Course A

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

Lesson 1: Debugging: Unspotted Bugs

Bug | Debugging | Persistence | Unplugged

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

Fail | Frustrated | Persistence | Unplugged

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: Programming Unplugged: Happy Maps

Unplugged | Algorithms | Sequencing

Lesson 6: Programming in Maze

Algorithms | Debugging | Program | Programming | Widget - Text Compression

Lesson 7: Common Sense Education: Going Places Safely

Common Sense Education | Unplugged

Lesson 8: Loops Unplugged: Happy Loops

Unplugged | Loop | Repeat

Lesson 9: Loops in Collector

Loop | Collector

Lesson 10: Loops in Artist

Loop | Artist

Lesson 11: Events Unplugged: The Big Event

Unplugged | Event

Lesson 12: Events in Play Lab

Play Lab | Event

Lesson 1: Debugging: Unspotted Bugs

Bug | Debugging | Persistence | Unplugged

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**

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

- [ ] Review the Unspotted Bugs Story (**Unspotted Bugs - Online Story**)
- [ ] 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** to each student

Links

For the Teacher

- **Marble Run** - Teacher Prep Guide
- **Unspotted Bugs** - Storybook (PDF)
- **Think Spot Journal**

For the Students

- **Unspotted Bugs** - Online Story

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?

Reflect:

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

- **AP** - Algorithms & Programming

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

Fail | Frustrated | Persistence | Unplugged

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 lession 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

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

- [ ] Pre-read "Stevie and the Big Project" to identify appropriate questions for your class.
- [ ] 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**.

Links

For the Teacher

- **Marble Run** - Teacher Prep Guide (**PDF** | **DOCX**)
- **Stevie and the Big Project** - Storybook (PDF) (**download**)
- **Think Spot Journal**

For the Students

- **Stevie and the Big Project** - Online Story

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)

Goal: 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

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

Journal Prompts:

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

- 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

▸ **AP** - Algorithms & Programming

Lesson 3: Real-Life Algorithms: Plant a Seed

Unplugged | Algorithms

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 - Assessment**

Extended Learning

- **Go Figure**

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 - Assessment** for each student.
- [ ] Make sure each student has a**Think Spot Journal**.

Links

For the Teacher

- **Plant a Seed** - Teacher Video
- **Real-Life Algorithms: Plant A Seed** - Worksheet
- **Real-Life Algorithms** - Assessment
- **Think Spot Journal**

Vocabulary

- **Algorithm** - A precise sequence of instructions for processes that can be executed by a computer

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 **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)

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?

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.

Journaling

Ask the students to go back to their desks to reflect individually on what they learned. Write a couple of the questions up above on a whiteboard. Ask the students to discuss these in their journal. Sample prompts include:

Journal Prompts:

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

Assessment (15 min)

Real-Life Algorithms - Assessment

- Hand out the **Real-Life Algorithms - 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

▸ **AP** - Algorithms & Programming

COURSE A 1 2 3 4 5 6 7 8 9 10 11 12 CODE

Lesson 4: Learn to Drag and Drop

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

Overview

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

Purpose

The main goal of this lesson is to build students' experience with computers. By covering the most basic computer functions such as clicking, dragging, and dropping, we are creating a more equal playing field in the class for future puzzles. This lesson also provides a great opportunity to introduce proper 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)

- **Course A Online Puzzles - Website**

Wrap Up (5 - 10 min)

- **Journaling**

Extension Activities

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 your own class section on Code.org 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**.

Links

For the Teacher

- **How to Make a Class Section on Code.org** - Teacher Video
- **Course A Online Puzzles** - Website
- **Unplugged Blockly Blocks (Grades 1 - 2)** - Manipulatives (**PDF** | **DOCX**)
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations (**PDF** | **DOCX**)
- **Computer Lab Expectations** - Teacher Prep Guide (**PDF** | **DOCX**)
- **Think Spot Journal**

For the Students

- **Pair Programming** - Student Video

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 be walk when we are in the computer lab?
- What should you do if you get stuck on a puzzle?
- If you get frustrated, will it help to hit the computer?
- When we're about to go to the computer lab, how should be 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 **Course A Online Puzzles - Website** corresponding to this lesson. 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)

Main Activity (20 - 30 min)

Course A Online Puzzles - Website

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.

