MODULE 1 (COMPUTER MODELING AND SIMULATION) INTRODUCTION

Module Name: Introduction to Computer Modeling and Simulation

Content of this Introduction:

1. Overview of the Module.
2. Prerequisite knowledge and assumptions encompassed by the Module.
3. Standards covered by the Module.
4. Materials needed for the Module.
5. Pacing Guides for 6 Lessons, including Learning Objectives and Assessment Questions.

1. Overview of the Module

This module introduces basic concepts in modeling complex systems through hands-on activities and participatory simulations. A scaffolded series of highly-engaging design and build activities guide students through developing their first computer model in StarLogo Nova, a modeling and simulation environment developed at Massachusetts Institute of Technology. Students practice designing and running experiments using a computer model as a virtual test bed.

2. Prerequisite knowledge and assumptions encompassed by the Module

There are no pre-requisites for Module 1. The module was designed to be an introduction to computer modeling and simulation for students with no prior background in the topic. It is necessary to complete this module prior to commencing the Earth, Life or Physical Science module.

3. Standards covered by the Module

Please see the Standards Document for a detailed description of Standards covered by this Module, Lesson by Lesson.

4. Materials needed for this Module

You will need the following materials to teach this module:

- Computer and projector
- What is a CAS? document [for reference]
- Characteristics of a Complex Adaptive System document [for reference]

- Feedback Loops document [for reference]
- Walk and Turn StarLogo Nova models [for Lesson 1 activity]
- Red, blue, and black pens for students [for Lesson 3 activities]
- Dice and paper cups for students [for Lesson 4 activities]
- Guided Introduction to StarLogo Nova document [for reference]
- StarLogo Nova Blocks CS Concepts guide document [for reference & student handout]
- StarLogo Nova Blocks Reference Guide [for reference & student handout]
- Scientific Practices with Modeling & Simulation [student handout]
- Experimental Design form document [student handout]
- Model observation form [student handout]
- Project design form [student handout]
- Model design form [student handout]
- Lesson plans for 6 lessons
- Slide presentation with instructions
- New commands and concepts sheets for each lesson [student handout]

5. **Pacing Guides for 6 Lessons, including Learning Objectives and Assessment Questions.**
   (See following pages.)

DAY 1- Introduction to Complex Adaptive Systems and Computer Modeling and Simulation

Pacing Guide

Getting Started (Assessment)	Pre-test / Assessment	10 min
Activity 1 (New Learning)	Walk & Turn: Participatory Simulation, Computer Model (Teacher-led demo), Correspondence between the real world and the virtual world, and Parts of a StarLogo Nova model.	25 min
Activity 2 (New Learning)	Complex Adaptive Systems: Video introduction, Characteristics of Complex Adaptive Systems.	10 min
Wrap up (Reflection)	What are computer models good for?	5 min

Learning objectives: Students will…

Complex Adaptive Systems	Experience being part of a complex adaptive system. (LO 1) Learn characteristics of complex adaptive systems. (LO 6) Learn that complex adaptive systems are 1) made of many interacting parts or agents, 2) each agent follows its own rules, 3) emergent patterns can result from the interaction of agents.
Modeling and Simulation	Compare and contrast a computer simulation vs. a real-world phenomenon. (LO 2) See a demo of using a computer model to run experiments. (LO 3) Speculate as to why computer models can be valuable scientific tools. (LO 5) Learn that models are representations of reality. Not all features of the real world are incorporated in to models. Models contain assumptions. Learn how to setup and run an experiment using a model as a test bed.
Computer Science	Investigate the parts of a StarLogo Nova user interface and paradigm. (LO 4) Learn that programs consist of simple instructions that are executed in a sequence. Each time the instructions are repeated in a loop. Each time through the loop is called an iteration.

Assessments of understanding:

Complex Adaptive Systems	What are four necessary characteristics of complex adaptive systems? [They are 1) made of many interacting parts or agents, 2) each agent follows its own rules, 3) emergent patterns can result from the interaction of agents and 4) hard to predict.]
Modeling and Simulation	Why are models useful? How can computer models be used to learn about the real world? What can be different about a model vs. the world?
Computer Science	What is an instruction? What is a loop? What is an iteration? What are the parts of the StarLogo Nova user interface?

DAY 2- Introduction to StarLogo NOVA and building Painting Turtles

Pacing Guide

Getting Started (Review)	Review of the previous day's lesson and concepts; connection to today's lesson.	5 min
Activity 1 (Guided Practice)	Guided Tour of StarLogo Nova: Guided Tour and Observations and Ethical considerations concerning remixing and sharing.	30 min
Activity 2 (Discovery/Creative)	Painting Turtles Challenge: Pair programming; new StarLogo Nova commands, and Painting Turtles Challenge.	10 min
Wrap up (Reflection)	What does Painting Turtles have to do with Modeling and Simulation?	5 min

Learning objectives: Students will…

Complex Adaptive Systems	Learn that in complex adaptive systems one type of interaction is that agents impact their environment.(LO7)
Modeling and Simulation	Create a model in which agents impact their environment. (LO8)
Computer Science	Get comfortable with the StarLogo Nova programming environment. (LO9) Create a program containing simple instructions that are executed in a loop. (LO10) Trace a program's execution. (LO11) Change variables to alter turtle movement. Use randomness. (LO12)
Mathematics	Turn angles, random function, relative vs. absolute position, and heading. (LO13)

Assessments of understanding:

Complex Adaptive Systems	Which characteristics of a complex adaptive system are seen in Painting Turtles? Is Painting Turtles a model of a complex adaptive system? Why or why not?
Modeling and Simulation	Is the painting made by turtles repeatable? If I run the program again, will it produce the same drawing? Why or why not?
Computer Science	What variables were used in Painting Turtles? What is the difference between right turn 90 degrees vs. right turn random 90 degrees?
Other	Math: turn angles, random function, relative vs. absolute.

DAY 3- Conditional branching with Trailblazer and Bumper Turtles

Pacing Guide

Getting Started (Review)	Review of the previous day's lesson and concepts; connection to today's lesson.	5 min
Activity 1 (Guided Practice)	Trailblazer: Blazing a Trail and Comparing Solutions; and New CS Concepts: branching and Boolean.	20 min
Activity 2 (Discovery / Creative)	Bumper Turtles Challenge: Introduce new StarLogo Nova commands and the Challenge.	20 min
Wrap up (Reflection)	What does Bumper Turtles have to do with Modeling and Simulation? Is Bumper Turtles a model of a complex adaptive system? Why or why not?	5 min

Learning objectives: Students will…

Complex Adaptive Systems	Learn that in complex adaptive systems one type of common interaction is that agents react to their environment. (LO14)
Modeling and Simulation	Create a simple model in which agents react to their environment. (LO15)
Computer Science	Learn CS concepts of booleans, logic, and conditionals. (LO18) Use booleans and conditional branching to implement agents that can react to their environment. (LO17) Trace a program's execution. (LO16) Compare solutions to Trailblazer using number of steps or number of instructions as a metric. (LO20)
Other	Practice Pair Programming and Iterative design, implement, and test cycle (LO19)

Assessments of understanding:

Complex Adaptive Systems	Which characteristics of complex adaptive systems do you see in Bumper Turtles?
Modeling and Simulation	Give an example of how agents reacting to their environment may be used to represent a behavior in the real-world. In your own words, how can if/then logic be used in a computer model?
Computer Science	How would you assess which Trailblazer solution is the best?
Other	Logic: What is the difference between a series of if/then statements and nested if/then/else statements?

DAY 4- Colliding Turtles and Probability in Models

Pacing Guide

Getting Started (Review)	Review of the previous day’s lesson and concepts; connection to today’s lesson.	5 min
Activity 1 (Guided Practice)	Probability with Dice and Data: “Chances Are” and “Wiggle Walk.”	20 min
Activity 2 (Discovery / Creative)	Colliding Turtles: New concepts and the Challenge (adding a behavior that takes place upon collision).	20 min
Wrap up (Reflection)	What could collisions represent in the real world? How does probability play a role in modeling and simulation?	5 min

Learning objectives: Students will…

Complex Adaptive Systems	Learn that in complex adaptive systems one type of common interaction is of agents interacting with other agents. (LO21)
Modeling and Simulation	Create a simple model in which agents interact with other agents upon collision. (LO22) Use a random function to implement probabilistic outcomes/behaviors. (LO23) Learn the concept of random numbers. (LO24)
Computer Science	Understand the concept of collisions and bounding boxes around objects. (LO25)
Other	Learn mathematical concepts: probability; distributions resulting from 1 die and 2 dice throws. (LO26)

Assessments of understanding:

Complex Adaptive Systems	Which characteristics of complex adaptive systems can you identify in Colliding turtles?
Modeling and Simulation	Give an example of how agents interacting with other agents may be used to represent something in the real world. How does using probability impact the outcome when running simulations?
Computer Science	Identify variables used in Colliding Turtles.
Other	What is the difference in the outcome between “right turn random 90 degrees” and “right turn random 90 degrees followed by a left turn random 90 degrees”?

DAY 5: Modeling the Spread of Disease

Pacing Guide

Getting Started (Review)	Review of the previous day's lesson and concepts; connection to today's lesson.	5 min
Activity 1 (New Learning)	Introduction to Epidemiology: Methicillin resistant Staphylococcus Aureus and modeling the spread of disease as a complex adaptive system phenomenon.	10 min
Activity 2 (Guided Practice / Discovery)	Modeling the Spread of Disease: Altering colliding turtles to make an epidemic model; adding a slider for transmission rate; customizing the model by adding another factor (such as recovery rate).	30 min
Wrap up (Reflection)	What can this model tell you? Can it be trusted to tell us anything about the real world? What other things move through a population like a disease?	5 min

Learning objectives: Students will…

Complex Adaptive Systems	Learn about epidemiology and how epidemics can be modeled as complex adaptive systems. (LO27)
Modeling and Simulation	Alter the Colliding Turtles model to create the contagion model then use the contagion model as an experimental test bed. Conduct experiments, collect and analyze data.
Computer Science	Learn new CS concepts: procedures and variables. (LO29) Create and use sliders to set variables and initial conditions. (LO30) Create procedures and call procedures. (LO31)
Other	Use the random function to simulate probabilistic outcomes. (LO32)

Assessments of understanding:

Complex Adaptive Systems	Which characteristics of complex adaptive systems can you identify in the contagion model?
Modeling and Simulation	What are other things that spread through a population like a disease? How does using probability impact the outcome when running simulations? What assumptions are made in this contagion model?
Computer Science	How would you change the model to one in which sick agents get healthy again after colliding with a healthy agent?
Other	If time allows, discuss how model might be modified to reflect real world.

DAY 6: Adding Instrumentation to your model and running experiments

Pacing Guide

Getting Started (Review)	Review of the previous day's lesson and concepts; connection to today's lesson.	5 min
Activity 1 (Guided Practice)	Add instrumentation to your model: review qualitative vs. quantitative data, and add a line graph. Then test your model.	10 min
Activity 2 (Guided Practice / Discovery)	Running experiments: designing experiments, running experiments, and collecting and analyzing data output from models. Introduce the concept of parameter sweeps.	30 min
Wrap up (Reflection)	What patterns did you uncover? What conditions led to each pattern? Is the result of a simulation always the same?	5 min

Learning objectives: Students will…

Complex Adaptive Systems	Students will understand that computer models are used by scientists to study and understand real-world problems. (LO38)
Modeling and Simulation	Learn how to instrument a model with a line graph. (LO34) Learn experimental design using a computer model. (LO35) Conduct experiments using a model as an experimental testbed. (LO36) Record and analyze results. (LO37)
Computer Science	Learn how computer science is integrated into science through scientific inquiry using computer models and simulation.
Other	Learn the difference between qualitative vs. quantitative results. (LO33) Ask questions that arise from observations of your model's behavior. (SEP)

Assessments of understanding:

Complex Adaptive Systems	Which characteristics of complex adaptive systems can you identify in your epidemic model?
Modeling and Simulation	What is included and what is missing from your model? Name two things that happen in real life that are not part of this model.
Computer Science	How might you use computer science to investigate whether a new fad will spread through your school?
Other	If time allows, discuss how the model might be modified to help you study a real world disease.

1

Lesson 1

Introduction to Complex Adaptive Systems and Computer Modeling and Simulation

50 minutes (1 day)

Lesson Overview

In this lesson students will participate in two activities. In the first activity is a participatory simulation called “Walk & Turn.” After playing the “game” consisting of following simple rules, students will see a computer model based on the same activity. Of particular interest is the comparison of real and virtual worlds and consideration of how a computer model might help us understand a phenomenon in the real world. Students will be exposed to characteristics of complex adaptive systems: many agents, simple rules, emergent patterns, and “adaptive” to change. In the second activity, students will be led “under the hood” and get an overview of some of the parts of a computer model.

Teaching Summary

Getting started – 10 minutes

1. Pre-test / assessment (short survey to assess existing knowledge)

Activity #1: Walk & Turn – 25 minutes (New Learning)

2. Participatory Simulation
3. Computer model, teacher-led demonstration
4. Correspondence between real-world and virtual world
5. Under the hood: parts of a StarLogo Nova computer model

Activity #2: Complex Adaptive Systems – 10 minutes (New Learning)

6. Video “Introduction to Complex Systems”
7. Characteristics of complex adaptive systems

Wrap-up – 5 minutes (Reflection)

8. What are computer models good for?
9. Review of new terms used

Lesson Objectives

The student will:

- ✓ Experience being part of a complex adaptive system (LO1)
- ✓ Compare and contrast a computer simulation with a real-world phenomenon (LO2)
- ✓ See a demo of using a computer model to run experiments (LO3)
- ✓ Investigate the parts of a computer model (LO4)
- ✓ Speculate as to why computer models can be valuable scientific tools (LO5)
- ✓ Learn characteristics of complex adaptive systems (LO6)

Teaching Guide

Materials, Resources and Preparation

For the Students

- Pre-test/survey

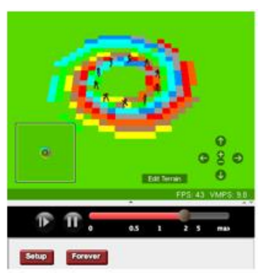
For the Teacher

- Large open space
- Computer and projector
- Walk & Turn StarLogo Nova model

Getting started - 10 min

1. Pre-test / assessment

Activity #1: Walk & Turn - 25 min



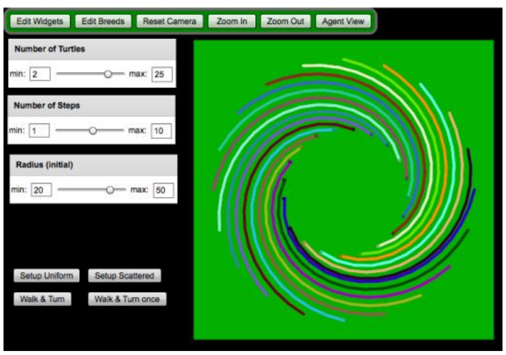
2. Participatory simulation

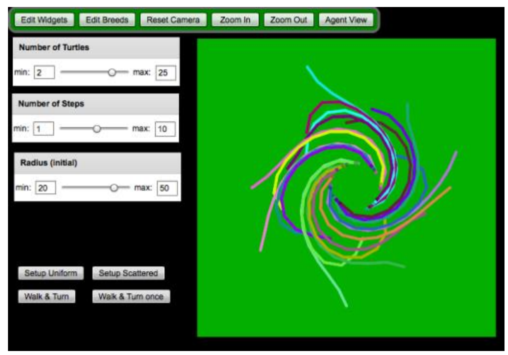
- Tell the students that in "Walk & Turn" each student will play the role of an "agent" following "simple rules." Ask students to form a circle standing at least 5 feet apart. (You can also ask students, "How would you go about forming a circle if you all started out bunched up in the center of the space?")
- Next, explain the following rules: "Turn to face the person on your left while keeping this heading, take three steps." Demonstrate how agents are supposed to keep the same heading rather than adjust the heading at each step. If necessary, have the "agents" close their eyes when taking steps to keep the heading.
- Have students try it once then ask for predictions about what will happen if they repeat the rule over and over again. (Use the term "iteration" as "doing something over and over again.") (Practice 1: Asking questions and defining problems)

- Have students design and conduct an experiment and test their predictions in real life.
- After playing a few times ask what they observed and if their predictions were correct.
- Ask students what questions they have about the phenomenon. (Practice 1: Asking questions and defining problems)
- If they don't offer one, ask "What if, instead of forming a circle, you spread out randomly?"
- Ask what the result of following the simple rules would be if, at the start, the agents were scattered randomly, rather than arranged in a circle. Have students make predictions then conduct the experiment and test their predictions in real-life.

  Teaching Tip *In a large class, select 8-12 students to play the part of agents in the Walk & Turn participatory simulation. The other students can be "observers."*

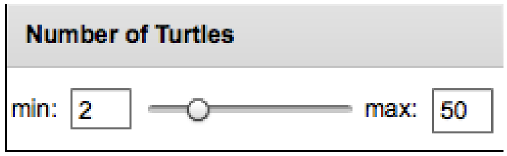
  Teaching Tip *To ensure that students keep their heading constant for three steps, have students close their eyes before taking three steps.*





3. Computer model, teacher-led demonstration

- Bring students back into the classroom and together look at a computer model of the Walk & Turn activity. Open the computer model and project it on a screen.
- Tell students that this is a computer model that was designed by a "modeler" to resemble the activity we just completed.
- Some aspects of the real world were included in the model; others were left out.
- Introduce StarLogo Nova as an agent-based modeling tool to be used by them. It has a user interface through which students will build a computer program.
- Demonstrate what is happening with agents starting in a uniform circle versus scattered.
- Using the computer model as a test bed, ask students what questions they would like to investigate or what experiments they would like to run, and what to change in the model in order to run experiments. (Practice 1: Asking questions and defining problems)
- Demonstrate running the experiment using the model as-is and change the settings one at a time. (Practice 2: Developing and using models)
- Demonstrate changing settings such as the number of agents or the size of the circle.
- Ask what other "variables" you could change. [number of steps to take, direction to turn (left or right)]



- Ask “What pattern do you see forming?” (CCC: Patterns)
- Ask “Does the same pattern emerge, regardless of the setting of the variable?”
- Ask “Does this model and its outcome match what we experienced in real-life?” and “What are some similarities and differences between the model and what we did in real life?” [Example: In the model, all people took the same size steps and they all followed directions, while in real life, people have different strides and don’t always follow directions.]
- Discuss the pattern or lack of pattern that the computer model produced. (CCC: Patterns)
- Ask “Was your prediction accurate?” (Practice: Analyzing and Interpreting Data)
- Does the real-life outcome match what the model generated? Why or why not?
- Ask “Did the outcome (the emergent pattern) match what happened the first time when students were in a circle?” and “Would we get the same pattern again if we scattered randomly again and followed the simple rules?”
- Explain that a system that adjusts to changes and produces similar patterns is called “adaptive” and that being adaptive is a characteristic of complex adaptive systems.

4. Correspondence between real-world and computer model?

- Discuss the relationship between the computer model and the real world phenomenon.
- What were some of the parts of the real world that were represented in the model?
- What was left out of the model? [stride length, people not following directions]
- What assumptions were made in the model?   [people form perfect circles]
- Review the new terms used in this activity: agent, simple rules, heading, iteration, prediction, emergent patterns, random, initial condition, outcome, phenomenon, and adaptive.

  Teaching Tip *Keeping a “word wall” with new terminology and definitions is highly recommended.*

  Teaching Tip *Some teachers find it helpful to relate simulations to computer games but do so with caution. Computer games often have pre-determined conclusions or predictable behavior that was “programmed in,” but, in contrast, computer models of complex adaptive systems do not.*

5. Under the Hood: Start with a new project

- StarLogo Nova is an agent-based modeling tool. StarLogo Nova has a user interface, or way that a user can interact with it. Within StarLogo Nova you can build your “computer model” by building a computer program. Your computer program is your “project” that is stored online.
- Show the three areas of a StarLogo Nova User Interface (info, display, code)
  - The top part, called **the Information area,** is a place for you to give your model a title and record notes about your model.
  - The middle part, called **Spaceland,** is a 3D world where the agents are displayed. It consists of a green flat plane (called the terrain). Note that it has several buttons that can be used to control the simulation. The “**setup**” button, “**forever**” button and “**score**” box are called “**widgets**” or “user interface elements.”
  - The bottom part, called the **workspace** or **blocks area**, contains drawers of blocks that

represent the programming commands, and pages where you snap the blocks together to write programs.

- Show some simple commands on blocks, and say that they snap together like Lego blocks.
- Show where these blocks are located in drawers and if time allows setup a simple sequence of steps and execute.

Activity #2: Complex Adaptive Systems - 10 min

6. Watch the video "Introduction to Complex Systems" (MIT_Video_CAStoABM.mp4)

7. Review the characteristics of complex adaptive systems (Characteristics of CASv.pdf)

Wrap-up - 5 min

8. What are computer models good for?

- Would you trust a computer model if your life depended on it?
  [What features of the real world are left out? Do those features matter?]
- What are models good for?
- New Terms: agent, simple rules, heading, iteration, prediction, emergent patterns, scatter, initial conditions, outcomes, phenomenon, and adaptive.

Assessment Questions

- How can computer models be used to learn about the real world?
- What kinds of things would you rather model on a computer than in real life?
- What are some key differences between a model and the real world?
- We've learned about three characteristics of complex adaptive systems (many interacting agents or parts, simple rules and emergent pattern). Given those characteristics, is a clock a complex adaptive system? Why or why not?

Background information (Optional reading assignment for students)

- What is a Complex Adaptive System? (What is a CAS.pdf)
- Characteristics of Complex Adaptive Systems (Characteristics of CAS.pdf)
- Feedback Loops (Feedback Loops.pdf)

Standards Addressed

NRC Scientific and Engineering Practice Standards

Practice 1: Asking questions and defining problems
1A: Ask questions that arise from careful observation of phenomena, models, or unexpected results.
1F: Ask questions that can be investigated within the scope of the classroom, outdoor environment, and based on observations and scientific principles.

Practice 2: Developing and using models
2A: Evaluate limitations of a model of a system (not for a proposed object or tool.)
2C: Use a model of simple systems with uncertain and less predictable factors.
2E: Use a model to predict and/or describe phenomena.

2G: Use a model to generate data to test ideas about phenomena in natural or designed systems, including those representing inputs and outputs, and those at unobservable scales.

Practice 3: Planning and carrying out investigations
3D: Produce data to serve as the basis for evidence to answer scientific questions or test design solutions under a range of conditions.
3E: Collect data about the performance of a proposed system under a range of conditions.

Practice 4: Analyzing and interpreting data
4B: Use graphical displays to identify temporal and spatial relationships.
4D: Analyze and interpret data to provide evidence for phenomena.

Practice 5: Using mathematics and computational thinking
5A: Use digital tools (e.g., computers) to analyze data sets for patterns and trends.
5B: Use mathematical representations to describe and/or support scientific conclusions and design solutions.

Practice 6: Constructing explanations and designing solutions
6B: Construct an explanation using models or representations.

Practice 7: Engaging in argument from evidence
7C: Construct and present an oral argument to support or refute a model for a phenomenon.

NRC Crosscutting Concepts

1. Patterns:
1D: Graphs, charts, and images can be used to identify patterns in data.

3. Scale, Proportion, and Quantity
3A: Time, space, and energy phenomena can be observed at various scales using models to study systems that are too large or too small.
3E: Phenomena that can be observed at one scale may not be observable at another scale.

4. Systems and Systems models
4B: Models can be used to represent systems and their interactions—such as inputs, processes and outputs—and energy, matter, and information flows within systems.
4C: Models are limited in that they only represent certain aspects of the system under study.

7. Stability and Change:
7A: Explanations of stability and change in natural or designed systems can be constructed by examining the changes over time and forces at different scales, including the atomic scale.

CSTA K-12 Computer Science Standards

CT	Modeling & simulation	2-11	Analyze the degree to which a computer model accurately represents the real world.
CT	Modeling & simulation	2-9	Interact with content-specific models and simulations to support learning and research.
CT	Modeling & simulation	3A-8	Use modeling and simulation to represent and understand phenomena.
CT	Modeling & simulation	3B-8	Use models and simulation to help formulate, refine, and test scientific hypotheses.
CT	Modeling & simulation	3B-9	Analyze data and identify patterns through modeling and simulation.

Responsiveness to Varied Student Learning Needs

In Project GUTS, we integrate teaching strategies found to be effective with learners with various backgrounds and characteristics such as economically disadvantaged students (EDS), students from groups that are underrepresented in STEM (URG), students with disabilities (DIS), English Language learners (ELL), girls and young women (FEM), students in alternative education (ALT), and gifted and talented students (GAT).

In each lesson we describe the accommodations and differentiation strategies that are integrated in the activities to support a wide range of learners.

Module 1 Lesson 1 Activity #1: Walk and Turn Participatory Simulation & Model

(EDS) *We validate the use of place [by situating the experiment within the school setting] to keep the students engaged and make a connection of science and school/neighborhood.*

(URG) *We use technology to present information in multiple modes of representations. We choose a modeling and simulation activity that involves student movement, a strategy that uses a multi-modal experience to increase student engagement.*

(DIS) *We use technology to present information in multiple modes of representations. We provide multiple means of action, expression, representation and engagement. These are all principles of Universal Design for Learning.*

Module 1 Lesson 1 Activity #2: Introduction to Complex Adaptive Systems

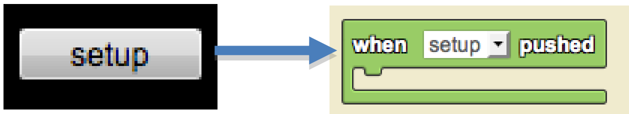
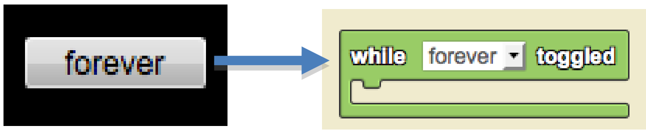
(EDS) *We validate the sense of place [by describing neighborhood phenomena such as traffic patterns and ecosystems as complex adaptive systems] to keep the students engaged and make a connection of science and neighborhood.*

(ELL) *We recommend using a word wall with words with photos or to represent concepts as a language support strategy for English language learners.*

(FEM) (URG) *We choose a curriculum topic, Complex Adaptive Systems, which has relevancy and real-world application, to interest and engage the girls and students from underrepresented groups in STEM in the class.*

StarLogo Nova Blocks introduced in Module 1 Lesson 1

Event Handling

	When “setup” (on the interface) is clicked, code blocks placed inside this block will run once.
	When “forever” (on the interface) is clicked it runs code blocks inside this block until “forever” is clicked again.

1

Lesson 2

Introduction to StarLogo Nova and Building Painting Turtles

50 minutes (1 day)

Lesson Overview

In this lesson students will participate in two activities. The first activity is a guided introduction to the StarLogo Nova simulation environment. In the second activity, students will learn a few simple commands and then create their first computer program. Students will progress from single turtle explorations to instructing many turtles to follow their commands in parallel.

Teaching Summary

Getting started – 5 minutes (Review)

1. Review of the previous day's lesson and concepts and connection to today's lesson.

Activity #1: Guided Tour of StarLogo Nova – 30 minutes (Guided Practice)

1. Guided tour
2. Observations and ethical considerations

Activity #2: Painting Turtles Challenge – 10 minutes (Discovery / Creative)

3. Pair programming
4. New StarLogo Nova commands
5. The Challenge

Wrap-up – 5 minutes (Reflection)

6. What does Painting Turtles have to do with Modeling and Simulation?

Lesson Objectives

The student will:

- ✓ Learn that in complex adaptive systems one type of interaction is that agents impact their environment (LO7)
- ✓ Make a model in which agents impact their environment (LO8)

- ✓ Get comfortable with the StarLogo Nova programming environment (LO9)
- ✓ Make a program that contains simple instructions that are executed in a loop (LO10)
- ✓ Trace a program's execution (LO11)
- ✓ Change variables to alter turtle movement, use randomness (LO12)
- ✓ Use turn angles, random function, relative vs. absolute position and heading (LO13)

Teaching Guide

Materials, Resources and Preparation

For the Students

- Computers
- Guided Introduction to StarLogo Nova
- Module 1 Lesson 1 & 2 Blocks reference
- Module 1 Lesson 1 & 2 CS Concepts

For the Teacher

- Computer and projector
- Slide presentation with simple commands

Getting started - 5 min

1. Review of previous day's lesson and link to where we are going today.

- What do computer models let you do that you cannot in real life?
- What was NOT captured in the Walk & Turn model that happened in real-life?

Activity #1: Guided Tour of StarLogo Nova - 30 min

2. Guided tour

- Assign partners to share a computer or work individually.
- Hand out the "Guided Introduction to StarLogo Nova" tutorial.
- Teacher-led overview of the first 5 pages of the Guided Introduction.
- Have students progress through the tutorial, completing each task before moving on.

3. Observations and ethical considerations

- What did you notice about the way the StarLogo Nova drawers are organized?
- What does it means to be a web-based environment – where does my program live?
- What does it mean to REMIX a program? Is it cheating?

Teaching Tip *This guided tour activity can be run as a group activity by asking a pair of students to come up to the projecting computer and attempt to complete a task in front of the class before moving on to the next task with a new pair of students demonstrating.*

Activity #2: Painting Turtles - 10 min

4. Pair programming

- Describe the roles of navigator and pilot.

- "We will be using a programming practice called 'Pair Programming.' In this practice two people are paired to work together on a project. One person takes the role of the navigator who tell the other person, the pilot, what to do. The pilot listens to the navigator and puts the blocks in place. After 10 minutes I will tell you to switch roles and keep on working."
- Let students know that they are to switch roles after 10 minutes on your command.

5. New commands

- Review the commands: create agents, forward, left by __ degrees, pen down, when ___ pushed, and while ___ toggled.
- Show new StarLogo Nova commands: right by __ degrees, backward, and random.
- Review a basic framework (setup and runtime) and tell students that they are going to create their own painting turtles program by programming or giving the agents instructions.
- Ask students to REMIX the starter Painting Turtles model from your public gallery. (Alternatively, start by remixing the Draw a Flower project from the Guided Introduction.)

6. The Painting Turtle Challenge

- Give the students the specifications for the program. It must do the following: Create turtles, have the turtles move around the space leaving trails, and have a wiggle to their walk. [Hint: use the random block inside of the right and left turn block to introduce randomness into a turtle's walk. We call this a "wiggle."]
  (CT-Algorithms 3A-3; CT-Data representation 3A-12)

- If time allows, have the students upload and share their projects.
- If time allows, talk about the execution model. Why does execution order matter?

Wrap-up – 5 min

7. What does Painting Turtles have to do with Modeling and Simulation?

- What could these trails represent? (CT-Modeling and simulation 2-9)
- Review the new terms used: agent, location, heading, steps, iteration, setup, runtime, random.

Assessment Questions

- Is Painting Turtles a model of a complex adaptive system? Why or why not?
- What variables were used in Painting Turtles?
- What is the difference between turning 90 and turning random 90?
- Is the painting made by turtles repeatable? If I run the program again, will it produce the same drawing? Why or why not? (CPP-Data collection & analysis 2-9; CT-Modeling & simulation 2-9)

Background Information

Guided Introduction to StarLogo Nova

Standards Addressed

CSTA K-12 Computer Science Standards

CT	Algorithms	3A-3	Explain how sequence, selection, iteration and recursion are the building blocks of algorithms.
CT	Modeling & simulation	2-9	Interact with content-specific models and simulations to support learning and research.
CPP	Data collection & analysis	2-9	Collect and analyze data that are output from multiple runs of a computer program.
CPP	Programming	2-5	Implement a problem solution in a programming environment using looping behavior, conditional statements, logic, expressions, variables and

Responsiveness to Varied Student Learning Needs

In Project GUTS, we integrate teaching strategies found to be effective with learners with various backgrounds and characteristics such as economically disadvantaged students (EDS), students from groups that are underrepresented in STEM (URG), students with disabilities (DIS), English Language learners (ELL), girls and young women (FEM), students in alternative education (ALT), and gifted and talented students (GAT).

In each lesson we describe the accommodations and differentiation strategies that are integrated in the activities to support a wide range of learners.

Module 1 Lesson 2: Guided Tour of StarLogo Nova and the Painting Turtles Challenge

(URG)(DIS) *We use technology to present information in multiple modes of representations. In the StarLogo Nova modeling and simulation environment students can present information as code blocks, text, visual imagery, and as data in tables and graphs.*

(FEM) *We recommend careful planning of partners for pair programming in the Guided Tour and Painting Turtles Challenge on-computer activities, a practice that encourages participation for the girls in science.*

Student Activity #1 Guide:

Guided Introduction to StarLogo Nova

Who is this for?
This is a Guided introduction to StarLogo Nova that was designed for teachers and students. Teachers participating in the Code.org Project GUTS Middle School Computer Science in Science Program are being asked to complete the tutorial before showing up to the face-to-face workshop. Students of participating teachers are being asked to complete the tutorial during Module 1 Lesson 2 (day 2 of module 1).

What is StarLogo Nova?
StarLogo Nova is an agent based modeling (ABM) environment that was created by the Scheller Teacher Education Program at Massachusetts Institute of Technology. StarLogo Nova is the newest version in a long line of Logo based educational environments. StarLogo Nova is special because it is entirely web-based. Since it is web-based, you do not need to download or install any software on your computer or school's server.

Where can I find it?
To get started in StarLogo Nova, all you need to do is go to slnova.org and request an account.

When will I use it?
In Code.org's Computer Science in Science modules we will use StarLogo Nova to build and experiment with models of complex adaptive systems within regular school day classes.

