Lesson 5: Programming with Scrat

38 minutes

Overview

In this **skill-building** lesson, students will continue to develop sequential algorithms.

Purpose

In this lesson, students will develop programming and debugging skills on a computer platform. The block-based format of these puzzles help students learn about sequence and concepts, without having to worry about perfecting syntax.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Agenda

Warm Up (3 minutes)
- **Vocabulary**
- **Review: Unplugged Activity**

Main Activity (30 minutes)
- **Programming With Scrat**

Wrap Up (5 minutes)
- **Reflection**

Extended Learning

Cross-Curricular Opportunity

Objectives

Students will be able to:

- Build a computer program from a set of written instructions
- Choose appropriate debugging practices when solving problems
- Construct a program by reorganizing sequential movements

Preparation

- Play through the puzzles to find any potential problem areas for your class.
- (Optional) Pick a couple of puzzles to do as a group with your class.
- Make sure each student has a reflection journal.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the teachers

- **CSF Course A Lesson Slides** - Slides (**Download**) ▾ Make a Copy

For the students

- **Debugging With the Step Button** - Video (**Download**)
- **Feeling Faces Emotion Image** - Resource ▾ Make a Copy
- **Pair Programming** - Student Video

Vocabulary

- **Algorithm** - A list of steps to finish a task.

- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Program** - An algorithm that has been coded into something that can be run by a machine.
- **Programming** - The art of creating a program.

Teaching Guide

Warm Up (3 minutes)

Vocabulary

Display: Show "Vocabulary" slide

Say:

- **Algorithm** - A list of steps to finish a task.
- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Program** - An algorithm that has been coded into something that can be run by a machine.
- **Programming** - The art of creating a program.

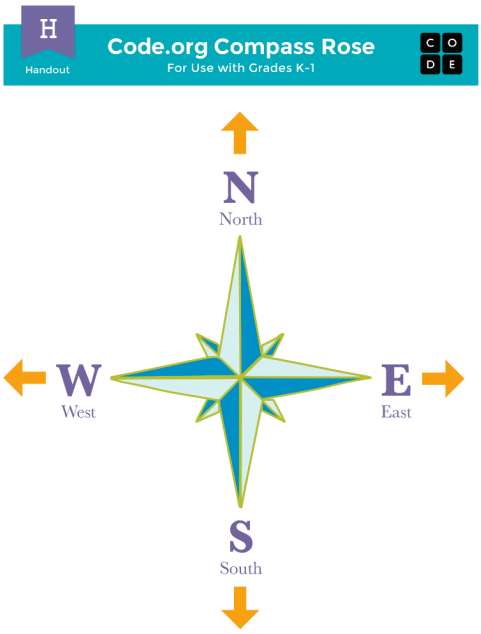
Review: Unplugged Activity

This lesson relies on many of the unplugged ideas that students have learned in the weeks leading up to this first online activity. It is important that you bring those concepts (such as persistence, debugging, algorithms, and programs) around full-circle so that your class can benefit from them in their online work as well.

Teaching Tip

Display: Show "Compass Rose" slide

If your class has already learned cardinal directions, then changing "Up" and "Down" to "North" and "South" shouldn't be a problem. If they have not, we have provided a handy worksheet with the Code.org Compass Rose that you can use to get students onboard. This conversion will come in handy for nearly all of the online puzzles aimed at kindergarten and first grade.



Let students know that they will see those letters in their online programs next to the direction arrows.

Display: Show "Review" slide

Display: Show students the map from the "Happy Maps" exercise that they completed in the lessons prior to this one.

Discuss: Ask students to recall the symbols used in "Happy Maps."

- *What happens when the flurb reads the "North" arrow?*
- *What happens when the flurb reads the "East" arrow?*
- *What would happen if we made a mistake when programming the flurb? What if there was a "bug" in our program? Would we throw the whole thing away and start over?*

Encourage students to think about the debugging tips:

- *Was everything right at the first step?*
- *How about the second?*
- *Where did it go wrong?*

Transition: Once you are satisfied that your students remember "Happy Maps", you can move into the Bridging Activity.

Preview: Programming in Maze (3 minutes)

To prepare students, preview an online puzzle (or two) as a class.

Display: Show "Preview" slide

Model: Reveal an entire online puzzle from the progression to come. We recommend Lesson 5, Puzzle 5. Point out the "Play Area" with *Scrat*, as well as the "Work Space" with the Blockly code. Explain that this Blockly code is now the language that students will be using to get *Scrat* to the acorn. Do they see any similarities to the exercise that they just did? What are the big differences?

Work with your class to drag code into the workspace in such a way that *Scrat* (eventually) gets to the acorn.

Transition: Students should now be ready to transition to computers to complete online puzzles on their own.

Main Activity (30 minutes)

Programming With Scrat

If you are looking for some extra puzzles to cover with your class, here are some "prediction" puzzles that will allow you to walk through existing code with your students to predict what *Scrat* will do. It is a good idea to cover them together before letting students loose on their own computers.

Prediction Levels:

- **Course B, Programming in Maze #1**
- **Course B, Programming in Maze #2**

Display: Show “Decompose it!” slide

Remind students to try coding one block at a time. Make sure they press “Run” frequently in order to break the following challenges down into small steps.

Display: Show “Video: Pair Programming” slide

Watch the ***Pair Programming** video with your students, then assign them to pairs. This should help students start off in the right direction.

1 **Video: Pair Programming**

Display: Show “Level 2-5: Skill Building” slide

2-5 **Skill Building**

2 3 4 5

Display: Show “Video: Debugging with the Step Button” slide

6 **Video: Debugging with the Step Button**

Display: Show “Level 7-8: Skill Building” slide

7-8 **Skill Building**

7 8

Display: Show “Level 9: Challenge” slide

9 **Challenge**

Display: Show “Level 10-12: Practice” slide

10-12 **Practice**

10 11 12

Display: Show “Lesson Extras” slide

13-14 **Lesson Extras**

Teaching Tip

Show the students the *right* way to help classmates by reminding them:

- Don’t sit in the classmate’s chair.
- Don’t use the classmate’s keyboard.
- Don’t touch the classmate’s mouse.
- Make sure the classmate can describe the solution to you out loud before you walk away.

Circulate: Teachers play a vital role in computer science education and supporting a collaborative and vibrant classroom environment. During online activities, the role of the teacher is primarily one of encouragement and support. Online lessons are meant to be student-centered, so teachers should avoid stepping in when students get stuck. Some ideas on how to do this are:

- Utilize ***Pair Programming** - student video whenever possible.
- Encourage students with questions/challenges to start by asking their partner.
- Unanswered questions can be escalated to a nearby group, who might already know the solution.
- Remind students to use the debugging process before you approach.
- Have students describe the problem that they’re seeing. What is it supposed to do? What does it do? What does that tell you?
- Remind frustrated students that frustration is a step on the path to learning, and that persistence will pay off.
- If a student is still stuck after all of this, ask leading questions to get the student to spot an error on their own.

Wrap Up (5 minutes)

Reflection

Display: Show "Reflection” slide

Do This: Having students write about what they learned, why it’s useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- Draw one of the ***Feeling Faces** that shows how you felt about today's lesson in the corner of your journal page.
- Draw *Scrat* and an acorn somewhere on your paper. Can you write a program to get Scrat to the acorn?

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

In small groups, let students design their own mazes on paper and challenge other students or groups to write programs to solve them. For added fun, make life-size mazes with students as Scrat and the acorn.

Cross-Curricular Opportunity

Saving Scrat's Acorns (70-90 minutes)

Computer Science + English Language Arts + Math + Science

Saving Scrat's Acorns is an optional activity aligned to Common Core ELA, Common Core Math and Next Generation Science Standards, written by our teacher community. Students will help *Scrat* count, compare, identify, design and build a tent as he makes his way across the ice to gather his acorns.

Standards Addressed:

- **CCSS.ELA-LITERACY.RI.K.2:**With prompting and support, identify the main topic and retell key details of a text.
- **CCSS.MATH.CONTENT.K.CC.A.3:**Write numbers from 0 to 20. Represent a number of objects with a written numeral 0-20 (with 0 representing a count of no objects).
- **CCSS.MATH.CONTENT.K.CC.C.6:** Identify whether the number of objects in one group is greater than, less than, or equal to the number of objects in another group, e.g., by using matching and counting strategies.
- **CCSS.MATH.CONTENT.K.CC.B.5:** Count to answer "how many?" questions about as many as 20 things arranged in a line, a rectangular array, or a circle, or as many as 10 things in a scattered configuration; given a number from 1-20, count out that many objects.
- **NGSS.K-PS3-2:** Use tools and materials to design and build a structure that will reduce the warming effect of sunlight on an area.

This work is available under a **Creative Commons License (CC BY-NC-SA 4.0)**.

If you are interested in licensing Code.org materials for commercial purposes **contact us**.