Pre-Express Course

Pre-readers and early readers can learn the basics of computer science and internet safety.

Teacher Links: **Teacher Videos Playlist**

Lesson 1: Debugging: Unspotted Bugs

Unplugged | Bug | Debugging | Persistence

Lesson 2: Persistence & Frustration: Stevie and the Big Project

Unplugged | Fail | Frustrated | Persistence

Lesson 3: Real-Life Algorithms: Plant a Seed

Unplugged | Algorithms

Lesson 4: Learn to Drag and Drop

Click | Double-Click | Drag | Drop | Pair Programming

Lesson 5: Common Sense Education: Your Digital Footprint

Common Sense Education | Unplugged

Lesson 6: Programming Unplugged: My Robotic Friends

Algorithms | Debugging | Unplugged

Using a set of symbols in place of code, students will design algorithms to instruct a "robot" to stack cups in different patterns. Students will take turns participating as the robot, responding only to the algorithm defined by their peers. This segment teaches students the connection between symbols and actions, the difference between an algorithm and a program, and the valuable skill of debugging.

Lesson 7: Programming in Maze

Algorithms | Debugging | Program | Programming

Lesson 8: Programming in Star Wars

Programming | Maze

Lesson 9: My Loopy Robotic Friends

Unplugged | Loop | Repeat

Building on the initial "My Robotic Friends" activity, students tackle larger and more complicated designs. In order to program their "robots" to complete these bigger designs, students will need to identify repeated patterns in their instructions that could be replaced with a loop.

Lesson 10: Loops in Collector

Loop | Collector

Lesson 11: Loops in Artist

Loop | Artist

Lesson 12: Events Unplugged: The Big Event

Event | Unplugged

Lesson 13: Events in Play Lab

Event | Play Lab

Lesson 14: Spelling Bee

Lesson 1: Debugging: Unspotted Bugs

Overview

This lesson will guide students through the steps of debugging. Students will learn the mantra: "What happened? What was supposed to happen? What does that tell you?"

Purpose

Research shows that some students have less trouble debugging a program than writing one when they first learn to code. In this lesson, we introduce the idea of debugging in a real world sense.

The goal in this lesson is to teach students steps to spot a bug and to increase persistence by showing them that it's normal to find mistakes. In later lessons, students will debug actual programs on Code.org.

Agenda

Warm Up (12 min)

- **Unspotted Bugs Vocabulary**

Marble Run Breakdown (10 - 20 min)

- **Debug the Run**

Wrap Up (10 - 20 min)

- **Journaling**

Extended Learning

- **Real Life Bug Hunting**

View on Code Studio

Objectives

Students will be able to:

- Express that they have noticed when something goes differently than what is expected.
- Identify what the expected result was before an error occurs.
- Determine and describe the difference between what was expected and what actually happened in the event of an error.

Preparation

- [ ] Watch the **Unspotted Bugs - Teacher Video**.
- [ ] Review the Unspotted Bugs Story (**resource unspotted-bugs-pdf not found**).
- [ ] Pre-read Unspotted Bugs to identify appropriate questions for your classroom.
- [ ] Follow instructions in the **Marble Run - Teacher Prep Guide** to make a Marble Run (which will be arranged incorrectly at the start).
- [ ] Give a **Think Spot Journal - Reflection Journal** to each student.

Links

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

For the Teachers

- **Unspotted Bugs** - Teacher Video
- **Marble Run** - Teacher Prep Guide

For the Students

- **Unspotted Bugs** - Storybook
- **First Computer Bug** - Student Video
- **Think Spot Journal** - Reflection Journal

Make a Copy

Vocabulary

- **Bug** - Part of a program that does not work correctly.

- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Persistence** - Trying again and again, even when something is very hard.

Teaching Guide

Warm Up (12 min)

Goal: Help students understand the steps involved in debugging.

Unspotted Bugs

This story can be presented in several ways, including:

- Circled up story time
- Projected with document camera / smartboard
- Pair shared with students at their computers

The story of Unspotted Bugs presents many of the ideas that students will need to understand the debugging process of coding. This warm-up is meant to tie a memorable story together with a concept that young kids often find to be difficult.

Read the book and discuss the techniques that JD used to discover and take care of bugs. Make sure those questions and tactics get repeated often enough that students can recall (if not recite) them without the story in hand.

Lesson Tip

Important ideas from the story:

- What happened?
- What was supposed to happen?
- What does that tell you?
- Did it work at the first step?
- Did it work at the second step?
- Where did it go wrong?

Potential Questions for Storytime:

- Page 3: What do you notice in the picture? What's wrong with the flower? (It's upside down!) What's wrong with the clock? (The hands aren't in the center) Why do you think there is something wrong with these items? (Because there are bugs on them!)
- Page 7: What's wrong with the picture? (The lamp is upside down) Why is that? (There's a bug)
- Page 11: What's wrong in this scene? (The car doesn't have wheels!) Why? (Because there are bugs on it!)
- What did JD find when he went looking for the bug? What was wrong? What does this mean? (JD found an upside down tree. This is wrong because the tree trunk should be touching the ground! This means there is a bug on the tree!)

Vocabulary

This lesson has three new and important vocabulary words:

- *Bug* - Say it with me - Buhh-g. Something that is going wrong. An error.
- *Debugging* - Say it with me: Dee-bug-ing. To find and fix errors.
- *Persistence* - Say it with me: Purr-siss-tense. Not giving up. Persistence works best when you try things many different ways, many different times.

Marble Run Breakdown (10 - 20 min)

Goal: Help students think critically about the difference between what is happening and what is expected.

Debug the Run

Now that students have been introduced to the idea of looking for problems, they can try to apply it to more places in the real world. This next activity gives them practice looking for bugs in Marble Runs (a project that they will be working with next week.)

Grab your sample marble run (built from our plans, or something similar.) Show the students how each piece works, then demonstrate putting them together (but put them together incorrectly, to prevent the ball from flowing properly from A to B.

The goal of this exercise is to help the students identify when something goes wrong, so if they don't catch it the first time, run it again, and again. It can help to make exaggerated frustration faces when the ball doesn't do what you would like it to do.

Let the students share hypotheses about what is going wrong, and how to fix it. Students should feel free to try things that you know will be incorrect. If students misidentify solutions, use the bug finding formula on their configurations. Repeat until you get a working run.

Encouragement is key here. If things don't work right away, praise the class for being so persistent and choosing not to give up. If they start to get frustrated, encourage them to persist a bit longer, promising them that they will get it soon if they just hang in there.

Lesson Tip

Say:

Great! You all are so good at this, maybe you can help me with my own problem!

See, I have this marble run that I made. It comes in two pieces. When I put the ball in here (input A) it's supposed to come out here (output A). When I put the ball in here (input B) it's supposed to come out here (output B). Now, when I slide them together, I should be able to put the ball in here (input A) and have it come out here (output B). But it doesn't work, watch.

[Slide the pieces together with output B facing output A.]

Watch what happens. [Drop ball at input A and notice that it does not come out output B.]

- BUG!

What happened?

- The ball fell on the table.

What was supposed to happen?

- The ball was supposed to drop from A into B.

What does that tell you?

- You should turn B around so that the ball goes into the right place!

Wrap Up (10 - 20 min)

Journaling

Goal: Students will start to understand the importance of the activity they just completed by reflecting on it verbally, then through drawing in their journals.

Clear your mind:

It can be distracting to a learner when they have unanswered questions or doubts. To end this lesson, we'll give everyone the chance to get those out so that they can reflect on what they've been taught.

Encourage students to share their thoughts and questions either with the whole class or with an elbow partner.

Lesson Tip

Say:

What do you think we learned in this lesson?

- Debugging
- How to solve a problem
- How to make a marble go
- How do you think that can help us in other places?

Journal Prompts:

Once they've had time to ponder their own thoughts, get the students thinking about the purpose of the lesson that they just learned. Why did you do this activity? How will it help them later? Can they think of buggy things that they've seen in the real world?

