Course A

Course A offers computer science curriculum for beginning readers around the kindergarten age range. Students will learn to program using commands like loops and events. The lessons featured in this course also teach students to collaborate with others meaningfully, investigate different problem-solving techniques, persist in the face of difficult tasks, and learn about internet safety. By the end of this course, students create their very own custom game or story from Play Lab that they can share.

Persistence

Computer science requires persistence. We've created two short picture books and an optional activity to model ways to approach struggle and problem solving in a way that young students can understand. Read these stories with your class any time you want to introduce or reinforce persistence. After reading, you can even complete Stevie's big "marble run" project with your class!

Picture Book: Unspotted Bugs Slides | Video

Picture Book: Stevie and the Big Project Slides | Video

Optional Activity: Marble Run Teacher Guide

Journaling

The lessons in this course include journaling prompts. Journals are also useful as scratch paper for building, debugging, and strategizing. Journals can become a fantastic resource for referencing previous answers when struggling with more complex problems.

Think Spot Journal Student Handout

Debugging

From beginners to professionals, debugging is an essential yet often underrated practice. It is likely that your students will find most of their "coding" time is actually spent fixing bugs! To encourage students to take ownership of this practice, we provide this handy reference they can use while coding. Please consult the "Debugging" section of our CS Fundamentals Curriculum Guide for more information on this, as well as other debugging facilitation strategies for your classroom.

Debugging Guide Student Handout

Chapter 1: Digital Citizenship

Lesson 1: Going Places Safely

Unplugged | Online Safety

Learn the rules to safely visit places online.

Chapter Commentary

Digital Citizenship

Chapter 2: Sequencing

Lesson 2: Learn to Drag and Drop

Skill Building | Sequencing

Click and drag to finish the puzzles.

Lesson 3: Happy Maps

Unplugged | Sequencing

Write instructions to get the Flurb to the fruit.

Lesson 4: Sequencing with Scrat

Skill Building | Sequencing

Program Scrat to reach the acorn.

Lesson 5: Programming with Scrat

Skill Building | Sequencing

Write programs to help Scrat reach the acorn.

Lesson 6: Programming with Rey and BB-8

Skill Building | Sequencing

Help BB-8 collect the scrap metal.

Chapter Commentary

Sequencing

Chapter 3: Loops

Lesson 7: Happy Loops

Unplugged | Loops

Help the Flurb solve bigger problems using loops.

Lesson 8: Loops with Scrat

Skill Building | Loops

Help Scrat cover more ground using loops.

Lesson 9: Loops with Laurel

Skill Building | Loops

Help Laurel the adventurer collect the treasure underground.

Lesson 10: Ocean Scene with Loops

Skill Building | Loops

Use loops and patterns to finish the images.

Chapter Commentary

Loops

Chapter 4: Events

Lesson 11: The Big Event Jr.

Unplugged | Events

Move and shout when your teachers presses buttons on a giant remote.

Lesson 12: On the Move with Events

Create your own game or story.

Chapter Commentary

Events

Lesson 1: Going Places Safely

Overview

In collaboration with Common Sense Media, this lesson helps students learn that many websites ask for information that is private and discusses how to responsibly handle such requests. Students also find out that they can go to exciting places online, but they need to follow certain rules to remain safe.

Purpose

Common Sense Education has created this lesson to teach kids the importance of being safe online. By relating places in the real world to websites on the internet, students will make important connections between safe websites and safe places in their own neighborhood.

Agenda

Warm Up (20 min)

- **Where We Go**

Main Activity (20 min)

- **Keep It Private**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Understand that being safe when they visit websites is similar to staying safe in real life
- Learn to recognize websites that are safe for them to visit.
- Recognize the kind of information that is private and understand that it should never be shared online.

Preparation

- [ ] Print one assessment for each student.
- [ ] Make sure each student has a journal.
- [ ] Review CSF Digital Citizenship resource list for more online safety content.

Links

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

For the Teachers

- **Going Places Safely** - Assessment Answer Key **Make a Copy**
- **Common Sense Education** - Website
- **CSF Digital Citizenship** - Resource List

For the Students

- **Going Places Safely** - Lesson Video (**download**)
- **Going Places Safely** - Assessment **Make a Copy**

Vocabulary

- **Digital Citizen** - Someone who acts safely, responsibly, and respectfully online.

Teaching Guide

Warm Up (20 min)

Where We Go

- Invite students to talk about places they have visited on a class field trip.
  - If students have limited experience with field trips, provide some examples of the types of places they could visit as a class, such as museums, science centers, or zoos.
  - Have students choose a place they would like to go on a class field trip.
- Have students take an imaginary field trip to their chosen place.
  - Narrate the preparations while having students pantomime what's happening – For example: put on your jacket; climb on/off the bus; get your ticket checked; go inside.
    - Have students describe what they think they might see and do once they arrive.
- Let the students sit back down, then ask: "What do you need to do to stay safe when you visit new places?"

Play **Going Places Safely - Lesson Video** .

What three rules does the character (Arms) follow when he goes places online?

1) Always ask your parent (or teacher) first

2) Only talk to people you know

3) Stick to places that are just right for you

Now, let's see what more we can do to keep ourselves safe.

Main Activity (20 min)

common sense education®

Keep It Private

- Invite students to give examples of information that they should keep private.
  - Write down their responses on the board or chart paper so that you can return to them later in the lesson.
- Make sure they understand that private information includes the following:
  - full name
  - age
  - address
  - telephone number
  - email address (or parents' email addresses)
  - where they go to school or after school
  - where their parents work
- Encourage students to discuss why it is important to keep this information private.
  - Stress that it is never safe to give out private information to people they don't know.
  - Students should always ask a parent or caregiver before they give out private information to anyone.
- Hand out the worksheet and allow students to complete the activity independently after the instructions have been well explained.
  - This should feel familiar, thanks to the video and discussion.

Wrap Up (15 min)

Flash Chat: What did we learn?

- What information should you always keep private when you are using the computer?
- What can the Internet be used for?
- What rules do we have for visiting places online?

Take the time to discuss again what is appropriate information to share on the Internet, and what is not:

Appropriate	Not Appropriate
Interests	Address
Hobbies	Full Name
First Name	Information that would hurt others

Lesson Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

Journaling

Goal: Help students reflect on the things they learned in this lesson

Give the students a journal prompt to help them process some of the things that they encountered during the day.

Journal prompts could include:

- What was today's lesson about?
- Draw some things that you should never talk to a stranger about on the internet. For example, draw your house to represent your address, draw your school, or draw your family.
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw a computer lab "Do" and a "Don't"

Extended Learning

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

Common Sense Education

- Visit **Common Sense Education - Website** to learn more about how you can keep your students safe in this digital age.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **IC** - Impacts of Computing

Lesson 2: Learn to Drag and Drop

Overview

This lesson will give students an idea of what to expect when they head to the computer lab. It begins with a brief discussion introducing them to computer lab manners, then they will progress into using a computer to complete online puzzles.

Purpose

The main goal of this lesson is to build students' experience with computers. By covering the most basic computer functions such as clicking, dragging, and dropping, we are creating a more equal playing field in the class for future puzzles. This lesson also provides a great opportunity to introduce basic computer hardware terminology, potentially including "mouse", "trackpad" or "touchscreen", depending on your devices.

Agenda

Warm Up (10 min)

- **Behaving in the Computer Lab Discuss**

Preview Online Puzzles (5 min)

Main Activity (20 - 30 min)

- **Learn to Drag and Drop**

Wrap Up (5 - 10 min)

- **Journaling**

Extension Activities

View on Code Studio

Objectives

Students will be able to:

- Recognize what is expected when students transition into the computer lab.
- Use appropriate terminology when referring to a computer mouse, trackpad, or touchscreen.

