments you can try
- **Extend the model**
Future enhancements



The screenshot shows the NetLogo interface for the "Wolf Sheep Predation" model. The title bar reads "* NetLogo — Wolf Sheep Predation". Below the title bar are three buttons: "Interface", "Info" (which is highlighted with a red box), and "Code". Underneath the buttons are two small icons: a magnifying glass labeled "Find..." and a pencil labeled "Edit".

WHAT IS IT?

This model explores the stability of predator-prey ecosystems. Such a system is called unstable if it tends to result in extinction for one or more species involved. In contrast, a system is stable if it tends to maintain itself over time, despite fluctuations in population sizes.

HOW IT WORKS

There are two main variations to this model.

In the first variation, the "sheep-wolves" version, wolves and sheep wander randomly around the landscape, while the wolves look for sheep to prey on. Each step costs the wolves energy, and they must eat sheep in order to replenish their energy - when they run out of energy they die. To allow the population to continue, each wolf or sheep has a fixed probability of reproducing at each time step. In this variation, we model the grass as "infinite" so that sheep always have enough to eat, and we don't explicitly model the eating or growing of grass. As such, sheep don't either gain or lose energy by eating or moving. This variation produces interesting population dynamics, but is ultimately unstable. This variation of the model is particularly well-suited to interacting species in a rich nutrient environment, such as two strains of bacteria in a petri dish (Gause, 1934).

The second variation, the "sheep-wolves-grass" version explicitly models grass (green) in addition to wolves and sheep. The behavior of the wolves is identical to the first variation, however this time the sheep must eat grass in order to maintain their energy - when they run out of energy they die. Once grass is eaten it will only regrow after a fixed amount of time. This variation is more complex than the first, but it is generally stable. It is a closer match to the classic Lotka Volterra population oscillation models. The classic LV models though assume the populations can take on real values, but in small populations these models underestimate extinctions and agent-based models such as the ones here, provide more realistic results. (See Wilensky & Rand, 2015; chapter 4).

The construction of this model is described in two papers by Wilensky & Reisman (1998; 2006) referenced below.