Students should finish by drawing or writing in their journal. Possible topics include:

- How do you feel when something that you are working on acts buggy?
- How many times do you think you should try to fix a bug before you give up?
- What would you do if you notice that something is buggy, but you don't know how to fix it?

Extended Learning

Real Life Bug Hunting

Take your students outside. Do you see any signs of bugs? What are they? Now look closer... can you find the actual bug?

Lesson Tip:

The signs of real-live bugs won't be as dramatic as upside down trees, but it might be dead leaves, spots on flowers, or slime on the sidewalk. Have the students brainstorm these before going outside to look for them.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming

Lesson 2: Persistence & Frustration: Stevie and the Big Project

Overview

When students run into a barrier while answering a question or working on a project, it's so easy for them to get frustrated and give up. This lesson will introduce students to the idea that frustration can be an important part of learning. Here, frustration is presented as a step in the creative process, rather than a sign of failure.

This lesson can be done over one or two class sessions. If you have more time, feel free to draw out the building and revising phase of the Marble Run activity.

Purpose

The goal of this lesson is to help students realize that failure and frustration are common when working on projects, but that doesn't mean that they should give up.

In this lesson, students will develop an understanding of what it means to be frustrated while working on a large project. It's possible that not every student will experience frustration with this activity, but there are many opportunities to open a discussion about moments in the past where students have felt frustrated but nevertheless persisted.

Agenda

Warm Up (15 min)

- **Stevie and the Big Project**
- **Vocabulary**

Marble Run (20 - 45 min)

- **Before the Project:**
- **Building the Marble Run:**
- **After the Marble Run:**

Wrap Up (5 min)

- **Journaling**

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Recognize and point out symptoms of frustration.
- Describe at least one reason why they will choose to be persistent in the face of frustration, rather than giving up.

Preparation

- [ ] Watch the **Stevie and the Big Project - Teacher Video**.
- [ ] Pre-read "Stevie and the Big Project" to identify appropriate questions for your class (**resource stevie-big-project-pdf not found**).
- [ ] Follow instructions in the **Marble Run - Teacher Prep Guide** to make a Marble Run.
- [ ] Print copies of the **Marble Run Ruler** (page 2 of teacher guide) for each student or pair of students
- [ ] Prepare a resource station with cardstock, safety scissors, tape, and anything else you think might be fun for students to build with. Include a stack of the **"Marble Run Hints"** pages from the Teacher Prep Guide, but do not advertise their existence.
- [ ] (Optional) Allow students to bring cardboard, popsicle sticks, string, or other tidbits from home to add to the resource station.
- [ ] 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

- **Stevie and the Big Project** - Teacher Video
- **Marble Run** - Teacher Prep Guide

For the Students

- **Stevie and the Big Project** - Storybook
- **Think Spot Journal** - Reflection Journal

Make a Copy

Vocabulary

- **F.A.I.L.** - First Attempt In Learning
- **Frustrated** - Feeling annoyed or angry because something is not the way you want it.
- **Persistence** - Trying again and again, even when something is very hard.

Teaching Guide

Warm Up (15 min)

Stevie and the Big Project

Goal: Introduce students to the idea that they don't have to give up just because they are frustrated.

This lesson begins with a story. Students will be introduced to several ideas on persistence and frustration through relatable struggles by fictional characters, including the idea that frustration is not a sign that someone should instantly give up.

This book can be presented in several ways, including:

- Circled up story time
- Projected with document camera / smartboard
- Pair share with students at their computers

Use the reading techniques that work in your classroom:

If your students like to discuss things that happen as they appear in the book, be sure to stop your class after large plot areas like when Stevie breaks her structure, or when Laurel explains frustration.

If your students like to sit through a whole story and discuss at the end, read through the book, then prompt their memory with some "Remember when…" type questions.

Lesson Tip

Sample Questions:

- How would you feel if you were given a project that feels much harder than what you are used to?
- Do you think it's okay to try something new, even if it doesn't work out the first time?
- Why do you think Stevie smashed her project?
  - Do you think that helped her or hurt her when it comes to reaching her goal?
  - What do you think Stevie should have done instead of breaking her project?
- Can somebody explain what frustration is?
- How do you think you can know when you are frustrated?
  - What face do you make when you are frustrated?
  - How can you make yourself feel better when you start to get frustrated?
  - We all get frustrated sometimes. Does that mean that we should give up?
- Can someone tell me what persistence is?
  - Why is it hard to learn if you're not persistent?
  - Can you tell me why you might be tempted not to be persistent?
  - What happened when Stevie decided to be persistent?
  - Do you think you can be persistent?

Vocabulary

- *Persistence* - Say it with me: Purr-siss-tense. Not giving up. Persistence works best when you try things many different ways, many different times.
- *Frustrated* - Say it with me: Frus - straight - ted. Feeling annoyed or angry because something is not the way you want it.
- *F.A.I.L.* - First Attempt in learning. When you try to do something, but you don't do it quite right.

Marble Run (20 - 45 min)

This activity is meant to highlight and normalize the feeling of frustration, while giving students a chance to be persistent.

Before the Project:

It is vitally important that students understand that this activity is meant to help them learn about frustration and persistence. This is not one of those times when we allow students to experience something, then give it a name afterward. Students need to know that they will be feeling some emotions, and that those emotions are okay.

Take a moment to relate the next activity back to the book that you just read. The class might be excited that they get to try the same project that Stevie did, but they might also be apprehensive at the thought of tackling something difficult.

Encourage your students to have their Think Spot Journals around during the activity so they can use them to plan, solve, and voice concerns.

Building the Marble Run:

Time to be an engineer!

Break students up into pairs and have them quickly come up with a team name. This should help to unify them in their work.

Next, point out the resource station that you have set up with all of the supplies and goodies that students will have access to. Make sure you are very clear about whether they are limited only to the items in the resource station or whether they are allowed to ask for other items for their creation.

Give students checkpoints for this activity. Make sure that they know that there is no penalty for not finishing on time.

Preplanning is optional, since prediction is not often a kindergartener's strong suit.

The first attempt at building will likely be hectic and a bit sloppy, but it should give students access to the feelings and opportunities for persistence that are being studied in this lesson.

Try to end the Marble Run build with an opportunity for groups to collaborate. This will improve the chances of success for students who have been struggling, without the need for teacher intervention.

After the Marble Run:

Time to do some damage control if any is needed.

Remind students that this activity was planned to teach students how to identify feelings of frustration and work past them to be persistent.

Discuss the difference between being successful for the purpose of this activity, and being successful at building their contraption. Is it possible to have done the first without the second?

Lesson Tip

Say:

Now, we're going to do something very fun, and very challenging! I am going to let you all try to make a Marble Run of your own!

This is **supposed** to be challenging. That's part of the fun! Your Marble Run probably won't work right the first time, and that's alright. The goal for this game is to practice being persistent.

Remember, Stevie showed us that this might be difficult, and sometimes difficult things are frustrating. It is okay if you get frustrated during this activity. Most of us probably will at some point. How should we handle those feelings?

- Count to 10
- Take deep breaths
- Journal about them
- Talk to a partner about them
- Ask for help

Lesson Tip

Checkpoint Suggestions:

- Pre-planning time (3-5 minutes)
- First attempt at building (10-15 minutes) -- For a longer (or two day) time period --
- Discuss with another group (3-5 minutes)
- Revision of structure (10-15 minutes) -- Wrap Up Work --
- Collaborative work time (5-15 minutes)

Teacher Tip

Tears are a very common byproduct when kindergarteners attempt lessons of this nature. You will likely want to have a pre-packaged prescription for students who become emotionally raw.

- Can you put into words what you are feeling right now?
- Stevie would be so proud of you. What do you think Laurel and Jorge would say if you told them how you feel?
- What would it be called if you said out loud that you are frustrated, but decided to keep working anyway?
  - Do you feel like you can be persistent with me today?

