Unit Name: Solar System Model	Duration: 2, 1-hour sessions (approx. 2 hours total)
Note to the Teacher	The first two sessions of Module 3 guide students through the next Scratch project, the Solar System Model. The example program is an animation that utilizes user input to model the revolutions of planets around the sun. Users are prompted by the program to enter for how many days they would like to see the planets move. They can type their answers directly into the interface (the stage). The user can then see how far around the sun each of the planets moves in that amount of days. This is the first time that the students will be programming in Scratch 2.0. Scratch 2.0 provides a new interface for the students, but they will not find it difficult to transition from the previous version of Scratch to this one. The students will be excited to see the different sprites and backgrounds that can be utilized in Scratch 2.0. If teachers want to give the students some time to explore these new items, they can. • Students are provided with a prepared program that has the necessary sprites and backgrounds but not the code. The sessions' PowerPoints carefully lead them through coding the program. Instructions for students have diminishing levels of guidance with detailed instructions for the first planet, Mercury, and independent coding of the remaining planets. • Students are first shown that the planets take different amounts of time to revolve around the sun. Students may have some misconceptions about this, or they may not have been exposed prior to information about the solar system. For example, the colors of the planets were selected based on the characteristics of that planet. So, Mars is red because of the rust in its rocks on the surface. The colors of the planets are explained in the PowerPoint slides. Students are also told how many days each planet requires to revolve around the sun, how to incorporate the user's input, how to use the pen command to trace the path of the planet, and how to use math in their program so each planet moves the correct distance. • The following terms are included in the Solar System Model: gravity, revolution, arc of a circle, elliptical, fraction, and division. • As part of developing the Solar System Model, students learn to add *comments* to their scripts. Comments were not available in Scratch 1.4 but are available to programmers in 2.0. A comment within the programming has no effect on the program itself. But, it helps the programmer and anyone reviewing the code to understand it better. Comments can answer questions like: What does this stack do? What is its purpose? Why is this stack built in this way?
Standards Addressed	**CSTA:** •L1:3.CT-01; •L1:3.CT-02; •L1:6.CT-01; •L1:6.CT-05; •L1:6.CT-06; •L1:3.CL-01; •L1:3.CL-02; •L1:6.CL-01; •L1:6.CL-02; •L1:6.CL-03; •L1:3.CPP-01; •L1:3.CPP-02; •L1:3.CPP-03; •L1:3.CPP-04; •L1:3.CPP-06; •L1:6.CPP-01; •L1:6.CPP-02; •L1:6.CPP-03; •L1:6.CPP-05; •L1:6.CPP-06; •L1:6.CPP-08; •L1:3.CD-01; •L1:6.CD-01; •L1:6.CD-03; L1:6.CD-04; •L1:3.CI-01; •L1:6.CI-01; •L1:6.CI-02; •L1:6.CI-03; •L1:6.CI-04 **Interdisciplinary:** •CCSS.ELA-Literacy.RI.3.3; •CCSS.ELA-Literacy.RI.4.3; •CCSS.ELA-Literacy.RI.3.4; •CCSS.ELA-Literacy.RI.4.4; •CCSS.ELA-Literacy.RI.4.7; •CCSS.ELA-Literacy.RI.5.7; •CCSS.ELA-Literacy.RI.4.9; •CCSS.ELA-Literacy.RI.5.9; •CCSS.ELA-Literacy.RI.3.10; •CCSS.ELA-Literacy.W.3.2; •CCSS.ELA-Literacy.W.4.2; •CCSS.ELA-Literacy.W.3.6; •CCSS.ELA-Literacy.W.3.7; •CCSS.ELA-Literacy.W.3.8; •CCSS.ELA-Literacy.SL.3.3; •CCSS.ELA-Literacy.SL.3.4; •CCSS.ELA-Literacy.SL.4.5; •CCSS.ELA-Literacy.SL.5.5; •CCSS.Math.Content.5.G.A.1; •MS-ESS1-1; •MS-ESS1-2;•MS-ES1-3

Objectives SWBAT:	• Understand Scratch as a tool for communicating and explaining • Apply Scratch skills learned in Levels 1, 2, and 3 • Follow instructions to create a model that animates the time it takes planets to revolve around the sun • Understand and apply pseudocode for planning purposes • Apply mathematics to improve the accuracy of the model • Apply coding concepts and practices • Understand that storyboards and other charts (design, Gantt, and rubrics) are useful tools for planning
Materials	• Scratch 2.0 • Solar System Model — finished program • Solar System Model — no code • PowerPoints for the Solar System Model (Parts 1 & 2) • Student Worksheets o The Collaboration and Project rubrics o Check Your Understanding questions for Module 3
Anticipatory Set	Before starting the first two sessions, remind students that earlier in the year, they created their own project in Scratch that also involved an area of science. That was the Water Cycle Quiz. Some teams might have created quizzes about other topics within science. Tell the students that they are going to do something similar with this project, but this project is going to be a model instead of a quiz. The model is going to be interactive, like the Water Cycle Quiz, but in a slightly different way. Students at this age may have many different ideas and thoughts about the solar system. Engage the students in a discussion about what they know and what they would like to know about the solar system. This can then be used to guide the students when it comes time for them to research information about the solar system on their own.