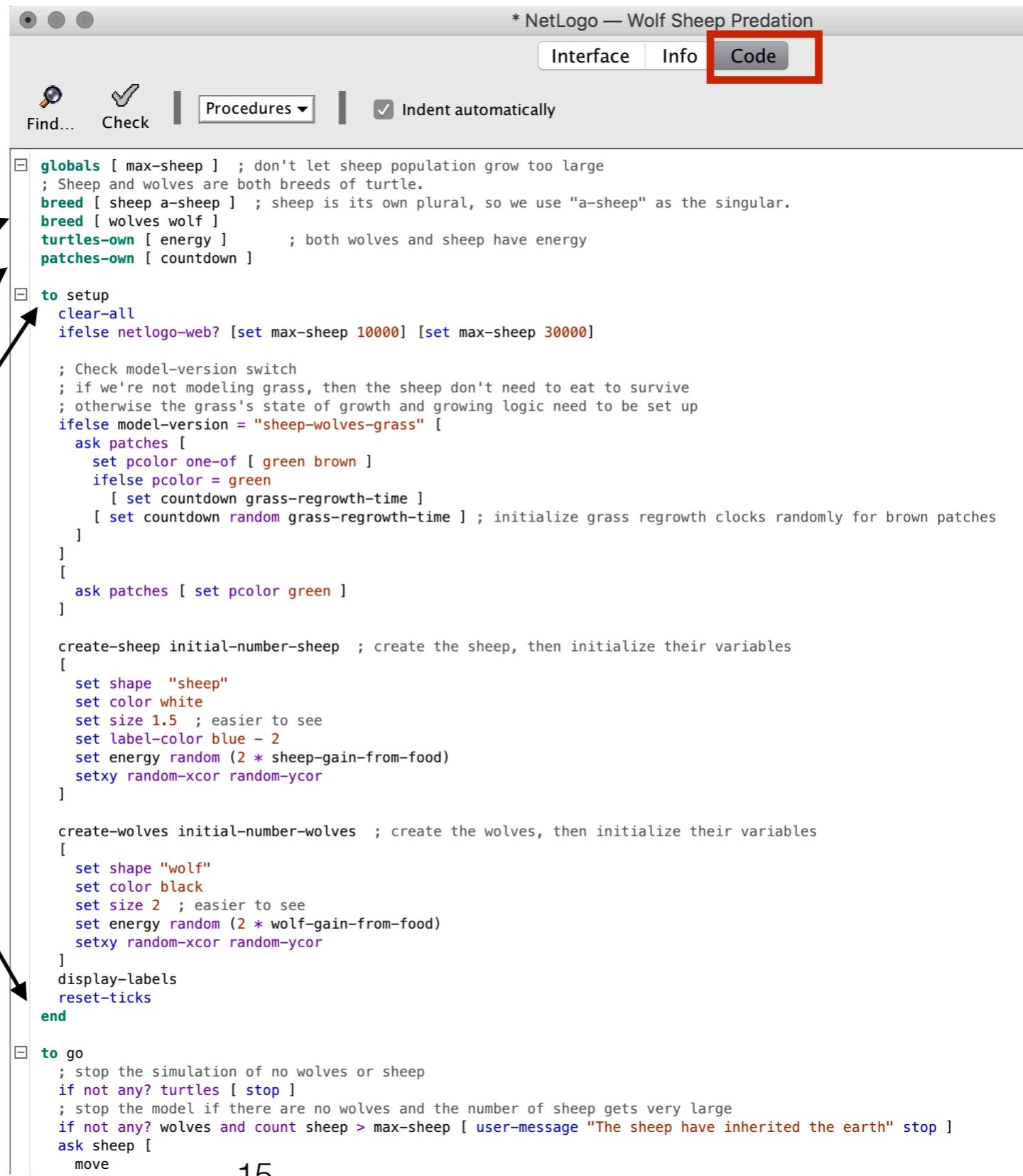
HOW TO USE IT

1. Set the model-version chooser to "sheep-wolves-grass" to include grass eating and growth in the model, or to "sheep-wolves" to only include wolves (black) and sheep (white).
2. Adjust the slider parameters (see below), or use the default settings.

Code for the Model

Code Tab:

- Extensions, Global variables
- Breeds (categories of agents)
- Attributes of agents, patches...
(specified by -own [])
- Button procedures and other function procedures
(begin with to and corresponding code is indented and ends with end)



The screenshot shows the NetLogo interface with the title bar "NetLogo — Wolf Sheep Predation". Below the title bar, there are three tabs: "Interface", "Info", and "Code", with "Code" being the active tab and highlighted with a red border. The main area displays the NetLogo code for the "Wolf Sheep Predation" model. The code is organized into several procedures:

- globals**:
 - max-sheep ; don't let sheep population grow too large
 - Sheep and wolves are both breeds of turtle.
 - breed [sheep a-sheep] ; sheep is its own plural, so we use "a-sheep" as the singular.
 - breed [wolves wolf]
 - turtles-own [energy] ; both wolves and sheep have energy
 - patches-own [countdown]
- to setup**:
 - clear-all
 - ifelse netlogo-web? [set max-sheep 10000] [set max-sheep 30000]
 - ; Check model-version switch
 - ; if we're not modeling grass, then the sheep don't need to eat to survive
 - ; otherwise the grass's state of growth and growing logic need to be set up
 - ifelse model-version = "sheep-wolves-grass" [
 - ask patches [
 - set pcolor one-of [green brown]
 - ifelse pcolor = green [
 - set countdown grass-regrowth-time
 - [set countdown random grass-regrowth-time] ; initialize grass regrowth clocks randomly for brown patches
 - [
 - ask patches [set pcolor green]
 - create-sheep initial-number-sheep ; create the sheep, then initialize their variables [
 - set shape "sheep"
 - set color white
 - set size 1.5 ; easier to see
 - set label-color blue - 2
 - set energy random (2 * sheep-gain-from-food)
 - setxy random-xcor random-ycor
 - create-wolves initial-number-wolves ; create the wolves, then initialize their variables [
 - set shape "wolf"
 - set color black
 - set size 2 ; easier to see
 - set energy random (2 * wolf-gain-from-food)
 - setxy random-xcor random-ycor
 - display-labels
 - reset-ticks
- end**
- to go**:
 - ; stop the simulation of no wolves or sheep
 - if not any? turtles [stop]
 - ; stop the model if there are no wolves and the number of sheep gets very large
 - if not any? wolves and count sheep > max-sheep [user-message "The sheep have inherited the earth" stop]
 - ask sheep [move

Code for the Model

* NetLogo — Wolf Sheep Predation

Interface Info **Code**

Find... Check Procedures ▾ Indent automatically

Turtles? →

Patches? ←

```
globals [ max-sheep ] ; don't let sheep population grow too large
; Sheep and wolves are both breeds of turtle.
breed [ sheep a-sheep ] ; sheep is its own plural, so we use "a-sheep" as the singular.
breed [ wolves wolf ]
turtles-own [ energy ] ; both wolves and sheep have energy
patches-own [ countdown ]
```

to setup
clear-all
ifelse netlogo-web? [set max-sheep 10000] [set max-sheep 30000]