Wrap Up (5 min)

Journaling

Allow students to reflect on the emotions and processes experienced during the lesson.

Finish out this lesson by asking students to spend some time in their **Think Spot Journal - Reflection Journal**.

Journal Prompts:

- Draw a picture of what you look like when you're frustrated.
- Draw a picture that shows things you can do to feel better when you're frustrated.

- What does persistence look like?

Extended Learning

- Add a third piece to the beginning of the Marble Run. Can students start a marble up even higher and get it to flow through the rest of their contraption?
- Talking through frustration. Can students think of things that they can say to classmates to help them be persistent when they are frustrated?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 3: Real-Life Algorithms: Plant a Seed

Overview

In this lesson, students will relate the concept of algorithms back to everyday, real-life activities by planting an actual seed. The goal here is to start building the skills to translate real-world situations to online scenarios and vice versa.

Purpose

In this lesson, students will learn that algorithms are everywhere in our daily lives. For example, it is possible to write an algorithm to plant a seed. Instead of giving vague or over-generalized instructions, students will break down a large activity into smaller and more specific commands. From these commands, students must determine a special sequence of instructions that will allow their classmate to plant a seed.

Agenda

Warm Up (10 min)

- **Vocabulary**
- **What We Do Daily**

Main Activity (20 min)

- **Real-Life Algorithms: Plant a Seed - Worksheet**

Wrap Up (10 - 20 min)

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

Assessment (15 min)

- **Real-Life Algorithms: Plant a Seed - Assessment**

Extended Learning

- **Go Figure**

View on Code Studio

Objectives

Students will be able to:

- Decompose large activities into a series of smaller events.
- Arrange sequential events into their logical order.

Preparation

- [ ] Watch the **Plant a Seed - Teacher Video**.
- [ ] Prepare supplies for planting seeds. You'll need seeds, dirt, and paper cups for each student or group.
- [ ] Print one **Real-Life Algorithms: Plant a Seed - Worksheet** for each student.
- [ ] Print one **Real-Life Algorithms: Plant a Seed - Assessment** for each 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

- **Real-Life Algorithms: Planting a Seed** - Unplugged Video (**download**)
- **Plant a Seed** - Teacher Video
- **Real-Life Algorithms: Plant a Seed** - Worksheet
- **Real-Life Algorithms: Plant a Seed** - Worksheet Answer Key
- **Real-Life Algorithms: Plant a Seed** - Assessment
- **Real-Life Algorithms: Plant a Seed** - Assessment Answer Key

For the Students

- **Think Spot Journal** - Reflection Journal

Make a Copy

Teaching Guide

Warm Up (10 min)

Vocabulary

This lesson has one vocabulary word that is important to review:

Algorithm - Say it with me: Al-go-ri-thm

A list of steps that you can follow to finish a task

What We Do Daily

- Ask your students what they did to get ready for school this morning.
  - Write their answers on the board
  - If possible, put numbers next to their responses to indicate the order that they happen
    - If students give responses out of order, have them help you put them in some kind of logical order
    - Point out places where order matters and places where it doesn't
- Introduce students to the idea that it is possible to create algorithms for the things that we do everyday.
  - Give them a couple of examples, such as making breakfast, tying shoes, and brushing teeth.
- Let's try doing this with a new and fun activity, like planting a seed!

Main Activity (20 min)

Real-Life Algorithms: Plant a Seed - Worksheet

You can use algorithms to help describe things that people do every day. In this activity, we will create an algorithm to help each other plant a seed. Directions:

- Cut out the steps for planting a seed from the **Real-Life Algorithms: Plant a Seed - Worksheet**.
- Work together to choose the six correct steps from the nine total options.
- Glue the six correct steps, in order, onto a separate piece of paper.
- Trade the finished algorithm with another person or group and let them use it to plant their seed!

Lesson Tip

You know your classroom best. As the teacher, decide if you should all do this together, or if students should work in pairs or small groups.

Wrap Up (10 - 20 min)

Lesson Tip

If deciding on the correct steps seems too difficult for your students, do that piece together as a class before you break up into teams.

Flash Chat: What did we learn?

- How many of you were able to follow your classmates' algorithms to plant your seeds?
- Did the exercise leave anything out?
  - What would you have added to make the algorithm even better?
  - What if the algorithm had been only one step: "Plant the seed"?
    - Would it have been easier or harder?
    - What if it were forty steps?
- What was your favorite part about that activity?

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 the seed you planted today.
- Write the algorithm you used to plant the seed.

Assessment (15 min)

Real-Life Algorithms: Plant a Seed - Assessment

- Hand out the worksheet titled **Real-Life Algorithms: Plant a Seed - Assessment** and allow students to complete the activity 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.

Go Figure

- Break the class up into teams.
- Have each team come up with several steps that they can think of to complete a task.
- Gather teams back together into one big group and have one team share their steps, without letting anyone know what the activity was that they had chosen.
- Allow the rest of the class to try to guess what activity the algorithm is for.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 4: Learn to Drag and Drop

Overview

In this lesson, students will relate the concept of algorithms back to everyday, real-life activities by planting an actual seed. The goal here is to start building the skills to translate real-world situations to online scenarios and vice versa.

Purpose

The main goal of this lesson is to build 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 appropriate computer lab behavior.

Agenda

Warm Up (10 min)

- **Behaving in the Computer Lab Discuss**
- **Vocabulary**

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

- **Dragging and Dropping Algorithms**
- **Previewing Online Puzzles as a Class**

Main Activity (20 - 30 min)

- **CSF Pre-Express Course**

Wrap Up (5 - 10 min)

- **Journaling**

Extension Activities

View on Code Studio

Objectives

Students will be able to:

- Recognize what is expected of them when they transition into the computer lab.
- Drag, drop, and click to complete Code.org puzzles.

Preparation

- [ ] Watch the **How to Make a Class Section on Code.org - Teacher Video**. Create a class section and make sure every student has a card with their passcode on it.
- [ ] Have the school IT person add a quick link for your class section to the computer desktop.
- [ ] 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

- **CSF Pre-Express Course**
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations Make a Copy

For the Students

- **Pair Programming** - Student Video
- **Unplugged Blockly Blocks (Grades K-1)** - Manipulatives
- **Think Spot Journal** - Reflection Journal Make a Copy

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

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 talk 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

Vocabulary

- Click: Pressing the mouse button
- Double-Click: Pressing the mouse button twice very quickly.
- Drag: Click your mouse button and hold as you move the mouse pointer to another location
- Drop: Releasing your mouse button to "let go" of the item that you are dragging.

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

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

Dragging and Dropping Algorithms

Print out a copy of **Real-Life Algorithms: Plant a Seed - Worksheet**. Cut out each of the squares representing tasks. On a projector or in front of the class practice "dragging and dropping" by pressing your finger on one of the paper squares and moving it across a table. Explain that you can "click" on this square by tapping your finger to the square, or you can "drag" the square by pressing your finger on the square and moving it. To "drop" the square, release your finger from the square.

After showing this to the class, ask for volunteers to put the algorithm in correct order by "dragging and dropping" the squares.

Previewing Online Puzzles as a Class

Project a puzzle from the online stage. 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)

CSF Pre-Express Course

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.

The vocabulary introduced in this lesson becomes relevant during this activity. Take some time to explicitly teach how to click, double-click, drag, and drop. It might work better for you to cover these words in the classroom environment where you can lead by example -- or it might make more sense to teach the words individually as students work on their puzzles in the lab. You will need to decide what you believe is best for your class.

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

Place kids in pairs and have them watch the **Pair Programming - Student Video** video at their stations. This should help students start off in the right direction.

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.

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:

- Can you draw a sequence for getting ready to go to the computer lab?
- Draw a computer lab "Do" and a "Don't"
- Use your journal to let me know how you felt about today's lesson plan

Extension Activities

If students complete the puzzles from this stage 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: Common Sense Education: Your Digital Footprint

Overview

In collaboration with **Common Sense Education - Website**, this lesson helps students learn about the similarities of staying safe in the real world and when visiting websites. Students will also learn that the information they put online leaves a digital footprint or “trail.” This trail can be big or small, helpful or hurtful, depending on how they manage it.