Model	The teacher may choose to model how the different planets in the solar system have years of different lengths at least partly because of the planets' different distances from the sun. The teacher may have the students act this out, with the students acting as planets and the sun. The teacher could also choose to show the user interface of Scratch 2.0, which is new to students. Students may be interested in exploring Scratch 2.0 and the teacher could use this as an opportunity to model the new interface.
Check for Understanding - Formative Assessment	Students answer the Check Your Understanding questions to assess what they understand from Sessions 1 and 2. They can be completed at any time during the session. Therefore, it might be easier to allow students to review the questions before completing the sessions.
Guided Practice	The PowerPoints for each session provide guided programming practice. As described elsewhere, more scaffolded guidance is replaced by prompts to complete the coding independently as students continue through the coding of the example project. If students are proceeding through the session PowerPoints independently at their computers, teachers should check that students continue to make progress, looking for signs that the student might be stuck, confused, or frustrated. If teachers are leading students through the PowerPoints, they should ask questions along the way to ensure that students are following. • The following questions can always be asked when presenting stacks of code: What does this block do? What sequence does the program follow with this stack? How does the programming of this sprite affect the program? How does this sprite's program connect to what happens before and after it?
Closure	After students have coded the example Solar System Model, they are ready to join teams for creating their own model. What they learned about programming the Solar System

	Model will be important for planning and programming the team's model. If students have any lingering questions about the program, they should ask them now.
Independent Practice	As more scaffolded guidance within the PowerPoints is replaced by prompts to complete the coding independently, students can apply and practice the skills presented earlier. In subsequent sessions they will gain independent practice within teams as they design and program their own models.
Enrichment/Extension Activity	The self-directed enrichment activities for the first two sessions challenge the students to plan unique programming solutions based upon what they have learned up until this point. • The first challenge (following Session 1) asks students to program that can process user input and provide feedback based upon that answer. The challenge is a guessing game in which a sprite tells the user that they need to guess a number between 1 and 100. Then, based upon the user's guess, the sprite will say if the guess is too high, too low, or the correct number. The user is not allowed an unlimited number of guesses. After they are out of guesses, the program tells them that they lost. This is a more difficult challenge than the activities presented in the first semester. • The second challenge (following Session 2) asks students to solve a maze challenge in which the player sprite is controlled by the user's presses of the arrow keys in an attempt to navigate through a maze. If the user contacts any of the walls, they are sent back to the start of the game. When the user reaches the goal, a congratulatory message is displayed. The goal is to 'save' a sprite within the maze. The player sprite needs only to touch the sprite to save it. There is also another sprite that serves as an obstacle. This sprite should move around the screen randomly at a chosen speed. If the player sprite touches the obstacle sprite, it must return to the beginning.
Final Project or Exam – Summative Assessment	The model project created by teams within the next six class sessions will serve as an assessment of students' abilities to create (design, plan, and program) using the skills learned previously (both programming and collaborative, and from Levels 1 and 2 as well as 3).