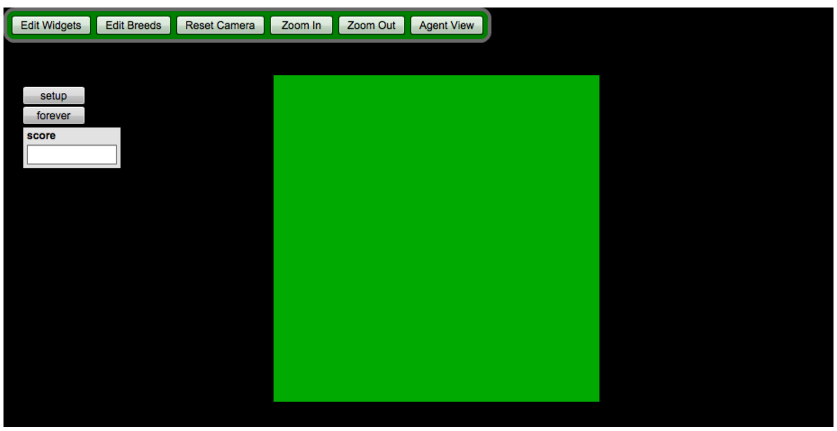
Why is this important?
StarLogo Nova gives students and teachers access to tools, similar to those used by scientists and researchers, with which to study systems in the world around them.

Now let's move on to some background information and a tour.

The StarLogo Nova User Interface:

The StarLogo Nova User Interface is made up of three parts.

The top part, called **the Information area,** is a place for you to give your model a title and record notes about your model.

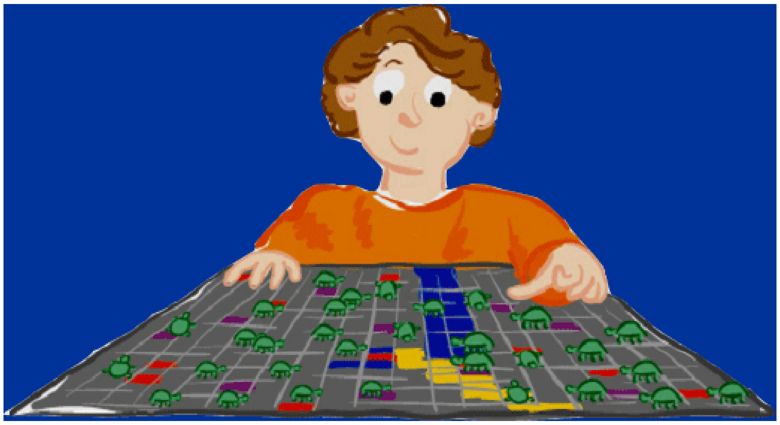


The middle part, called **Spaceland,** is a 3D world where the agents are displayed. It consists of a green flat plane (called the terrain). Note that it has several buttons that can be used to control the simulation. The "**setup**" button, "**forever**" button and "**score**" box are called "**widgets**" or "use interface elements".

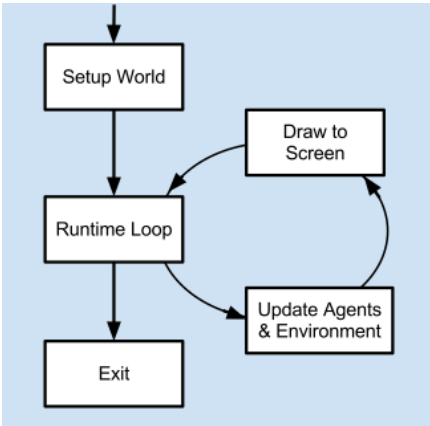
The bottom part, called the **Workspace or blocks area**, contains drawers of blocks that represent the programming commands, and pages where you snap the blocks together to write programs. Programming is basically putting together a logical sequence of instructions.

The StarLogo Nova Modeling and Simulation Paradigm

In StarLogo Nova, a model consists of the environment or terrain, the agents who represent individual characters in the model (whether they are water molecules, fish, or even humans), and interactions. Interactions can take place between agents and their environment or agents and other agents.
A "model" is the virtual world you set up in StarLogo Nova. It can contain agents, the environment and behaviors that have been specified in computer code. We call "running the model" forward in time "Simulation".



There are two main phases in Simulation. The first phase is called the "Setup" or "Initialization" phase in which the world and the agents are created. The second phase is called the "forever" or "runtime" phase. In this second phase, each agent follows its set of instructions then waits for the other agents to finish their set of instructions before taking another turn. Before taking another turn, time moves forward, and Spaceland gets redrawn with the updated state of the world and the agents.



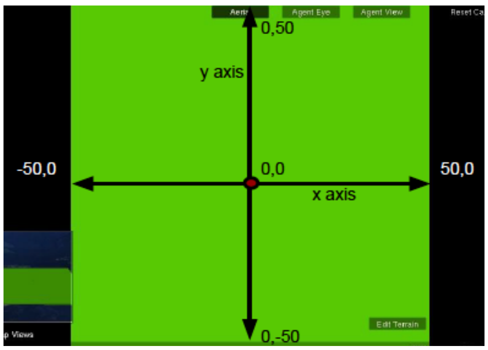
There are three major parts of any agent-based model: agents with rules that they follow, the environment in which they coexist, and time. In StarLogo TNG, the first two are easy to see – the agents are the different turtles and the environment is made up of square tiles called patches. Time is harder to see, instead it can be thought of as a series of time slices or "ticks". At each tick, all of the agents have a chance to update their position or state. Ticks or time slices are not the same as seconds because it may take more or less than one second to update all of the agents.

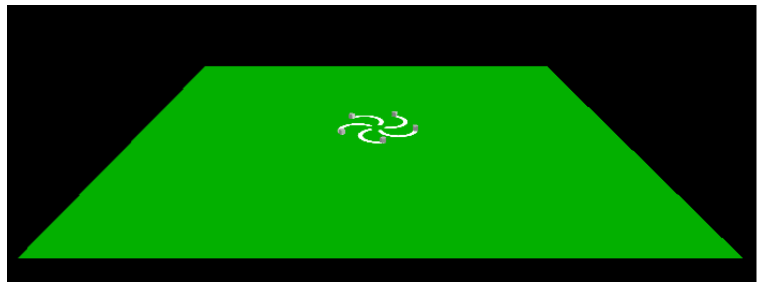
As a modeler, you decide what gets included (and what gets left out) of the model, and you get to write the instructions that tell each agent what behaviors to carry out.

StarLogo Nova Components:

Terrain

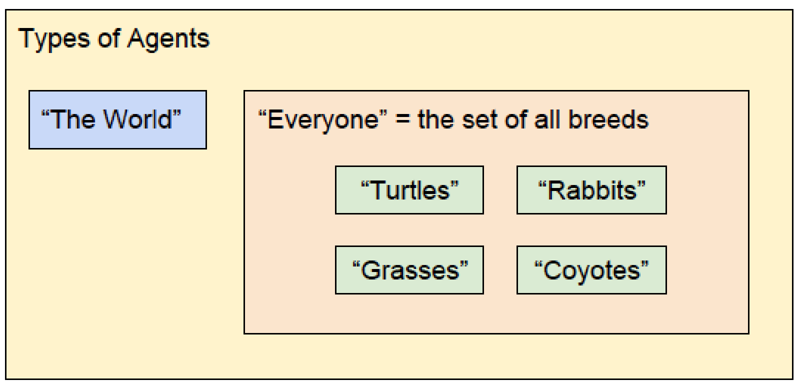
The Terrain is a coordinate plane that is 101 tiles long by 101 tiles wide. The center of the terrain has coordinate (0,0). You can see different views of the terrain by maneuvering the "camera" using the control, option, and command keys. Simply click on the key and click and drag on the terrain. Try seeing the world from various perspectives.



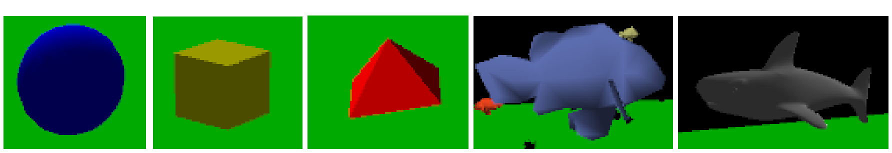


Agents and Breeds

Agents can take on different characteristics and behaviors. We often call agents "turtles" for historical reasons – the first agents in Logo were robot like "turtles". Each agent can be represented by a 3D model. You can select shapes for your agents from the library of 3D models.



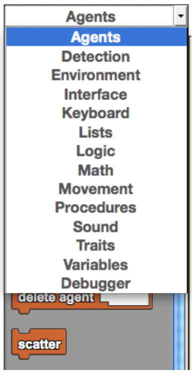
There can be many different types of agents within a model. Each different type of agent is called a breed. For example, in a model of rabbits and coyotes, the rabbits can be one breed of agents while the coyotes can be another breed of agents. "Everyone" refers to all of the agents except for "The World". "The World" is a special "super" agent that is responsible for setting up the world and creating the other agents.



Agents can be represented by 3D models from basic shapes to creatures.

Blocks and Drawers

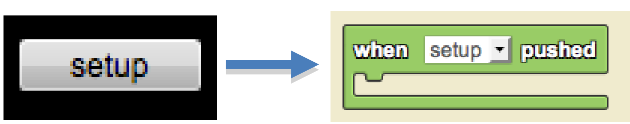
Blocks are the instructions that make up the program. There can be many different kinds of blocks that make up a program. Blocks are organized in drawers by function.

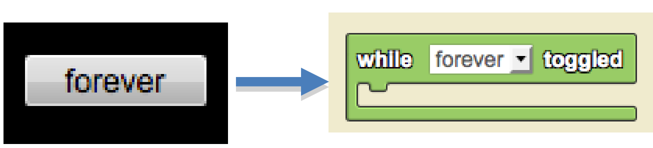


Pages and Procedures

Pages are areas where the user organizes blocks to control the different agents. There are initially three different pages: Turtle, Everyone, and the World. The blocks in the Turtle page are the instructions the turtle agents follow. When you create new breeds, new pages appear for them. Turtles are the starter breed of agents so there is a "Turtle" page by default.

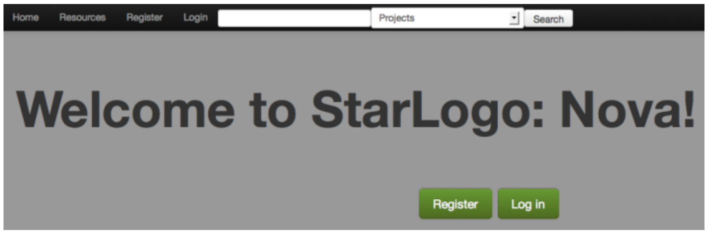
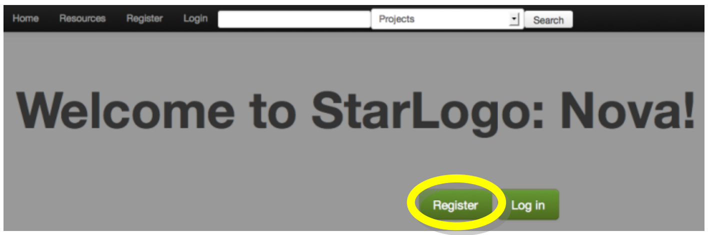
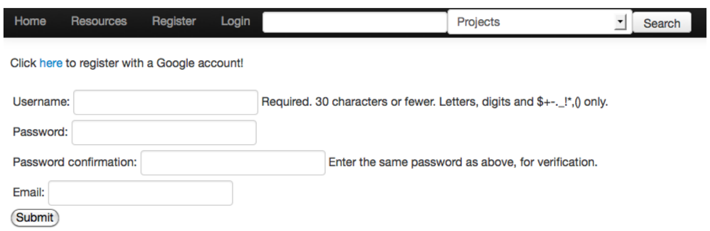
Procedures are stacks of blocks that are executed or "run" when either the user clicks a button, or when the program asks the agents to take their turn.



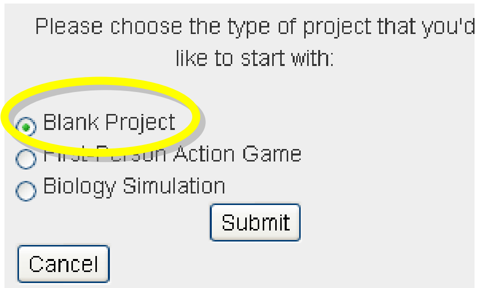
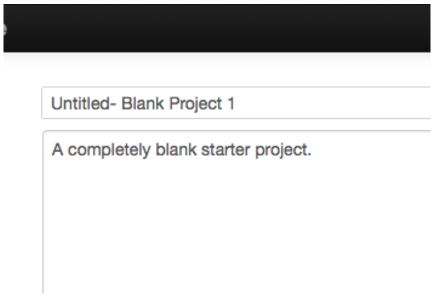
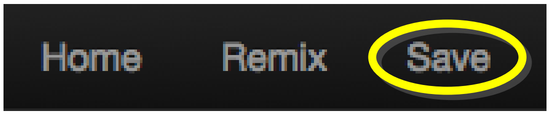


Now that you know the basics, let's get started on the Tutorial

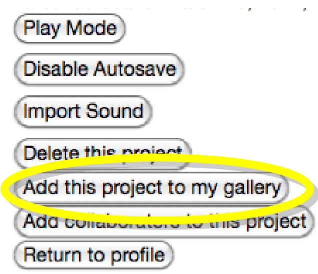
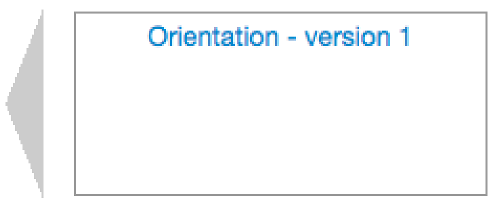
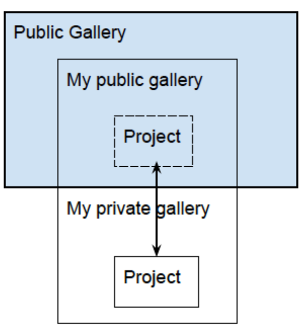
Part 1: Create a StarLogo NOVA account and log in

<table>
<tr><td>
</td><td>1) In a web browser with an internet connection, open a browser window and navigate to www.slnova.org the StarLogo Nova home page.</td></tr>
<tr><td>
</td><td>2) If you don't already have a StarLogo Nova account, click the Register link and follow the instructions.</td></tr>
<tr><td>
</td><td>3) Enter a username, a password, and an email address, then click "Submit".</td></tr>
<tr><td colspan="2">Congratulations! You now have an account on StarLogo Nova! When you set up your account you get your own public and a private gallery. A gallery is a place to store and display your projects. You also get your own personal "profile" page. Through your profile, you can access the projects you make and projects that have been shared with you.</td></tr>
</table>

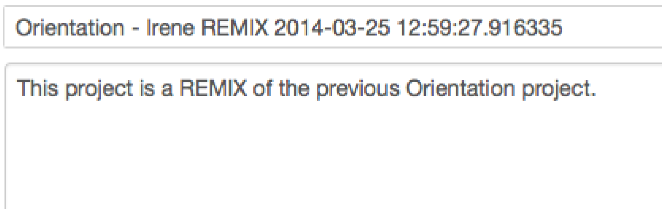
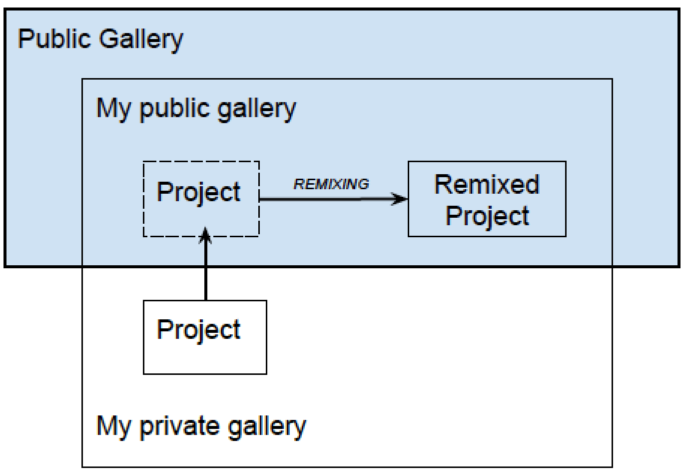
Part 2: Create a StarLogo NOVA project.

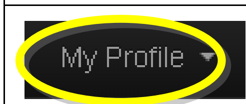
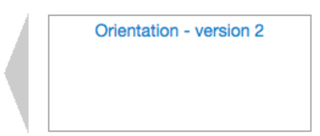
	1) After logging in successfully, click on “My Profile” in the upper right hand corner of the information area. You will see options for editing user settings.
	2) Click the “Create a new project” button to open a new project.
	3) Choose “Blank Project” and click “Submit”.
	4) You will see a new project appear with the title “Untitled-Blank Project 1”. Replace the title with “Orientation – version 1” followed by your name and your partner’s name. Then where it says “A completely blank starter project.” type something descriptive into the text box.
	5) Hit “Save”. This project is now saved in your Private Gallery. (All new projects get saved in the Private Gallery by default.)
Congratulations! You have now created and saved your first StarLogo Nova project.	