; Check model-version switch
; if we're not modeling grass, then the sheep don't need to eat to survive
; otherwise the grass's state of growth and growing logic need to be set up
ifelse model-version = "sheep-wolves-grass" [
ask patches [
set pcolor one-of [green brown]
ifelse pcolor = green
[set countdown grass-regrowth-time]
[set countdown random grass-regrowth-time] ; initialize grass regrowth clocks randomly for brown patches
]
[
ask patches [set pcolor green]
]

create-sheep initial-number-sheep ; create the sheep, then initialize their variables
[
set shape "sheep"
set color white
set size 1.5 ; easier to see

Defining Agents and the Environment in Netlogo

In Netlogo:

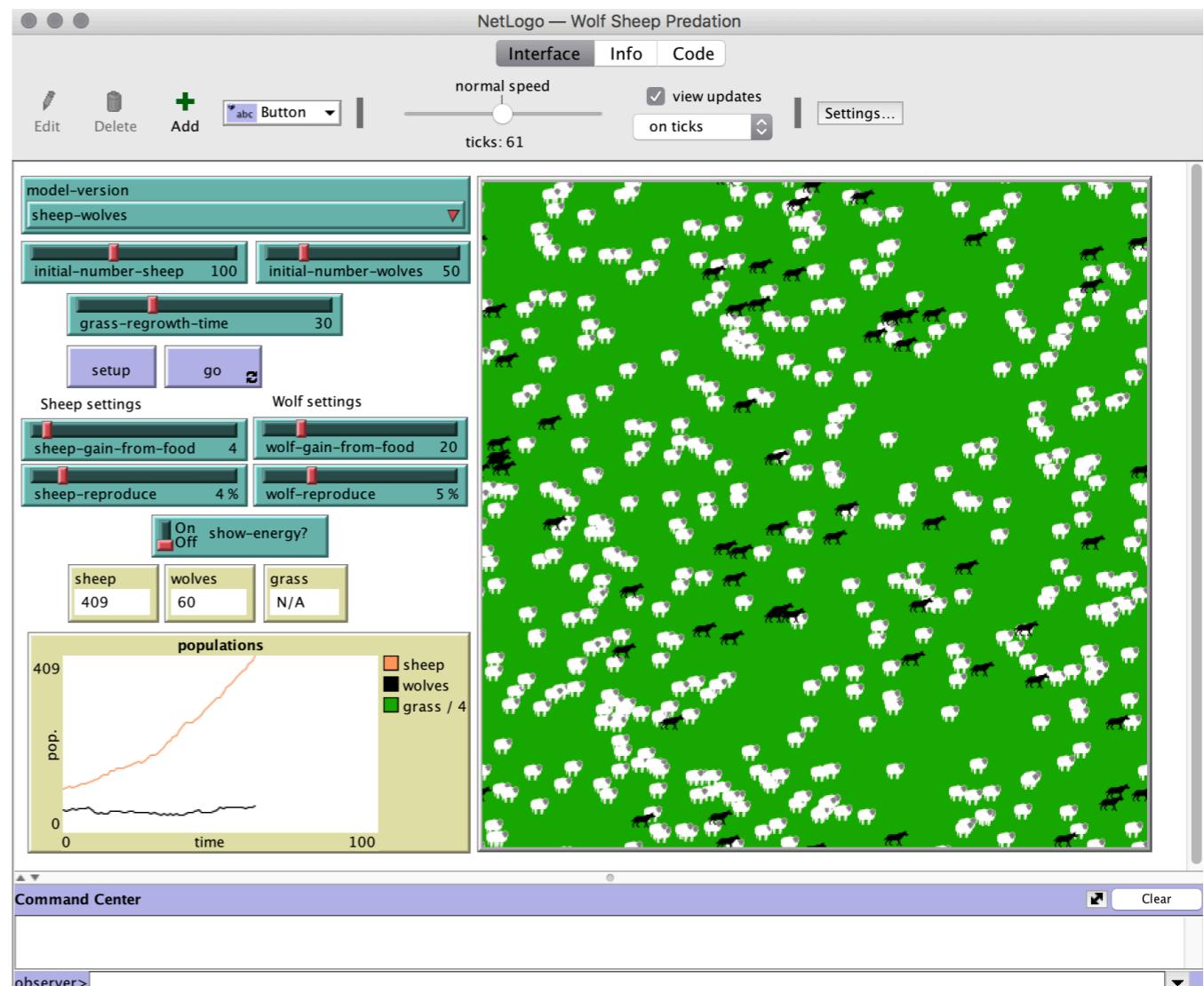
- Agents are called **turtles**
- The environment consists of **patches**
- Think of it as agents, like gnomes, riding around on turtles across patches of ground.