Preparation

- [ ] Create your own class section on Code.org and make sure every student has a card with their **passcode** on it.
- [ ] Make sure students will be able to access Lesson 2 from their devices. Consider whether you want to hide future lessons to prevent students from moving ahead too quickly.
- [ ] Have the school IT person add a quick link for your class section to the computer desktop.
- [ ] Review the Common Sense Education website for more online safety content.

Links

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

For the Students

- **Pair Programming** - Student Video

Teaching Guide

Warm Up (10 min)

Behaving in the Computer Lab

Goal: This discussion will teach students what to expect and how to behave when they enter the computer lab.

Discuss

Have a good discussion around the computer lab expectations to make sure that students understand the rules. Some topics of discussion might include:

- Is running in the computer lab okay?
- How loudly should we walk when we are in the computer lab?
- What should you do if you get stuck on a puzzle?
- If you get frustrated, will it help to hit the computer?
- When we're about to go to the computer lab, how should we get ready?

Discussion Goals:

- Use calm bodies in the lab
- Remember not to chew gum or candy
- Sanitize your hands
- Sit with your partner at one computer
- Make sure that the first "driver" can reach the mouse
- When you get frustrated, don't hit or shake the computer or monitor
- Follow the **20/20/20 - Website** rule
- How to deal with the **Wiggles** every 20-30 minutes (requires a free login on GoNoodle)
- Ask your partner before you ask the teacher
- Keep volume down so everyone else can hear their partners
- Use your journal for keeping track of feelings and solutions

Preview Online Puzzles (5 min)

Project a puzzle from lesson 2. Show the class how to click on the picture and place it in the correct spot by dragging and dropping. Purposely make mistakes such as clicking the background or dropping the image before it's at the right spot. Ask for help from volunteers in the class when you run into these problems.

Main Activity (20 - 30 min)

Learn to Drag and Drop

Goal: This will teach students how to use Code.org to complete online puzzles.

This stage was designed to give students the opportunity to practice hand-eye coordination, clicking, and drag & drop skills. Students will also play with sequence.

🎓 Take some time to explicitly teach how to click, drag, and drop. Take time to introduce the language around the devices students will be using when they work on the puzzles. If you have tablets, students will be using a **touch screen**. If you have laptops, they will likely be using a **trackpad**. Desktop computers like you might find in a lab will rely on the use of the **mouse**.

Place kids in pairs and have them watch the pair programming video as a class or at their stations. This should help students start off in the right direction.

💡

Content Corner

Considering having students break down the steps of dragging and dropping and record them on the board. For example:

1. Move the arrow to the block.
2. Click and hold the mouse button.
3. Move the mouse.
4. Let go of the button.

Doing this will give students practice with creating an **algorithm**, which is a concept that will be explored in upcoming lessons.

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

Teacher Tip

Show the students the right way to help classmates:

- 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

- Utilize pair programming whenever possible during the activity.
- 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 - 10 min)

Journaling

Goal: Help students reflect on the things they learned in this lesson

Give the students a journal prompt to help them process some of the things that they encountered during the day.

Journal prompts could include:

- Draw one of the feeling faces that shows how you felt about today's lesson in the corner of your journal page.
- Can you draw a sequence for getting ready to go to the computer lab?
- Draw a computer lab "Do" and a "Don't"
- ◦ Draw and label the name of the computer part you used when clicking and dragging during the puzzles. (mouse button, touch screen, trackpad)

Extension Activities

If students complete the puzzles from Stage 4 early, have them spend some time trying to come up with their own puzzles in their **Think Spot Journal - Reflection Journal**.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▸ **AP** - Algorithms & Programming
- ▸ **CS** - Computing Systems
- ▸ **IC** - Impacts of Computing

Lesson 3: Happy Maps

Overview

This unplugged lesson brings together teams with a simple task: get the "flurb" to the fruit. Students will practice writing precise instructions as they work to translate instructions into the symbols provided. If problems arise in the code, students should also work together to recognize bugs and build solutions.

Purpose

The bridge from algorithms to programming can be a short one if students understand the difference between planning out a sequence and encoding that sequence into the appropriate language. This activity will help students gain experience reading and writing in shorthand code.

Agenda

Warm Up (5 min)

- **Step-by-Step**

Activity (40 min)

- **Happy Maps Programming**
- **Students' Turn**

Wrap Up (8 min)

- **Flash Chat**
- **Journaling**

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Translate an algorithm into a program
- Decode and run a program created by someone else
- Identify and address bugs or errors in sequenced instructions

Preparation

- [ ] Print one **Happy Map Cards - Worksheet** and one **Happy Map Game Pieces - Manipulatives** per group of 2-3.
- [ ] Make sure each student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teachers

- **Happy Map Cards** - Worksheet Answer Key **Make a Copy**

For the Students

- **Happy Maps** - Unplugged Video (**download**)
- **Happy Map Cards** - Worksheet **Make a Copy**
- **Happy Map Game Pieces** - Manipulatives **Make a Copy**

Vocabulary

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

Teaching Guide

Warm Up (5 min)

Step-by-Step

This warm-up is meant to get the class thinking about how to break a big problem down into a list of individual steps.

Model: Start by asking the class for step-by-step directions on how to get to the chalkboard. Make sure that you ask students to break apart any large instructions, like "Walk to the chalkboard," into smaller instructions like "Step forward."

When you reach the board, ask for instructions to draw a smiley face and try to get them to keep their instructions equally small.

Say: Well done! You just gave me a list of steps to finish a task. In computer science, that's called an Algorithm! [Follow up with your typical method of introducing vocabulary: word wall, repeat-after-me, etc.]

Teaching Tip

If the class starts shouting simultaneously, explain that you can only hear one instruction at a time. Call on students individually if that helps.

Activity (40 min)

Happy Maps Programming

In this exercise, the class will get map cards that have a pre-defined start space (Flurb) and end space (fruit). Students will need to get the Flurbs to the fruit on each card, using the arrows provided.

Model: Select one of the intermediate maps from the Happy Maps Cards worksheet (e.g., #3). Display it for the class and work through the puzzle together.

Have students look at the puzzle, then think-pair-share their solution for how they would get the Flurb to the fruit.

Think: This Flurb needs to take two steps to get to the fruit. Work with your elbow partner to decide what you think those two steps are.

Pair: Have students discuss with neighbors for about 90 seconds.

Share: Ask a few students to describe their algorithm to the class. Move your finger along the displayed map as the students read their steps. Once you have a solution, ask if anyone else came up with a different idea that also works.

Now, share with the students that the magic step of changing an algorithm into a "program" happens when the code is written down using symbols. Do the students see any symbols on the display?

Think: Challenge students to encode the algorithm that they came up with before into symbols, and to write those symbols down in their journals (or on a piece of paper.)

Pair: Once students have written down their symbols, ask them to swap with a partner to see if they can follow each others' instructions.

Share: Ask for volunteers to come draw their arrows on the board. If the original code doesn't work, spend some time debugging as a class. Students should be familiar with the idea of "debugging" from previous lessons, so be sure to use the vocabulary to get them comfortable with it.

Once the code has been successfully written on the board, congratulate the class on writing their first program together!

Students' Turn

Group: If your class is comfortable, place students into small groups of 2-3. Otherwise, you can continue solving these problems as a class and having them think-pair-share to write programs.

Distribute: Pass out one of the images from the **Happy Map Cards - Worksheet** to each group as needed. **(Optional)** If you start noticing that students are ready for more, use the provided **Happy Map Game Pieces - Manipulatives** and let students choose their own start and finish destinations on the blank map.

Encourage the students to follow these steps:

- Discuss an algorithm to get the Flurb to the fruit.
- Encode the algorithm into arrows in their journals.
- Try their code to see if everything works as expected.
- Debug any issues and fix their code until it works correctly.

Share: When the lesson is done, offer to let groups share out the most difficult maps that they solved. If you had time, ask them to share their solutions as well.