Part 3: Share your project.

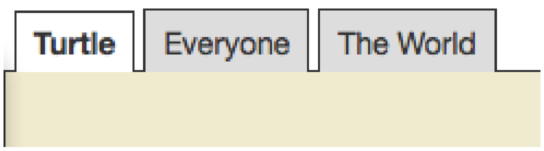
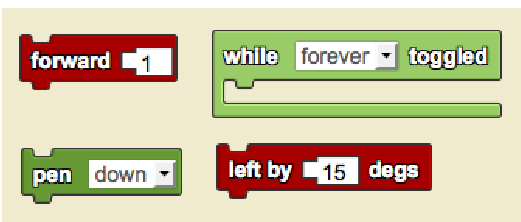
	1) Click on “Add this project to my gallery” and select your “public gallery” as the destination for this project. You will notice that your project is now shared between your public gallery and your private gallery. Once a project is in your public gallery, it can become a shared project where multiple collaborators can edit the project. Check that you can now see your project in your public gallery.
	2) Click on “My Profile” to get back to your profile page.
	3) Once there, you can see the project “Orientation – version 1” in your Public Gallery. Clicking on the project title will open the project.
	4) Let’s review what you just accomplished. You created a new project in your private gallery then you added the project to your public gallery. Your project can now be shared with others. AND you can see other user’s projects in their public galleries as well. Note that there is only one version of the project. It can be reached from both your public and private galleries.
Congratulations, you have shared your StarLogo Nova project through your public gallery! *We are asking you to store your projects in your public galleries so we can provide you with assistance as requested.	

Part 4: Remix your project.

<table>
<tr><td>
</td><td>1) Click “Remix” to create a new version of your project.</td></tr>
<tr><td>
</td><td>2) Remix is similar to “Save As” – it preserves your previous version and saves you a new version with the name “REMIX” and a timestamp at the end of the project name. Change the name of the project to “Orientation - version 2” and edit the info text as well.</td></tr>
<tr><td>
</td><td>3) Each time you remix a model, the remixed version appears in the same gallery where the original was located.

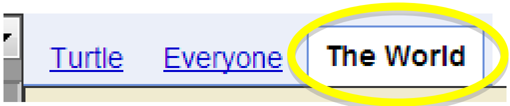
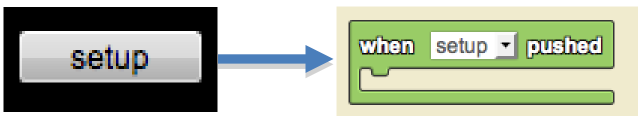
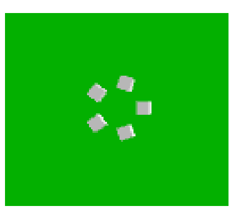
So if you remix a project in your public gallery, the remixed version is also in your public gallery. But if you remix a project in your private gallery, the remixed version is also in your private gallery.</td></tr>
<tr><td>
</td><td>4) Click on “My Profile” to get back to your profile page.</td></tr>
<tr><td>
</td><td>5) Once there, click on the project “Orientation – version 2”, this is the remix of version 1 that was saved to your Public Gallery.</td></tr>
<tr><td colspan="2">Hooray, you’ve learned how to remix a project.</td></tr>
</table>

Part 5: Find and pull out Blocks onto Pages

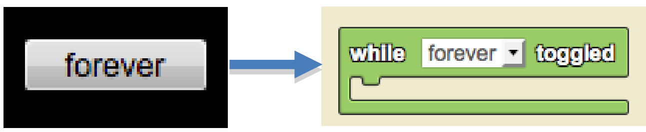
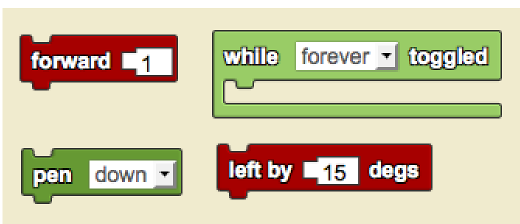
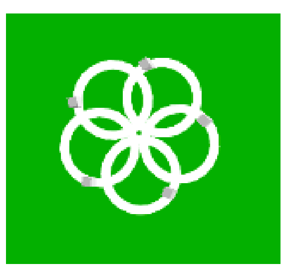
<table>
<tr><td colspan="2">Now we will be making changes in “Orientation – version 2”.</td></tr>
<tr><td>
</td><td>1. Click through the different “drawers” at the left of the Workspace or blocks area to view the “blocks” contained within each drawer. Notice how the drawers are arranged. What do all of the blocks in the “Movement” drawer have in common? What about the blocks in the “Interface” drawer?</td></tr>
<tr><td>
</td><td>2. Then click through the “Page” tabs and you will see that there are different pages for blocks. Click on the tabs to move between pages. You can tell which page you are on because it has a white tab.</td></tr>
<tr><td>
</td><td>3. Find the following blocks in the drawers and drag them out onto the “Turtle” page. What do you think these blocks do?</td></tr>
<tr><td>
</td><td>4. Find the following blocks and drag them out onto “The World” page (click on the “The World” tab first to switch pages).
What do you think these blocks do?</td></tr>
<tr><td colspan="2">Make notes in the information area on where you found the blocks. Were there any patterns or cues that helped you find the blocks? [In StarLogo Nova, each drawer holds a set of related blocks a programmer may use. For example the Interface drawer holds all “user interface” blocks.]

When you are done, remember to save the project. It will be saved in your public gallery.</td></tr>
</table>

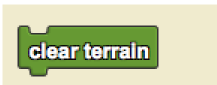
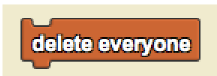
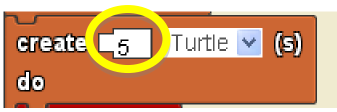
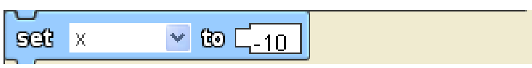
Part 6: Create agents

Remix “Orientation – version 2” and rename it as “Orientation – version 3” in your public gallery. In this part of the tutorial, you will program “The World” to create some turtle agents when the user clicks the setup button.	
	1) Click on the page tab to switch to “The World” page. You already dropped blocks onto this page in the last task.
	2) Now it is your turn to put the blocks together into a procedure that the World will carry out when the setup button is pushed.
	3) Here’s what the blocks do: The “**when setup pushed”** block on the World Page tells the world what to do when the setup button is pushed. The “**create-do**” block allows users to create particular numbers of agents of certain type or breed, then have each of the newly created agents perform the commands within the “do” area right away. The “**clear terrain**” block erases any drawing on the terrain. The “**delete everyone**” block gets rid of any agents in Spaceland. The “**forward**” block moves the agent forward.
	4) Try combining the blocks so that when the setup button is pushed, five turtles are created and each of the turtles moves forward 2 steps away from the center of Spaceland. After making changes to your program, always click “Save” to save your changes, and always click “Run Code” before running your project.
 	5) Check Your Progress: Scroll/Swipe back up to Spaceland, and click the “setup” button. When you’ve successfully assembled these blocks, you should see a set of white blocks (the agents you created) in the center of Spaceland.
Congratulations! You’ve created your first agents. Notice that they were all “born” in the center of Spaceland at coordinate (0, 0) facing outward, then they took 2 steps forward just as you instructed them to.	
Save your “Orientation – version 3” project in your public gallery.	

Part 7: Draw a flower

Remix "Orientation – version 3" and rename it as "Orientation – version 4". In this part of the tutorial, you will program the turtle agents to draw a flower when the user clicks the "forever" button.	
	1) Click on the "Turtle" tab to switch to the "Turtle" page. The turtle agents will run the blocks you assemble here.
	2) Now it is your task to put the blocks together into a procedure so that the turtle agents will each draw a circular flower petal when the forever button is toggled on. (A toggle is like a light switch, it is either on or off.)
	3) Let's review what the blocks do: The "**when forever toggled"** block on the Turtle Page tells the turtles what to do when the "forever" button is toggled to the ON position. The "**pen down**" block tells each turtle agent to leave a trail wherever it goes. The "**forward**" block moves the agent forward. The "**left by __degrees**" block tells each turtle to turn left by some number of degrees.
	4) Try combining the blocks so that when the forever button is pushed, each of the five turtles draws a petal of the flower. Don't forget to click on "Save" to save your changes. To test your changes, click "Run Code" and then use the user-interface buttons (setup and forever) in Spaceland.
 	5) Check Your Progress: Scroll back down to Spaceland, and click the "forever" button. You should see a set of white blocks (the agents you created) drawing a flower.
Congratulations! Your turtle agents drew a flower!	
Save your "Orientation – version 4" project in your public gallery.	

Part 8: Extensions (for those who want an additional challenge)

If you'd like to try these extensions, remix "Orientation – version 4" and rename it "Orientation – version 5".	
	Use "clear terrain" to erase any existing drawing on the terrain between drawings.
	Use "delete everyone" to get rid of all existing agents between drawings.
	Extension #1: Try to change the number of flower petals.
	Extension #2: Draw a flower with square petals instead of round petals. [replace the number of steps and the degrees to turn]
	Extension #3: Change the color of the flowers.
	Extension #4: Change the starting location (x, y) of agents.
Define or identify the following terms: Workspace, SpaceLand, Information area, Terrain, Page, Drawer, Blocks, Buttons, Widgets, Agents.	

End of the Guided Introduction Tutorial

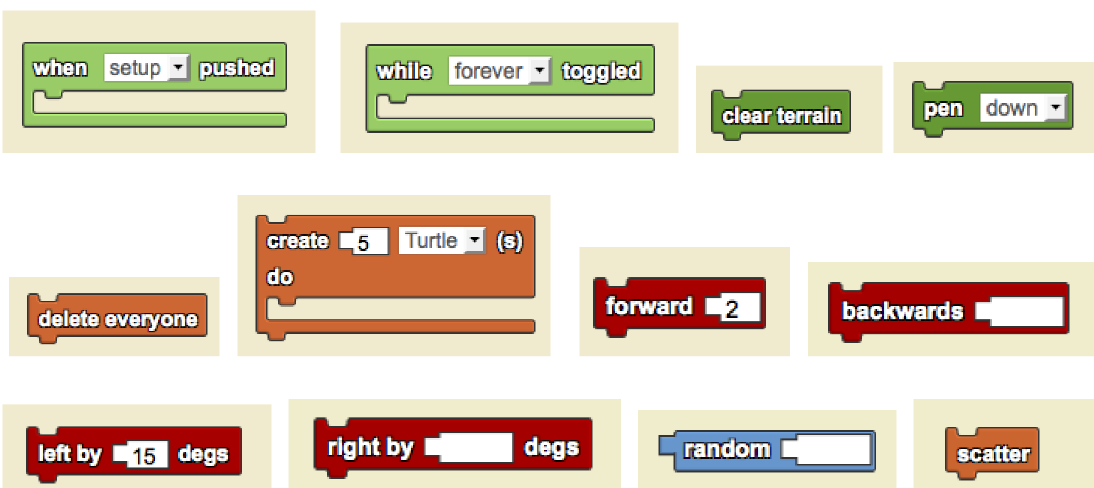
Student Activity #2 Guide:

The Painting Turtles Challenge: Paint a masterpiece!
Your challenge is to make the turtles draw paths on Spaceland and to experiment with different kinds of turtle movement.

Note: DO NOT forget to REMIX the project before making changes.

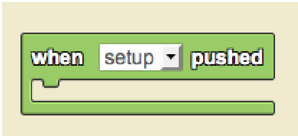
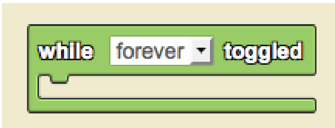
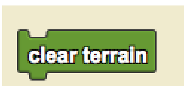
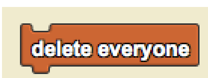
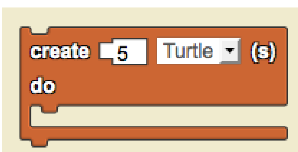
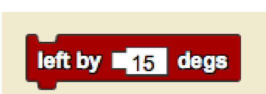
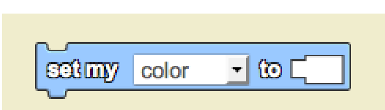
Guidelines:
Remix the “Draw a Flower” (Orientation – version 4) project from the Guided Introduction.
Don’t forget to put both partners’ names in the project title.
Create many turtles when the setup button is pressed (see the Guided Tutorial.)
Have the turtles move and leave trails with their pens down when the forever button is toggled on.
When you are done, save and share your project.

The new command blocks to be used are on page 1 of the StarLogo Nova Blocks Guide.
(Refer to the StarLogo Nova Blocks Reference Guide provided)



An extension:
Experiment with adding a slight wiggle to the turtle’s walk using the random command block. The random block can be used in place of a number anywhere a number could fit.

StarLogo Nova Blocks introduced in Module 1 Lesson 2

	Executes commands when the push button widget is pressed. Widgets can be created using the Edit Widgets feature.
	Executes commands when the toggle button widget is turned on. When the toggle button widget is turned off, nothing occurs. Widgets can be created using the *Edit Widgets* feature.
	Removes all stamping or drawing modifications done to the terrain.
	Adjusts the pen tool. If the pen is down, the agent stamps the terrain at each position as it moves along the terrain. If the pen is up, the agent does not stamp the terrain.
	Deletes all agents.
	Creates a specific number of agents of the specific breed. Attach other blocks to specify traits and properties of these agents. Additional breeds are created in the interface.
	Moves an agent forward by a certain number of steps.
	Moves an agent backwards by a certain number of steps.
	Rotates an agent left by a certain number of degrees.
	Rotates an agent right by a certain number of degrees.
	Sets an agent's trait to a particular value.

	Places agents in random positions on the terrain.
	Returns a random integer from 0 to the input but NOT including the number entered (non-inclusive).
	Random can be used in place of a number anywhere a number would fit. In this example instead of having a constant turn angle, an agent following this command would turn different amounts each time this command was executed

1

Lesson 3

Conditional branching with Trailblazer and Bumper Turtles

50 minutes (1 day)

Lesson Overview

In this lesson students will participate in two activities. In the first they will learn about conditional branching, logic, and Boolean expressions as they guide an agent through a puzzle. They also consider the best path / solution and the degree of universality of their solution. In the second activity, students will learn a few simple commands and then starting with a basic project, implement a Bumper Turtles program in which agents respond to their environment.

Teaching Summary

Getting started – 5 minutes (Review)

1. Review of the previous day's lesson and concepts and connection to today's lesson

Activity #1: Trailblazer – 20 minutes (Guided Practice)

2. Blazing a Trail and Comparing solutions
3. New Concepts: branching and Booleans

Activity #2: Bumper Turtles Challenge – 20 minutes (Discovery / Creative)

4. Introduce new StarLogo Nova commands
5. The Challenge

Wrap-up – 5 minutes (Reflection)

6. What does Bumper Turtles have to do with Modeling and Simulation?
7. What does Bumper Turtles have to do with complex adaptive systems?

Lesson Objectives

The student will:

- ✓ Learn that in complex adaptive systems one type of common interaction is that agents react to their environment (LO14)
- ✓ Create a model in which agents react to their environment (LO15)
- ✓ Trace a program's execution (LO16)
- ✓ Experiment with adding branching and selection to their agents' behavior (LO17)
- ✓ Learn computer science concepts of Booleans, logic, and conditionals (LO18)
- ✓ Practice Pair Programming and Iterative design, implement, test cycle (LO19)
- ✓ Compare solutions to a problem using number of steps or number of instructions used (LO20)

Teaching Guide

Materials, Resources and Preparation

For the Students

- Student Activity Sheets (Trailblazer and Bumper Turtles Challenge)
- Three colors of pens (black, red and blue)
- Pencils and erasers
- Computers
- Module 1 Lesson 3 CS Concepts reference
- Module 1 Lesson 3 Blocks reference

For the Teacher

- Computer and projector
- Slide presentation

Getting started - 5 min

1. Review of previous day's lesson and link to where we are going today.

- What commands enabled turtles to impact their environment?
  [pen down, delete everyone, clear terrain]
- What could an agent leaving trails represent in a real world scenario?
- Next we are going to learn how turtles can **react** to their environment.

Activity #1: Trailblazer - 20 min

Learning how to build a StarLogo Nova model requires students to learn how to translate an initial idea of what the turtles should do into specific instructions of how they should do it. In this activity, participants will specify directions to help others recreate the safe path through a tangled maze without ever seeing it. Later we will draw a direct analogy to the conditional instruction used in programming the same behavior.

2. Blaze a trail (see student activity worksheet for details)

- Pass out Student Activity #1 handout.

- Assign partners to work together.
- Have students start at the designated start arrow position and heading.
- Have students draw the path in order to pick up all the gold while avoiding the hazards.
- Instruct the students that it is important that they use a pencil and that they draw the path in the middle of the squares.
- Have students draw landmarks as necessary to "pick up" gold while avoiding the "hazards."
- Players should create colored landmarks on the paper telling the other player where to go according to the following rules:
  - **Take a step forward.**
  - **If you are standing on a RED square, then turn right by 90 degrees**
  - **If you are standing on a BLUE square, then turn left by 90 degrees**
  - **If you are standing on a BLACK square, then turn right by 180 degrees**

  The goal is to place enough landmarks so that the turtle will follow the entire path from beginning to end.
- When a student has completed laying down landmarks, have him/her trade with the partner.
- The partner's task is to follow the trail and make it to the exit, picking up all the gold and avoiding the hazards. If not, what went wrong?
- When all participants have finished, ask them to compare their results.
- Conclude by asking: How many people were able to accurately follow their partner's path? Were there any mistakes made? (Did some people forget certain color blocks or use the wrong colors?)
- What strategies resulted in the most successful path followings?
- Is there more than one right solution?
- How would you decide which solution is the best?

  Teaching Tip *This activity can be demonstrated on a large tiled floor using cans and objects as landmarks.*

3. New concepts: branching and Booleans

- Introduce a new CS construct called branching.
- Introduce the commands IF/THEN and IF/THEN/ELSE.
- Introduce Booleans as statements that evaluate to TRUE or FALSE.
- Demonstrate how Booleans and conditionals work together to enable branching.
- Have students come up with example IF/THEN statements from everyday life.

BRANCHING WITH CONDITIONALS

In addition to simple stacks of commands as we did in the Drawing a Flower and Painting Turtles projects, we can instruct a computer to execute different stacks of commands depending on the circumstance or condition. For example, we might want to write different behaviors for a coyote agent based on a condition. If it runs into a rabbit then chase the rabbit BUT if it runs into a mountain lion, run away from the mountain lion.

To do this, we use a statement called a CONDITIONAL statement. It looks like this:

IF (something is true) THEN (do this)

So using our example, our conditional statements would look like this:

IF (Rabbit nearby) THEN (Chase the Rabbit)
IF (Mountain Lion nearby) THEN (Run away from Mountain Lion)

The (something is true) part is called a BOOLEAN expression or an expression that evaluates to TRUE or FALSE.

(Rabbit nearby) could evaluate to TRUE or FALSE.
(Mountain Lion nearby) could evaluate to TRUE or FALSE.

CONDITIONAL STATEMENTS and BOOLEAN expressions work together. The THEN part of the CONDITIONAL STATEMENT only gets executed if its BOOLEAN expression evaluates to TRUE.

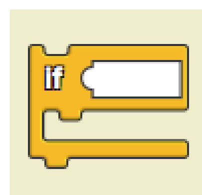
Another type of CONDITIONAL STATEMENT is the IF/THEN/ELSE STATEMENT. With this statement the THEN part gets executed if the BOOLEAN expression evaluates to TRUE. The ELSE part gets executed if the BOOLEAN expression evaluates to FALSE.

Think about when this might come in handy: What if you were the coyote and had a rabbit and a mountain lion in front of you at the same time. How would you use the IF/THEN/ELSE command to make sure you take the right action?

Activity #2: Bumper Turtles - 20 min

4. Introduce new commands

- Show new StarLogo Nova command blocks (slide presentation).
- Review commands used during last class.



5. The Bumper Turtle Challenge

- Open up the Bumper Turtles starter program.
- Remix and change title.
- Don't forget to put both partners' names in the project title.
- Create four turtles.
- Then have the turtles react to the colored squares.
- Trace through code and talk about the order in which commands are executed.
- Optional: Discuss the difference between if/then statements in a row vs. nested if/then/else statements. Act out a turtle's actions following the two different command structures.
- Does execution order matter?

Wrap-up - 5 min

6. What does Bumper Turtles have to do with Modeling and Simulation?

- What could these trails and bumpers represent?
- Review the new terms used in this activity: conditional, Boolean, branching.

7. **What does Bumper Turtles have to do with complex systems?**

Assessment Questions

- Is Bumper Turtles a model of a complex adaptive system? Why or why not?
- How would you assess the “best” solution? [time taken, steps taken, fewest instructions]
- What is the difference between if/then blocks in a row vs. nested if/then/else statements?
- In your own worlds, how can the if/then logic block be used in a computer model to give an agent a behavior?
- Is the path taken by turtles repeatable? If I run the program again, will it produce the same drawing? Why or why not?

Standards Addressed

NRC Scientific and Engineering Practice Standards

Practice 1: Asking questions and defining problems
1A: Ask questions that arise from careful observation of phenomena, models, or unexpected results.
1D: Ask questions to clarify and/or refine a model, an explanation, or an engineering problem.

Practice 2: Developing and using models
2A: Evaluate limitations of a model for a proposed object or tool.
2B: Develop or modify a model—based on evidence – to match what happens if a variable or component of a system is changed.
2E: Develop and/or use a model to predict and/or describe phenomena.
2G: Develop and/or use a model to generate data to test ideas about phenomena in natural or designed systems, including those representing inputs and outputs, and those at unobservable scales.

Practice 4: Analyzing and interpreting data
4B: Use graphical displays (e.g., maps, charts, graphs, and/or tables) of large data sets to identify temporal and spatial relationships.
4E: Apply concepts of statistics and probability (including mean, median, mode, and variability) to analyze and characterize data, using digital tools when feasible.

Practice 5: Using mathematics and computational thinking
5A: Use digital tools (e.g., computers) to analyze data sets for patterns and trends.
5C: Create algorithms (a series of ordered steps) to solve a problem.

NRC Crosscutting Concepts

1. Patterns:
1C: Patterns can be used to identify cause and effect relationships.
1D: Graphs, charts, and images can be used to identify patterns in data.

2. Cause and Effect:
2A: Relationships can be classified as causal or correlational, and correlation does not necessarily imply causation.
2B: Cause and effect relationships may be used to predict phenomena in natural or designed systems.

4. Systems and Systems models
4B: Models can be used to represent systems and their interactions—such as inputs, processes and outputs—and energy, matter, and information flows within systems.
4C: Models are limited in that they only represent certain aspects of the system under study.

CSTA K-12 Computer Science Standards

CT	Algorithms	1:6-2	Develop a simple understanding of algorithms using computer-free exercises.
CT	Algorithms	2-4	Evaluate ways that different algorithms may be used to solve the same problem.
CT	Algorithms	3A-3	Explain how sequence, selection, iteration and recursion are the building blocks of algorithms.
CT	Data representation	2-8	Use visual representation of problem state, structure and data.
CT	Modeling & simulation	2-9	Interact with content-specific models and simulations to support learning and research.
CPP	Data collection & analysis	2-9	Collect and analyze data that are output from multiple runs of a computer program.
CPP	Programming	2-5	Implement a problem solution in a programming environment using looping behavior, conditional statements, logic, expressions, variables and
CPP	Programming	3A-3	Use various debugging and testing methods to ensure program correctness.

Responsiveness to Varied Student Learning Needs

In Project GUTS, we integrate teaching strategies found to be effective with learners with various backgrounds and characteristics such as economically disadvantaged students (EDS), students from groups that are underrepresented in STEM (URG), students with disabilities (DIS), English Language learners (ELL), girls and young women (FEM), students in alternative education (ALT), and gifted and talented students (GAT).

In each lesson we describe the accommodations and differentiation strategies that are integrated in the activities to support a wide range of learners.

Module 1 Lesson 3: Trailblazer Activity and Bumper Turtle Challenge

(URG) *We involve students in working in small groups for the Trailblazer table-top game/activity, a strategy that uses a multi-modal experience to increase student engagement.*

(DIS) *Across both the Trailblazer and Bumper Turtles Challenge, we provide multiple means of action, expression, representation and engagement by introducing a game and a design and build programming activity that build on the same underlying concepts. These are all principles of Universal Design for Learning.*

(URG)(DIS) *We use technology to present information in multiple modes of representations. In the StarLogo Nova modeling and simulation environment students can present information as code blocks, text, visual imagery, and as data in tables and graphs.*

(FEM) *We recommend careful planning of partners for pair programming in the Bumper Turtle Challenge activity, a practice that encourages participation for the girls in science.*

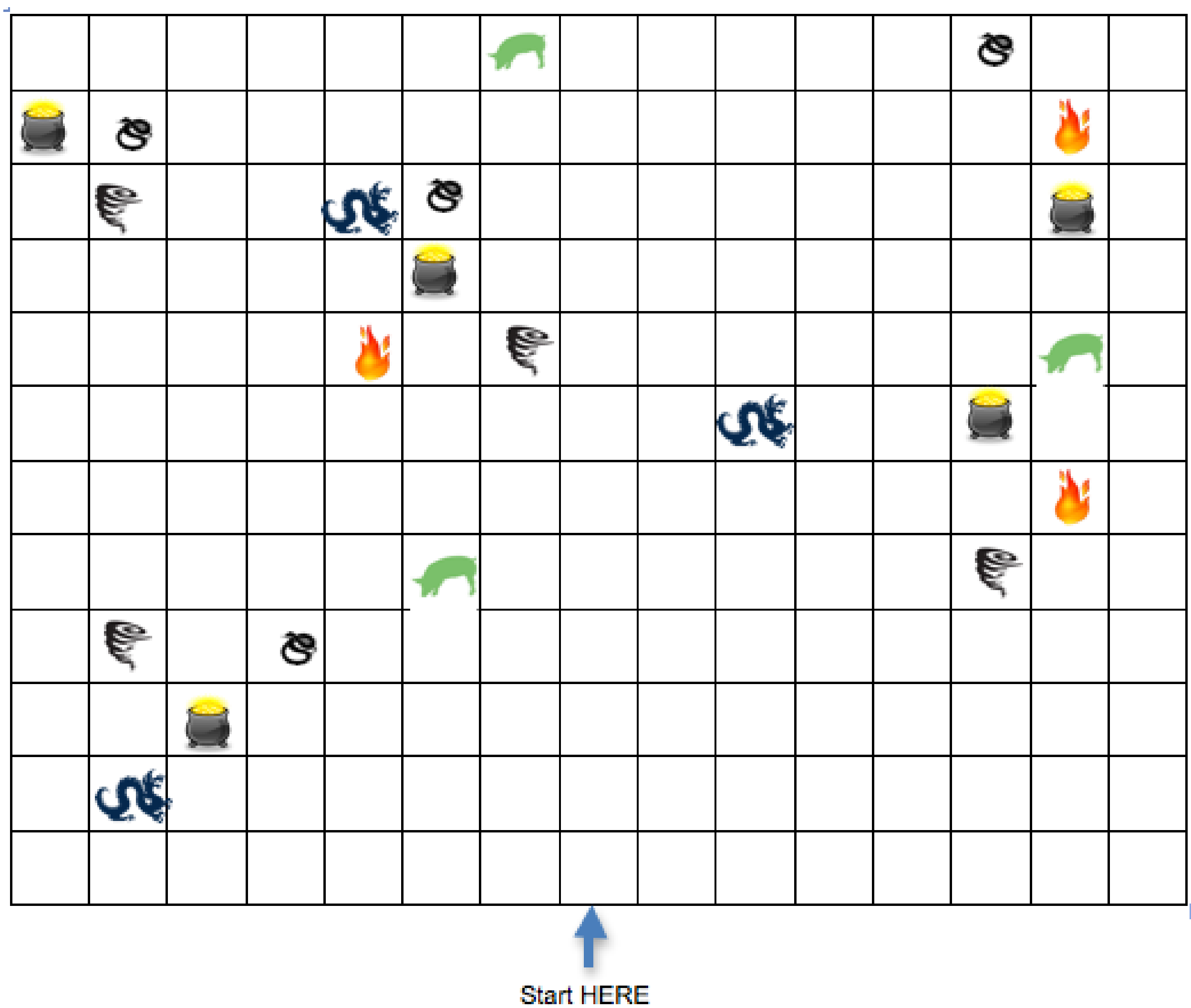
Student Activity #1 Guide

TRAILBLAZER Instructions and playing board

How to play: On the map, START at the designated position and heading in the direction of the arrow. Using a pencil, draw the path in the CENTER of squares. Pick up ALL the gold while avoiding the hazards, ending at START. Color the squares as necessary according to the following rules:

- **Take a step forward.**
- **If you are standing on a RED square, then turn right by 90 degrees**
- **If you are standing on a BLUE square, then turn left by 90 degrees**
- **If you are standing on a BLACK square, then turn right by 180 degrees**

Trade your map with a partner and figure out if following the landmarks leads you along the path to collect all of the gold while avoiding the hazards.



Student Activity #2 Guide

The Bumper Turtle Challenge
Start with the model “Bumper Turtles starter”. (Teacher provides the link)
This starter model already has a button called “Paint Landmarks” and some coding associated with it.
Click on the “Paint Landmarks” push button and see the program execute the code provided.

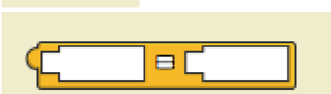
Your challenge is to make the turtles react to the landmarks created by the “Paint Landmarks” procedure, following the rules laid out in the Trailblazer activity.

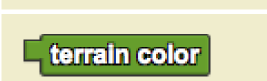
Note: DO NOT make changes to the “Paint Landmarks” procedure.

<u>Guidelines:</u>

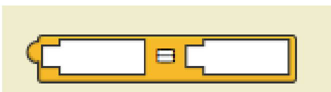
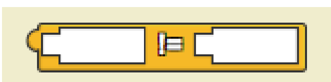
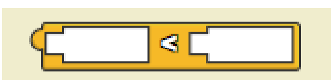
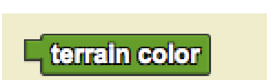
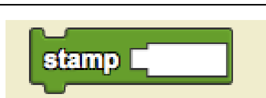
Remix the “Bumper Turtles starter” model
Don’t forget to put both partners’ names in the project title.
Create 4 turtles (Refer to Lesson 2)
Have the turtles check the terrain color they are standing on
Instruct your turtles to react to the landmarks according to the rules in the Trailblazer activity
When you are done, save and share your project.

The new command blocks to be used in addition to the blocks you used in Lesson 1 and 2 are:
(Refer to the Blocks Reference Guide provided)





StarLogo Nova Blocks introduced in Module 1 Lesson 3

	Carries out the specific procedures if the condition is true. This conditional block ONLY runs the main block IF the condition after the "if" is met.
	Carries out the specific procedures if the condition is true. Otherwise (if the condition is false) carry out procedures in the second block.
	Determines whether the two inputs are equivalent and returns true or false. Usually used after an "if" in a conditional block.
	Determines whether the two inputs are unequal and returns true or false. Usually used after an "if" in a conditional block.
	Determines if the first input is less than the second input and returns true or false. Usually used after an "if" in a conditional block.
	Determines if the first input is greater than the second input and returns true or false. Usually used after an "if" in a conditional block.
	Determines if the first input is less than or equal to the second input and returns true or false. Usually used after an "if" in a conditional block.
	Determines if the second input is greater than or equal to the second input and returns true or false. Usually used after an "if" in a conditional block.
	Returns an agent's trait.
	Returns the color of the terrain the agent is currently on.
	Colors a grid square on the terrain in the agent's current position.
	Colors a circle on the terrain in the agent's current position.

1

Lesson 4

Probability with Dice and Data and Colliding Turtles

50 minutes (1 day)

Lesson Overview

In this lesson students will participate in two activities. In the first activity, students will learn about probability, how it is implemented in StarLogo Nova, and use probability to implement chance behavior in agent movement. In the second activity, students will remix their Bumper Turtles project into a Colliding Turtles project in which turtles react to other turtles upon collision. The wrap-up discussion covers what probability is used for in computer modeling and simulation.

Teaching Summary

Getting started – 5 minutes (Review)

1. Review of the previous day's lesson and concepts and connection to today's lesson.

Activity #1: Probability with Dice and Data – 20 minutes (Guided Practice)

2. Chances Are – one die
3. Wiggle Walk – two dice

Activity #2: Colliding Turtles – 20 minutes (Discovery / Creative)

4. New Concepts: Agent-agent interactions
5. Add a behavior that takes place upon collision

Wrap-up – 5 minutes (Reflection)

6. What could turtles colliding represent in the real world?
7. How does probability play a role in modeling and simulation?

Lesson Objectives

The student will:

- ✓ Learn that in complex adaptive systems one type of common interaction is of agents interacting with other agents (LO21)
- ✓ Create a simple program with agents interacting with other agents (LO22)
- ✓ Use the random function to implement probabilistic outcomes / behaviors (LO23)
- ✓ Learn computer science concepts of random numbers (LO24)
- ✓ Understand the concept of collisions (LO25)
- ✓ Learn mathematical concepts of probability and distributions (LO26)

Teaching Guide

Materials, Resources and Preparation

For the Students

- Computers
- StarLogo Nova saved model from Bumper Turtles or Colliding Turtles Starter model
- Two dice and a paper cup for each group of 2-3 students

For the Teacher

- Computer and projector
- Slide presentation with simple commands
- Teacher sample code
- Graph paper or Excel spreadsheet projected on a screen

Getting started - 5 min

1. Review of previous day's lesson and link to where we are going today.

- What commands enabled agents to react to their environments?
  [See slide presentation: if/then and if/then/else commands; example of using and if/then to react to the environment.]

Activity #1: Dice and Data - 20 min

2. Chances Are

- Introduce the topic: Probability plays a large role in models of complex adaptive systems. We've programmed our agents to mimic the movement of creatures in the real world. There are also chance events that occur upon when agents interact, such as the passing of a contagion from one person to the other.
- Form small groups of 2-3 students.
- Hand out two dice and one cup per small group. Hand out activity sheet.
- Instruct the students to roll 1 die 50 times while a partner marks down the results for all 50 rolls using the chart provided with the possible rolls (1 through 6).

Roll	Area for recording marks	
1		Sum:
2		Sum:
3		Sum:
4		Sum:
5		Sum:
6		Sum:

- Ask each group "What was the distribution of the results? Was one number more common? Why might that be?"
- If time allows, pool all of the data from the class and see what the sums are across groups. Graph this data as a bar chart. Ask what the graph look like.
- As the groups report their data, put the class information together as a bar graph.
- Discussion: What do the numbers look like now? What's different about this data compared to the data of just one group?
- Note: the distribution should be even, theoretically – the chances of rolling each of the sides is equal.

3. Wiggle Walk

- Next we are going to roll two die. Do you predict we will have the same flat distribution?
- This time, ask students to roll 2 dice simultaneously and collect the data from 50 rolls.
- Record the outcome on the activity sheet with a mark next to the sum of the dice rolled. After 50 rolls, sum up each row and record the sum in the column on the right.

Roll	Area for recording marks	
2		Sum:
3		Sum:
4		Sum:
5		Sum:
6		Sum:
7		Sum:
8		Sum:
9		Sum:
10		Sum:
11		Sum:
12		Sum:

- What number gets rolled the most often? Why?
- Ask "How many ways are there to make 2?", "How many ways can you roll a 12?" What about other configurations?

- Given this chart, what should be the most commonly rolled number? Why?
- If time allows, graph this data as a bar chart; what shape does the graph look like? (Draw it.)
- To help students understand this phenomenon, describe the underlying statistics:
  We know that with one die we have equal probability of rolling a 1, 2, 3, 4, 5, or 6.
- Have students fill in the chart in their activity sheet that shows the outcome of rolling two dice. The numbers down the column represent the number rolled on one die; the numbers rolled across the top represent the numbers rolled on the other die. Add them together just like you would in an addition table.

Result of the roll of two dice

+	1	2	3	4	5	6
1	2					
2						
3						
4						
5						
6						

Notice that the most common number in the matrix is a 7 [because there are many different combinations that add up to 7. That's why it is called "Lucky Seven".]

- Next relate this to our wiggle walk. Fill in the result of a left turn of random 6 followed by a right turn of random 6. (Note that instead of addition, we use subtraction in this case.)

Right turn

Left turn

--	1	2	3	4	5	6
1	0					
2						
3						
4						
5						
6						

Notice the most common number in the matrix is now a 0. A 0 represents no change in heading.

So whereas it is common to stay going straight, there is a smaller chance of turning at a large angle.

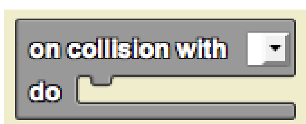
- In the wiggle walk we use this left turn random and right turn random to make an agent walk mostly in a straight line but sometimes wander a bit off.
- Using the PowerPoint presentation, walk through the execution of the wiggle code.

Activity #2: Colliding Turtles - 20 min

In this activity students will remix their Bumper Turtles project OR use the Colliding Turtles Starter model to create a model with colliding turtles.

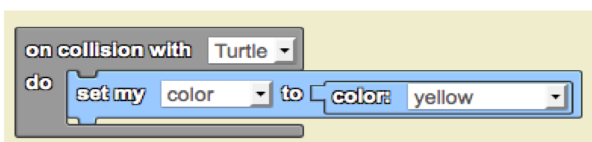
4. Agent-agent interactions

- Introduce collisions: In this activity we are going to implement agents interacting with other agents upon collision. Collisions occur when two agents bump into one another. (They do not need to be centered on the same patch, just touching.) This is different from bumper turtles – In bumper turtles, agents were responding to colored patches in their environment, not other agents.
- We will use a collision block that looks like this. Use the pull down arrow to select what type of object to collide with. Then put the commands that should run or execute when the collision occurs in the area below the notch.

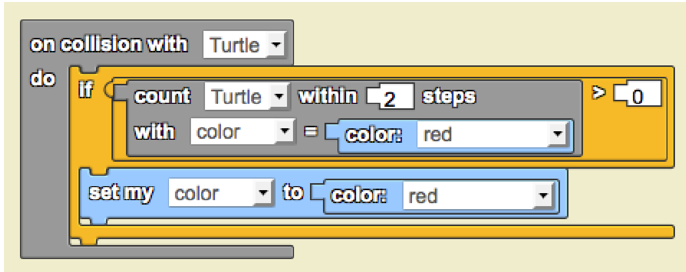


5. Adding a behavior that takes place upon collision with another turtle

- Here's an example of programming a reaction to a collision. In this example, when the agent collides with another agent, the first agent's color turns yellow.



- Below is another example showing the use of a collision block with a conditional instruction. The set color command only executes if the condition is true. The condition asks whether or not there is a red turtle within 2 steps of the current position. If true then the first agent becomes red; if not, nothing happens.



Teaching Tip *If time allows, have students show their projects to each other.*

Wrap-up – 5 min

6. **What does Colliding Turtles have to do with Modeling and Simulation?**
   What could these collisions represent?
7. **How do we use probability in models?**

Assessment Questions

- Which characteristics of complex adaptive systems can you identify in Colliding Turtles?
- Give an example of how agents interacting with other agents may be used to represent something in the real world.
- How does using probability impact the outcome when running simulations?
- What variables were used in Colliding Turtles?
- What is the difference between “right turn random 90” and “right turn random 90 left turn random 90”?

Standards Addressed

NRC Scientific and Engineering Practice Standards

Practice 1: Asking questions and defining problems
1A: Ask questions that arise from careful observation of phenomena, models, or unexpected results.
1B: Ask question to identify and/or clarify evidence and/or the premise(s) of an argument.
1E: Ask questions that require sufficient and appropriate empirical evidence to answer.
1F: Ask questions that can be investigated within the scope of the classroom, outdoor environment, and based on observations and scientific principles.

Practice 2: Developing and using models
2C: Use and/or develop a model of simple systems with uncertain and less predictable factors.
2E: Develop and/or use a model to predict and/or describe phenomena.

Practice 3: Planning and carrying out investigations
3D: Collect data to produce data to serve as the basis for evidence to answer scientific questions or test design solutions under a range of conditions.

Practice 4: Analyzing and interpreting data
4A: Construct, analyze, and/or interpret graphical displays of data and/or large data sets to identify linear and nonlinear relationships.
4B: Use graphical displays (e.g., maps, charts, graphs, and/or tables) of large data sets to identify temporal and spatial relationships.

4E: Apply concepts of statistics and probability (including mean, median, mode, and variability) to analyze and characterize data, using digital tools when feasible.
4F: Consider limitations of data analysis (e.g., measurement error), and/or seek to improve precision and accuracy of data with better technological tools and methods (e.g., multiple trials).

Practice 5: Using mathematics and computational thinking
5B: Use mathematical representations to describe and/or support scientific conclusions and design solutions.
5C: Create algorithms (a series of ordered steps) to solve a problem.
5D: Apply mathematical concepts and/or processes (e.g., ratio, rate, percent, basic operations, simple algebra) to scientific and engineering questions and problems.

Practice 6: Constructing explanations and designing solutions
6D: Apply scientific ideas, principles, and/or evidence to construct, revise and/or use an explanation for real-world phenomena, examples, or events.

NRC Crosscutting Concepts

1. Patterns:
1D: Graphs, charts, and images can be used to identify patterns in data.

2. Cause and Effect:
2C: Phenomena may have more than one cause, and some cause and effect relationships in systems can only be described using probability.

4. Systems and Systems models
4B: Models can be used to represent systems and their interactions—such as inputs, processes and outputs—and energy, matter, and information flows within systems.

CSTA K-12 Computer Science Standards

CT	Algorithms	1:6-2	Develop a simple understanding of algorithms using computer-free exercises.
CT	Data representation	2-8	Use visual representation of problem state, structure and data.
CT	Data representation	3A-12	Describe how mathematical and statistical functions, sets, and logic are used in computation.
CT	Modeling & simulation	2-9	Interact with content-specific models and simulations to support learning and research.
CT	Modeling & simulation	3A-8	Use modeling and simulation to represent and understand natural phenomena.
CPP	Programming	2-5	Implement a problem solution in a programming environment using looping behavior, conditional statements, logic, expressions, variables and

Responsiveness to Varied Student Learning Needs

In Project GUTS, we integrate teaching strategies found to be effective with learners with various backgrounds and characteristics, such as economically disadvantaged students (EDS), students from groups that are underrepresented in STEM (URG), students with disabilities (DIS), English Language learners (ELL), girls and young women (FEM), students in alternative education (ALT), and gifted and talented students (GAT).

In each lesson we describe the accommodations and differentiation strategies that are integrated in the activities to support a wide range of learners.

Module 1 Lesson 4 Activity #1: Dice and Data (Chances Are and Wiggle Walk)

(URG) *In Dice and Data, we use a class modeling activity that involved student movement and interaction, a strategy that uses a multi-modal experience to increase student engagement.*

(DIS) *Across both the Chances Are and Wiggle Walk activities, we provide multiple means of action, expression, representation and engagement by introducing a game and a design-and-build programming activity that build on the same underlying concepts. These are all principles of Universal Design for Learning.*

Module 1 Lesson 4 Activity #2: Colliding Turtles Challenge

(URG)(DIS) *We use technology to present information in multiple modes of representations. In the StarLogo Nova modeling and simulation environment, students can present information as code blocks, text, visual imagery, and as data in tables and graphs.*

(FEM) *We recommend careful planning of partners for pair programing in the Colliding Turtle Challenge activity, a practice that encourages participation for the girls in science.*

Student Activity #1 Guide:

Name ____________________

Chances Are and Wiggle Walk

Introduction

Probability plays a large role in models of complex adaptive systems. We'll be programming our agents to mimic the movement of creatures in the real world. There are also chance events that occur upon when agents interact such as the passing of a contagion from one person to the other.

PART 1: " Chances Are"

Directions

- In your group, roll 1 die 50 times in a cup while the partner marks down the results for all 50 rolls using the chart provided with the possible rolls (1 through 6).

For each roll of the die, record the outcome with a mark below next to the number rolled. After 50 rolls, sum up each row and record the sum in the column on the right.

1		Sum:
2		Sum:
3		Sum:
4		Sum:
5		Sum:
6		Sum:

Discussion:

What was the distribution of the results?

Was one number more common?

Why might that be?

Extension:

If time allows, pool all of the data from the class and see what the sums are across groups.

If you graphed this data as a bar chart, what would the graph look like? (draw it below.)

PART 2: Wiggle Walk

Next imagine that you were going to roll two dice instead of one die. Do you think we will still get a flat distribution (in which the probability of rolling each outcome would be the same)?

Directions

- For each roll of two dice in the cup, record the outcome with a mark below next to the number rolled. After 50 rolls, sum up each row and record the sum in the column on the right.

1		Sum:
2		Sum:
3		Sum:
4		Sum:
5		Sum:
6		Sum:
7		Sum:
8		Sum:
9		Sum:
10		Sum:
11		Sum:
12		Sum:

Discussion:
Was one number rolled more often than the others?
If so, what was the most frequent outcome when rolling two dice?
Why might that be?

Extension:
If time allows, graph this data as a bar chart, what shape does the graph look like? (draw it below.)

To understand this phenomenon, let's look at the underlying statistics:
We know that with one die we have equal probability of rolling a 1, 2, 3, 4, 5, or a 6.

Fill in the chart below that shows the outcome of rolling two dice. The numbers down the column represent the number rolled on one die, the numbers rolled across the top represent the numbers rolled on the other die. Add them together just like you would in an addition table.

Result of the roll of two dice

+	1	2	3	4	5	6
1	2					
2						
3						
4						
5						
6						

Discussion:
How many ways are there to make 2?
How many ways can you roll a 12?
What about other numbers?
What is the most often seen number in the matrix?
How many times is it seen?

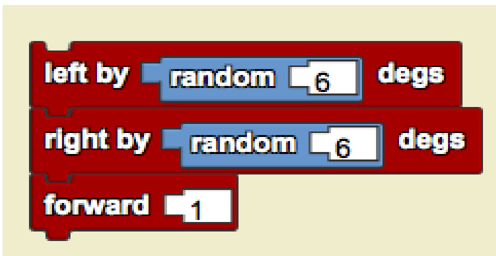
Next, fill in the result of a left turn of random 6 followed by a right turn of random 6. (note that instead of addition, we use subtraction in this case.

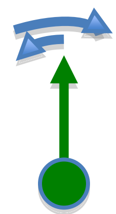
Right turn (columns); *Left turn* (rows)

--	1	2	3	4	5	6
1	0					
2						
3						
4						
5						
6						

Discussion:
How many ways are there to make 0?
How many ways can you roll a +6?
How many ways can you roll a -6?
What about other numbers?
What is the most often seen number in the matrix?
How many times is it seen?

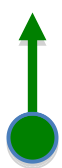
Finally, consider rolling two dice where one represents the degrees to turn to the left and the other die represents the degrees to turn to the right. That's similar to using a left turn random 6 followed by a right turn random 6, then taking a step forward.



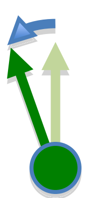


Turtle seen from above with initial heading.

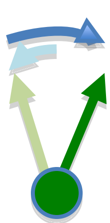
What is actually going on?



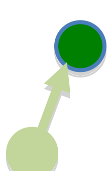
1. Agent has an initial heading.



2. Agent turns left by some random number of degrees between 1 and 6. In this case, it is 2 degrees. (Image not to scale)



3. Agent turns right by some random number of degrees between 0 and 5. In this case, it is 5 degrees, so the final heading is 3 degrees to the right of the original heading. (Image not to scale)

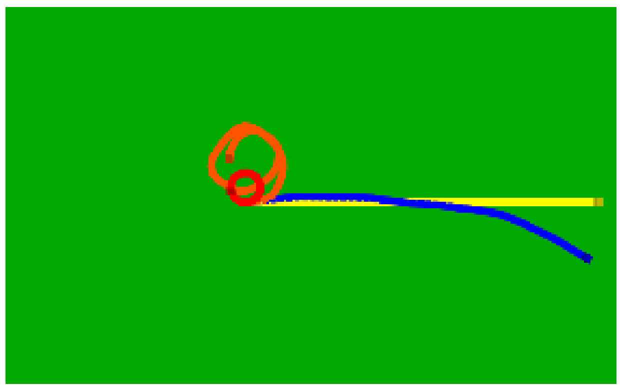


4. Agent takes one step forward at new heading. (Image not to scale)

In StarLogo Nova:

In StarLogo Nova, we use the random command to simulate the roll of a die. Random 6 would give me the result of rolling a 60-sided die with numbers 1 through 6 on the sides. Random functions can also be used within other commands to implement random behavior.

Self-test:

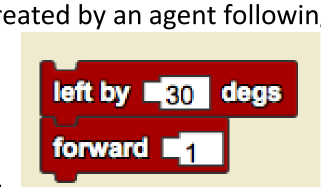


Match the command blocks a, b, c, and d with the path created by an agent following those blocks.

a.

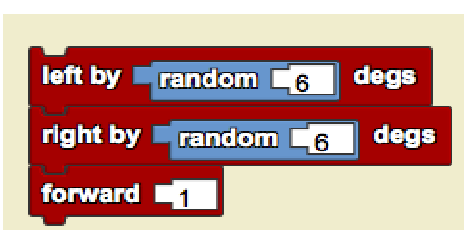
color: _________

c.



color: _________

b.



color: _________

d.

color: _________

Student Activity #2 Guide:

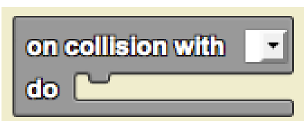
Name ____________________

Collisions

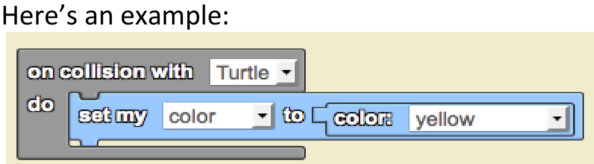
Introduction

In this activity we are going to implement agents interacting with other agents upon colliding. Collisions occur when two agents bump into one another. (They do not need to be centered on the same patch, just touching.) This is different from Bumper Turtles – In Bumper Turtles, agents were responding to colored patches in their environment, not other agents.

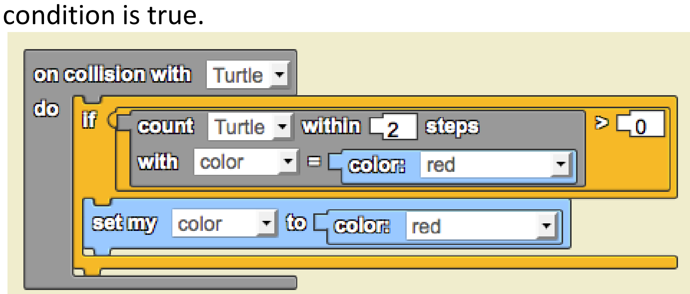
We will use a collision block that looks like this. Use the pull down arrow to select what to type of object to collide with. Then put the commands that should run or execute when the collision occurs in the area below the notch.



Here's an example:



Here's another example with a conditional instruction. The set color command only executes if the condition is true.



Your challenge is to make the turtles react to another agent upon collision.

<u>Guidelines:</u>

Remix the "Colliding Turtles starter" model

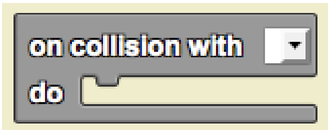
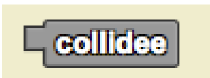
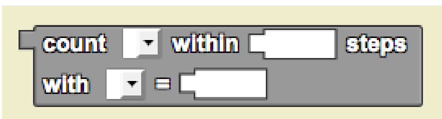
Don't forget to put both partners' names in the project title.

Create 50 turtles (Refer to Lesson 2)

Have the turtles change a trait after colliding with another turtle.

When you are done, save and share your project.

StarLogo Nova Blocks introduced in Module 1 Lesson 4

	Specifies a series of actions to be executed after a collision between two breeds. If the two breeds are the same, each agent in the collision must perform the same action after the collision. If the two breeds are different, each agent in the collision can be given different commands.
	Specifies the other agent in the collision.
	Counts the number of agents of a specific breed in a given radius with a specific trait.

1

Lesson 5

Modeling the Spread of Disease

50 minutes (1 day)

Lesson Overview

In this lesson students will convert their Colliding Turtles model into a simple Epidemic model by adding slider widgets and recovery. The Contagion model represents a very simplified version of an epidemic or spread of a disease. Two variables will be created: transmission rate and recovery rate. Students will later use this model to run experiments to determine if disease will spread throughout a virtual population in different scenarios.

Teaching Summary

Getting started – 5 minutes (Review)

1. Review of the previous day's lesson and concepts and connection to today's lesson

Activity #1: Introduction of Epidemiology – 25 minutes (New Learning)

2. Epidemiology and the spread of disease

Activity #2: Modeling the Spread of Disease – 25 minutes (Guided Practice and Discovery)

3. New concepts and commands
4. Altering Colliding Turtles to an Epidemic Model and adding widgets
5. Customizing your model [adding in recovery]
6. Test your model

Wrap-up – 5 minutes (Reflection)

7. What does this model tell you? Can it be trusted?
8. What other things move through a population like a disease? [Rumors, ideas, etc.]

Lesson Objectives

The student will:

- ✓ Learn about epidemiology and how it can be modeled as a complex system (LO27)
- ✓ Create a simple model in which agents pass a contagion from one to another (LO28)
- ✓ Learn the CS concepts of procedures and variables (LO29)
- ✓ Create and use sliders to set variables and initial conditions (LO30)
- ✓ Create procedures and call procedures (LO31)
- ✓ Use the random function to simulate probabilistic outcomes (LO32)

Teaching Guide

Materials, Resources and Preparation

For the Students

- Computers
- StarLogo Nova model of Colliding Turtles

For the Teacher

- Computer and projector
- Slide presentation with new commands and concepts

Getting started - 5 min

1. Review of previous day's lesson and link to where we are going today

- What does the Colliding Turtles model you built last time remind you of? What could the blue and red represent?
- Thinking about the wiggle walk, what would be the result if the left and right turn were given a different range for the random function?

Activity #1: Introduction to Epidemiology - 10 min

2. Epidemiology and spread of disease

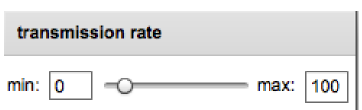
- "Scientists who study epidemics are called epidemiologists. Epidemiologists study the spread of disease such as the swine flu. Epidemics are hard to study because they are made up of many connected and inter-related parts interacting in different ways. They can be hard to predict and sometimes the way a disease spreads creates patterns that cannot be predicted even if you know something about the people involved. These are characteristics of what we call "complex systems." (Many interacting parts, hard to predict, patterns emerge that are not predictable even if you know about individuals)"
- "One way epidemiologists try to understand epidemics is to run computer simulations of how diseases spread and test different measures that could be taken to stop an epidemic."
- Introduce the example of MRSA (Methicillin-resistant Staphylococcus aureus) using the slide presentation. Discuss how agents go from one state to another such as from healthy to infected.

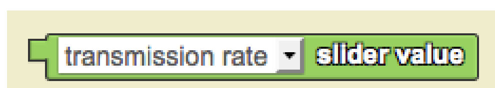
Activity #2: Modeling the Spread of Disease - 30 min

3. **New CS Concept and Commands** (see slides on Variables and Procedures)

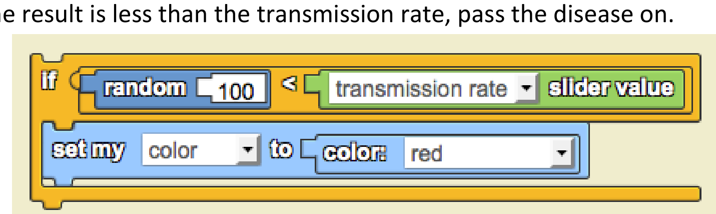
4. **Altering Colliding Turtles to create an Epidemic Model**
(See student activity handout.)

- What we have already is a model in which there are agents and collisions between agents have an outcome.
- Now we are going to alter our Colliding Turtles to make it into a model of an epidemic.
- To do this, we are going to create two variables: transmission rate and recovery rate. Transmission rate is the probability that an agent will get sick upon colliding with a sick agent. Recovery rate is the probability that a sick agent gets healthy at any turn. Ask if these are independent or dependent variables.
- We will learn how to create widgets called "sliders." These sliders are variables that hold values. These values are inputs into our model.
- In the colliding turtles model, turtles always reacted to a collision. In our epidemic model, we will incorporate a transmission rate so only some of the time a disease is passed from one agent to another.
- Demonstrate how to create a widget or show students using the slide presentation. [Edit Widget, New Widget, Create Widget by specifying name and type.]
- Change the max value of the "Transmission rate" to 100 by typing in 100 next to max and hitting the return key.





- Click on "Edit Widgets" again to get out into editing mode and back into play mode.
- We now have a way to set the transmission rate through the user interface with a slider.
- We can get the value of transmission rate in code using the slider value command.
- How do we "pass the disease to the healthy agent 40% of the time after a collision?"
  (Hint: remember the dice rolls.)
- We can use the random function to roll a 100-sided die.
- If the result is less than the transmission rate, pass the disease on.

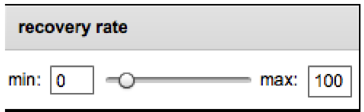


- Have students save and test their models.

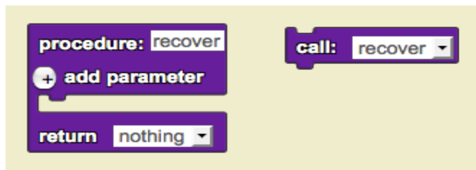
5. Customizing your model. (Add in recovery.)

As it currently exists, every agent who collides with a sick agent gets sick and no one recovers. Is that realistic?

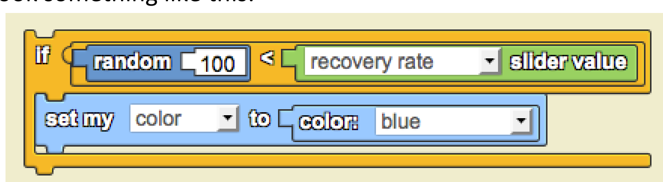
- Introduce new command blocks: the “procedure” block and “call” procedure block.
- Creating and calling the recovery procedure.
- Next, ask students to add the recovery rate slider in the same way they added the transmission rate slider.



- Then guide them in creating a recovery procedure. At each step, a sick person has a chance of recovering so we will need recovery to be called when the forever button is toggled.



- Note: the “call recovery block” needs to be within the “when forever toggled” loop.
- Within the recovery procedure, students will need to use the random function to mimic rolling the 100-sided die, as we did in the transmission case.
- It could look something like this:



6. Testing your model

- Have the students save and test their models.
- Try changing the recovery rate. Did you see any new outcomes or patterns?

Wrap-up – 5 min

7. What does your model tell you?

- What assumptions are made in your model? Was there any key factor that was left out that might be important in the real world?

8. What other things move through a population like a disease?

Assessment Questions

- Is the epidemic model a model of a complex adaptive system? Why or why not?
- What variables were used in your epidemic model?
- How would you test whether the spread of disease is impacted by an agent’s frequency of travel?
- How many times do I need to run each experiment at each setting?

Standards Addressed

NRC Scientific and Engineering Practice Standards

Practice 1: Asking questions and defining problems
1A: Ask questions that arise from careful observation of phenomena, models, or unexpected results.
1C: Ask questions to determine relationships between independent and dependent variables and relationships in models.

Practice 2: Developing and using models
2A: Evaluate limitations of a model for a proposed object or tool.
2B: Develop or modify a model—based on evidence – to match what happens if a variable or component of a system is changed.
2C: Use and/or develop a model of simple systems with uncertain and less predictable factors.
2D: Develop and/or revise a model to show the relationships among variables, including those that are not observable but predict observable phenomena.
2E: Develop and/or use a model to predict and/or describe phenomena.
2F: Develop a model to describe unobservable mechanisms.
2G: Develop and/or use a model to generate data to test ideas about phenomena in natural or designed systems, including those representing inputs and outputs, and those at unobservable scales.

Practice 3: Planning and carrying out investigations
3D: Collect data to produce data to serve as the basis for evidence to answer scientific questions or test design solutions under a range of conditions.
3E: Collect data about the performance of a proposed object, tool, process or system under a range of conditions.

Practice 4: Analyzing and interpreting data
4A: Construct, analyze, and/or interpret graphical displays of data and/or large data sets to identify linear and nonlinear relationships.
4B: Use graphical displays (e.g., maps, charts, graphs, and/or tables) of large data sets to identify temporal and spatial relationships.
4D: Analyze and interpret data to provide evidence for phenomena.

Practice 5: Using mathematics and computational thinking
5A: Use digital tools (e.g., computers) to analyze very large data sets for patterns and trends.
5B: Use mathematical representations to describe and/or support scientific conclusions and design solutions.
5C: Create algorithms (a series of ordered steps) to solve a problem.
5D: Apply mathematical concepts and/or processes (e.g., ratio, rate, percent, basic operations, simple algebra) to scientific and engineering questions and problems.

Practice 6: Constructing explanations and designing solutions
6A: Construct an explanation that includes qualitative or quantitative relationships between variables that predict(s) and/or describe(s) phenomena.
6B: Construct an explanation using models or representations.
6D: Apply scientific ideas, principles, and/or evidence to construct, revise and/or use an explanation for real-world phenomena, examples, or events.
6F: Apply scientific ideas or principles to design, construct, and/or test a design of an object, tool, process or system.
6G: Undertake a design project, engaging in the design cycle, to construct and/or implement a solution that meets specific design criteria and constraints.

NRC Crosscutting Concepts

1. Patterns:
1B: Patterns in rates of change and other numerical relationships can provide information about natural and human designed systems.
1C: Patterns can be used to identify cause and effect relationships.
1D: Graphs, charts, and images can be used to identify patterns in data.

2. Cause and Effect:

2A: Relationships can be classified as causal or correlational, and correlation does not necessarily imply causation.
2B: Cause and effect relationships may be used to predict phenomena in natural or designed systems.
2C: Phenomena may have more than one cause, and some cause and effect relationships in systems can only be described using probability.

3. Scale, Proportion, and Quantity
3A: Time, space, and energy phenomena can be observed at various scales using models to study systems that are too large or too small.
3B: The observed function of natural and designed systems may change with scale.
3D: Scientific relationships can be represented through the use of algebraic expressions and equations.
3E: Phenomena that can be observed at one scale may not be observable at another scale.

4. Systems and Systems models
4A: Systems may interact with other systems; they may have sub-systems and be a part of larger complex systems.
4B: Models can be used to represent systems and their interactions—such as inputs, processes and outputs—and energy, matter, and information flows within systems.
4C: Models are limited in that they only represent certain aspects of the system under study.

CSTA K-12 Computer Science Standards

CT	Abstraction	2-12	Use abstraction to decompose a problem into sub problems.
CT	Abstraction	3B-10	Decompose a problem by defining new functions and classes.
CT	Algorithms	3A-3	Explain how sequence, selection, iteration and recursion are the building blocks of algorithms.
CT	Connections to other fields	2-15	Provide examples of interdisciplinary applications of computational thinking.
CT	Data representation	2-8	Use visual representation of problem state, structure and data.
CT	Data representation	3A-12	Describe how mathematical and statistical functions, sets, and logic are used in computation.

Responsiveness to Varied Student Learning Needs

In Project GUTS, we integrate teaching strategies found to be effective with learners with various backgrounds and characteristics such as economically disadvantaged students (EDS), students from groups that are underrepresented in STEM (URG), students with disabilities (DIS), English Language learners (ELL), girls and young women (FEM), students in alternative education (ALT), and gifted and talented students (GAT).

In each lesson we describe the accommodations and differentiation strategies that are integrated in the activities to support a wide range of learners.

Module 1 Lesson 5 Activity #1: Introduction to Epidemiology

(EDS) *In the Introduction to Epidemiology we elicit students' prior knowledge of getting a communicable disease and build on their funds of knowledge as a resource for further questioning and investigating.*

(EDS) *We validate the sense of place [aspects of the neighborhood and school] to keep the students engaged* *and make a connection of science and neighborhood by using the example of Community Associated MRSA. We refer to the cultural context of school and community, neighborhoods, when discussing the model of CA-MRSA and what was included in the model.*

(URG)(DIS) *We use technology to present information in multiple modes of representations. In the StarLogo Nova modeling and simulation environment, students can present information as code blocks, text, graphical display of the simulation, and as data in tables and graphs.*

(FEM) *We chose a curriculum topic, epidemiology, which had relevancy and real-world application, a strategy to interest and engage girls and students from groups that are underrepresented in STEM.*

(ALT) *We focus on career connections, the work of epidemiologists, a life-skills strategy that is promoted in alternative education.*

Module 1 Lesson 5 Activity #2: Modeling the Spread of Disease (Contagion model)

(EDS)(URG)(ELL) *We ask students to apply what they know, specific to their cultural and/or socio-economic context, when addressing issues related to the spread of disease and the realism of models. We can ask what is assumed in this model, what is realistic to the students, and what is not. This practice can also highlight home culture connection to science by capitalizing on funds of knowledge from the students' homes and communities.*

(URG)(DIS) *We use technology to present information in multiple modes of representations. In the StarLogo Nova modeling and simulation environment students can present information as code blocks, text, graphical display of the simulation, and as data in tables and graphs.*

(URG) *We can connect science to locally meaningful issues that could promote community involvement and social activism. When discussing what else spreads like a disease, note that other trends and phenomena are studied as contagions. Examples include gossip, gun violence, fads and fashions, etc.*

(ELL) *The "place-based" nature of this lesson establishes connections between school science and the students' community and lives.*

(FEM) *We chose a curriculum topic, epidemiology, which had relevancy and real-world application, a strategy to interest and engage girls and students from groups that are underrepresented in STEM. Careful planning of partners for the on-computer activity is a strategy that encourages participation for the girls in science.*

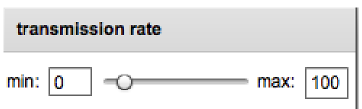
Student Activity #2 Guide:

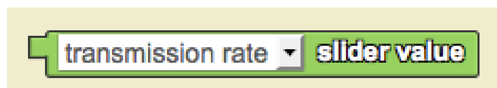
Part 1: Altering Colliding Turtles to create an Epidemic Model

Your challenge is to make the turtles spread disease to one another upon collision.

<u>Guidelines:</u>

Remix the “Epidemic Model starter” model
Don’t forget to put both partners’ names in the project title.
Create 300 turtles (Refer to Lesson 2)
Create a transmission rate slider, set the maximum to 100.
Use the value in the transmission rate slider as the probability of passing on the disease.
Save and test your model.

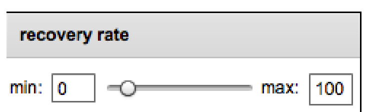




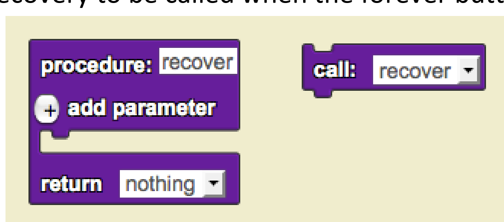
Part 2: Customizing your model. [adding in recovery]

Your challenge is to make the turtles recover from the disease.

Create a recovery rate slider.
Create a recovery procedure.
Use the value in the recovery rate slider as the probability that recovering from the disease at each step.



Then guide them in creating a recovery procedure.   At each step, a sick person has a chance of recovering so we will need recovery to be called when the forever button is toggled.



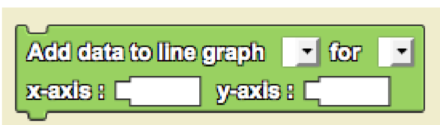
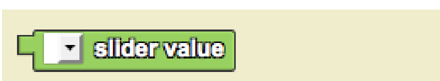
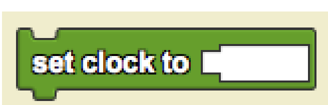
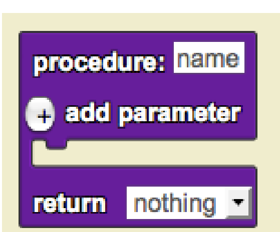
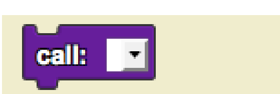
Note: the “call recovery block” needs to be within the “when forever toggled” loop.
Within the recovery procedure, students will need to use the random function to mimic rolling the 100-sided die as we did in the transmission case.

Testing your model

- Save and test your model.
- Try changing the recovery rate. Did you see any new outcomes or patterns?

When you are done, save and share your project.

StarLogo Nova Blocks introduced in Module 1 Lessons 5

	Push this button to get access to the widgets. Widgets are user-input and output elements like buttons, data output, tables and graphs.
	This button appears AFTER clicking “Edit Widgets”. Select the type of widget you’d like to add. Give the widget a name. “Push button” – a button that runs once or pops up after being pushed. “Toggle button” – a button that stays on until turned off. “Data Box” – a data display box. “Label” – a place for the use to enter data? “Horizontal Slider” – a slider to control the value of a variable. “Table” – a configurable data table with rows and columns. “Line Graph” – a configurable line graph with one or more lines. “Bar Graph” – a configurable bar graph with one or more bars.
	After creating a data box, use this block to set its value to a number or a color.
	After creating a line graph widget and naming a series (on the interface), this block adds data to the line graph. It will plot the point x, y based on the blocks that you provide. Usually these values are variables that you have defined.
	After setting up a slider widget, you can use the value of the slider in your code using this block.
	Returns the current value of the clock. This can be used when you plot data vs. time.
	Changes the value of the clock. This block is particularly useful for resetting the clock on setup.
	Creates a procedure with a name and a list of commands. Procedures are useful for organizing code into reusable modules. Some procedures just make changes to the agents and the world and return nothing. Other procedures, calculate values and return the result.
	Calls the procedure selected.

1

Lesson 6

Adding Instrumentation to your Model and Running Experiments

50 minutes (1 day)

Lesson Overview

In this lesson students will add instrumentation to their model so they can collect quantitative data on the spread of disease. Students will use this model to run experiments to determine if disease will spread throughout a virtual population under different initial conditions and different scenarios.

Teaching Summary

Getting started – 5 minutes

1. Review of the previous day's lesson and concepts and connection to today's lesson

Activity 1: Instrumenting your model – 10 minutes

2. Review qualitative vs. quantitative data
3. Add a line graph
4. Test your model

Activity 2: Running experiments – 30 minutes

5. Designing experiments
6. Running experiments
7. Collecting and analyzing data

Wrap-up – 5 minutes

8. What patterns did you uncover? What conditions or settings led to each pattern?
9. When you run the model with the same input setting (for transmission rate and recovery rate) do you always get the same result or outcome? Why or why not?

Lesson Objectives

The student will:

✓ Learn the difference between qualitative vs. quantitative results (LO33)

- ✓ Learn how to instrument a model with a line graph (LO34)
- ✓ Learn experimental design using a computer model (LO35)
- ✓ Conduct experiments using a model as an experimental test bed (LO36)
- ✓ Record and analyze results (LO37)
- ✓ Ask questions that arise from observations of your model's behavior (LO38)

Teaching Guide

Materials, Resources and Prep

For the Students

- Computers
- StarLogo Nova Epidemic model
- Experimental Design handout

For the Teacher

- Computer and projector
- Slide presentation with new commands and concepts

Getting started - 5 min

1. Review of previous day's lesson and link to where we are going today

- What are variables and procedures?
- How are they used in models?

Activity #1: Instrumenting your model - 10 min

2. Qualitative vs. quantitative data

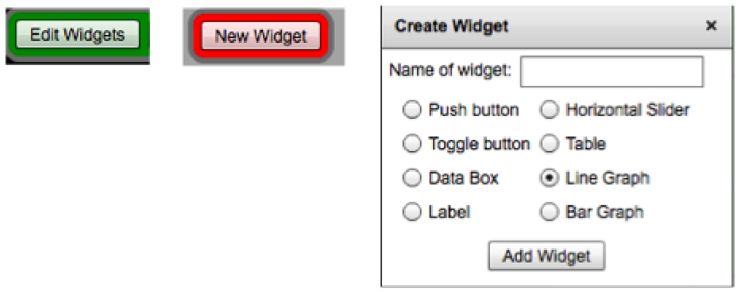
- Using the epidemic model, students can make qualitative observations of the changes that occur as we change the transmission rate. But sometimes, we need quantitative data to get a better sense of change over time.
- Definitions: Qualitative means relating to, measuring, or measured by the *quality* of something (its size, appearance, value, etc.) rather than its quantity. Quantitative means relating to, measuring, or measured by the *quantity* of something rather than its quality.
- In order to collect and visualize quantitative data we need to add a line graph in StarLogo Nova. The graph will collect data on the time elapsed since the model started running and the cumulative number of healthy and sick individuals over time.
- Then we can compare patterns in the collected data.

3. Adding a line graph

- We need some way of tracking the spread of disease; what data should we collect?
- What do you want to find out using the model? [Does everyone get sick? If so, how fast? For example, if we are interested in seeing how fast the disease spreads, we will need to know when everyone is sick.]
- If we want to know if anyone remained healthy throughout the epidemic, we need to keep track

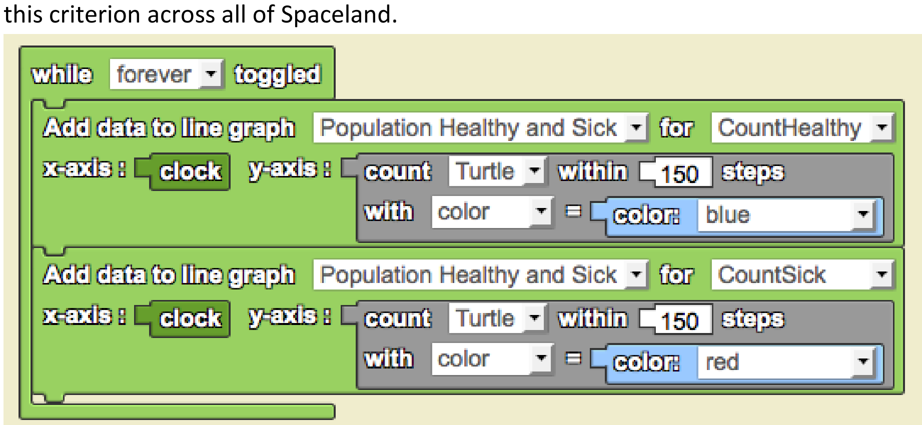
of the number of healthy people.

- Let's create a new line graph called "Population Healthy and Sick"
- How can we track the sick and healthy people? What is different about how they are displayed? [healthy are blue and sick are red]
- Demonstrate how to create a line graph using the slide presentation or StarLogo Nova.
- In the Colliding Turtles model, turtles always reacted to a collision. In our epidemic model, we will incorporate a transmission rate so only some of the time a disease is passed from one agent to another.
- Demonstrate how to create a widget or show students using the slide presentation.



[Edit Widget, New Widget, Create Widget by specifying name and select Line Graph, click on Add Widget. You can reposition the Line Graph in Edit Widgets mode.]

- Drag the line graph off to the side of Spaceland.
- Double click on New Series and change its name to "CountHealthy" then select blue as its line color.
- Add another Series and change its name to "CountSick" then select red as its line color.
- Finally, click "Edit Widgets" to leave editing mode and return to play mode.
- Next, we want "The World" to update the line graph each time through the forever loop, so we need to add a while forever toggled command to the page labeled "The World."
- Notice that we need the "clock" along the x-axis and the count of blue turtles on the y-axis. We can get the count of turtles that are blue using the count command. Since the whole Spaceland is less than 150 steps square, saying "within 150 steps" will get us the number of turtles that fit this criterion across all of Spaceland.



4. Test your model

- Does your line graph work?
- What patterns can you now see that were difficult to see without a line graph?

Activity #2: Designing and Running Experiments - 30 min

5. Design your experiment

- Assign students to work in small groups.
- Hand out the experimental design form and review its contents.
- Give students time to plan and describe their experiment using the form.
- Ask, “What are the dependent and independent variables?”
- Ask, “What are the variables that already exist in this model?” [#initially sick, density of population, transmission rate, population movement, recovery rate.] Choose one variable to experiment with, then use the “Experimental Design handout” to describe your experiments and record data from your experiments.
- Ask, “How many trials do you need to run at each setting of the variable?”

6. Run your experiment

- Students are to run the experiment they described in the form and collect data.
- What is the range of values for the variable you chose?
- How many trials will you run at each setting?
- How will you capture the data?
- Currently the “data output to file” function is not available in StarLogo Nova. An alternative is to grab screen shots of the graph and label them with the settings.
- Show students how to do a screen grab and name their file.
- Another alternative is to keep the line graph recording all experiments without clearing the line graph.

7. Analyzing data and describing the results of your experiment

- Ask, “What patterns do you see in the collected data?”
- Ask, “What correlations do you think exist between the variable setting and the outcomes?”

Wrap-up – 5 min

8. What does your model tell you?

- What assumptions are made in your model? Was there any key factor that was left out that might be important in the real world?

Assessment Questions

- Is the epidemic model a model of a complex adaptive system? Why or why not?
- What variables were included in your epidemic model?
- What was included and what was missing from your model? Name two things that happen in real life that are not included in this model.
- How would you change the model to one in which sick agents get healthy again after colliding into a healthy agent?
- How would you modify the model to help you study a real-world disease?

Standards Addressed

NRC Scientific and Engineering Practice Standards

Practice 1: Asking questions and defining problems
1A: Ask questions that arise from careful observation of phenomena, models, or unexpected results.
1B: Ask question to identify and/or clarify evidence and/or the premise(s) of an argument.
1C: Ask questions to determine relationships between independent and dependent variables and relationships in models.
1F: Ask questions that can be investigated within the scope of the classroom, outdoor environment, and based on observations and scientific principles.

Practice 2: Developing and using models
2C: Use and/or develop a model of simple systems with uncertain and less predictable factors.
2E: Develop and/or use a model to predict and/or describe phenomena.
2G: Develop and/or use a model to generate data to test ideas about phenomena in natural or designed systems, including those representing inputs and outputs, and those at unobservable scales.

Practice 3: Planning and carrying out investigations
3A: Plan an investigation individually and collaboratively, and in the design: identify independent and dependent variables and controls, what tools are needed to do the gathering, how measurements will be recorded, and how many data are needed to support a claim.
3B: Conduct an investigation and/or evaluate and/or revise the experimental design to produce data to serve as the basis for evidence that meet the goals of the investigation.
3D: Collect data to produce data to serve as the basis for evidence to answer scientific questions or test design solutions under a range of conditions.

Practice 4: Analyzing and interpreting data
4B: Use graphical displays (e.g., maps, charts, graphs, and/or tables) of large data sets to identify temporal and spatial relationships.
4D: Analyze and interpret data to provide evidence for phenomena.
4E: Apply concepts of statistics and probability (including mean, median, mode, and variability) to analyze and characterize data, using digital tools when feasible.
4F: Consider limitations of data analysis (e.g., measurement error), and/or seek to improve precision and accuracy of data with better technological tools and methods (e.g., multiple trials).
4G: Analyze and interpret data to determine similarities and differences in findings.

Practice 5: Using mathematics and computational thinking
5A: Use digital tools (e.g., computers) to analyze very large data sets for patterns and trends.
5B: Use mathematical representations to describe and/or support scientific conclusions and design solutions.
5D: Apply mathematical concepts and/or processes (e.g., ratio, rate, percent, basic operations, simple algebra) to scientific and engineering questions and problems.

Practice 6: Constructing explanations and designing solutions
6D: Apply scientific ideas, principles, and/or evidence to construct, revise and/or use an explanation for real-world phenomena, examples, or events.
6E: Apply scientific reasoning to show why the data or evidence is adequate for the explanation or conclusion.

Practice 7: Engaging in argument from evidence
7C: Construct, use, and/or present an oral and written argument supported by empirical evidence and scientific reasoning to support or refute an explanation or a model for a phenomenon or a solution to a problem.

Practice 8: Obtaining, evaluating, and communicating information
8E: Communicate scientific and/or technical information (e.g. about a proposed object, tool, process, system) in writing and/or through oral presentations.

NRC Crosscutting Concepts

1. Patterns:
1B: Patterns in rates of change and other numerical relationships can provide information about natural and human designed systems.
1C: Patterns can be used to identify cause and affect relationships.
1D: Graphs, charts, and images can be used to identify patterns in data.

2. Cause and Effect:
2C: Phenomena may have more than one cause, and some cause and effect relationships in systems can only be described using probability.

3. Scale, Proportion, and Quantity
3A: Time, space, and energy phenomena can be observed at various scales using models to study systems that are too large or too small.

4. Systems and Systems models
4A: Systems may interact with other systems; they may have sub-systems and be a part of larger complex systems.
4B: Models can be used to represent systems and their interactions—such as inputs, processes and outputs—and energy, matter, and information flows within systems.
4C: Models are limited in that they only represent certain aspects of the system under study.

CSTA K-12 Computer Science Standards

CT	Connections to other fields	2-15	Provide examples of interdisciplinary applications of computational thinking.
CT	Data representation	2-8	Use visual representation of problem state, structure and data.
CT	Modeling & simulation	1:6-4	Describe how a simulation can be used to solve a problem.
CT	Modeling & simulation	2-11	Analyze the degree to which a computer model accurately represents the real world.
CT	Modeling & simulation	2-9	Interact with content-specific models and simulations to support learning and research.
CT	Modeling & simulation	3A-8	Use modeling and simulation to represent and understand natural phenomena.
CT	Modeling & simulation	3B-9	Analyze data and identify patterns through modeling and simulation.
CPP	Data collection & analysis	2-9	Collect and analyze data that are output from multiple runs of a computer program.
CPP	Data collection & analysis	3B-7	Use data analysis to enhance understanding of complex natural and human systems.
CPP	Programming	3A-3	Use various debugging and testing methods to ensure program correctness.
CPP	Programming	3A-4	Apply analysis, design and implementation techniques to solve problems.

Responsiveness to Varied Student Learning Needs

In Project GUTS, we integrate teaching strategies found to be effective with learners with various backgrounds and characteristics such as economically disadvantaged students (EDS), students from groups that are underrepresented in STEM (URG), students with disabilities (DIS), English Language learners (ELL), girls and young women (FEM), students in alternative education (ALT), and gifted and talented students (GAT).

In each lesson we describe the accommodations and differentiation strategies that are integrated in the activities to support a wide range of learners.

Module 1 Lesson 6: Adding Instrumentation and Running Experiments with Your Model

(URG) *Students are given "agency" as the creators of their own models, and as researchers seeking to answer a question or understand a phenomenon. The models that students build are their own creations.*

(FEM) *Careful planning of partners for the on-computer activity is a strategy that encourages participation for the girls in science.*

(DIS)(GAT) *There is ample opportunity to extend level of content and time for practice by compacting areas already mastered and to allow more time for students to complete previous experiments and customizations. This differentiation strategy of pacing can benefit a wide range of students, including those with learning disabilities and/or gifted and talented students.*

(GAT) *The teacher can promote autonomy and the forging of authentic connections to the epidemiology content and to the practice of computer modeling.*

(GAT) *Grouping students of similar interests and ability, incorporating standards from a higher grade, and providing opportunities for self-directed projects are successful strategies for gifted and talented students.*