Wolf Sheep Predation Model

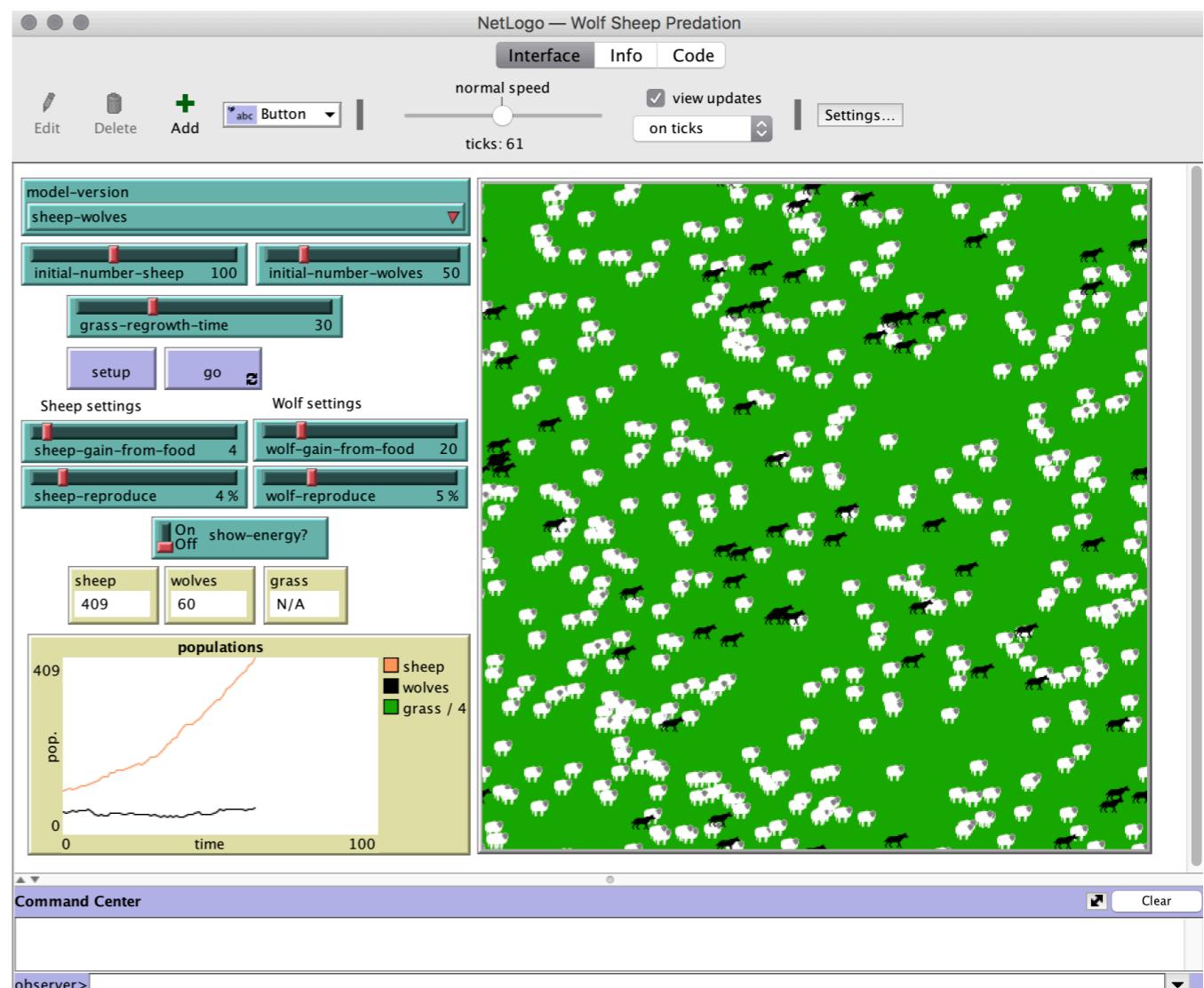
Run the model, adjust the parameters, and explore the info and code and tabs.

- What is this a model of and what does it show?
- What are the attributes and behaviors of the agents and patches?
- How is time represented in this model?
- Describe the interactions between agents and agents-environment.



Wolf Sheep Predation Model

What are some things you would change with this model to bring in more spatial related behaviors or constraints or make it more realistic?



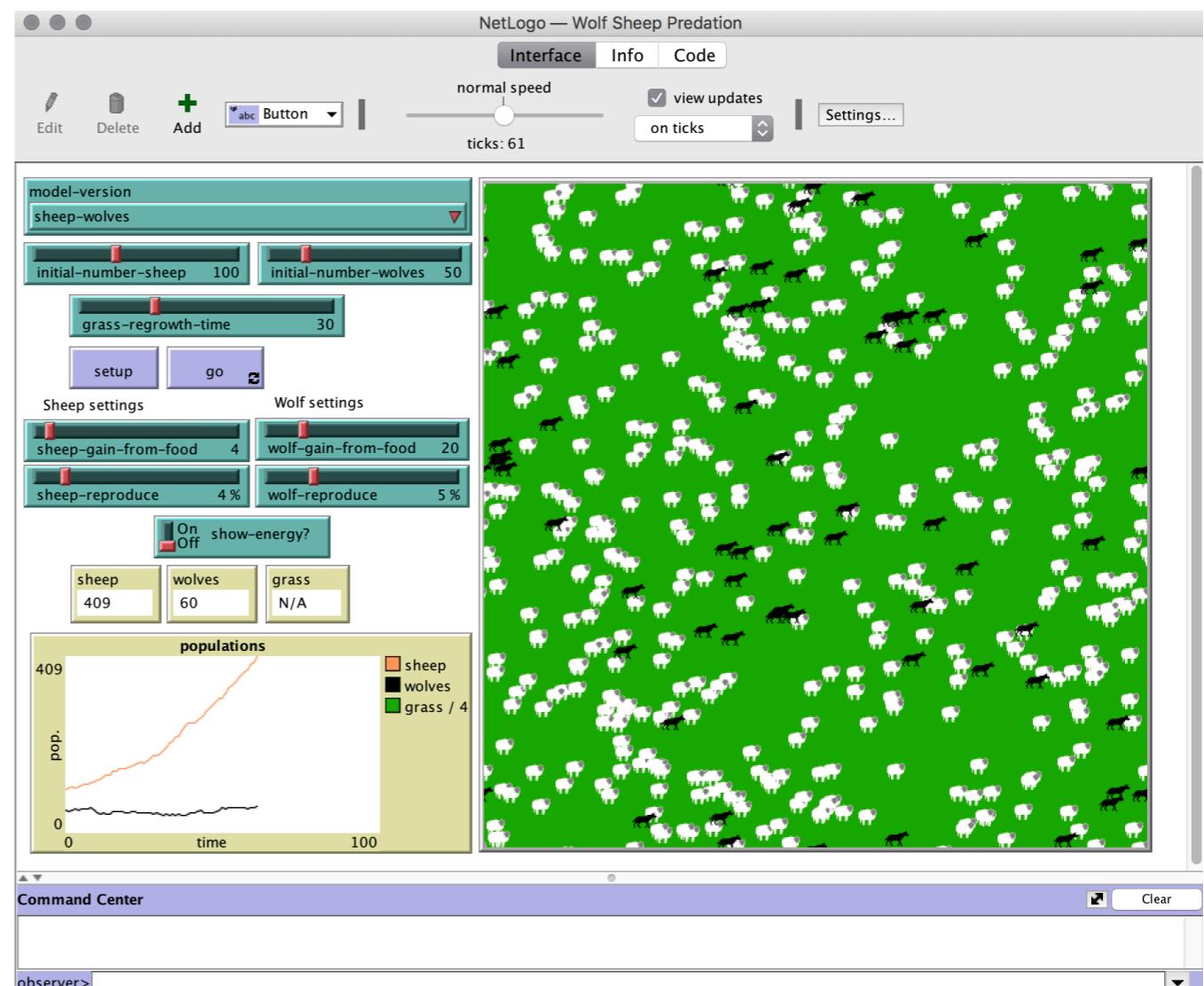
Wolf Sheep Predation Model

What are some things you would change with this model to bring in more spatial related behaviors or constraints or make it more realistic?

What if the sheep were constrained by a fence.

How can we change the model to reflect that this display is bounded by geography or an obstacle like a fence?

One way is to change the Settings so the world does not wrap



Models Library

Take a few minutes to explore some of the models in the library.

Read the info tab, adjust the parameters, peak at the code.