Wrap Up (8 min)

Flash Chat

Discuss: When it's time to wind down, ask students if they can tell the difference between an algorithm and a program. Both are a list of steps, but a program (code) has been encoded in a way that can be run by a machine (or a kindergartener!)

- Do you think that someone who speaks another language would be able to run your program?
- Why or why not?

Journaling

Students should be encouraged to capture their thoughts in their journal after each activity (with text or images.) Choose a journal prompt that will help students remember the purpose of this exercise.

Journal Prompts:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Can you draw your own Flurb map?
- What would the code be to solve your map?

Extended Learning

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

Stuffie Maps

- Create a life-size grid on the rug with tape and have student bring stuffed animals to school.
- Now students can program friends to move their actual stuffies as directed in the programs.

Create Your Own

- Have students create their own maps.
- Have other students solve them using programs.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▸ **AP** - Algorithms & Programming

Lesson 4: Sequencing with Scrat

Overview

Using Scrat from the Ice Age franchise, students will develop sequential algorithms to move a squirrel character from one side of a maze to the acorn at the other side. To do this they will stack code blocks together in a linear sequence.

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.

Agenda

Warm Up (10 min)
- **Behaving in the Computer Lab**

Bridging Activity - Drag and Drop (10 - 15 min)
- **Dragging and Dropping Algorithms**
- **Previewing Online Puzzles as a Class**

Main Activity (20 - 30 min)
- **Online Puzzles**

Wrap Up (5 - 10 min)
- **Journaling**

Extension Activities

View on Code Studio

Objectives

Students will be able to:

- Model proper computer lab behaviors
- Experiment with standard block-based programming actions such as: clicking, drag and drop, etc.

Preparation

- [ ] Cut out direction blocks from **Unplugged Blockly Blocks (Grades K-1) - Manipulatives** to use with the Happy Maps bridging activity.
- [ ] Make sure each student has a journal.

Links

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

For the Students

- **Pair Programming** - Student Video
- **Unplugged Blockly Blocks (Grades K-1)** - Manipulatives

Vocabulary

- **Click** - Press the mouse button
- **Double-Click** - Press the mouse button very quickly
- **Drag** - Click your mouse button and hold as you move the mouse pointer to a new location
- **Drop** - Release your mouse button to "let go" of an item that you are dragging

Teaching Guide

Warm Up (10 min)

Behaving in the Computer Lab

Review expectations and how to behave when they enter the computer lab.

Discuss:

Have a good discussion around your computer lab expectations to make sure that students understand the rules. Some topics of discussion might include:

- Is running in the computer lab okay?
- How loudly should we walk when we are in the computer lab?
- What should you do if you get stuck on a puzzle?
- If you get frustrated, will it help to hit the computer?
- When we're about to go to the computer lab, how should we get ready?

Some possible things to cover:

- Use calm bodies in the lab
- Remember not to chew gum or candy
- Sanitize your hands
- Sit with your partner at one computer
- Make sure that the first "driver" can reach the mouse
- When you get frustrated, don't hit or shake the computer or monitor
- Follow the **20/20/20 - Website** rule
- How to deal with the **Wiggles** every 20-30 minutes (requires a free login on GoNoodle)
- Ask your partner before you ask the teacher
- Keep volume down so everyone else can hear their partners
- Use your journal for keeping track of feelings and solutions

Bridging Activity - Drag and Drop (10 - 15 min)

This activity will help bring the unplugged concepts from Happy Maps into the online world that the students are moving into. Choose **one** of the following to do with your class:

Choose **one** of the following to do with your class:

Dragging and Dropping Algorithms

Project one of the maps from the "Happy Maps" activity and display it for the students to see. On a projector or in front of the class, put some direction blocks from the **Unplugged Blockly Blocks (Grades K-1) - Manipulatives** in random order and practice "dragging and dropping" by pressing your finger on one of the paper pieces and moving it across the screen. Explain that you can "click" to select this block by tapping your finger on it, or you can "drag" the block by pressing your finger on it and moving it. To "drop" the block, release your finger.

After showing this to the class, ask for volunteers to create an algorithm for the Happy Map by "dragging and dropping" the necessary blocks.

Previewing Online Puzzles as a Class

Project a puzzle from the lesson. Show the class how to click on a block and place it in the correct spot by dragging and dropping. Purposely make mistakes such as clicking the background or dropping the image before it's at the right spot. Ask for help from volunteers in the class when you run into these problems, and help them use the skills that they developed in the last unplugged lesson to make things right.

Main Activity (20 - 30 min)

Online Puzzles

This will teach students how to use Code.org to complete online puzzles.

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

Teacher Tip

Show the students the right way to help classmates:

- 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

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 whenever possible during the activity.
- 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.

Code Studio levels

Maze Intro: Programming with Blocks

1 *(click tabs to see student view)*

Practice

2 | 3 | 4 | 5 | 6 | 7 *(click tabs to see student view)*

Wrap Up (5 - 10 min)

Journaling

Give the students a journal prompt to help them process some of the things that they encountered during the day.

Journal Prompts:

- Can you draw a sequence for getting ready to go to the computer lab?
- Draw a computer lab "Do" and a "Don't"
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.

Extension Activities

If students complete the puzzles from this lesson early, have them spend some time trying to come up with their own puzzles in their **Think Spot Journal - Reflection Journal**.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 5: Programming with Scrat

Overview

Using characters from the Ice Age, students will develop sequential algorithms to move Scrat from one side of a maze to the acorn at the other side. To do this they will stack code blocks together in a linear sequence, making them move straight, turn left, or turn right.

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.

Agenda

Warm Up: The Unplugged Foundation (3 min)
- **Review Unplugged Activity**

Online Foundation: Preview Programming in Maze (3 min)

Main Activity (30 min)
- **Online Puzzles**

Wrap Up (5 - 10 min)
- **Journaling**

Extended Learning

View on Code Studio

Objectives

Students will be able to:

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

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 **Think Spot Journal - Reflection Journal**.

Links

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

For the Students

- **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: The Unplugged Foundation (3 min)

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.

Display: Show students 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?
- How about the "East" arrow?

Blend in some context from the story "Unspotted Bugs" as well.

- 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?

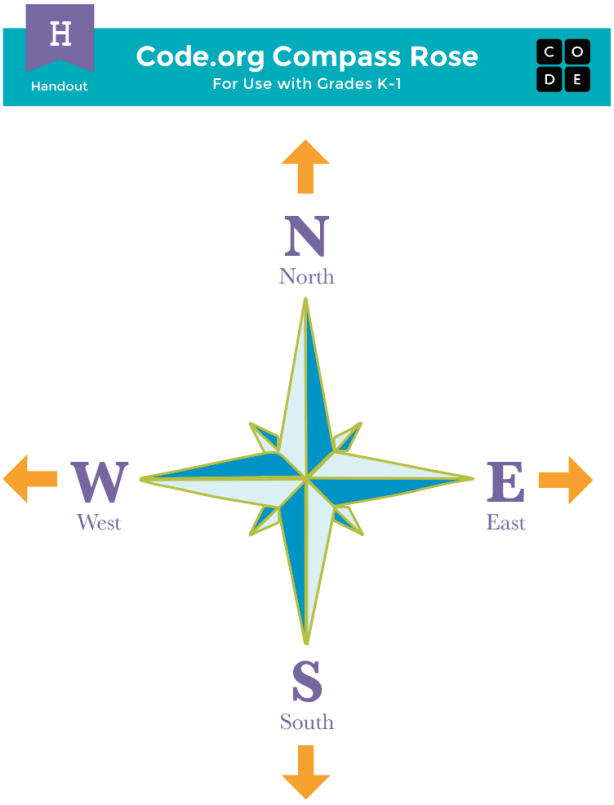
Encourage students to think about the debugging tips:

- 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" and "Unspotted Bugs", you can move into the Bridging Activity.