Purpose

Common Sense Education has created this lesson to teach kids the importance of understanding the permanence of something posted on the internet. By relating footprints on a map to what a student might post online, students will make important connections between being tracked by a physical footprint on a path and being tracked based on information posted online.

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Pause and Think**

Main Activity (20 min)

- **Your Digital Footprint - Digital Trail Squares**

Wrap Up (15 min)

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

Assessment (5 min)

- **Your Digital Footprint - Assessment**

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 if they should ask an adult they trust before they visit a particular website.
- Explore what information is appropriate to be put online.

Preparation

- [ ] Watch this **Your Digital Footprint - Teacher Video**.
- [ ] Prepare to show **Your Digital Footprint - Lesson Video**.
- [ ] (Optional) Prepare to show **Pause and Think Online - Video**.
- [ ] Common Sense Education's **Your Digital Footprint - Digital Trail Squares** game.
- [ ] Print one **Animal Tracks** chart (page 7) for each student.
- [ ] Print one **Your Digital Footprint - Assessment** for each student.

Links

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

For the Teachers

- **Your Digital Footprint** - Teacher Video
- **Common Sense Education** - Website
- **Your Digital Footprint** - Assessment Answer Key Make a Copy

For the Students

- **Your Digital Footprint** - Lesson Video
- **Your Digital Footprint** - Digital Trail Squares Make a Copy

- **Your Digital Footprint** - Assessment
  Make a Copy
- **Think Spot Journal** - Reflection Journal
  Make a Copy

Vocabulary

- **Digital Footprint** - The information about someone on the Internet.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important phrase:

- **Digital Footprint** - Say it with me: Dih-jih-tal Foot-print

The information about someone on the internet.

Pause and Think

- Ask What does it mean to be safe?
- When you walk down the street or play in your neighborhood without a trusted adult there, how do you stay safe?
- Tell students that just as they should stay safe in the real world, they should stay safe when they go into the online world (visiting websites). Make parallels between the answers students gave you about their neighborhood and the online world.

Play the **Your Digital Footprint - Lesson Video**.

- Introduce the idea that there are three different kinds of websites that students may have the opportunity to visit.
  - Green: A “green” website is:
    - A good site for kids your age to visit
    - Fun, with things for you to do and see
    - Has appropriate words
    - Doesn’t let you talk to people you don’t know
  - Yellow: A “yellow” website is:
    - A site you are not sure is right for you
    - One that asks for information such as who you are, where you live, your phone number or email address, etc.
    - A place where you are allowed to communicate freely with others
  - Red: A “red” website is:
    - A site that is not right for you
    - A place you might have gone to by accident
    - Filled with things that are for older kids or adults
  - Discuss examples of each of these kinds of sites.

Lesson Tip

If you have access to a computer, feel free to navigate to sites that might showcase each of these types (using extreme caution with your RED selection).

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

Main Activity (20 min)

common sense education®

Your Digital Footprint - Digital Trail Squares

- Peruse the **Your Digital Footprint - Digital Trail Squares** lesson on the Common Sense Education webpage.
- Give each student an **Animal Tracks Chart** (page 7).

	Mizzle the Mouse	Electra the Elephant
Whose full name do you know?		
Whose house could you find?		
Whose birth date do you know?		
Whose user name and password do you know?		
Who let out a secret on the internet?		
Which animal can you describe better from his or her photo?		

Directions:

- Place the *Digital Trail Squares* on the ground, face down, in two different trails, keeping Mizzle the Mouse and Electra the Elephant's trails separate from one another.
- Share the stories of Mizzle and Electra. These animals decided it would be fun to put some information about themselves online. They went onto **www.wildkingdom.com** and posted information. The only problem is that they forgot to ask their parents if it was okay first.
- Explain to students that they are from the "Things Big and Small" Detective Agency. A hunter has hired them to find out as much as possible about Mizzle the Mouse and Electra the Elephant. The more the detectives learn, the better for their plan to take over the animal kingdom.
- Divide students into groups of four. Tell them that each group should have a detective that will keep detailed notes.
- Invite students to go on a hunt for information. Let them know that the information that Mizzle and Electra post can be seen by anyone, including the detectives. Each group should follow the digital trail of both animals, starting with the mouse and then the elephant. Stagger the groups so they are on the trail at slightly different times. Students should fill out their handout as they go.

Lesson Tip

If your students have trouble writing, feel free to do this activity as a group and have students raise their hand when they find clues. This will allow you (or a teacher aide) to help communicate and record the information being shared.

For more in-depth modules, you can find additions to this curriculum at the **Common Sense Education - Website** page on Scope and Sequence.

Wrap Up (15 min)

Flash Chat: What did we learn?

- Who can the detectives find out more about, and why?
- Which animal has a bigger digital footprint?
- Mizzle says some interesting things about himself on the Internet. What are they?
- Is there anything that Electra posted on the Internet that could become a problem for her? If so, what and why?

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.

Take the time to discuss 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

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?
- How did you feel during today's lesson?
- 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.

Assessment (5 min)

Your Digital Footprint - Assessment

Hand out the assessment worksheet and allow students to complete the activity 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.

Common Sense Education

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

Lesson 6: Programming Unplugged: My Robotic Friends

Overview

Using a set of symbols in place of code, students will design algorithms to instruct a "robot" to stack cups in different patterns. Students will take turns participating as the robot, responding only to the algorithm defined by their peers. This segment teaches students the connection between symbols and actions, the difference between an algorithm and a program, and the valuable skill of debugging.

Purpose

This unplugged lesson brings the class together as a team with a simple task to complete: get a "robot" to stack cups in a specific design. Students will work to recognize real world actions as potential instructions in code. The designing of precise instructions will also be practiced, as students work to translate worded instructions into the symbols provided. If problems arise in the code, students should work together to recognize bugs and build solutions. This activity lays the groundwork for the programming that students will do throughout the course, as they learn the importance of defining a clearly communicated algorithm.

Agenda

Warm Up (5 min)

- **Talking to Robots**

Main Activity (45 min)

- **Introduction and Modeling**
- **Programming Your Robots**

Wrap Up (10 min)

- **Journaling**

View on Code Studio

Objectives

Students will be able to:

- Attend to precision when creating instructions
- Identify and address bugs or errors in sequenced instructions

Preparation

- [ ] Watch the **My Robotic Friends - Teacher Video**.
- [ ] (Optional) Print out one **My Robotic Friends - Symbol Key** per group or 2-3. Alternatively, find a place to display this information where students can reference throughout the lesson.
- [ ] Prepare a stack of 10 disposable cups per group of 2-3 students, OR
- [ ] (Optional) print and cut out **resource paper-trapezoid-template not found** for each group if your class is not going to use cups.
- [ ] Print out one set of **Stacking Cup Ideas - Manipulatives** per group.
- [ ] 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

- **My Robotic Friends** - Teacher Video
- **My Robotic Friends** - Teacher Prep Guide

For the Students

- **My Robotic Friends** - Unplugged Video (**download**)
- **My Robotic Friends** - Symbol Key Make a Copy
- **Stacking Cup Ideas** - Manipulatives Make a Copy

- **My Robotic Friends** - Paper Trapezoid Template Make a Copy
- **Think Spot Journal** - Reflection Journal Make a Copy

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.

Teaching Guide

Warm Up (5 min)

Talking to Robots

Display: Watch one of the videos below to give students context for the types of things that robots can do:

- **Asimo by Honda** (3:58)
- **Egg drawing robot** (3:15)
- **Dancing Lego Robot** (1:35)

Discussion Goal

The goal of this quick discussion is to call out that while robots may seem to behave like people, they're actually responding only to their programming. Students will likely refer to robots from movies and TV that behave more like humans. Push them to consider robots that they've seen or heard of in real life, like Roombas, or even digital assistants like Amazon Alexa.

Discuss: Refer to the video that you chose and ask students how they think that the robot knew what to do. Does a robot really “understand” what you say? Is it worried about getting in trouble if it doesn't do what it's told?

Say: Robots can only do what they've been told to do, but we don't just tell them using words. In order to do something, a robot needs to have a list of steps that it can read. Today, we are going to learn what it takes to make that happen.

Main Activity (45 min)

Introduction and Modeling

Set Up: Have stacks of cups or cut paper trapezoids available for groups.

Display: Display the **My Robotic Friends - Symbol Key** or write the allowed actions on the board - make sure these are in a place where they can be seen for the whole activity. Explain to the class that these will be the only six actions that they can use for this exercise. For this task, they will instruct their “robot” friend to build a specific cup stack using only the commands listed on the key.

Model: In order to explain how the instructions are intended to work, model for the class how to create and follow an algorithm for replicating a simple pattern. Place a single stack of cups in front of you to start.

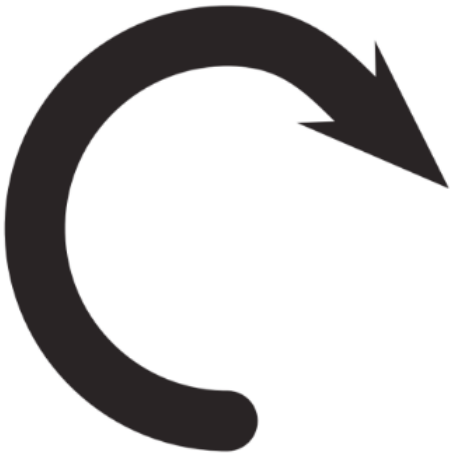
Display: Hold up the pattern you plan to model. A simple three cup pattern is a great place to start.

Pick Up Cup

Put Down Cup

Step Forward

Step Backward

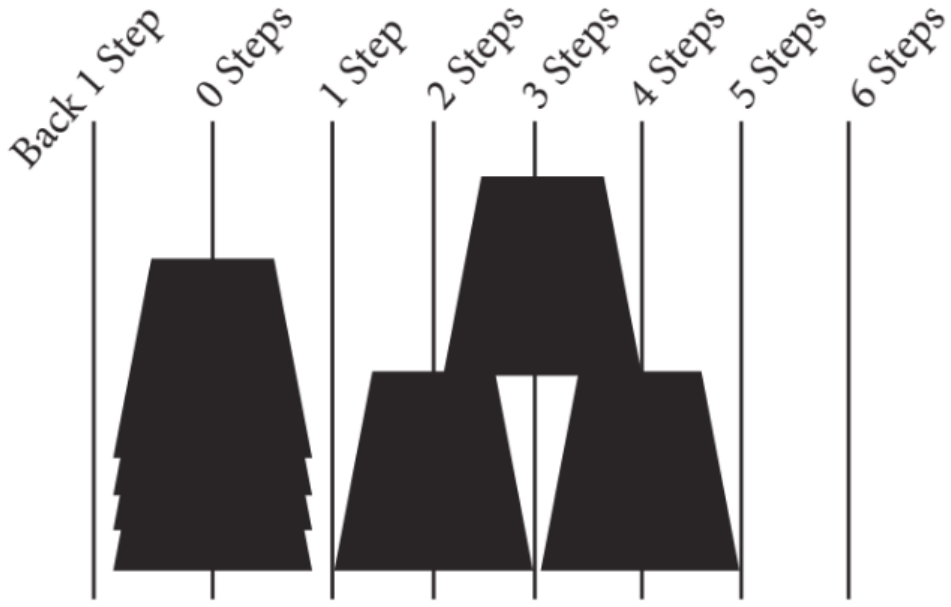


Turn Cup Right 90°

Turn Cup Left 90°

Prompt: Ask the class what the first instruction should be, using *only the six instructions allowed*. The first move should be to "pick up cup." If students suggest something else from the list, perform that action and allow them to see their error. If they suggest something not from the list, make a clear malfunction reaction and let them know that the command is not understood.

With cup in hand, ask the class to continue giving you instructions until the first cup is placed. This is a great place to clarify that a "step forward" and "step backward" each imply moving half a cup width. See the image below for reference.

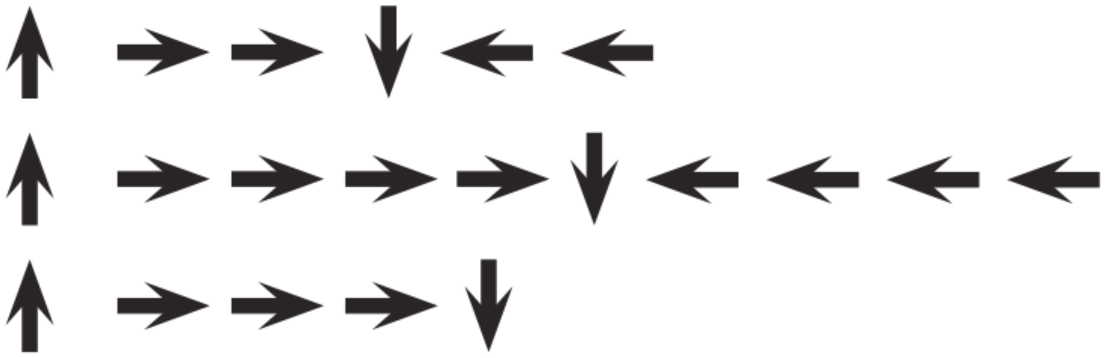


Continue asking for instructions from the classroom until you have completed the entire design.

Once your stack is complete, point out that they just gave you a list of steps for completing a task. That's an algorithm. Algorithms are great for sharing ideas, but spelling them out word by word can take a long time. That's what the symbols are for! When you change an algorithm into symbols that a robot (or computer) understands, that's called programming.

Ask the class to help you write the "program" for that first move, and then the rest of the moves necessary to complete the pattern. Depending on the confidence of your students, you might switch back and forth frequently between acting as the "robot" and writing down the code, or you might push them to write the whole program

before you will implement it. One possible solution looks like this:



Volunteer: Once the class has completed the model program, ask one of the students to come up and act as the "robot" to ensure that the program really works. Encourage them to say the instructions out loud as they "run" the code.

Programming Your Robots

Group: Place students into groups of 4. Each group should then further break down into two pairs - each pair will develop their own program to be "run" by the other pair.

Distribute: Give each group one stack of cups or paper cutouts.

Display: Show **Stacking Cup Ideas - Manipulatives** to the class or hand out individual copies for groups to use. Have each pair (not group) choose which idea they would like their robots to do. Try to push for an easier idea for the first time, then have them choose a more complex design later on. Encourage pairs to keep their choice secret from the other half of their group.

Discuss: Give each pair time to discuss how the stack should be built, using only the provided symbols. Make sure each group writes down the "program" somewhere for the "robot" to read later.

Do: Once both of the group's pairs have decided on their algorithms, they can take turns being "robots" for each other by following the instructions each pair wrote. Encourage students to watch their "robot" closely to ensure that they are following instructions. If a student sees a bug and raises their hand, have the robot finish the instructions to the best of their ability. Afterward, have the students discuss the potential bug and come up with a solution. Continue repeating until the stack is built properly.

Teaching Tip

Enforcing the rules: While the robot is working on the stack make sure that the class knows:

- Programmers are not allowed to talk when the robot is working. This includes blurting out answers or pointing out when the robot has done something wrong.
- Programmers should raise their hand if they see a bug.

Circulate: Look for groups who are trying to take shortcuts by adding things (like numbers) to their code. Praise them for their ingenuity, but remind them that for this exercise, the robots do not understand *anything* but the provided symbols. If you like, you can hint that they should save their brilliant solution for the next time they play this game, since they might get the chance to use it soon!

Iterate: Depending on your time available, mix up the pairs and give them a chance to do a different pattern. Each time groups repeat the process, encourage them to choose a more challenging pattern.

Discuss: After everyone has had a chance to be the robot, bring the class back together to discuss their experience. In particular, discuss as a class:

- What was the most difficult part of coming up with the instructions?
- Did anyone find a bug in your instructions once your robot was following them?
  - What was the bug?
  - Why do you think you didn't notice it when writing the program?
- When you were the robot, what was the hardest part of following the instructions you were given?

Discussion Goal

Sense making: The goal of this discussion is to give students space to make sense of their experience both as robot and programmer. The questions are intentionally broad, but designed to get students thinking about the challenges of writing a clear program and the constraints of a robot or computer in interpreting your instructions.

Wrap Up (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?
- How did you feel during today's lesson?
- Draw a stack of cups that the robot made today.
- Draw a stack of cups that you would like a robot to make someday!

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 7: Programming in Maze

Overview

In this series of online puzzles, students will build on the understanding of algorithms, debugging, and general computer literacy. Featuring characters from the game Angry Birds, students will develop sequential algorithms to get the bird to the pig without crashing into walls or TNT. Debugging puzzles have also been mixed into this stage for added practice with problem solving and critical thinking.

Purpose

In this lesson, students will be practicing their debugging and programming skills on a computer platform. When someone starts *programming* they piece together instructions in a specific order using something that a machine can read. Through the use of programming, students will develop an understanding of how a computer navigates instructions and order. *Debugging* is a concept that is very important to computer programming. Computer scientists have to get really good at facing all of the bugs in their own programs. Debugging forces the students to recognize problems and overcome them while building critical thinking and problem solving skills.

Agenda

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

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

Main Activity (30 min)
- **CSF Pre-Express Course**

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

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Order movement commands as sequential steps in a program.
- Represent an algorithm as a computer program.
- Develop problem solving and critical thinking skills by reviewing debugging practices.

Preparation

- [ ] Play through the **CSF Pre-Express Course** in stage 7 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 Teachers

- **CSF Pre-Express Course**
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations Make a Copy

For the Students

- **Unplugged Blockly Blocks (Grades K-1)** - Manipulatives
- **Think Spot Journal** - Reflection Journal Make a Copy

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

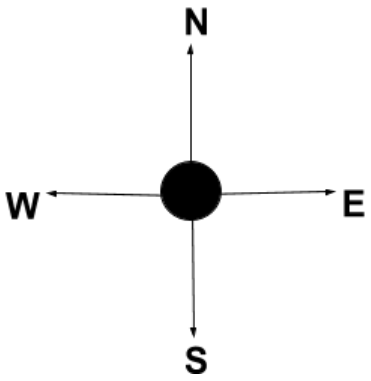
Warm Up (10 min)

Introduction

Ask your students if they are familiar with the game or the movie "Angry Birds". Explain that they will be writing programs to help Red, from Angry Birds locate a pig.

Teacher Tip:

Review cardinal directions with your class.



Let students know that they will see those letters in their programs next to the direction arrows. We recommend drawing the directions somewhere that the students can look back up to review.

Bridging Activity - Programming (10 min)

This activity will help bring the unplugged concepts from My Robotic Friends 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

Select a pattern from **Stacking Cup Ideas - Manipulatives** from the My Robotic Friends unplugged activity. Using arrows from the [course-ab-blockly-blocks], have the students program a "robot" from their desks to get the correct stacking of the cups. Make sure that they understand that the blocks need to go from top to bottom and they all need to touch! Have the students pair share to check answers and resolve any questions or bugs that may come up.

Previewing Online Puzzles as a Class

Pull a puzzle from the corresponding online puzzles. We recommend puzzle 8. Give students small note cards to draw out the various commands from the My Robotic Friend **My Robotic Friends - Symbol Key**. Have students lay out a pattern with the arrows they created that will get the bird to the pig. Ask the students to share. See how many other students had the same answer!

Main Activity (30 min)

Teacher Demonstration

We've included some multiple choice prediction levels that are difficult for non-readers. Like the puzzles in the bridging activity, 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:

- **Pre-Express Course, Programming in Maze #1**
- **Pre-Express Course, Programming in Maze #2**

CSF Pre-Express Course

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?

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

- 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?
- How did you feel during today's lesson?
- Write a program that gets you from your desk to the front of the classroom.
- What is a bug? What does a bug do to your program?

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 the pig and bird.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 8: Programming in Star Wars

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)
- **CSF Pre-Express Course**

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 **CSF Pre-Express Course** 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 Teachers

- **CSF Pre-Express Course**
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations Make a Copy

For the Students

- **Think Spot Journal** - Reflection Journal Make a Copy

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)

CSF Pre-Express Course

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?
- How did you feel during today's lesson?
- 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 9: My Loopy Robotic Friends

Overview

Building on the initial "My Robotic Friends" activity, students tackle larger and more complicated designs. In order to program their "robots" to complete these bigger designs, students will need to identify repeated patterns in their instructions that could be replaced with a loop.

Purpose

This lesson serves as a reintroduction to loops, using the now familiar set of "robot" programming instructions. Students will develop critical thinking skills by looking for patterns of repetition in the movements of classmates and determining how to simplify those repeated patterns using loops.

Agenda

Warm Up (10 min)

- **My Robotic Friends Review**

Activity (30 min)

- **Introduction and Modeling**
- **Looping Your Robots**

Wrap Up (5 min)

- **Journaling**

Extension Activities

View on Code Studio

Objectives

Students will be able to:

- Identify repeated patterns in code that could be replaced with a loop
- Write instructions that use loops to repeat patterns

Preparation

- [ ] Watch the **My Loopy Robotic Friends - Teacher Video**.
- [ ] (Optional) Print out one **My Robotic Friends - Symbol Key** per group or 4 students. Alternatively, find a place to display this information where students can reference throughout the lesson.
- [ ] Prepare a stack of 20 paper cups for each group of 4 students. OR
- [ ] (Optional) print and cut out **resource paper-trapezoid-template not found** for each group if your class is not going to use cups.
- [ ] Print out one set of **Stacking Cup Ideas - Manipulatives** per group.
- [ ] 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

- **My Loopy Robotic Friends** - Teacher Video

For the Students

- **My Robotic Friends** - Paper Trapezoid Template Make a Copy
- **My Robotic Friends** - Symbol Key Make a Copy
- **Stacking Cup Ideas** - Manipulatives Make a Copy

Vocabulary

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

Teaching Guide

Warm Up (10 min)

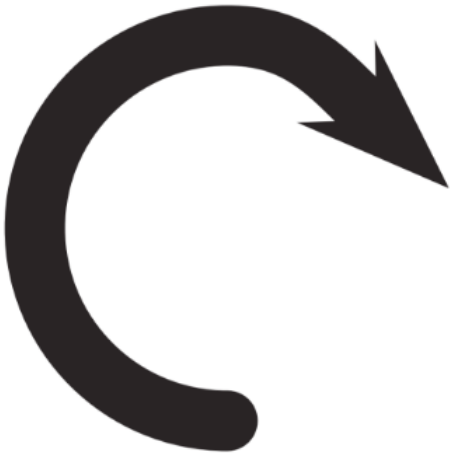
My Robotic Friends Review

Pick Up Cup

Put Down Cup

Step Forward

Step Backward



Turn Cup Right 90°

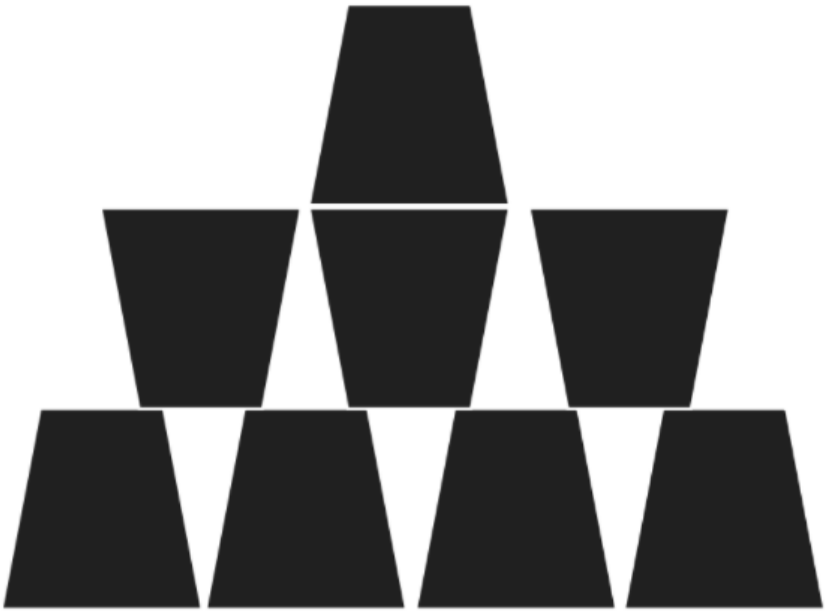
Turn Cup Left 90°

Goal: This review will refresh the students' minds about how quickly programs for the "My Robotic Friends" activity can get intense.

Display: Show the **My Robotic Friends - Symbol Key** that we used in My Robotic Friends. For each of the six symbols, ask students to show you what it looks like for a robot to follow that instruction.

Model: With the class together as a group, pull an easy puzzle from the "My Robotic Friends" Cup Stack Pack and program with each other as a reminder of rules and terminology.

Next, pull a puzzle that's slightly harder, but also requires a lot of steps like the one below.



Volunteer: Ask a volunteer (or a group of volunteers) to come forward to help program this one on the board. If you make them stick strictly to the "no symbols other than those on the key" rule, it will probably take a while!

Display: Now, bring up this image:

What is the reaction of the class?

Prompt: Give students the opportunity to brainstorm shorter ways to relay the code that they're about to create. (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.)

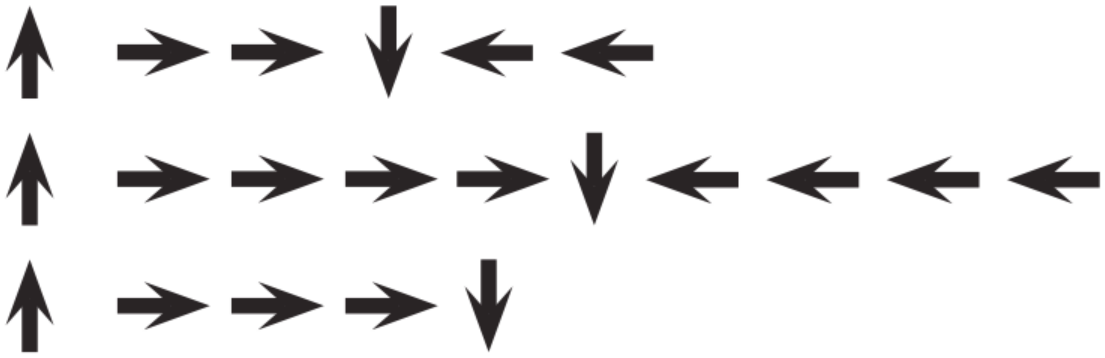
Once students have put together the idea of "repeating" code, give them the vocabulary around it. Make sure to share with them that often the terms "repeat something" and "loop something" are often used interchangeably.

Activity (30 min)

Introduction and Modeling

Set Up: Have stacks of cups or cut paper trapezoids available for groups.

Display: Take the program from one of your previous cup stacks and display it for the class, or use the one below.

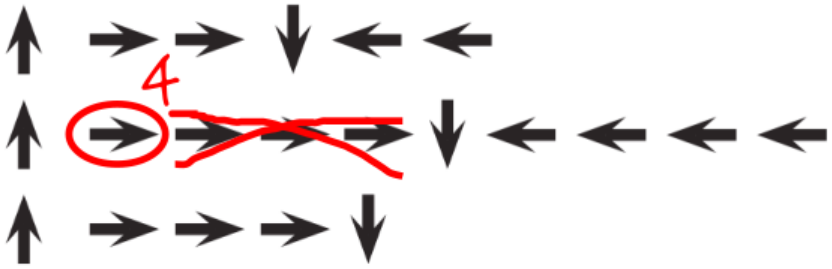


Think: Ask students to think quietly about where in this program they can find a pattern of instructions that repeat uninterrupted (one repetition after another).

Pair: Turn to a neighbor and share one of the repeating patterns you found.

Share: Ask a few students to share out the patterns they identified. Try to pull out different approaches to grouping patterns. For each pattern, ask students to identify how many times the pattern repeats.

Model: Using one of the repeating patterns that the class identified, model how Circle the instruction or pattern that repeats, write the number of loops near that circle, then cross out the rest of the arrows.



Repeat this until the entire program has been shortened, then re-write the program in a way where students can see how much more simple the resulting instructions are.

Looping Your Robots

Group: Place students into groups of 4. Each group should then further break down into two pairs - each pair will develop their own program "run" on the other pair.

Distribute: Give each group one stack of cups or paper cutouts.

Display: Show **Stacking Cup Ideas - Manipulatives** to the class or hand out individual copies for groups to use.. Have each pair (not group) choose which idea they would like their robot to do. Encourage pairs to select a more complicated pattern this time around.

Discuss: Let each group discuss how the stack should be built, then instruct each group to translate the algorithm into the symbols. Make sure each group writes down the symbol algorithm somewhere for the "robot" to read later. As students are working on their programs, remind them to be on the lookout for opportunities to replace a repeating pattern with a loop.

Do: When groups have finished their instructions, have each pair take turns "running" their code with another pair. Remind students to be on the lookout for bugs in their code, but not to interrupt a robot until it's finished running the program.

Discuss: When all of the pairs have had a chance to run their programs, ask a few to share their solutions with the class. Use this opportunity to discuss how groups came up with different solutions to the same puzzle. In particular, you might ask of each program:

- How did they identify the loops?
- Are there other ways those loops could have been written?
- How much shorter is the program with loops than it would be without?
- Is the program easier to understand with loops, or written out longhand? Why?

Teaching Tip

Looking for Loops: Be sure to keep your eyes open for students using loops. Try to avoid correcting their overall algorithms or prescribing a solution, but feel free to direct students towards patterns that could be shortened by using a repeat circle.

Watch students as they run through the code. 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 (5 min)

Journaling

Goal: Allow students to reflect on the activity that they just experienced.

Flash Chat:

Here are some possible topics:

- Do you feel like loops make programming easier or harder?
- What other kinds of things in life do we repeat?
  - Eating - put food in mouth, chew 20 times
  - Brushing hair - brush through hair 35 times
  - Routines - Wake up, go to school, come home, go to bed

Journal Prompts:

- Journal time! Ask students to draw a feeling face in the corner of their journal page to remind them how they felt about this lesson.
- Have the students write or draw something in their journal that will remind them later what loops are. This can come from a prompt like:
  - What does "repeat" mean to you?
  - Draw a picture of you repeating something.

Extension Activities

- Have students draw their own cup stacking creations for someone else to code.
- Provide students with algorithms that utilize repeats, then have them expand the program back out to a full step-by-step version.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 10: Loops in Collector

Overview

Building on the concept of repeating instructions from "My Loopy Robotic Friends," this stage will have students using loops to collect treasure 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 the Code.org puzzles, students will learn to add instructions to existing loops, gather repeated code into loops, and recognize patterns that need to be looped.

Agenda

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

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

Main Activity (30 min)
- **CSF Pre-Express Course**

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

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Identify the benefits of using a loop structure instead of manual repetition.
- Breakdown a long sequence of instructions into the largest repeatable sequence possible.
- Create a program for a given task which loops a sequence of commands.
- Employ a combination of sequential and looped commands to reach the end of a maze.

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.
- [ ] 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

- **CSF Pre-Express Course**
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations Make a Copy

For the Students

- **Unplugged Blockly Blocks (Grades K-1)** - Manipulatives
- **Think Spot Journal** - Reflection Journal Make a Copy

Vocabulary

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

- **Repeat** - To do something again.

Teaching Guide

Warm Up (10 min)

Introduction

Review with students the My Loopy Robotic Friends activity:

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

Bridging Activity - Loops (10 min)

This activity will help bring the unplugged concepts from "My Loopy Robotic Friends" 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

Select a pattern from **Stacking Cup Ideas - Manipulatives** from the My Robotic Friends unplugged activity 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). Remember to choose a cup pattern that doesn't use any upside down cups, because there is no `turn` command online for K-1 students. Next, have the students program a "robot" (a partner or the teacher) from their desks to get the correct stacking of the cups. Make sure that they understand that the blocks need to go from top to bottom and they all need to touch! Have the students pair share to check answers and resolve any questions or bugs that may come up.

Previewing Online Puzzles as a Class

Pull a puzzle from the corresponding online puzzles. We recommend puzzle 8. Using the **My Robotic Friends - Symbol Key** from the My Loopy Robotic Friends Unplugged Activity, have students draw out a pattern that they think will get Laurel the Adventurer to collect all the treasure. Ask the students to share. See how many other students had the same answer!

Main Activity (30 min)

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:

- **Pre-Express Course, Loops in Collector**

CSF Pre-Express Course

As students work through the puzzles, see if they can figure out how many fewer blocks they use with a loop vs. not using 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?
- How did you feel during today's lesson?
- Draw a maze that you can solve using loops.
- 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 11: Loops in Artist

Overview

Returning to loops, students learn to draw images by looping simple sequences of instructions. In the previous online lesson, loops were used to traverse a maze and collect treasure. Here, students use loops to create 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 will test their critical thinking skills by evaluating given code and determining what needs to be added in order to solve the puzzle. 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)
- **CSF Pre-Express Course**

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 **CSF Pre-Express Course** before the lesson 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

- **CSF Pre-Express Course**
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations Make a Copy

For the Students

- **Think Spot Journal** - Reflection Journal Make a Copy

Vocabulary

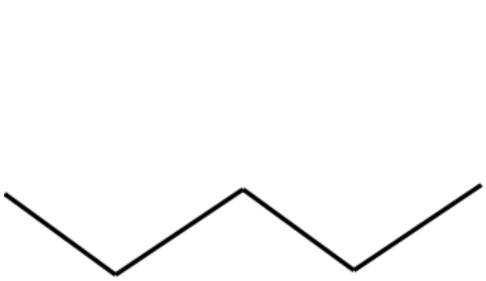
- **Loop** - The action of doing something over and over again.

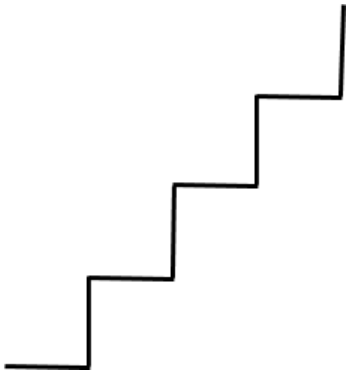
Teaching Guide

Warm Up (10 min)

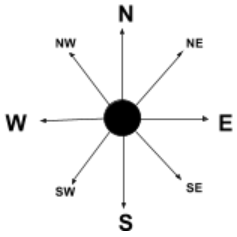
Introduction

- 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)

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:

- **Pre-Express Course, Loops in Artist**

CSF Pre-Express Course

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**.

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?
- How did you feel during today's lesson?
- Draw the patterns you made with a loop.
- Draw a pattern that you would like to make with a loop.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 12: Events Unplugged: The Big Event

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)

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

Assessment (10 min)

- **The Big Event - Assessment**

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

- [ ] Watch the **The Big Event - Teacher Video**.
- [ ] Print one **The Big Event (Courses A, B) - Controller Image**.
- [ ] Print one **The Big Event - Assessment** for each 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** - Teacher Video
- **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
- **Think Spot Journal** - Reflection Journal 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 Red, the Angry Bird find the pig?
  - In that exercise, you knew in advance exactly where you wanted Red to end up, so you could make a program that took the bird 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.

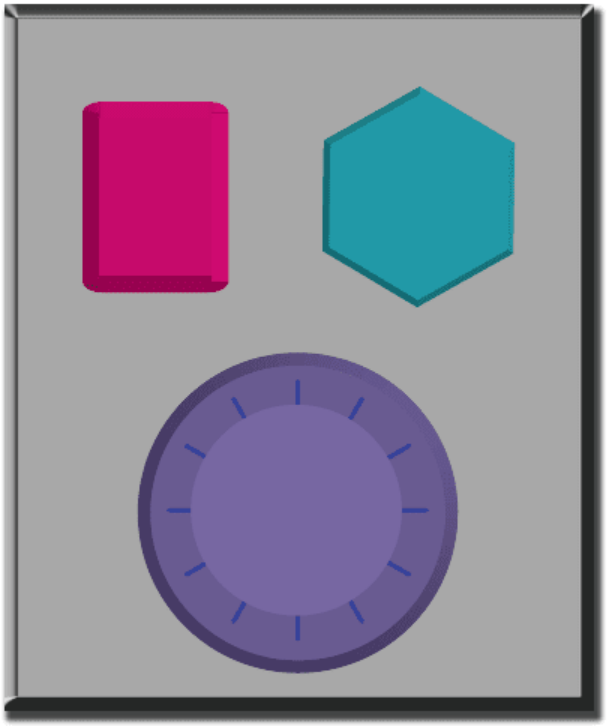
Lesson Tip

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 bird 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 Event Controller 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)

Flash Chat: What did we learn?

- 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?

Journaling

Journal Prompts:

- What was today’s lesson about?
- How did you feel during today’s lesson?
- Draw an event that caused an action today.
- Draw an action that was caused by an event that happened today.

Assessment (10 min)

The Big Event - Assessment

- Hand out the assessment activity and allow students to complete the activity 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 13: Events in Play Lab

Overview

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

Purpose

In this online activity, students will learn how to use events in Play Lab. They will start by training the knight to move when an arrow key is pressed, then end with the opportunity to showcase the rest of the skills that they learned throughout this course, including sequence and looping, as part of the final freeplay puzzle.

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)
- **CSF Pre-Express Course**

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 sequences and event-handlers.
- Share a creative artifact with other students.

Preparation

- [ ] Play through the **CSF Pre-Express Course** in stage 12 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

- **CSF Pre-Express Course**
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations Make a Copy

For the Students

- **Unplugged Blockly Blocks (Grades K-1)** - Manipulatives
- **Think Spot Journal** - Reflection Journal Make a Copy

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 **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 online puzzles. We recommend puzzle 4 of this stage. Call on different students to make a funny face representing a mood when you click on Daisy. Explain this is an event that they are reacting to and Daisy can be coded to change moods when you click on her.

Main Activity (30 min)

CSF Pre-Express Course

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.

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?
- How did you feel during today's lesson?
- 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

Lesson 14: Spelling Bee

Overview

Part puzzle, part word search, the Spelling Bee asks students to program a bee to find common words in a grid.

Agenda

Getting Started

Introduction

Activity

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Arrange sequential movement commands to search for and identify target words within a grid of letters.
- Practice spelling age-appropriate words

Teaching Guide

Getting Started

Introduction

- Students should be able to read and identify the following words for this activity:
  - North
  - South
  - East
  - West
  - Jump
  - Code
  - Debug
  - Above
  - Below
  - Story
  - Move
  - Square

Activity

Spelling Bee

Very young students or struggling readers may need additional support finding the words - using manipulatives (like Scrabble tiles) can help students see what words look like in different directions.

Extended Learning

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

Vocab Hop Scotch

Using class vocab words, create a floor-sized word search. The whole class can then "program" a student, or teacher, to spell out words by creating sequences of cardinal directions.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming