

StarLogoTNG Biology Curriculum Teacher Guide

Overview

The StarLogoTNG Biology Curriculum is a set of activities that centers on the themes of ecology and evolution. It uses the simple interactions of carrots, rabbits, and wolves, along with various abiotic factors, to highlight the concepts inherent in decentralized systems that are traditionally difficult for students to understand. The projects require students to learn simple programming with the aim of fostering computer usage in science, but the major focus of each activity is system observation and data analysis.

The curriculum is intended as a set of supplementary activities for the normal ecology and evolution units. While it may be used as a stand-alone unit, there may be details regarding each topic that have been removed for modeling purposes. Thus, it is recommended that teachers intersperse traditional teaching days between activities to cover any missing detail, as well as to give students time to digest the material.

Unit Summaries

Lesson 1 begins with just carrots, focusing on the relationships between the carrots and the abiotic elements of the ecosystem. Because the rest of the lessons deal focus primarily on biotic factors, this introductory lesson serves as a reminder that the abiotic factors are just as important. There is no programming involved; instead, the lesson introduces the ideas of modeling and simulation by allowing students to play with sliders and observe the graphs. The main goal of the lesson is to foster the idea of methodical scientific inquiry, whereby students must form a hypothesis and support/ contradict it with collected data. The activity also gives students an easy transition into the StarLogoTNG user interface.

Lesson 2 takes a step back from the biology theme with the VANTS (virtual ants) activity. This activity allows students to create visual designs on the StarLogoTNG SpaceLand using only very few simple commands. On the programming side, VANTS shows that the computer requires specific commands written in a specific way; on the biology side, VANTS highlights the emergent behaviors that arise from each agent performing its own task. With this activity, students are introduced to organization of StarLogo blocks, block canvas, and language.

Lesson 3 begins the ecosystem modeling, requiring the students to program their own rabbits. During this process, students learn the essential procedures that rabbits in the model follow (hop, eat, reproduce, die) that are necessary for a simple functioning ecosystem. While the rabbits are not as complicated as some of the later models, the step-by-step programming tasks give a behind-the-scenes look at the code and teach the students the thought process involved in building a model. After programming, students can observe a classic predator-prey relationship between the rabbits and carrots.

Lesson 4 introduces a gaming aspect that serves the dual purpose of entertaining the students while bringing up the role of humans in the ecosystem. The code now includes a small wolf population, and students program a first person hunter character that shoots the wolves. The model begins with a stable configuration, which students disturb by simply decreasing the age required for wolves to reproduce. The resultant overpopulation tends to overeat and wipe out the rabbits, so the hunters are used to stabilize the population. This lesson brings up ethical issues of hunting and the effects of human interactions with the ecosystem.

Lesson 5 introduces the ideas of competition and selection. For simplicity, the model has returned to being only carrots and rabbits. The activity begins with rabbits having a “color gene” that gives them coats of varying shades of blue. The color gene has no other effect and thus creates no selective advantage. Over time the rabbit population still converges to a single color due to genetic drift, but the color that survives changes between trials. Then students link the color gene first to rabbit speed and then to energy loss, demonstrating directional and stabilizing selection. There is also a “flood” button that speeds up the process by killing 90% of the population, creating a population bottleneck. The goal of this lesson is to emphasize the chance and competition as two very different driving forces behind evolution.

Lesson 6 brings together the driving forces behind evolution with the actual mechanism of mutation. In this project students can change the mutation rate and temperature via sliders. The energy loss in this scenario is now also related to temperature, with exponentially higher energy loss at lower temperatures. The key concept in this lesson is that genetic variation is important as a cushion against changes in the environment, and what is considered “fit” in one set of conditions may be considered “unfit” in another. The code is written such that functionality for more genes can be easily added, so the lesson can be extended to cover other concepts. A possible activity is to allow students to create their own gene and incorporate it into the model using the experience of the previous lessons, but it is important to gauge the level of understanding that they have gained.

Teaching Tips and Notes

Biology Modeling:

- The ecosystem models are *very* temperamental due to the nature of small populations. If you would like to change the model, don't be discouraged if it isn't stable on the first try. Varying parameters makes a *huge* difference on the model's performance.
- It is important to remind the students that the models are not a perfect depiction of real life, so always take the opportunity to discuss, “How is this model a good representation? How is it bad?”
- Tie in complex systems concepts whenever possible and try to bring these concepts back into other units during the course of the year. Diffusion/osmosis, proteins/DNA,

human physiology, etc. are all great examples of emergent behaviors from decentralized systems.

- Think of ways to test the students at the end of the unit. It is important for the students to feel like these exercises are not a waste of time.

StarLogoTNG:

- Students really enjoy being creative when using StarLogoTNG, so it is important to take every opportunity to make the projects their own. While it is difficult with these models to stray too far while getting the same message across, little touches like allowing the students to change terrain or program monsters and cartoons instead of rabbits and wolves already gets them excited. Exploration time also allows students to discover their own tricks and satisfy their curiosities.
- You may want to ask students to work in pairs at a computer, designating roles of “driver” and “navigator” that switch every 15-20 minutes. The driver controls the mouse and keyboard, while the navigator verbally directs the driver’s actions using the activity handouts. This will be particularly useful for activities like programming the rabbits, where a support system can help sort out many confusions of programming for the first time. Activities like the hunter may be done in pairs or alone, to give students more individual play-time. However, if you choose to have students working on individual computers, you may still want to designate “buddies” so that students can share projects and ask questions.
- Instruct students to shut off their monitors when you need their undivided attention.
- Encourage students to “Save Next Version” if they create any code, just in case the program freezes unexpectedly.

About the Guide

Each lesson includes the following:

- **Goals:** The overarching theme of the activity.
- **Biology Concepts:** Concepts in the biology curriculum that are covered.
- **StarLogoTNG Programming Concepts:** Programming concepts and elements of StarLogoTNG that are covered.
- **Materials:** Materials needed for the activity.
- **About the Model:** Technical information about the model and code. This description is very detailed and for reference only.
- **Possible Modifications:** Changes that can be easily made to the model to support different goals (of course there are many other possible modifications of varying difficulty to implement).
- **Suggested Teacher Guide:** Outline of a suggested lesson.
- **Student Worksheet (printout):** Student worksheets for distribution.

Project files include the following:

- ***project.sltnng*:** Starter code for students to do the activity.
- ***project-sol.sltnng*:** Solutions code for teacher reference. Contains all completed code in programming portions, as well as detailed commenting of all procedures.