Teaching Tip

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.

Online Foundation: Preview Programming in Maze (3 min)

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

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 min)

Online Puzzles

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 machines.

Prediction Levels:

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

Code Studio levels

Pair Programming

1 *(click tabs to see student view)*

Practice

2 | 3 | 4 | 5 *(click tabs to see student view)*

Debugging with the Step Button

6 *(click tabs to see student view)*

Practice

7 | 8 *(click tabs to see student view)*

Challenge

9 *(click tabs to see student view)*

Practice

10 | 11 | 12 *(click tabs to see student view)*

Levels

Extra | Extra *(click tabs to see student view)*

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:

Teacher Tip:

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

- 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

- 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 - 10 min)

Journaling

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 - Emotion Images** 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 to get Scrat to the acorn?

Extended Learning

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.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 6: Programming with Rey and BB-8

Overview

In this lesson, students will use their newfound programming skills in more complicated ways to navigate a tricky course with BB-8.

Purpose

With transfer of knowledge in mind, this lesson gives students a new environment to practice the skills that they have been cultivating. Star Wars fans will jump for joy when they see these puzzles. Each puzzle in this series has been added to provide a deeper understanding of the basic concepts that they will be using throughout the rest of this course.

Agenda

Warm Up (15 min)
- **Introduction**

Main Activity (30 min)
- **Online Puzzles**

Wrap Up (15 min)
- **Journaling**

View on Code Studio

Objectives

Students will be able to:

- Sequence commands in a logical order.
- Recognize problems or "bugs" in a program and develop a plan to resolve the issues.

Preparation

- [ ] Play through the puzzles corresponding with this lesson to find any potential problem areas for your class.
- [ ] Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.
- [ ] Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

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

For the Students

- **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 (15 min)

Introduction

Ask the students how they felt about the last lesson.

- Which puzzles were too hard or too easy?
- Which puzzles were frustrating or a lot of fun?
- If they were to teach the lesson to a friend, which part of the lesson would they want to review?

Use these questions to form a brief review of programming and debugging. If you think the class could benefit from it, you can go over the vocabulary words and definitions from the last lesson.

If you feel comfortable, also give a brief introduction to BB-8 from Star Wars. Many students may already be familiar with the lovable robot, but the introduction will surely build excitement.

Main Activity (30 min)

Online Puzzles

Code Studio levels

Programming with Rey and BB-8 1 *(click tabs to see student view)*

Practice 2 3 4 5 6 7 8 *(click tabs to see student view)*

Challenge 9 *(click tabs to see student view)*

Practice 10 11 12 *(click tabs to see student view)*

As we mentioned in the last lesson, we highly recommend viewing and using **Pair Programming - Student Video** as a class. Pair programming stimulates a discussion that can answer questions, review basic concepts, and build confidence with the subject.

Wrap Up (15 min)

Journaling

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:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw a picture of BB-8 you guided through the maze today and add a list of the commands that you used.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming

Lesson 7: Happy Loops

Overview

This activity revisits Happy Maps. This time, student will be solving bigger, longer puzzles with their code, leading them to see utility in structures that let them write longer code in an easier way.

Purpose

This lesson serves as an introduction to loops. Loops allow for students to simplify their code by grouping commands that need to be repeated. Students will develop critical thinking skills by noticing repetition in movements of their classmates and determining how many times to repeat commands. By seeing "Happy Maps" again, students will get the chance to relate old concepts such as sequencing to the new concept of repeat loops.

Agenda

Warm-Up (5 min)
- **Happy Maps Review**

Main Activity (20 min)
- **Happy Loops**

Wrap Up (8 min)
- **Journaling**

Extension Activities

View on Code Studio

Objectives

Students will be able to:

- Identify repeating code and shorten multiple actions into a single loop.
- Interpret a program with loops as a series of multiple actions.

Preparation

- [ ] Print out a **Happy Map Cards - Worksheet** to display for the class
- [ ] Per each group of 2-3, print one of each: **Happy Map Game Pieces - Manipulatives**, **Happy Map Cards XL - Worksheet**, and **Happy Map Game Pieces Bonus Pack - Manipulatives**.
- [ ] Make sure each student has a**Think Spot Journal - Reflection Journal**.

Links

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

For the Students

- **Happy Map Cards** - Worksheet **Make a Copy**
- **Happy Map Cards XL** - Worksheet **Make a Copy**
- **Happy Map Game Pieces** - Manipulatives **Make a Copy**
- **Happy Map Game Pieces Bonus Pack** - Manipulatives **Make a Copy**

Vocabulary

- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again

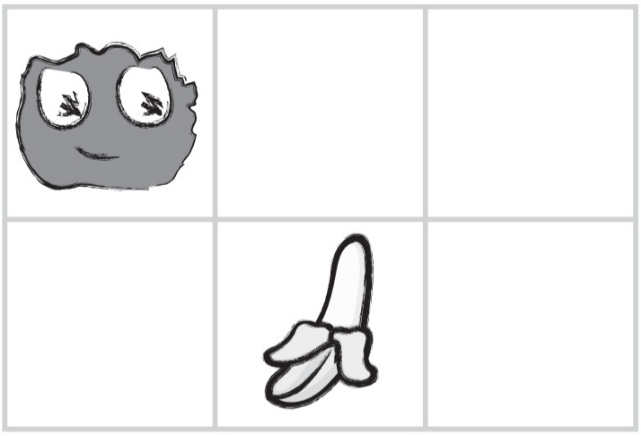
Teaching Guide

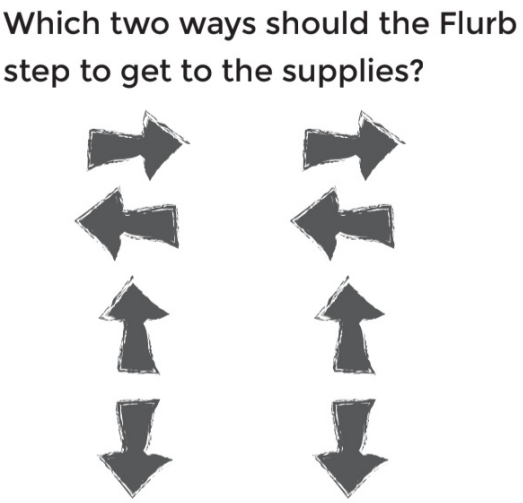
Warm-Up (5 min)

Happy Maps Review

This lesson builds off of the Happy Maps actvity from earlier in the year. Students might benefit from a quick refresher before you hop into the difficult stuff.

Display:





Model: Select one of the maps that your class covered together in the last lesson.

Have students look at the puzzle, then think-pair-share their solution for how they would get the Flurb to the fruit (using the symbols from last time).

Think: Do you remember writing programs to get the Flurbs to the fruit? What would this program look like coded with arrows?

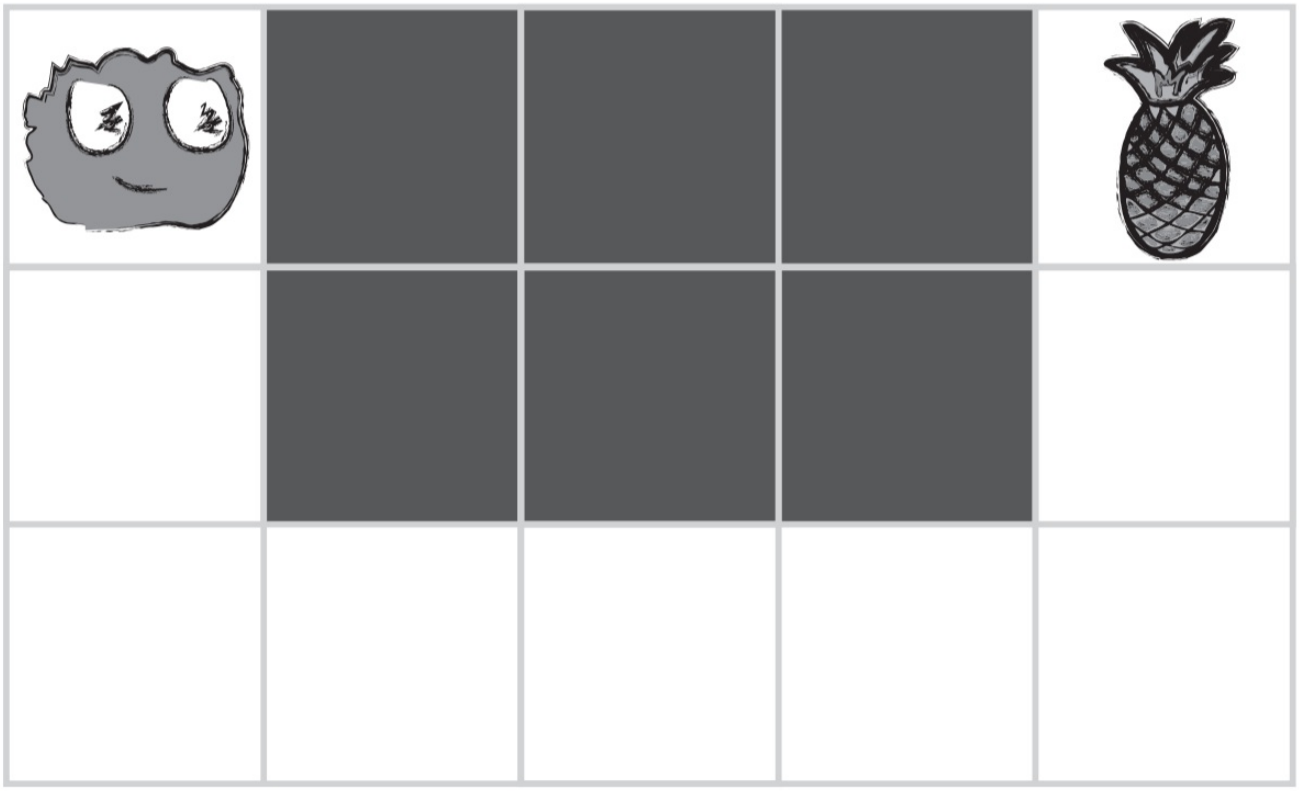
Pair: Have students discuss for about 90 seconds.

Share: Ask a student to use their fingers to point in the direction of the arrows that they chose to solve this puzzle.

Main Activity (20 min)

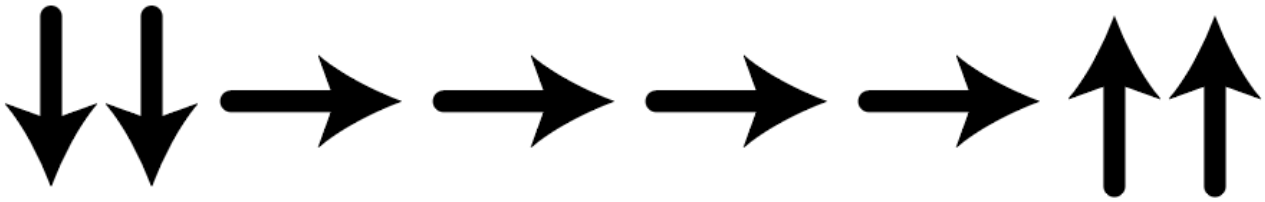
This portion of the lesson should help students see that there is an easier way to handle repetitive code than to brute force a solution with dozens of the same symbols.

Display:



Model: Once you are confident that your students remember Happy Maps, pull up one of the new -- and much longer -- **Happy Maps XL**.

Can your students help you program these maps? The resulting code might look something like this:



It's a bit longer, isn't it?

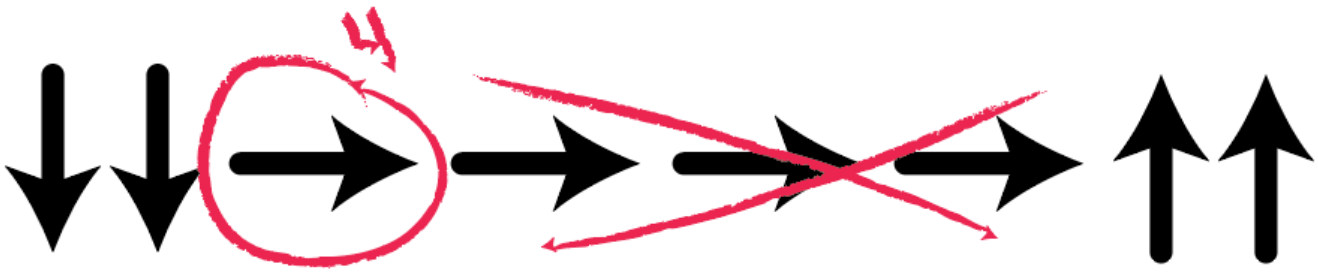
Discuss:

Give students the opportunity to brainstorm shorter ways to relay the code that they're creating. (This bit can be skipped over if your students start saying things like: "Move forward 6 times." Since that will open the discussion about how to show "six times" with symbols.)

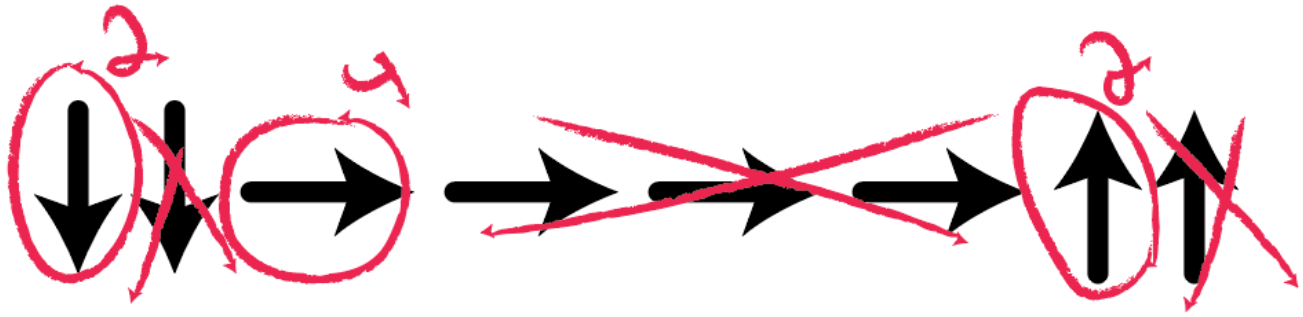
Discussion

The point of this discussion is to get students to see that, sometimes, many symbols are repeated and they can lump all of those movements into a single icon.

Ideally, students will land on a method that looks like this one:



Which can be reduced even further to this:



Eventually, students can write programs like this on the fly:

Define: Once students have put together the idea of “repeating” code, give them the vocabulary around it. Make sure to share with them that often the terms “repeat something” and “loop something” will be used interchangeably in Code Studio.

Happy Loops

Now that students are familiar with the ability to repeat lots of code using a single loop, select an XL map and let them help you write code for the situation. Do this as many times together as a class as you need, then set students off in groups to solve some problems on their own. You may also need to add the **Happy Map Game Pieces Bonus Pack - Manipulatives** to adapt this activity for loops.

Circulate: Make sure to walk around and have students run through their code with you watching. Are there any bugs? Use the debugging questions to help them find a solution.

- What does it do?
- What is it supposed to do?
- What does that tell you?
- Does it work at the first step?
- Does it work at the second step?
- Where does it stop working

Wrap Up (8 min)

Journaling

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 - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Have the students write or draw something in their journal that will remind them later what loops are. Prompts include:

- What does "repeat" mean to you?
- Draw a picture of you repeating something.

Extension Activities

- Create a life-size grid on the rug with tape and have student bring stuffies to school. Now students can program friends to move their actual stuffies as directed in the programs.
- Have students create their own maps for other students to solve using loops.
- Draw a program on the board that uses several sets of repeated commands and have students take turns coming to the front to swap symbols for repeat loops.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 8: Loops with Scrat

Overview

Building on the concept of repeating instructions from "Happy Loops," this stage will have students using loops to get to the acorn more efficiently on Code.org.

Purpose

In this lesson, students will be learning more about loops and how to implement them in Blockly code. Using **loops** is an important skill in programming because manually repeating commands is tedious and inefficient. With these Code.org puzzles, students will learn to add instructions to existing loops, gather repeated code into loops, and recognize patterns that need to be repeated.

Agenda

Warm Up - The Unplugged Foundation (10 min)
- **Review Unplugged Activity**

Bridging Activity - Choose One
- **Unplugged Activity Using Paper Blocks**
- **Previewing Online Puzzles as a Class**

Online Foundation: Preview Loops in Ice Age

Main Activity (30 min)
- **Online Puzzles**

Wrap Up (5 - 10 min)
- **Journaling**

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Construct a program using structures that repeat areas of code
- Improve existing code by finding areas of repetition and moving them into looping structures

Preparation

- [ ] Review the previous unplugged lesson and develop questions to remind students why loops are used.
- [ ] (Optional) Pick a couple of puzzles to do as a group with your class.
- [ ] Gather supplies from previous Happy Loops to reuse for warm up
- [ ] Make sure each student has a journal.

Links

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

For the Students

- **Feeling Faces** - Emotion Images **Make a Copy**
- **Happy Map Cards** - Worksheet **Make a Copy**
- **Happy Map Game Pieces Bonus Pack** - Manipulatives **Make a Copy**
- **Happy Map Game Pieces** - Manipulatives **Make a Copy**
- **Unplugged Blockly Blocks (Grades K-1)** - Manipulatives

Vocabulary

- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again

Teaching Guide

Warm Up - The Unplugged Foundation (10 min)

Review Unplugged Activity

This lesson relies on the concept of repeat loops that students learned in the previous unplugged activity, Happy Loops. It is important to bring this idea from the real world into digital form so that students understand how to use Blockly blocks to repeat a task multiple times.

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

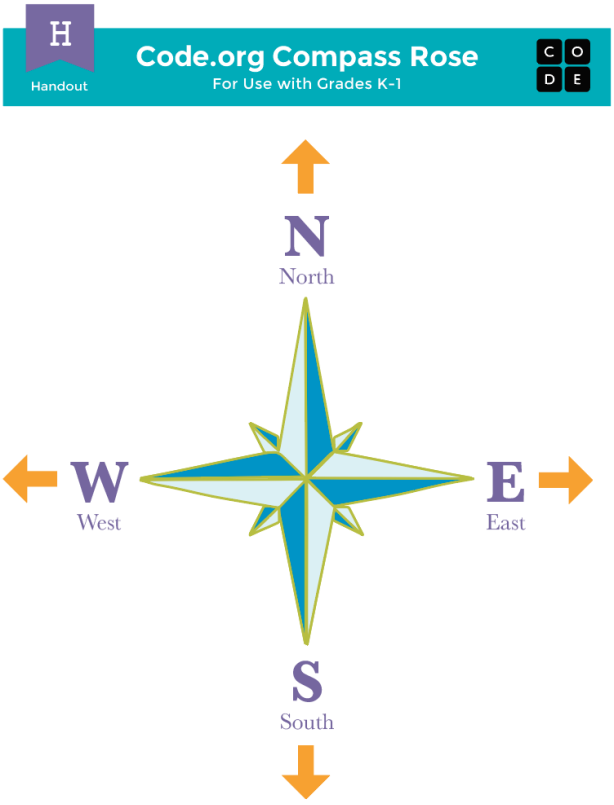
Discuss: Ask students to recall the symbols used in "Happy Loops."

- What happens when "East" arrow is circled with the number 3? (It moves E 3 times)
- What is it called when we circle an arrow and add a number? (A repeat loop)

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

Teaching Tip

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.

Bridging Activity - Choose One

Choose **one** of the following to do with your class:

Unplugged Activity Using Paper Blocks

Select an empty Flurb map from the**Happy Map Cards - Worksheet** and give students **Unplugged Blockly Blocks (Grades K-1) - Manipulatives** prefilled with the collect command, a repeat loop, and the cardinal commands like E → (East) and W ← (West). Now, have the students program the Flurb from their desks using the paper Blockly blocks to get the Flurbs to collect the fruit. Make sure that they understand that the blocks need to go from top to bottom and they all need to touch!

Teacher Tip:

If you predict that your students will have trouble with the idea of using repeat loops to not only move, but also pick corn, you can introduce this idea in the Bridging Activity. This will help students understand that loops can have many different uses.

Previewing Online Puzzles as a Class

Pull a puzzle from the corresponding online puzzles. We recommend puzzle 4. Using arrows from the**Happy Map Game Pieces - Manipulatives** and **Happy Map Game Pieces Bonus Pack - Manipulatives** , have students lay out a pattern that they think will get the Harvester to pick all the corn. Ask the students to share. See how many other students had the same answer!

Online Foundation: Preview Loops in Ice Age

To finish the connection, preview an online puzzle (or two) as a class.

Model: Reveal an entire online puzzle from the progression to come. We recommend Lesson 8, Puzzle 5. Point out the "Play Area" with Scrat and the acorn, as well as the "Work Space" with the Blockly code. Explain that this Blockly code is now the language that the class will be using to help Scrat get to the acorn. Do students 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 min)

Online Puzzles

As students work through the puzzles, see if they can figure out how many blocks they use with a loop vs. without a loop.

Code Studio levels

Practice 1 2 *(click tabs to see student view)*

Ice Age Loops 3 *(click tabs to see student view)*

Practice 4 5 6 7 8 9 *(click tabs to see student view)*

Challenge 10 *(click tabs to see student view)*

Practice 11 12 *(click tabs to see student view)*

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:

Teacher Tip:

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

- 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

- 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 - 10 min)

Journaling

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:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw Scrat and an acorn.
- Draw yourself using a loop to do an everyday activity, like brushing your teeth.

Extended Learning

So Moving

- Give the students pictures of actions or dance moves that they can do.
  - Have students arrange moves and add loops to choreograph their own dance.
- Share the dances with the rest of the class.

Connect It Back

- Find some YouTube videos of popular dances that repeat themselves.
- Can your class find the loops?
- Try the same thing with songs!

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 9: Loops with Laurel

Overview

In this lesson, students continue learning the concept of loops. In the previous lesson, students were introduced to loops by moving through a maze and picking corn. Here, loops are used to collect treasure in open cave spaces.

Purpose

This lesson gives students more practice with loops and introduces a new block, `get treasure`. The block works just like `pick corn` did in Harvester. These puzzles are more open, giving students more flexibility for their final solutions.

Agenda

Warm Up (10 min)
- **Introduction**

Main Activity (30 min)
- **Online Puzzles**

Wrap Up (5 - 10 min)
- **Journaling**

View on Code Studio

Objectives

Students will be able to:

- Identify the benefits of using a loop structure instead of manual repetition.
- Break down a long sequence of instructions into the smallest repeatable sequence possible.

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.
- [ ] Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**
- [ ] Make sure each student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Students

- **Unplugged Blockly Blocks (Grades K-1)** - Manipulatives

Vocabulary

- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again

Teaching Guide

Warm Up (10 min)

Introduction

Quickly review the definition of a loop, the action of doing something over and over again.

- What are loops?
- Why do we use them?

Main Activity (30 min)

Online Puzzles

Teacher Demonstration

We've included some multiple choice prediction levels that are difficult for non-readers. These levels are optional for you to review with your class to help prepare for the puzzles to come. Alternatively, these could be used after finishing the stage as a review for the class.

Prediction Levels:

- **Course A, Loops in Collector #1**
- **Course A, Loops in Collector #2**

Lesson Tip:

Some students may be curious about what happens when you add more blocks inside a repeat loop. Make sure to explain that the repeat loop goes through every block inside it once, then starts back over from the top and repeats that. It does not just repeat each block inside that many times and move to the next one.

Code Studio levels

The Collector 1 *(click tabs to see student view)*

Practice 2 3 *(click tabs to see student view)*

Using the Repeat Block 4 *(click tabs to see student view)*

Practice 5 6 7 8 9 10 *(click tabs to see student view)*

Challenge 11 *(click tabs to see student view)*

Practice 12 13 14 *(click tabs to see student view)*

Levels Extra Extra *(click tabs to see student view)*

As students work through the puzzles, see if they can figure out how many blocks they use with a loop vs. without a loop.

Wrap Up (5 - 10 min)

Journaling

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:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- How did loops make your program easier to write?
- Draw something that uses loops.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 10: Ocean Scene with Loops

Overview

Returning to loops, students learn to draw images by looping simple sequences of instructions. In the previous plugged lesson, loops were used to traverse a maze and collect treasure. Here, loops are creating patterns. At the end of this stage, students will be given the opportunity to create their own images using loops.

Purpose

This lesson gives a different perspective on how loops can create things in programming. Students can also reflect on the inefficiency of programming without loops here because of how many blocks the program would require without the help of `repeat` loops.

Agenda

Warm Up (10 min)
- **Introduction**

Main Activity (30 min)
- **Online Puzzles**

Wrap Up (5 - 10 min)
- **Journaling**

View on Code Studio

Objectives

Students will be able to:

- Count the number of times an action should be repeated and represent it as a loop.
- Decompose a shape into its largest repeatable sequence.
- Create a program that draws complex shapes by repeating simple sequences.

Preparation

- [ ] Play through the puzzles to find any potential problem areas for your class.
- [ ] Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.
- [ ] Make sure each student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teachers

- **Pause and Think Online** - Video

Vocabulary

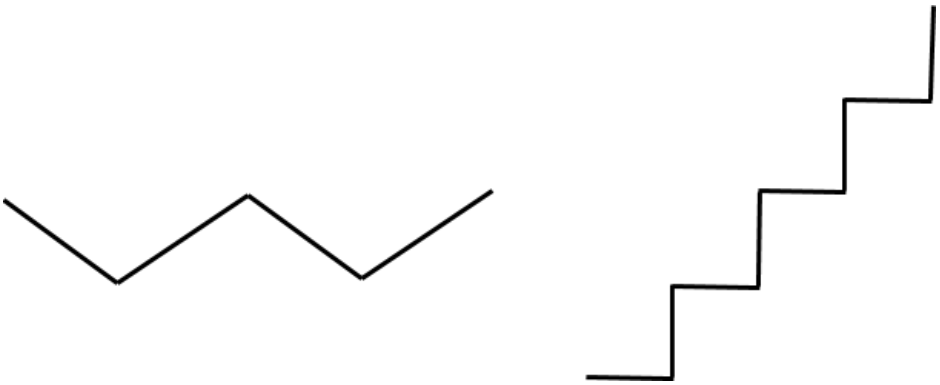
- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again

Teaching Guide

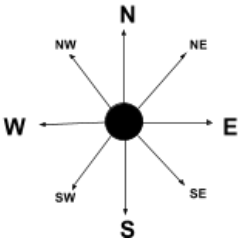
Warm Up (10 min)

Introduction

- Quickly review the definition of a loop, the action of doing something over and over again.
- Discuss different patterns like zigzags and stairsteps.
  - How would you explain to someone how to draw that pattern?
  - How could you draw this using a loop?



In the artist levels students will be using 45 degree angles described as northwest, northeast, southwest, southeast. We recommend briefly discussing these directions with the class and drawing an image for students to refer back to.



Main Activity (30 min)

Online Puzzles

Teacher Demonstration

We've included some multiple choice prediction levels that are difficult for non-readers. These levels are optional for you to review with your class to help prepare for the puzzles to come. Alternatively, these could be used after finishing the stage as a review for the class.

Teacher Tip

Remind the students to only share their work with their close friends or family. For more information watch or show the class **Pause and Think Online - Video** .

Prediction Levels:

- **Course A, Loops in Artist**

Code Studio levels

The Artist in Code Studio

1 *(click tabs to see student view)*

Practice

2 3 4 5 6 *(click tabs to see student view)*

Repeat Blocks with the Artist 7 *(click tabs to see student view)*

Practice 8 9 10 *(click tabs to see student view)*

Challenge 11 *(click tabs to see student view)*

Practice 12 13 *(click tabs to see student view)*

Free Play 14 *(click tabs to see student view)*

Levels Extra Extra *(click tabs to see student view)*

Wrap Up (5 - 10 min)

Journaling

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:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw some stairs. Imagine the loop needed to draw this.
- Draw something else in your life that uses loops.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 11: The Big Event Jr.

Overview

Events are a great way to add variety to a pre-written algorithm. Sometimes you want your program to be able to respond to the user exactly when the user wants it to. That is what events are for.

Purpose

Today, students will learn to distinguish events from actions. The students will see activities interrupted by having a "button" pressed on a paper remote. When seeing this **event**, the class will react with a unique action. Events are widely used in programming and should be easily recognizable after this lesson.

Agenda

Warm Up (15 min)

- **Vocabulary**
- **A Series of Events**

Main Activity (15 min)

- **The Big Event**

Wrap Up (10 min)

- **Reflection**

Assessment (10 min)

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Repeat commands given by an instructor.
- Recognize actions of the teacher as signals to initiate commands.
- Practice differentiating pre-defined actions and event-driven ones.

Preparation

- [ ] Prepare to project **The Big Event (Courses A, B) - Controller Image**.
- [ ] Print one **The Big Event - Assessment** per student.
- [ ] Make sure each student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teachers

- **The Big Event** - Assessment Answer Key **Make a Copy**

For the Students

- **The Big Event** - Unplugged Video (**download**)
- **The Big Event (Courses A, B)** - Controller Image **Make a Copy**
- **The Big Event** - Assessment **Make a Copy**

Vocabulary

- **Event** - An action that causes something to happen.

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has one new and important vocabulary word:

- **Event** - Say it with me: E-vent

An action that causes something to happen.

A Series of Events

- Prep your class to answer a question:
  - "I'm going to ask you a question. I want you to raise your hand if you want me to call on you for the answer."
  - Ask a simple question that most of your students should be able to answer, such as:
    - How many thumbs do I have?
    - What is bigger, a bird or a horse?
  - Call on a student who has their hand raised and let them give their answer.
  - Upon finishing that display, ask the class how you knew that the student wanted you to call on them.
    - Your class will likely mention the raising of the hand.
  - Explain to everyone that when students raise their hand, it is an "event" that causes you to know that they want to be called on.
- Ask the class if they can think of any other events that give signals.
  - You may need to remind them that you're not talking about an event like a birthday party or a field trip.
  - If they have trouble, you can remind them that an event is an action that causes something to happen.
    - What about an alarm clock going off? What does that make happen?
    - What about pressing "Start" on the microwave? What does that do?
    - What about pressing the power button on your tv remote?
- Today, we're going to create programs with events.

Main Activity (15 min)

The Big Event

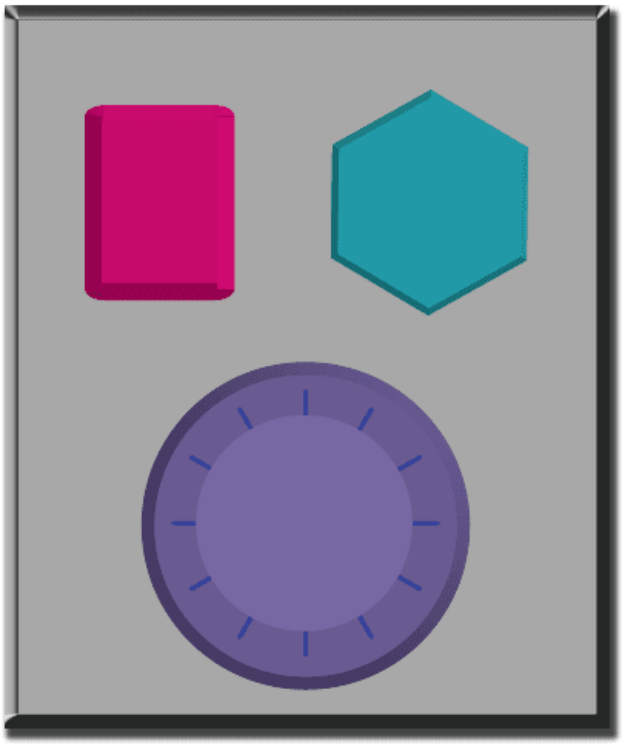
- Do you remember helping the Flurbs find fruit?
  - In that exercise, you knew in advance exactly where you wanted your Flurb to end up, so you could make a program that took them from start to finish without any interruptions.
  - In most real programs, we can't do that because we want to have options, depending on what the user needs.
    - Say that I only want my character to move when my finger is on the screen of my phone. I would need to program the character to only move when I put my finger on the screen of my phone.
    - Putting my finger on the screen would then become an "event" that tells my character to move.

> **Lesson Tip**
>
> If your students seem confused, talk about their favorite games and all of the ways that they let the characters know what they're supposed to do. Point out how the game would be really boring if it ran from start to finish without any events required.

In earlier lessons, we created algorithms that allowed us to control a friend or Flurb for several steps at a time. It was fun and useful, but what happens when you don't know everything that you want your friend to do in advance? This is where events come in!

Directions:

- Project **The Big Event (Courses A, B) - Controller Image** onto your classroom screen.



- Decide with your class what each button does. We suggest:
  - Pink Button -> Say “Wooooo!”
  - Teal Button -> “Yeah!”
  - Purple Dial -> “Boom!”
- Practice tapping the buttons on the overhead and having your class react.
- Add some button sequences into the mix and have the students try to keep up with their sounds.
- Let your class know that every time you push a button, it is an “event” that lets them know what they are expected to do next.
- Get the class started on a planned task before interrupting them again with the buttons. We suggest:
  - Counting to 10
  - Singing “Old MacDonald”
- Once their plan is underway, interject button presses sporadically.
- Continue the blend until they understand the difference between actions that are guided by a plan and those that are event driven.

Wrap Up (10 min)

Reflection

Discuss: Ask students to reflect on what they have learned through the following prompts:

- Why do we need to be able to handle events in a program?
- What are some other kinds of events that you can think of?

Journal: In their **Think Spot Journals**, ask students to write and draw with the following questions in mind:

- What was today’s lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw an event that caused an action today.
- Draw an action that was caused by an event that happened today.

Assessment (10 min)

Distribute: Hand out one **The Big Event - Assessment** to each student and allow them to complete it independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities.

Extended Learning

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

One Person's Event is Another One's Reaction

Assign each student an event to watch out for, and an appropriate reaction to that event. Chain the actions so that each child's reaction becomes an event that triggers the reaction of another student. Keep assigning until everyone has something to do and everyone makes someone react.

Eventopalooza

Break the class up into groups. Using the Events Controller, assign each group a different reaction to the same button. Do this for all three buttons, then watch the chaos!

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

Lesson 12: On the Move with Events

Overview

In this online activity, students will have the opportunity to learn how to use events in Play Lab and to apply all of the coding skills they've learned to create an animated game. It's time to get creative and make a story in the Play Lab!

Purpose

Students will further develop their understanding of events using Play Lab today. Events are very common in most computer programs. In this activity, students will use events to make a character move around the screen, make noises, and change backgrounds based on user-initiated events.

Agenda

Warm Up (10 min)
- **Introduction**

Bridging Activity - Events (10 min)
- **Unplugged Activity Using Paper Blocks**
- **Previewing Online Puzzles as a Class**

Main Activity (30 min)
- **Online Puzzles and Free Play**

Wrap Up (5 - 10 min)
- **Journaling**

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Identify actions that correlate to input events.
- Create an animated, interactive story using sequence and event-handlers.
- Share a creative artifact with other students.

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.
- [ ] Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.
- [ ] Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

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

For the Teachers

- **Pause and Think Online** - Video
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations **Make a Copy**

For the Students

- **The Big Event (Courses A, B)** - Controller Image **Make a Copy**
- **Unplugged Blockly Blocks (Grades K-1)** - Manipulatives

Vocabulary

- **Event** - An action that causes something to happen.

Teaching Guide

Warm Up (10 min)

Introduction

Review "The Big Event" activity with students:

- What did we "program" the button events to do?

Now we're going to add events to our code. Specifically, we're going to have an event for when two characters touch each other.

- When have you seen two characters touch each other as an event in games?

Bridging Activity - Events (10 min)

This activity will help bring the unplugged concepts from "The Big Event" into the online world that the students are moving into. Choose **one** of the following to do with your class:

Unplugged Activity Using Paper Blocks

Using the remote from the **The Big Event (Courses A, B) - Controller Image** and blocks from **Unplugged Blockly Blocks (Grades K-1) - Manipulatives**, gather your class to reprise the activity from the previous lesson. Ask the class "when the teal button is pushed, what do we do?" then fill in one of the `when` event blocks and one of the blue action blocks accordingly. Make sure that the students understand that the `when` blocks need to be on top of the blue block and they need to touch in order for the program to run.

Lesson Tip

Students will have the opportunity to share their final product with a link. This is a great opportunity to show your school community the great things your students are doing. Collect all of the links and keep them on your class website for all to see!

Remind the students to only share their work with their close friends or family. For more information watch or show the class **Pause and Think Online - Video** .

Previewing Online Puzzles as a Class

Pull a puzzle from the corresponding puzzles online. We recommend puzzle 7. Call on different students to make a funny noise when you click on Jorge. Explain this is an event that they are reacting to and Jorge can be coded to make noise when you click on him.

Main Activity (30 min)

Online Puzzles and Free Play

This is the most free-form plugged activity of the course. At the final stage students have the freedom to create a story of their own. You may want to provide structured guidelines around what kind of story to write, particularly for students who are overwhelmed by too many options.

Code Studio levels

Create a Story 1 *(click tabs to see student view)*

Free Play 2 *(click tabs to see student view)*

Practice 3 *(click tabs to see student view)*

Mini-project: Jorge the Dog 4 5 6 7 8

(click tabs to see student view)

Levels Extra Extra *(click tabs to see student view)*

Wrap Up (5 - 10 min)

Journaling

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:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw an event you used in your program today.
- Imagine that you have a remote controlled robot. What would the remote look like? Draw a picture of what you think you could make the robot do.

Extended Learning

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

Look Under the Hood

When you share a link to your story, you also share all of the code that goes behind it. This is a great way for students to learn from each other.

- Post links to completed stories online.
  - Make a story of your own to share as well!
- When students load up a link, have them click the "How it Works" button to see the code behind the story.
- Discuss as a group the different ways your classmates coded their stories.
  - What surprised you?
  - What would you like to try?
- Choose someone else's story and click Remix to build on it. (Don't worry, the original story will be safe.)

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming