Introduction

For thousands of years, people have been creating models to help them better understand the world around them. Leonardo DaVinci built models of flying machines that some claim were inspired by his desire to understand the flight of birds. Sir Isaac Newton described the behavior of physical systems with sets of equations. Jacques Vaucanson built a mechanical duck that actually ingested (and eliminated!) its food (Bedini, 1964). These models not only helped their creators better understand the phenomena that they were studying, but also helped them convey their new ideas to other people.

Throughout history, most people, like the pioneers mentioned above, have created models out of wood, paper, metal, and mathematical expressions. In more recent times, computers have provided a new medium for building, analyzing, and describing models. Using computers, economists build models of the stock market, biologists build models of cell division, and historians build models of ancient civilizations. Computers also make it easier for novices to build and explore their own models—and learn new scientific ideas in the process.

Research has shown that the process of creating models (as opposed to simply using models built by someone else) not only fosters model-building skills but also helps develop a greater understanding of the concepts embedded in the models (Confrey & Doerr, 1994; diSessa, 1986; Talsma, 2000). When you build your own models, you can decide what topic you want to study and how you want to study it. As your investigation proceeds, you can determine which aspects of the system you want to focus on, and refine your model as your understanding of the system grows. Perhaps most important, building your own models helps you develop a sound understanding of both how a system works and why it works that way.

The StarLogo language was designed to enable people to build their own models of complex, dynamic systems. Unlike many other modeling tools, StarLogo supports a tangible process of building, analyzing, and describing models that does not require advanced mathematical or programming skills. Using StarLogo, you can build and explore models—and in the process you can develop a deeper understanding of patterns and processes in the world around you.

New technologies, like StarLogo, can shape both what and how people learn. But, too often, people mistakenly believe that the mere presence of a new technology will be sufficient to cause change. The ability to learn new ideas through computer simulations, for example, is greatly influenced not only by the technologies used but also by the ways in which the simulations are presented. Creating an environment that supports learners' intellectual curiosity is just as important as providing the tools for building and exploring new phenomena. The open-ended Challenges and Activities in this book help you create such an environment by providing a means for understanding complex systems through the design and creation of your own dynamic models.

CREATING MODELS

During the past 10 years, simulation modeling, especially as it helps people to understand complex systems, has become a mainstream use of computational technology. The widespread popularity of "edutainment" software like SimCity and Civilization gives a clear indication of the extent to which simulation games have permeated popular culture. While it can be useful to experiment with prebuilt models like SimCity, a deeper understanding comes through building and manipulating models whose underlying structure is accessible (Feurzeig & Roberts, 1999; Resnick, Bruckman, & Martin, 1996). Just as a young child learns more by building a bridge out of blocks instead of merely playing with a prefabricated bridge, designing and creating your own models provide richer learning experiences than simply playing with prebuilt models. This learning process is critically important in domains that require an understanding of complex systems, from economics and mathematics to physics and biology. In addition, this learning process fosters the kinds of higher-order thinking and problem-solving skills that are called for in science, mathematics, and technology standards. This book focuses on learning about complex systems and developing the ability to create your own models in StarLogo, from the conceptualization of an idea through the final implementation, analysis, and presentation of a model.

Several common modeling programs, including Model-It, Stella, and MatLab, enable the design and creation of your own models. To model a system in one of these environments, you need to describe how the entire system changes. StarLogo approaches model building from a different perspective. In StarLogo, you write simple rules for individual behaviors. For instance, you might create rules for a bird that describe how fast it should fly and when it should fly towards another bird. When you watch many birds simultaneously following those rules, you can observe how patterns in the system, like flocking, arise out of the individual behaviors. Building up models from the individual, or "bird," level enables you to develop a better understanding of the system, or "flock," level behaviors.

While building your own models can be a great learning experience, it raises many questions about what to include in your model. We have found that people new to model building often believe that a model is only "good" or "useful" when it incorporates every possible aspect of a system. It is important to keep in mind that, even in the scope of "scientific modeling," people build a wide variety of models that encompass varying levels of detail and utilize a variety of methods. Depending on the goals you have, some approaches to model building will be more suitable than others. Choosing the one that best suits your personal goals and ideas can be a tricky process. Chapter 2 provides a framework for thinking about which approach is most appropriate for a particular set of goals.

INTERDISCIPLINARY NATURE

In recent years, there has been a surge of interest in the study of complex systems—that is, systems in which complex patterns and behaviors arise from simple rules and interactions. Researchers who study complex systems model economics as well as ecology, chemistry as well as population dynamics. Similarly, this book is not directed toward any single discipline. In StarLogo, you can model the dynamics of predator-prey relationships or experiment with the formation of traffic jams; you can investigate the demographics of neighboring countries or explore the collective behavior of groups of ants. Employing the principles covered in this book, you can design and create dynamic, complex models of systems from a variety of disciplines.

The conceptual organization of this book encourages you to draw on many perspectives as you create and analyze models. Consider a model of a rain forest. A biologist might study the model with an eye towards the interactions among the individuals, while an environmental chemist might focus on the accumulation of pollutants in different groups of creatures, and a mathematician might investigate the patterns of population growth and decline. Integrating these perspectives in both the creation and investigation phases of model building can enrich your models and help illuminate the underlying connections between seemingly disparate fields.

STRUCTURED EXPLORATION

One of our primary motivations when we wrote this book was to achieve a balance between structure and exploration. We designed the materials in this book to foster an exploratory, creative spirit while at the same time providing adequate structure for learning how to build models. In our experience, novices find this structured exploration to be engaging and more tractable than either didactic or completely open-ended learning environments.

The core of this book is a series of Activities and Challenges. Each Challenge is a problem statement intended to get your creative juices flowing. The Challenges encourage you to explore model design and construction while learning about the principles of complex systems. At the same time, they build familiarity with the StarLogo environment. The Challenges provide specific programming information when it is needed, thus minimizing the need for direct instruction or a lengthy manual. By structuring each Challenge to focus on key concepts, like how to move creatures or use variables, this book enables everyone to cultivate model-building skills in StarLogo.

Though "on-screen" computer modeling is one focus of this book, "off-screen" Activities provide another way to connect abstract notions of dynamic processes and complexity to personal experience. Participation in life-sized simulations helps learners develop deeper understandings of dynamic systems (Colella, 2000). The group Activities in the book allow participants to think about concepts like exponential growth, local versus global information, and group decision making from a personal perspective. Integrating the group Activities with the Challenges is an excellent way to learn about building models and exploring their content.

AUDIENCE

We designed this book primarily for educators. We use the term "educator" in its broadest sense. Whether you are teaching yourself, students, teachers, or your own child, you can use this book as an introduction to model building. This text can also provide an opportunity to learn about and implement new ways of teaching and learning. This book emphasizes the design and creation of modeling projects in an environment that promotes the exploration and explanation of new ideas. In addition, this book provides an innovative structure for teaching concepts from specific subject areas. By incorporating modeling into your classes, you and your students can gain new perspectives on important ideas and build deeper understandings of domain-specific content.

This book can be used in a variety of settings. For example, a ninth-grade social studies teacher can use it to enhance a unit on demographics. A professor can use it as a main or supplemental text in a course at a graduate school of education. Parents can use it to explore innovative uses of computers with their children. A boys and girls club mentor can use it to teach kids about scientific investigation through model construction and analysis. Workshop leaders can use it in a professional development seminar to draw connections between using technology and addressing the new science standards. As these examples illustrate, this book can be used in a wide variety of settings, not just to teach StarLogo, but also to incorporate facets of modeling into more traditional subjects or to provide the foundation for interdisciplinary explorations. In other words, this book is a flexible resource that can be applied in many environments.

ORGANIZATION

This book consists of three main sections. The first six chapters explain the philosophy behind StarLogo, introduce modeling concepts, describe the book structure, propose several different ways to use this book, and give a quick tour of the StarLogo world. The Activities and Challenges in the second section form the core of the Adventures curriculum. The Activities are group exercises that help participants gain a first-person perspective on complex, dynamic models. The Challenges are sequential lessons that help people develop both model-building skills and a better understanding of complex systems. Just after the Activities and Challenges, you will find Student Handouts for each Challenge. Finally, the appendices in the last section of the book contain notes about MacStarLogo, collected hints from the Challenges, and an explanation of common StarLogo error messages.

CHAPTER 1

Travels with StarLogo

There is an old saying that goes something like this: "Give a person a hammer, and the whole world looks like a nail." Indeed, the ways we see the world are deeply influenced by the tools that we use.

For centuries, educators have used many different tools to help their students learn. When developing his concept of kindergarten in the early 1800s, Friedrich Froebel gave children a set of objects, which he called "gifts," to stimulate certain kinds of exploration (Brosterman, 1997). For example, colored balls and blocks encouraged children to explore shape and color. Montessori extended many of Froebel's ideas, creating materials that are still found in many elementary classrooms today.

Just as different Froebel gifts help children explore different concepts, different computer programs support different types of explorations and learning. The use of StarLogo in this book influences the types of projects that you work on. Even more important, it influences the type of thinking you do as you work on those projects.

StarLogo is designed especially for helping people create models of *decentralized systems*—that is, systems in which patterns arise from interactions among lots of individual objects. For example, StarLogo is well designed for exploring how bird flocks arise from interactions among individual birds, or how traffic jams arise from interactions among individual cars.

Decentralized systems are very common in the world. But it turns out that most people have great difficulty understanding the workings of such systems. That's why we see StarLogo as such a powerful tool. With StarLogo, you can create models of many phenomena that are familiar and important but traditionally difficult to understand.

In this chapter, we will give a brief introduction to the StarLogo approach to modeling—and the new ways of thinking that are nurtured by this type of modeling.

THE CENTRALIZED MINDSET

A flock of birds sweeps across the sky. How do birds keep their movements so orderly, so synchronized? Most people assume that birds play a game of Follow the Leader: The bird at the front of the flock leads, and the others follow. But that's not so. Most bird flocks don't have leaders at all. Rather, each bird follows a set of simple rules, reacting to the movements of the nearby birds. Orderly flocking patterns arise from these simple, local interactions. The bird in front is not a "leader" in any meaningful sense—it just happens to end up there. The flock is organized without an organizer, coordinated without a coordinator.

So why do most people assume that bird flocks have leaders? The problem extends beyond bird flocks. When people see patterns in the world, they tend to assume that someone or something is in charge of creating the pattern. When people see traffic jams, for example, they tend to assume an overturned truck or a broken bridge must be at fault; in fact, many traffic jams are caused by simple interactions among neighboring cars.

This tendency to assume centralized control, which we call the *centralized mindset*, makes it difficult for people to understand the workings of many phenomena in the world. The recurrent questioning of evolutionary theories is another example: When people see complex living systems in the world, they assume that someone or something must have explicitly designed them; instead, these livings systems are the products of millions of incremental changes over time. Both experts and novices fall into the centralized mindset. Until recently, even scientists assumed that bird flocks must have leaders. A similar bias toward centralized theories can be seen throughout the history of science.

One of the goals of StarLogo is to help people move beyond the centralized mindset, helping them learn how patterns can arise from lots of simple interactions among simple objects, without any object in charge. With StarLogo, you can model (and gain insights into) many real-life phenomena, such as bird flocks, traffic jams, ant colonies, and market economies. By writing simple rules for individual creatures, you are able to investigate the collective behavior of systems consisting of thousands of creatures. StarLogo models often exhibit behaviors at the system level that are not evident in the rules you write for individual behavior. These system-level behaviors are referred to as *emergent* because they arise out of the interactions among individuals. For instance, an ant colony can exhibit coordinated behavior as it forages for food, even though the rules describe only individual behavior, not colony-level behavior.

TOOLS FOR DECENTRALIZED THINKING

People already have lots of experience with decentralized systems; they observe decentralized systems all the time. But observation does not necessarily lead to deep understanding: People observed bird flocks for thousands of years before anyone suggested that flocks are leaderless. To develop a deeper understanding, people need opportunities not just to observe decentralized systems but to design their own models of decentralized systems.

Scientists have developed a variety of computational tools for this type of decentralized modeling. But most of these tools are designed for people with advanced mathematical or programming experience. StarLogo is an exception: It is designed explicitly for nonexperts.

StarLogo is an extension of Logo, a programming language that is generally used in elementary and secondary schools (Papert, 1980). In traditional versions of Logo, children create geometric patterns and animations by giving commands to a graphical *turtle* on the computer screen. The Logo turtle can be used to represent any type of object in the world: An ant, a car, a molecule. But traditional versions of Logo typically have only a few turtles. To support decentralized modeling, StarLogo has thousands of turtles, and all of the turtles can perform their actions at the same time, in parallel. Moreover, StarLogo turtles have better "senses" so that they can detect things (such as other turtles) in their local environment. Finally, the turtles' world is divided into small square sections called *patches*. For example, each patch can grow "food," which the turtles can search for and eat.

Unlike many other modeling programs, StarLogo allows you to directly observe both individual actions and the group patterns that emerge from those actions. The spatial nature of the program and its visual representation enable you to look at things (like creatures and their environment) instead of just looking at abstractions of things (like equations or graphs). This presentation of a model makes it easier to interpret what is happening. StarLogo makes it possible for everyone—even people without an extensive background in mathematics or programming—to design and build their own models.

AN EXAMPLE: TURTLE ECOLOGY

The legendary baseball manager Casey Stengel once said, "If you don't know where you're going, you might end up somewhere else." Our experiences with computer-based modeling activities have taught us a corollary: "Even if you think you know where you're going, you'll probably end up somewhere else."

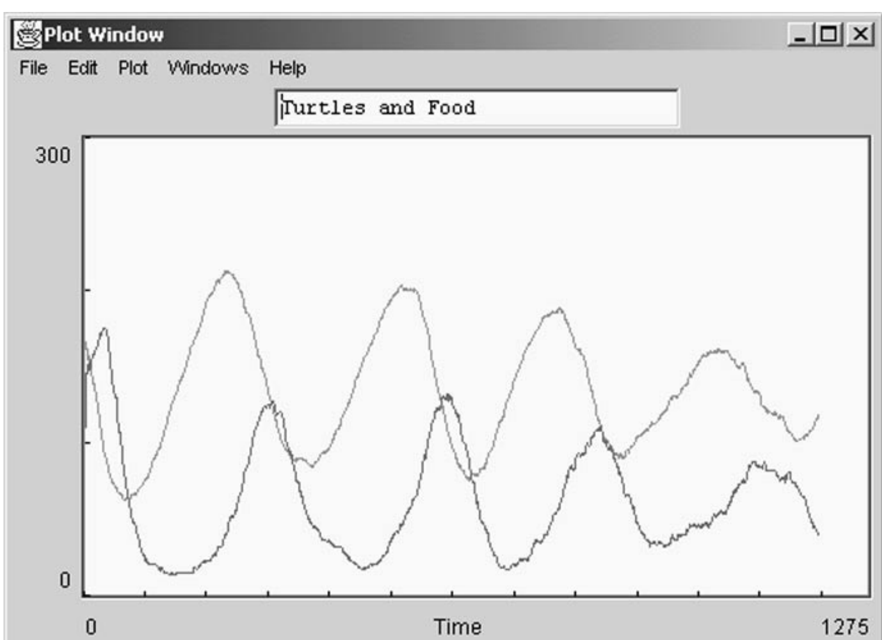
That's what happened to Benjamin, a high school student, when he set out to create a StarLogo program that would simulate evolution by natural selection. At the core of Benjamin's simulation were turtles and food. His basic idea was simple: Turtles who eat a lot of food reproduce, and turtles who don't eat enough food die. Eventually, he planned to add "genes" to his turtles. Different genes could provide turtles with different levels of "fitness" (perhaps different capabilities for finding food). But, as often happens in the course of building models, Benjamin became fascinated by the fundamental behaviors that his model generated. On the road to evolution, Benjamin embarked on an interesting exploration of ecological systems (in particular, predator-prey systems).

Benjamin began by making food grow randomly throughout the StarLogo world. Then he created some turtles. The turtles had very meager sensory capabilities. They could not "see" or "smell" food at a distance. They could sense food only when they bumped directly into it. So the turtles followed a very simple strategy: Wander around randomly, eating whatever food you bump into.

Benjamin gave each turtle an energy level. Every time a turtle ate some food, its energy increased. Every time it took a step, its energy decreased a bit—and if its energy dipped to zero, it died. As he ran his model, Benjamin observed that sometimes all of the turtles died. After some experimenting, Benjamin added reproduction to his model: Whenever a turtle's energy increased above a certain threshold, the turtle "cloned" itself and split its energy with its new twin.

Benjamin assumed that his new rule for cloning would somehow balance the rule for dying, leading to some sort of equilibrium and preventing the extinction of his turtles. Benjamin started the model running again. But the rules didn't balance out as Benjamin had predicted—the turtle population repeatedly grew and declined. Meanwhile, the amount of available food also fluctuated, but out of phase with the turtles.

On each cycle, the turtles overgrazed the food supply, leading to a scarcity of food, and many of the turtles died. But then, with fewer turtles left to eat the food, the food became more abundant. The few surviving turtles thus found a plentiful food supply, and each of them rapidly increased its energy. When a turtle's energy surpassed a certain threshold, it cloned, increasing the turtle population. But as the population grew too high, food again became scarce, and the cycle started again. Visually, the oscillations were striking (see plot below). Turtles and food were always present, but the density of each continually changed.



A Plot Window showing the changing densities of turtles and food.

The Lotka-Volterra Equations:

The changes in the population density of the prey (n_1) and the population density of the predator (n_2) can be described with the following differential equations:

$$\frac{dn_1}{dt} = n_1 (b - k_1 n_2)$$

$$\frac{dn_2}{dt} = n_2 (k_2 n_1 - d)$$

where b is the birth rate of the prey, d is the death rate of the predators, and k_1 and k_2 are constants.

The oscillating behavior in Benjamin's project is characteristic of ecological systems with predators (in this case, turtles) and prey (in this case, food). Traditionally, scientific (and educational) explorations of predator-prey systems are based on sets of differential equations, known as the Lotka-Volterra equations (see box at left). It is a straightforward matter to write a computer program based on those equations that computes how the population densities of the predators and prey vary with time.

This approach is typical of the way that scientists have traditionally modeled and studied the behaviors of all types of systems (physical, biological, and social). Scientists write down sets of equations and then attempt to solve them either analytically or numerically. Approaches like this one require advanced mathematical training—training that is usually available only at the university level.

The StarLogo approach to modeling systems (exemplified by Benjamin's predator-prey project) is sharply different. StarLogo makes systems-related ideas much more accessible to students by providing them with a stronger personal connection to the underlying models. Traditional differential-equation approaches are "impersonal" in two ways. The first is obvious: They rely on abstract symbol manipulation (accessible only to people with advanced mathematical training). The second is more subtle: Differential equations deal in aggregate quantities. In the Lotka-Volterra system, for example, the differential equations describe how the overall populations (not the individual creatures) change over time. There are now some very good computer modeling tools—such as Stella and Model-It—that are based on differential equations. These tools eliminate the need to manipulate symbols, focusing on more qualitative and graphical descriptions. But they still rely on aggregate quantities.

In StarLogo, by contrast, students think about the actions and interactions of individual objects or creatures. StarLogo projects describe how individual creatures (not overall populations) behave. Thinking in terms of individual creatures seems far more intuitive and is accessible to students of all levels. Students can imagine themselves as individual creatures and think about what they might do in a certain situation. By describing and observing the dynamics at the level of the individual creatures, rather than at the aggregate level of population densities, students can more easily think about and understand the patterns that arise. Just as Benjamin charted a new path to understanding, this book will enable you to explore new territory and, perhaps, view the world a little differently.