> 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

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.

Place kids in pairs and have them watch the**Pair Programming - Student 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 could include:

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

Standards Alignment

CSTA K-12 Computer Science Standards

▶ **AP** - Algorithms & Programming

COURSE A 1 2 3 4 5 6 7 8 9 10 11 12 CODE

Lesson 5: Programming Unplugged: Happy Maps

Unplugged | Algorithms | Sequencing

Overview

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

Purpose

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

Agenda

Warm Up (15 - 20 min)

- **Step-by-Step**
- **Discuss**
- **Vocabulary**

Main Activity (15 - 20 min)

- **Happy Maps Programming**

Wrap Up (8 min)

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

Extension Activities

Objectives

Students will be able to:

- Translate an algorithm into a program.
- Decode and run a program created by someone else.

Preparation

- [ ] Watch the **Happy Maps - Teacher Video**.
- [ ] Print out **Happy Map Cards - Worksheet** for each group.
- [ ] Print out **Happy Map Game Pieces - Manipulatives** for each group.
- [ ] Make sure each student has a**Think Spot Journal**.

Links

For the Teacher

- **Happy Map Cards** - Worksheet (**PDF** | **DOCX**)
- **Happy Map Game Pieces** - Manipulatives (**PDF** | **DOCX**)
- **Happy Maps** - Teacher Video
- **Think Spot Journal**

Vocabulary

- **Program** - An algorithm that has been coded into something that can be run by a machine.

Teaching Guide

Warm Up (15 - 20 min)

Step-by-Step

Goal: This portion of the lesson will set the stage for making the connection between an algorithm and a program.

- Ask your students for directions to the chalkboard.
  - If they start shouting simultaneously, explain that you can only hear one instruction at a time. Call on students individually if that helps.
- When you reach the board, ask for instructions to draw a smiley face.
  - Again, request this one step at a time
- Explain that many tasks can be described using a specific list of instructions. That list is called an algorithm.

This is where we introduce the activity. In your **Happy Map Cards**, there are single and double step maps. Show the students that the point of these maps is to figure out how to get the Flurb to the fruit. Students should then use their words to solve these puzzles in small groups. Example algorithms include:

- Move the Flurb up
- Move the Flurb North, then West

Discuss

- Ask if any students want to share their algorithm for one of the mazes.
  - Can everyone see how the volunteer came up with that answer?
  - Is there any debugging that needs to be done?
- Now, what if we tried to move our Flurb through a 10-step maze?
  - Could we remember all of the steps?
  - What if we had to write all of the steps down in words?
  - How could we make this easier?

Lesson Tip:

If you have time and motivation, get your students to bring their stuffies into school for this lesson and have them create programs to move stuffies from square to square, outlined in tape on the carpet.

Show students how to represent the algorithms that they just created using arrows (either drawn, or cut from the **Happy Maps Game Pieces**). Have a short discussion about how quick and easy this “code” makes the process of getting the Flurb where it needs to be.

Vocabulary

- **Program (or Code)**: an algorithm that has been encoded into something that can be run by a machine.

Main Activity (15 - 20 min)

Happy Maps Programming

Goal: Happy Maps Programming will help students translate an algorithm into code.

Now that the students have had some practice encoding algorithms, have them work on some larger maps.

Encourage the students to follow these steps:

- Discuss an algorithm to get the Flurb to the fruit.
- Encode the algorithm into arrows to share with the class.
- Try their code to see if everything works as expected.

- Debug any issues and fix their code until it works correctly.

Make sure to bring the class back together at least a couple of times to allow students to share their code or the things that they have learned.

Wrap Up (8 min)

Flash Chat: What did we learn today?

When it's time to wind down class, ask students if they can tell the difference between an algorithm and code.

Both are a list of steps, but code (a program) has been encoded in a way that can be run by a machine (or a kindergartener!)

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

Journaling

Journal Prompts:

Students should be encouraged to capture their thoughts in their journal after each activity (with text or images.)

Choose a journal prompt that will help students remember the purpose of this exercise. These could include:

- In this game, we made programs for **people** to run. What else can read a program?
- How did you know if there was a bug in your program?

Extension Activities

- Create a life-size grid on the rug with tape and have student bring stuffies to school. Now students can program friends to move their actual stuffies as directed in the programs.
- Have students create their own maps for other students to solve using programs.

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

COURSE A 1 2 3 4 5 6 7 8 9 10 11 12 CODE

Lesson 6: Programming in Maze

Algorithms | Debugging | Program | Programming | Widget - Text Compression

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 (5 min)

- **Introduction**

Bridging Activity - Programming (10 min)

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

Main Activity (30 min)

- **Course A Online Puzzles - Website**

Wrap Up (5 - 10 min)

- **Journaling**

Extended Learning

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 **Course A Online Puzzles - Website** in stage 6 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**.

Links

For the Teacher

- **Course A Online Puzzles** - Website
- **Unplugged Blockly Blocks (Grades 1 - 2)** - Manipulatives (**PDF** | **DOCX**)
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations (**PDF** | **DOCX**)
- **Think Spot Journal**

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

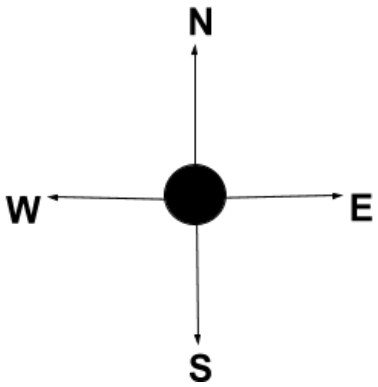
Warm Up (5 min)

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 Happy Maps 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 an empty Flurb map from the **Happy Map Cards - Worksheet** and give students **Unplugged Blockly Blocks (Grades 1 - 2) - Manipulatives** prefilled with cardinal commands like E → (East) and W ← (West). Now, have the students program the Flurb from their desks using the paper Blockly blocks to get the Flurbs to the fruit. Make sure that they understand that the blocks need to go from top to bottom and they all need to touch!

Previewing Online Puzzles as a Class

Pull a lesson from the puzzles corresponding to this lesson. We recommend puzzle 7. Using arrows from the **Happy Map Game Pieces - Manipulatives**, have students lay out a pattern that they think 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:

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

Course A Online Puzzles - Website

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

- Utilize **Pair Programming - Student Video** whenever possible

- Encourage students with questions/challenges to start by asking their partner
- Unanswered questions can be escalated to a nearby group, who might already know the solution
- Remind students to use the debugging process before you approach
- Have students describe the problem that they're seeing. What is it supposed to do? What does it do? What does that tell you?
- Remind frustrated students that frustration is a step on the path to learning, and that persistence will pay off.

Teacher Tip:

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

- Don't sit in their chair
- Don't use their keyboard
- Don't touch their mouse
- Make sure the classmate can describe the solution before you walk away

- 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?
- What's something you shouldn't do while helping a friend with a puzzle?
- Why do you think debugging is important?

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.

COURSE A 1 2 3 4 5 6 7 8 9 10 11 12 CODE

Lesson 7: Common Sense Education: Going Places Safely

Common Sense Education | Unplugged

Overview

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

Purpose

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

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Where We Go**

Main Activity (20 min)

- **Keep It Private**

Wrap Up (15 min)

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

Assessment (5 min)

- **Keep It Private - Assessment**

Extended Learning

Objectives

Students will be able to:

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

Preparation

- [ ] Watch the **Going Places Safely - Teacher Video**.
- [ ] Prepare to show the **Going Places Safely - Lesson Video**.
- [ ] Live access or print-off of **SecretBuilders** sign-up page (Click “New Player,” select an age, and then select “I’m a Girl” or “I’m a Boy.”).
- [ ] Print one **Keep It Private - Assessment** for each student.
- [ ] Make sure each student has a **Think Spot Journal**.

Links

For the Teacher

- **Going Places Safely** - Teacher Video
- **Going Places Safely** - Lesson Video
- **Keep It Private** - Assessment
- **Common Sense Education** - Website
- **Think Spot Journal**

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

Username - Say it with me: Yews-er-naym

A name you make up so that you can see or do things on a website, sometimes called a “screen name”

Where We Go

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

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

What three rules does Jeremiah follow when he goes places online?

1) Always ask your parent (or teacher) first

2) Only talk to people you know

3) Stick to places that are just right for you

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

Main Activity (20 min)

common sense education®

Keep It Private

Access **SecretBuilders** sign-up page live, or project a print-out on the board for the class to see.

- Invite students to give examples of information that they should keep private.
  - Write down their responses on the board or chart paper so that you can return to them later in the lesson.
- Make sure they understand that private information includes the following:
  - full name
  - age
  - address
  - telephone number
  - email address (or parents’ email addresses)
  - where they go to school or after school
  - where their parents work
- Encourage students to discuss why it is important to keep this information private.
  - Stress that it is never safe to give out private information to people they don’t know.

    - Students should always ask a parent or caregiver before they give out private information to anyone.
- Refer back to the sign-up page.
    - Ask "Do you think you should use your real name, or something that includes your real name, when you make up a username?"

Guide students through the following rules and tips for creating usernames:

Rules:

- Ask a parent or other trusted adult before you create a username.
- Never include any private information in your username, such as your real name, age, birthday, the name of your school or hometown, parts of your address or phone number, or email address.
- Avoid using symbols or spaces, as they are usually not allowed in usernames.

Tips

- Include the name of something that will help you remember your username, like your favorite animal, character, or toy. You might have to combine this with other words or numbers.
- If the username you create is already taken, you will have to come up with another one.
- Write down your username and password and, with the help of a parent, find a safe place to keep it in case you forget them.

Distribute paper and place students in pairs.

Directions:

1. Have students interview their partner using the following questions, and write down their responses:
2. What is your favorite pet or animal?
3. What is your favorite TV show, book, or movie character?
4. What are your favorite numbers?
5. Instruct students to make up three safe usernames for their partner using information from their interview responses.
6. They should not include their partner's name, age, school, email address, birthday, or any other private information.
7. Invite students to share one or more of their usernames with the class.
8. Encourage students to respond to one another's usernames, confirming that each name follows the rules they have learned.

Lesson Tip

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?

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

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

Appropriate	Not Appropriate
Interests	Address
Hobbies	Full Name

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.

Appropriate	Not Appropriate
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)

Keep It Private - 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.

COURSE A 1 2 3 4 5 6 7 8 9 10 11 12 CODE

Lesson 8: Loops Unplugged: Happy Loops

Unplugged | Loop | Repeat

Overview

Loops are a very helpful and powerful tool in programming. To understand how helpful loops can be, students will need to be driven to want an easier way to solve mundane problems.

Purpose

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

Agenda

Warm Up (10 - 15 min)

- **Such a Long Walk**
- **Discuss**
- **Vocabulary**

Main Activity (15 - 20 min)

- **Happy Loops**

Wrap Up (8 min)

- **Journaling**

Extension Activities

Objectives

Students will be able to:

- Identify repetitive code and convert a series of multiple actions into a single loop.
- Decode loops into a series of multiple actions.

Preparation

- [ ] Print out **Happy Map Cards - Worksheet** for each group
- [ ] Print out **Happy Map Game Pieces - Manipulatives** for each group
- [ ] Print out **Happy Map Cards XL - Worksheet** for each group
- [ ] Print out **Happy Map Game Pieces Bonus Pack - Manipulatives** for each group
- [ ] Make sure each student has a **Think Spot Journal**.

Links

For the Teacher

- **Happy Map Cards** - Worksheet (**PDF** | **DOCX**)
- **Happy Map Cards XL** - Worksheet
- **Happy Map Game Pieces** - Manipulatives (**PDF** | **DOCX**)
- **Happy Map Game Pieces Bonus Pack** - Manipulatives
- **Think Spot Journal**

Vocabulary

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

Teaching Guide

Warm Up (10 - 15 min)

Such a Long Walk

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

Recall the "Happy Maps Programming" activity with your students. Recall the limited pieces that they had to work with (up, down, left, right). Then, pull up one of the new -- and much longer -- **Happy Maps XL**.

Can your students help you program these maps? It takes a while, doesn't it? What can you do if you run out of arrow pieces?

Discuss

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

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

Vocabulary

- **Loop**: the act of doing something over and over and over.
- **Repeat**: doing something again.

Notice: Loops **repeat** a step over and over again.

Main Activity (15 - 20 min)

Happy Loops

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

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

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

Wrap Up (8 min)

Journaling

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

Journal Prompts:

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

Extension Activities

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

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

COURSE A 1 2 3 4 5 6 7 8 9 10 11 12 CODE

Lesson 9: Loops in Collector

Loop | Collector

Overview

Building on the concept of repeating instructions from "Happy Loops," 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)

- **Course A Online Puzzles - Website**

Wrap Up (5 - 10 min)

- **Journaling**

Extended Learning

Objectives

Students will be able to:

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

- [ ] Play through the **Course A Online Puzzles - Website** in stage 9 to find any potential problem areas for your class.
- [ ] (Optional) Pick a couple of puzzles to do as a group with your class.
- [ ] Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**
- [ ] Make sure each student has a**Think Spot Journal**.

Links

For the Teacher

- **Course A Online Puzzles** - Website
- **Unplugged Blockly Blocks (Grades 1 - 2)** - Manipulatives (**PDF** | **DOCX**)
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations (**PDF** | **DOCX**)
- **Think Spot Journal**

Vocabulary

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

Teaching Guide

Warm Up (10 min)

Introduction

Review the Happy Loops activity with your students:

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

Bridging Activity - Loops (10 min)

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

Unplugged Activity Using Paper Blocks

Select an empty Flurb map from the **Happy Map Cards - Worksheet** and give students **Unplugged Blockly Blocks (Grades 1 - 2) - Manipulatives** prefilled with the `collect` command, a `repeat` loop, and the cardinal commands like `E →` (East) and `W ←` (West). Now, have the students program the Flurb from their desks using the paper Blockly blocks to get the Flurbs to collect the fruit. Make sure that they understand that the blocks need to go from top to bottom and they all need to touch!

Previewing Online Puzzles as a Class

Pull a puzzle from the corresponding online puzzles. We recommend puzzle 7. Using arrows from the **Happy Map Game Pieces - Manipulatives** and **Happy Map Game Pieces Bonus Pack - Manipulatives** , have students lay out a pattern that they think will get 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:

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

Course A Online Puzzles - Website

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

Wrap Up (5 - 10 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- How did loops make your program easier to write?
- Draw something that uses loops.

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

▶ **AP** - Algorithms & Programming

Lesson 10: Loops in Artist

Loop | Artist

Overview

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

Purpose

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

Agenda

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

Main Activity (30 min)
- **Course A Online Puzzles - Website**

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

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 **Course A Online Puzzles - Website** in stage 10 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**.

Links

For the Teacher

- **Course A Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations (**PDF** | **DOCX**)
- **Think Spot Journal**

Vocabulary

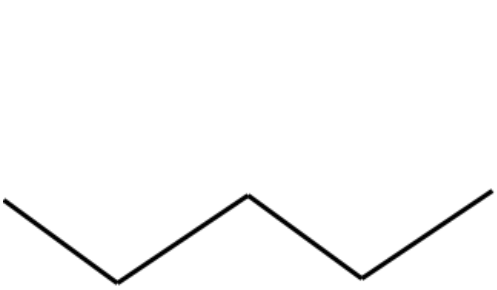
- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again

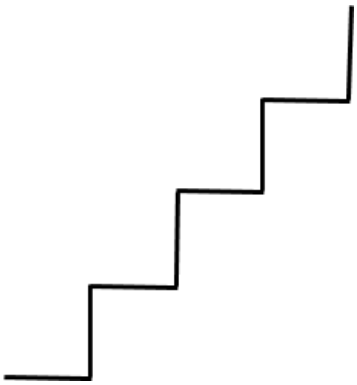
Teaching Guide

Warm Up (10 min)

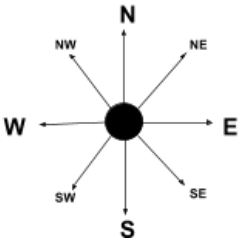
Introduction

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





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



Main Activity (30 min)

Teacher Demonstration

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

Teacher Tip

Remind the students to only share their work with their close friends or family. For more information watch or show the class **Resource not found**.

Prediction Levels:

- **Course A, Loops in Artist**

Course A Online Puzzles - Website

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 some stairs. Imagine the loop needed to draw this.
- Draw something else in your life that uses loops.

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 11: Events Unplugged: The Big Event

Unplugged | 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 (15 min)

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

Assessment (10 min)

- **CSF The Big Event Activity - Assessment**

Extended Learning

Objectives

Students will be able to:

- 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 **CSF The Big Event Activity - Worksheet**.
- [ ] Print **CSF The Big Event Activity - Assessment** for each student.
- [ ] Make sure every student has a**Think Spot Journal**.

Links

For the Teacher

- **The Big Event** - Teacher Video
- **CSF The Big Event Activity** - Worksheet
- **CSF The Big Event Activity** - Assessment
- **Think Spot Journal**

Vocabulary

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

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has one new and important vocabulary word:

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

An action that causes something to happen.

A Series of Events

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

Main Activity (15 min)

The Big Event

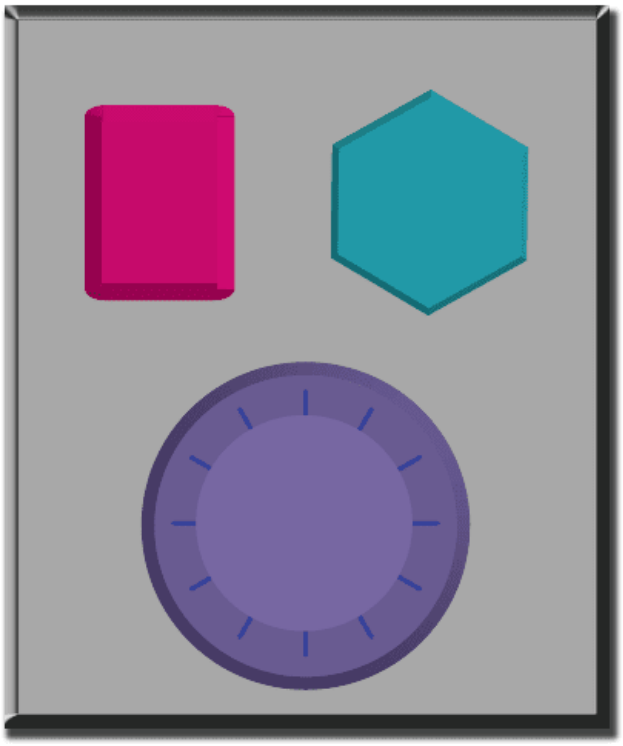
- Do you remember helping the Flurbs find fruit?
  - In that exercise, you knew in advance exactly where you wanted your Flurb to end up, so you could make a program that took them from start to finish without any interruptions.
  - In most real programs, we can't do that because we want to have options,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

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

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

Directions:

- Project the 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 (15 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)

CSF The Big Event Activity - 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

▶ **AP** - Algorithms & Programming

COURSE A 1 2 3 4 5 6 7 8 9 10 11 12 CODE

Lesson 12: Events in Play Lab

Play Lab | Event

Overview

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

Purpose

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

Agenda

Warm Up (10 min)

- **Introduction**

Bridging Activity - Events (10 min)

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

Main Activity (30 min)

- **Course A Online Puzzles - Website**

Wrap Up (5 - 10 min)

- **Journaling**

Extended Learning

Objectives

Students will be able to:

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

Preparation

- [ ] Play through the **Course A Online Puzzles - Website** 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**.

Links

For the Teacher

- **Course A Online Puzzles** - Website
- **Unplugged Blockly Blocks (Grades 1 - 2)** - Manipulatives (**PDF** | **DOCX**)
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations (**PDF** | **DOCX**)
- **Think Spot Journal**

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 **CSF The Big Event Activity - Worksheet** and **Unplugged Blockly Blocks (Grades 1 - 2) - Manipulatives**, gather your class to reprise the activity from the previous lesson. Ask the class "when the teal button is pushed, what do we do?" then fill in one of the `when` event blocks and one of the blue action blocks accordingly. Make sure that the students understand that the `when` blocks need to be on top of the blue block and they need to touch in order for the program to run.

Lesson Tip

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

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

Previewing Online Puzzles as a Class

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

Main Activity (30 min)

Course A Online Puzzles - Website

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

▸ **AP** - Algorithms & Programming