Course F

The last course in CS Fundamentals is tailored to the needs of students in the fifth grade. In these lessons, students will create programs with different kinds of loops, events, functions, and conditionals. They will also investigate different problem-solving techniques and discuss societal impacts of computing and the internet. By the end of the curriculum, students create interactive stories and games that they can share with their friends and family.

Journaling

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

Think Spot Journal **Student Handout**

Debugging

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

Debugging Guide **Student Handout**

Chapter 1: Ramp Up

Lesson 1: Functions in Minecraft

Skill Building | Ramp Up

Can you figure out how to use functions for the most efficient code?

Lesson 2: Swimming Fish with Sprite Lab

Skill Building | Ramp Up

Learn how to create and edit sprites.

Lesson 3: Alien Dance Party with Sprite Lab

Skill Building | Ramp Up

Create an interactive project that can be shared with classmates.

Lesson 4: Drawing with Loops

Skill Building | Ramp Up

In this lesson, loops make it easy to make even cooler images with Artist!

Lesson 5: Nested Loops in Maze

Skill Building | Ramp Up

Loops inside loops inside loops. What does this mean? This lesson will teach you what happens when you place a loop inside another loop.

Chapter Commentary

Ramp Up

Chapter 2: Digital Citizenship

Lesson 6: The Power of Words

Unplugged | Cyberbullying

Bullying is never okay. This lesson will teach you about what is and isn't okay to say online.

Chapter Commentary

Digital Citizenship

Chapter 3: Variables

Lesson 7: Envelope Variables

Unplugged | Variables

Envelopes and variables have something in common: both can hold valuable things. Here you will learn what variables are and the awesome things they can do.

Lesson 8: Variables with Artist

Skill Building | Variables

Don't forget to bring creativity to class! In these puzzles you will be making fantastic drawings using variables.

Lesson 9: Changing Variables with Bee

Skill Building | Variables

This bee loves variables!

Lesson 10: Changing Variables with Artist

Skill Building | Variables

In this lesson, you'll make drawings using variables that change as the program runs.

Chapter Commentary

Variables

Chapter 4: Data

Lesson 11: Simulating Experiments

Unplugged | Data

Run simulations on the computer and experiment by changing variables.

Lesson 12: AI For Oceans

Tutorial Summary: First students classify objects as either "fish" or "not fish" to attempt to remove trash from the ocean. Then, students will need to expand their training data set to include other sea creatures that belong in the water. In the second part of the activity, students will choose their own labels to apply to images of randomly generated fish. This training data is used for a machine learning model that should then be able to label new images on its own.

Lesson 13: The Internet

Unplugged | Internet

Ever wondered how information travels across the internet? It's not magic! This lesson will teach you the basics of how the internet works.

Chapter Commentary

Data

Chapter 5: For Loops

Lesson 14: For Loop Fun

Unplugged | For Loops

You're going to have loads of fun learning about `for` loops!

Lesson 15: For Loops with Bee

Skill Building | For Loops

Buzz buzz. In these puzzles you will be guiding a bee to nectar and honey using `for` loops!

Lesson 16: For Loops with Artist

Skill Building | For Loops

Get ready to make your next masterpiece. Here you will be using `for` loops to make some jaw-dropping pictures.

Chapter Commentary

For Loops

Chapter 6: Sprites

Lesson 17: Behaviors in Sprite Lab

Skill Building | Behaviors

Learn to program your own sprite behaviors!

Lesson 18: Virtual Pet with Sprite Lab

Application | Sprites

In this lesson, students will create an interactive Virtual Pet that looks and behaves how they wish. Students will use Sprite Lab's "Costumes" tool to customize their pet's appearance. They will then use events, behaviors, and other concepts they have learned to give their pet a life of its own!

Chapter Commentary

Sprites

Chapter 7: End of Course Project

Lesson 19: End of Course Project

End of Course Project

Projects this big take time and plenty of planning. Find your inspiration, develop a plan, and unleash your creativity!

Chapter Commentary

End of Course Project

Lesson 1: Functions in Minecraft

Overview

Students will begin to understand how functions can be helpful in this fun and interactive Minecraft adventure!

Purpose

Students will discover the versatility of programming by practicing functions in different environments. Here, students will recognize reusable patterns and be able to incorporate named blocks to call pre-defined functions.

Agenda

Warm Up (10 min)

Introduction

Bridging Activity - Functions (15 min)

Unplugged Activity Using Some Blockly

Preview of Online Puzzles

Main Activity (30 min)

Online Puzzles

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Use functions to simplify complex programs.
- Use pre-determined functions to complete commonly repeated tasks.

Preparation

□ Play through the puzzles to find any potential problem areas for your class.

□ Make sure every student has a journal.

Vocabulary

- **Function** - A piece of code that you can easily call over and over again.

Teaching Guide

Warm Up (10 min)

Introduction

Help the class understand that functions are simply a chunk of code that has a name. Once defined, you can use that name over and over in your program to tell the computer to run the chunk of code that you assigned to it.

Bridging Activity - Functions (15 min)

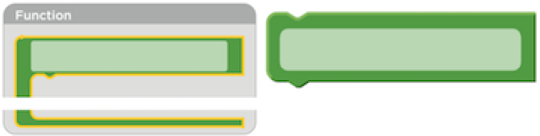
This activity will help bring the unplugged concepts from "Functions Unplugged: Songwriting" into the online world that the students are moving into. Choose *one* of the following to do with your class:

Unplugged Activity Using Some Blockly

Pick a song to play that the students enjoy and print out the lyrics. You can use the same song from "Functions Unplugged: Songwriting." Break your class into groups or pairs. Pass out the printed out lyrics (including the repeated chorus) and the basic function blocks from **Unplugged Blocks (Courses C-F) - Manipulatives** to each group or pair of students. See lesson tip for details.

Lesson Tip

Function blocks:



The block to the left is a function declaration, a block that students will name and use to fill in the function. The block to the right is a function call, a block that makes the function code run. Students will need multiple of the function call blocks.

Ask the students to cross out any part of the song that can be made into a function (the chorus is a good example) and put it into the function blocks provided. Students should fill in the function declaration with a function name and the words of the repeated lyrics. Once the function declaration is done, ask the students to fill in the function calls and place them on top of the crossed out lyrics.

Once every group or pair is done, ask the class where they put their functions and why. Did everyone make the same function? How often is the function repeated?

Preview of Online Puzzles

Pull up a puzzle 9 of this lesson. As a class, work through the puzzle without using functions. Once you have gotten the solution, display it on a white board or overhead. Ask the class to point to the repeated code. Ask the class how they would simplify the program. Why can you not just use a loop?

On the white board or overhead, rewrite the program without the repeated code, but leaving one line space. In that/those line space(s), call a function. Off to the side, declare the function like the left example block in the lesson tip. Ask the class what they think the code will do now.

Open up a discussion with the class on why functions could be useful in programming. Invite students to discuss the difference between functions and loops.

Main Activity (30 min)

Online Puzzles

Code Studio levels

Practice 1 2 3 4 5 6 7 8 9 10

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

Free Play

🖥 **12** ***(click tabs to see student view)***

We recommend providing paper and pencils for students to write (or draw) out ideas. Also, if students are having trouble recognizing patterns, have them work with a partner on the harder puzzles.

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.

- What was today's lesson about?
- How do you feel about today's lesson?
- What did your functions do in the programs you wrote today? How did that help you?
- When should you use a function instead of a loop?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 2: Swimming Fish with Sprite Lab

Overview

In this lesson, students will learn about the two concepts at the heart of Sprite Lab: sprites and behaviors. Sprites are characters or objects on the screen that students can move, change, and manipulate. Behaviors are actions that sprites will take continuously until they are stopped.

Purpose

This lesson is designed to introduce students to the core vocabulary of Sprite Lab, and allow them to apply concepts they learned in other environments to this tool. By creating a fish tank, students will begin to form an understanding of the programming model of this tool, and explore ways they can use it to express themselves.

Agenda

Warm Up (10 min)

Introduction

Main Activity (20 min)

Online Puzzles

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Define “sprite” as a character or object on the screen that can be moved and changed.
- Create new sprites and assign them costumes and behaviors.

Preparation

☐Play through the puzzles to find any potential problem areas for your class.
☐Make sure every student has a journal.

Vocabulary

- **Behavior** - An action that a sprite performs continuously until it’s told to stop.
- **Sprite** - A graphic on the screen with a location, size, and appearance.

Introduced Code

- `set background color`
- `make new sprite`
- `set sprite property`

Teaching Guide

Warm Up (10 min)

Introduction

Today students will learn how to work with sprites in Sprite Lab.

Display: Pull up a previous puzzle from Code.org, ideally one containing a "main character" like Scrat from Ice Age or one of the Angry Birds.

Discuss: Let the students know that this character on the screen is a "sprite." It is a graphic that is controlled by a program. In this lesson, students will have the opportunity to choose their own sprites to control.

Display: Begin by showing Puzzle 1 from today's Online Puzzles to your students.

Think/Pair: Ask them to predict what will happen when the code is run, and to discuss with their neighbors. Run the code, and discuss the outcome.

Before heading into the Main Activity, introduce or review today's lesson vocabulary.

Main Activity (20 min)

Online Puzzles

Goal: Today, students will be programming their own Fish Tank. They’ll begin by learning how to put some sprites on the screen, then they will make them move. Finally, they’ll customize their fish tank to add whatever creatures and objects they want.

💡 **Transition:** Move students to their machines. Encourage students to follow the instructions for each puzzle. Help them realize that this is a creative activity, intended to help them learn Sprite Lab. It is not an assessment activity of any sort.

💡 Teaching Tip

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. Have students describe the problem that they’re seeing:

- What is it supposed to do?
- What does it do?
- What does that tell you?

🖥 Code Studio levels

Introducing Sprite Lab 🎥 1 ***(click tabs to see student view)***

Prediction 🖥 2 ***(click tabs to see student view)***

Mini-Project: Swimming Fish 🖥 3 🖥 4 🖥 5 ***(click tabs to see student view)***

How to Make a Sprite 🎥 6 ***(click tabs to see student view)***

Mini-Project: Swimming Fish (continued) 🖥 7 🖥 8

(click tabs to see student view)

Free Play 🖥 **9** ***(click tabs to see student view)***

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 do you feel about today's lesson?
- How did it feel to make a scene that was more creative?
- Was it difficult to finish a lesson where there was no clear "right" and "wrong"?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 3: Alien Dance Party with Sprite Lab

Overview

This lesson features Sprite Lab, a platform where students can create their own interactive animations and games. In addition to behaviors, today students will incorporate user input as events to create an "alien dance party".

Purpose

Students will use events to make characters move around the screen, make noises, and change backgrounds based on user input. This lesson offers a great introduction to events in programming and even gives students a chance to show creativity! At the end of the puzzle sequence, students will be presented with the opportunity to share their projects.

Agenda

Warm Up (5 min)

- **Introduction**

Main Activity (30 min)

- **Online Puzzles**

Wrap Up (10 min)

- **Journaling**

View on Code Studio

Objectives

Students will be able to:

- Identify actions that correlate to input events.
- Create an interactive animation using sprites, behaviors, and events.

Preparation

☐Play through the puzzles to find any potential problem areas for your class.
☐Make sure every student has a journal.

Vocabulary

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

Introduced Code

- `set background`
- `random location`
- `location picker`
- `change color`
- `remove color`
- `sprite clicked`
- `sprite touches sprite`

Teaching Guide

Warm Up (5 min)

Introduction

Today students will visit *events* in programming.

Demo: Ask the students to raise their hands in the air.

What you did was declare an event. When you say "raise your hands in the air" the students responded by raising their hands. In coding, you would declare this by saying something like "when I say 'raise your hands,' you raise your hands".

You can also think of cities as declaring events. There are laws that say "when there is a green light, cars move through the intersection".

Discuss: Ask the students why they think this is an event.

Today, students will play in Sprite Lab, but the events they will be working on will be more like the video games they are used to playing. Events will take the form of actions, such as clicking the screen or two characters running into each other.

Display: Begin by showing Puzzle 1 to your students.

Think/Pair: Ask them to predict what will happen when the code is run, and to discuss with their neighbors. Run the code, and discuss the outcome.

Main Activity (30 min)

Online Puzzles

Goal: Today, students will be creating their own alien dance party! They'll begin by reviewing how to put sprites on the screen, then they will assign them behaviors and learn to change those behaviors when an event is initiated.

Transition: Move students to their machines. Encourage students to follow the instructions for each puzzle. Help them realize that this is a creative activity, intended to help them learn Sprite Lab. It is not an assessment activity of any sort.

Teaching Tip

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. Have students describe the problem that they're seeing:

- What is it supposed to do?
- What does it do?
- What does that tell you?

Code Studio levels

Prediction 1 ***(click tabs to see student view)***

Video 2 ***(click tabs to see student view)***

Mini-Project: Alien Dance Moves 3 4 5 6

(click tabs to see student view)

Mini-Project: Alien Dance Party 7 8 9

(click tabs to see student view)

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 do you feel about today's lesson?
- How did it feel to have control over what your characters were able to do?
- Did you change the program in any way to make it feel more like your own?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 4: Drawing with Loops

Overview

Watch student faces light up as they make their own gorgeous designs using a small number of blocks and digital stickers! This lesson builds on the understanding of loops from previous lessons and gives students a chance to be truly creative. This activity is fantastic for producing artifacts for portfolios or parent/teacher conferences.

Purpose

This series highlights the power of loops with creative and personal designs.

Offered as a project-backed sequence, this progression will allow students to build on top of their own work and create amazing artifacts.

Agenda

Warm Up (15 min)

Introduction

Main Activity (30 min)

Online Puzzles

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Identify the benefits of using a loop structure instead of manual repetition.
- Differentiate between commands that need to be repeated in loops and commands that should be used on their own.

Preparation

□ Play through the puzzles to find any potential problem areas for your class.

□ Make sure every student has a journal.

Vocabulary

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

Teaching Guide

Warm Up (15 min)

Introduction

Students should have had plenty of introduction to loops at this point. Based on what you think your class could benefit from, we recommend:

- Creating a new stack design with loops just like in "My Loopy Robotic Friends"
- Reviewing how to use Artist by playing through a puzzle from "Programming in Artist"
- Previewing a puzzle from this lesson

All of these options will either review loops or the artist, which will help prepare your class for fun with the online puzzles!

Main Activity (30 min)

Online Puzzles

Some students may discover where to add `repeat` loops by writing out the program without loops then circling sections of repetitions. If the students in your class seem like they could benefit from this, have them keep paper and pencils beside them at their machines. Students might also enjoy drawing some of the shapes and figures on paper before they program it online. (When drawing stamps, it can be easier to symbolize those with simple shapes like circles and squares.)

🖥 Code Studio levels

Artist Intro with JR Hildebrande 🎥 1 ***(click tabs to see student view)***

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

Loops with the Artist 🎥 4 ***(click tabs to see student view)***

Practice 🖥 5 🖥 6 ***(click tabs to see student view)***

Challenge 🖥 7 🖥 8 ***(click tabs to see student view)***

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

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

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

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?
- What was the coolest shape or figure you programmed today? Draw it out!
- What is another shape or figure you would like to program? Can you come up with the code to create it?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 5: Nested Loops in Maze

Overview

In this online activity, students will have the opportunity to push their understanding of loops to a whole new level. Playing with the Bee and Plants vs Zombies, students will learn how to program a loop to be inside of another loop. They will also be encouraged to figure out how little changes in either loop will affect their program when they click `Run`.

Purpose

In this introduction to *nested loops*, students will go outside of their comfort zone to create more efficient solutions to puzzles.

In earlier puzzles, loops pushed students to recognize repetition. Here, students will learn to recognize patterns *within* repeated patterns to develop these *nested loops*. This stage starts off by encouraging students try to solve a puzzle where the code is irritating and complex to write out the long way. After a video introduces *nested loops*, students are shown an example and asked to predict what will happen when a loop is put inside of another loop. This progression leads into plenty of practice for students to solidify and build on their understanding of looping in programming.

Agenda

Warm Up (10 min)

Introduction

Main Activity (30 min)

Online Puzzles

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Break complex tasks into smaller repeatable sections.
- Recognize large repeated patterns as made from smaller repeated patterns.
- Identify the benefits of using a loop structure instead of manual repetition.

Preparation

□ Play through the puzzles to find any potential problem areas for your class.
□ Make sure every student has a journal.

Vocabulary

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

Teaching Guide

Warm Up (10 min)

Introduction

Briefly review with the class what loops are and why we use them.

- What do loops do?
  - Loops repeat a set of commands. (see vocabulary on command if students don't recognize it)
- How do we use loops?
  - We use loops to create a pattern made of repeated actions.

Tell the class that they will now be doing something super cool: using loops inside loops. Ask the class to predict what kinds of things we would be using a loop inside of a loop for.

"If a loop repeats a pattern, then looping a loop would repeat a pattern of patterns!"

Students don't need to understand this right away, so feel free to move on to the online puzzles even if students still seem a little confused.

Main Activity (30 min)

Online Puzzles

We highly recommend **Pair Programming - Student Video** in this lesson. This may not be an easy topic for the majority of your students. Working with a partner and discussing potential solutions to the puzzles might ease the students' minds.

Also, have paper and pencils nearby for students to write out their plan before coding. Some puzzles have a limit on the number of certain blocks you can use, so if students like to write out the long answer to find the repeats, paper can be useful.

🖥 Code Studio levels

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

Nested Loops with the Bee 📹 3 ***(click tabs to see student view)***

Prediction 🖥 4 ***(click tabs to see student view)***

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

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

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

Prediction 🖥 13 ***(click tabs to see student view)***

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 about today's lesson?
- What is a nested loop?
- Can you draw a puzzle that would use a nested loop? Try coding the solution to your own puzzle.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 6: The Power of Words

Overview

common sense education®

This lesson was originally created by **Common Sense Education**.

As kids grow, they'll naturally start to communicate more online. But some of what they see could make them feel hurt, sad, angry, or even fearful. Help your students build empathy for others and learn strategies to use when confronted with cyberbullying.

Purpose

Common Sense Education created this lesson to teach students what they should do when someone uses mean or hurtful language on the internet.

Agenda

Warm Up: Same Word. Different Meaning. (5 min)

Key Vocabulary

Watch: The Power of Words (35 min)

Investigate: What's the Problem? (15 min)

Wrap Up: Crossing the Line (10 min)

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Understand that it's important to think about the words we use, because everyone interprets things differently.
- Identify ways to respond to mean words online, using S-T-O-P.
- Decide what kinds of statements are OK to say online and which are not.

Preparation

□Review instructional materials.
□Print handout(s) for each student.
□Prepare a long string (or rope) and paper for class.

Links

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

For the Teachers

- **The Power of Words: Lesson Slides** - Slide Deck
- **The Power of Words: The Power of Words** - Lesson Video (**download**)
- **The Power of Words: Lesson Quiz Answer Key** - Website

For the Students

- **The Power of Words: Words Can Hurt** - Student Handout Make a Copy ▾
- **The Power of Words: Lesson Quiz** - Form

Teaching Guide

Warm Up: Same Word. Different Meaning. (5 min)

Key Vocabulary

- **empathy**: to imagine the feelings that someone else is experiencing
- **interpret**: to understand something based on our point of view

Before the lesson: As an optional activity before the lesson, have students play the **E-volve** game in Digital Passport™ by Common Sense Education. This will help introduce key concepts of this lesson. To see more, check out the **Digital Passport Educator Guide**.

Project Slide 4 and ask students: *Imagine when you walked in, I said to you* [in a friendly tone, with a big smile] *"Hello!" Which emoji would fit?*

Have students show their answer by holding up fingers (1, 2, 3, or 4). If anyone holds up a number other than 1, ask them to share out why.

Repeat this two more times, saying *Hello* differently each time: once with an angry tone and a scowl, and once with a nervous or scared tone.

Project Slide 5 and ask: *What about if you're just reading it? How does it make you feel now?*

Point out how when something is just in text, it can be even more difficult to interpret what someone means. Explain that their responses might be different depending on who they are because we all interpret the world a little differently. Clarify that interpret means *to understand something based on our point of view.* (**Slide 6**)

Say: *Because words can be interpreted differently by different people, it's important to think about the words we use and how they affect people. We might say something just to be funny, but someone might take it seriously, and it might hurt their feelings. This can happen in person, and it can also happen online. It may even happen more online, because the words are just text -- we're not actually face-to-face with other people.*

Watch: The Power of Words (35 min)

Ask: *Who has heard of the saying "Sticks and stones may break my bones, but words can never hurt me"?*

Invite students to respond. Answers will vary. Follow up by asking students if they think the saying is true.

Project Slide 7 and say: *We're going to watch a video now called "The Power of Words." As we watch, continue to think about whether that statement is true or not.*

Show the video **The Power of Words** and then call on students to share out what it says about whether the statement is true or not. Emphasize that sometimes it is hard to ignore what someone is saying when it's a mean name. Names CAN make you feel sad or hurt.

Ask: *Why do you think those other players said those things to Guts?* (**Slide 8**)

Invite students to respond. Answers will vary, but possible reasons include *they were trying to joke around* or *they feel badly themselves and are taking it out on someone*.

Ask: *What does Guts do in response to the mean words?*

Invite students to respond and clarify that Guts talked to someone he trusted, who was able to empathize with him and give him some good advice. Define empathy as *imagining the feelings that someone else is experiencing*. (**Slide 9**)

Project Slide 10 and review the acronym S-T-O-P for responding to mean words online.

Investigate: What's the Problem? (15 min)

Distribute the **Words Can Hurt Student Handout** and read the directions aloud. Allow pairs five minutes to complete the activity.

Invite pairs to share their answers. Refer to the **Teacher Version** for possible responses.

Wrap Up: Crossing the Line (10 min)

Place a string across the length of the classroom. Mark one half of the room "OK." Mark the other half "Not OK." Have students all stand on the OK side of the line.

Project Slide 11 and read the directions aloud.

Have students read the statements on **Slides 12-15**. After each statement, allow time for students to move or stay. Call on students to explain their thinking. If possible, choose students from each side to debate reasons why the statement is or isn't okay.

For the examples where students say it's Not OK, follow-up by asking them what they could do if they received that message from someone.

Review that kids like to go online to chat, watch videos, send messages, play games, and do homework. Sometimes people say mean or scary things. Messages that make people feel bad cross the line.

Have students complete the **Lesson Quiz**.

Extended Learning

Here are additional resources you can provide students to enhance their learning:

- **Family Activity**
- **Family Tips**
- **Prepare** slips of paper with comments that say mildly unkind things, like:
    - I can't believe you like superhero shows. Only first-graders do.
    - You are weird.
    - Why are you wearing THAT shirt?

  Pass out one slip to each student and pair them up. Have students trade slips and read them silently. Then have them trade back and take turns saying the comment aloud to the other person. Ask them what the difference was between writing something not-so-nice and being accountable for what they were saying to someone's face. Emphasize that people sometimes hide behind the anonymity of their online activity. Remind students that if they wouldn't say something to another person's face, it's not OK to say it in their online activities.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **NI** - Networks & the Internet

Lesson 7: Envelope Variables

Overview

Variables are used as placeholders for values such as numbers or words. Variables allow for a lot of freedom in programming. Instead of having to type out a phrase many times or remember an obscure number, computer scientists can use variables to reference them. This lesson helps to explain what variables are and how we can use them in many different ways. The idea of variables isn't an easy concept to grasp, so we recommend allowing plenty of time for discussion at the end of the lesson.

Purpose

Variables are very helpful in programming. Students will be introduced to this topic using envelopes to represent variables that have been given names. The value of the variable will be written on a card inside of an envelope. This lesson helps students understand how names can be a placeholder for values in the physical world, so that programming with variables will seem less confusing in the virtual world.

Agenda

Warm Up (10 min)

- **Vocabulary**
- **Introduction**

Main Activity (20 min)

- **Envelope Variables - Worksheet**

Wrap Up (10 min)

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

Assessment (10 min)

- **Envelope Variables - Assessment**

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Identify variables and determine their values.
- Define and call variables in the context of real-life activities.
- Create situations which require the use of variables.

Preparation

□Obtain 6 or more blank envelopes for warm up plus some for the main activity.
□Print one worksheet per student.
□Print one assessment for each student.
□Provide students with envelopes, paper, pens & pencils.
□Make sure every student has a journal.

Links

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

For the Teachers

- **Variables in Envelopes** - Unplugged Video (**download**)
- **Envelope Variables** - Worksheet Answer Key Make a Copy ▾
- **Envelope Variables** - Assessment Answer Key Make a Copy ▾

For the Students

- **Envelope Variables** - Worksheet Make a Copy ▾
- **Envelope Variables** - Assessment Make a Copy ▾

Vocabulary

- **Variable** - A label for a piece of information used in a program.

Teaching Guide

Warm Up (10 min)

Vocabulary

This lesson has one important word:

- **Variable** - Say it with me: Vayr-ee-ah-buhl

A placeholder for a piece of information that can change.

Introduction

Call four volunteers to the front of the room and line them up. Let the students know that you are going to write a poem for each of them.

On the board (or under your document camera) write the sentence for your first student (suppose it's Bill):

"My student Bill, standing proud
is a fine example for the crowd"

Encourage the students to clap at your abilities and thank Bill for volunteering. Allow Bill to sit down (or go to the back of the line) as you erase the board, then call the next volunteer (we'll say that she's called Annie).

"My student Annie, standing proud
is a fine example for the crowd"

Again, accepting applause, erase the board and invite the next volunteer.

"My student Jenny, standing proud
is a fine example for the crowd"

As you call the final volunteer, inquire as to whether everyone in the class would like a poem written about each of them. Maybe the everyone in the whole school? Goodness, that's going to take a while! Pose the question to your students:

"How could I do this more quickly?"

Your students will likely pick up on the fact that only one word is changing, and that word is simply a person's name. Help them see the location by circling Jenny's name on the board and writing "firstName" next to it.

"It would take a long time to write a poem for everyone in the school if I couldn't start until I knew who I was writing it about, wouldn't it?"

- How long do you think it would take to make a video game if they couldn't start until they knew your username?
- How expensive would video games be if they had to be created separately for each person?
- How do you think we can get around that?

By this time, it's quite likely that your class will come up with the idea of having a placeholder. With that, they're most of the way into understanding where this lesson goes.

- What would we call that placeholder?
    - We need to call it something that makes sense. We wouldn't want to call it "age" if it was a placeholder for their name, right?

Now, let's add some more volunteers. Give them each a piece of paper to write their name on, and have them tuck it inside individual envelopes labeled firstName.

This time, put the poem on the board with a blank space labeled "firstName" where the student's name will go.

- Have the first student in line (likely the last student from the previous example) pull their name from the envelope and that's what you'll write in the space.
- When you erase the board, only erase the portion with the last student's name in it.
- Call the next student to show their variable.

- Repeat as many times as is entertaining

Now it's time for the main activity.

Main Activity (20 min)

Envelope Variables - Worksheet

Once the students understand how the envelopes relate to the sentences, pass out the activity worksheet and let them prepare some variables of their own.

Directions:

- Divide students into groups of 2-4.
- Have students design (draw) a robot.
- After 10-15 minutes, request that the students fill their envelopes with important details about their robot such as its name, height, and purpose.
- Collect each group's envelopes, then bring them to the front of the room to share.
- Write on the board, "My robot's name is robotName, it is numUnitsTall tall, and it's purpose is purpose."
- Use the envelopes to fill the appropriate variable in the sentence, then ask each group to stand when they hear the sentence that describes their creation.

Wrap Up (10 min)

Flash Chat: What did we learn?

- What did we learn today?
- Can you think of anywhere that you have seen variables before?
- There is at least one variable at the top of most homework hand outs? Can you think of what it could be?
- Why do you think that professionals do not put spaces in variable names?
  - What would happen if there was a variable "eye" a variable "color" and a variable "eye color"?
- Variables can be used to store numbers, too.
  - Suppose I have envelopes labeled num1 and num2, then I write num1+num2?
  - What happens if the "num1" envelope contains the number 4 and "num2" contains the number 5?

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 do you feel about today's lesson?
- What is a variable?
- Why do you think variables are important in programming?

Assessment (10 min)

Envelope Variables - Assessment

Allow students enough time to finish this assessment. If you are willing to spare more time, go over the answers as a class.

Extended Learning

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

What's in the box?

- Draw boxes on a piece of paper with simple mathematical operators between them.
  - For instance [] + [] = []
- Have similar size squares with numbers between 1 & 20.
- Ask one student to come create a true equation, using the numbers provided.
- Once the student has finished (and the class verifies the equation) exchange one of the numbers with another one, then remove a second number entirely.
  - Tell the students that there is a hidden number in the empty box that makes that equation true again.
  - What number is in the box?
- Play this game over and over again until you can remove the number from any location and the students can figure out what it is supposed to be.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 8: Variables with Artist

Overview

In this lesson, students will explore the creation of repetitive designs using variables in the Artist environment. Students will learn how variables can be used to make code easier to write and easier to read, even when the values don't change at runtime.

Purpose

This stage teaches the most basic use for variables, as a constant that reoccurs frequently in a program.

Agenda

Warm Up (15 min)

- **Introduction**

Bridging Activity - Variables (15 min)

- **Unplugged Activity Using Variables as Constant**
- **Preview of Online Puzzles as a Class**

Main Activity (30 min)

- **Online Puzzles**

Wrap Up (15 min)

- **Journaling**

View on Code Studio

Objectives

Students will be able to:

- Assign values to existing variables.
- Utilize variables in place of repetitive values inside of a program.

Preparation

□ Play through the puzzles to find any potential problem areas for your class.
□ Make sure every student has a journal.

Links

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

For the Students

- **Variables** - Student Video (**download**)
- **Unplugged Blocks (Courses C-F)** - Manipulatives

Vocabulary

- **Constant** - A variable used throughout a program that never changes value
- **Variable** - A label for a piece of information used in a program.

Teaching Guide

Warm Up (15 min)

Introduction

This is the first online lesson dealing with variables, so it might be a worthwhile exercise to review the "Envelope Variables" unplugged activity from last time, as well as the vocabulary that was introduced in that lesson.

Discussion:

- What is a variable? (A placeholder for a piece of information that can change.)
- When can a variable be helpful? (When you don't know what information is going to be used in a certain place until runtime, or when you have lots of places that one piece of information will be used, but that information might change someday.)

Ask the class when they could see a variable being helpful in programming. When would they NOT want to use a variable?

If the class seems interested, continue the discussion. Otherwise, move on to one of the bridging activities.

Bridging Activity - Variables (15 min)

This activity will help bring the unplugged concepts from "Envelope Variables" into the online world that the students are moving into.

Unplugged Activity Using Variables as Constant

Discuss: Remember our robot article? It used each variable only once...but what if I wanted to mention the robot's name several times?

Display: Write a paragraph on the board that refers back to the name of a specific robot several times. Ask the students what happens when you need to make the article about a different robot.

Think/Pair: Ask students to work together to see if they can come up with an idea to make changing the article for each robot easier.

Share: Work with students until you eventually get to the place where you have defined a variable called **robotName** somewhere before the paragraph, then set **robotName** equal to the robot that you are writing the article about. Replace all specific naming instances for the previous robot with the variable **robotName**.

Preview of Online Puzzles as a Class

Demo: Display a puzzle for the class. We recommend the 6th puzzle. Go over the code with the students to make sure they understand what's happening before they help you convert the code to use variables. Can they think of something that might happen that would make them really glad that they used variables instead of hardcoded numbers?

Transition: Now it's time for your students to move to their own machines and get started!

Main Activity (30 min)

Online Puzzles

Notice that this stage first covers the idea of a variable as a constant (a variable that you use in many places, but it does not change.) This might be something that students find helpful as they're creating their own projects.

Watch out for puzzle #6. It is the first time that students will be expected to set a variable on their own. This can be tricky if they don't have a true grasp on the concept. If they're having trouble, send them back to the prediction level (#5) and have them explain to their partners why the answer ended up what it was. Once both partners are convinced, let them continue back to puzzle #6.

🖥 Code Studio levels

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

Artist: Variables 🎥 **3** ***(click tabs to see student view)***

Prediction 🖥 **4** ***(click tabs to see student view)***

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

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?
- What is a variable? Why is it helpful in programming?
- How well do you think you understand variables? (Answer on a scale from 1-5 or with an emoticon.) If you're having troubles, can you put into words what you don't understand?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 9: Changing Variables with Bee

Overview

This lesson will help illustrate how variables can make programs more powerful by allowing values to change while the code is running.

Purpose

You don't always know what a value is going to be before you begin your program. Sometimes, values change while your code is running. This lesson will illustrate how code with changing values can be helpful.

Agenda

Warm Up (15 min)

Introduction

Main Activity (30 min)

Online Puzzles

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Identify areas where they can use variables to modify quantities during runtime.
- Examine code to find places where variables can be substituted for specific values.

Preparation

☐Play through the puzzles to find any potential problem areas for your class.
☐Make sure every student has a journal.

Vocabulary

- **Variable** - A label for a piece of information used in a program.

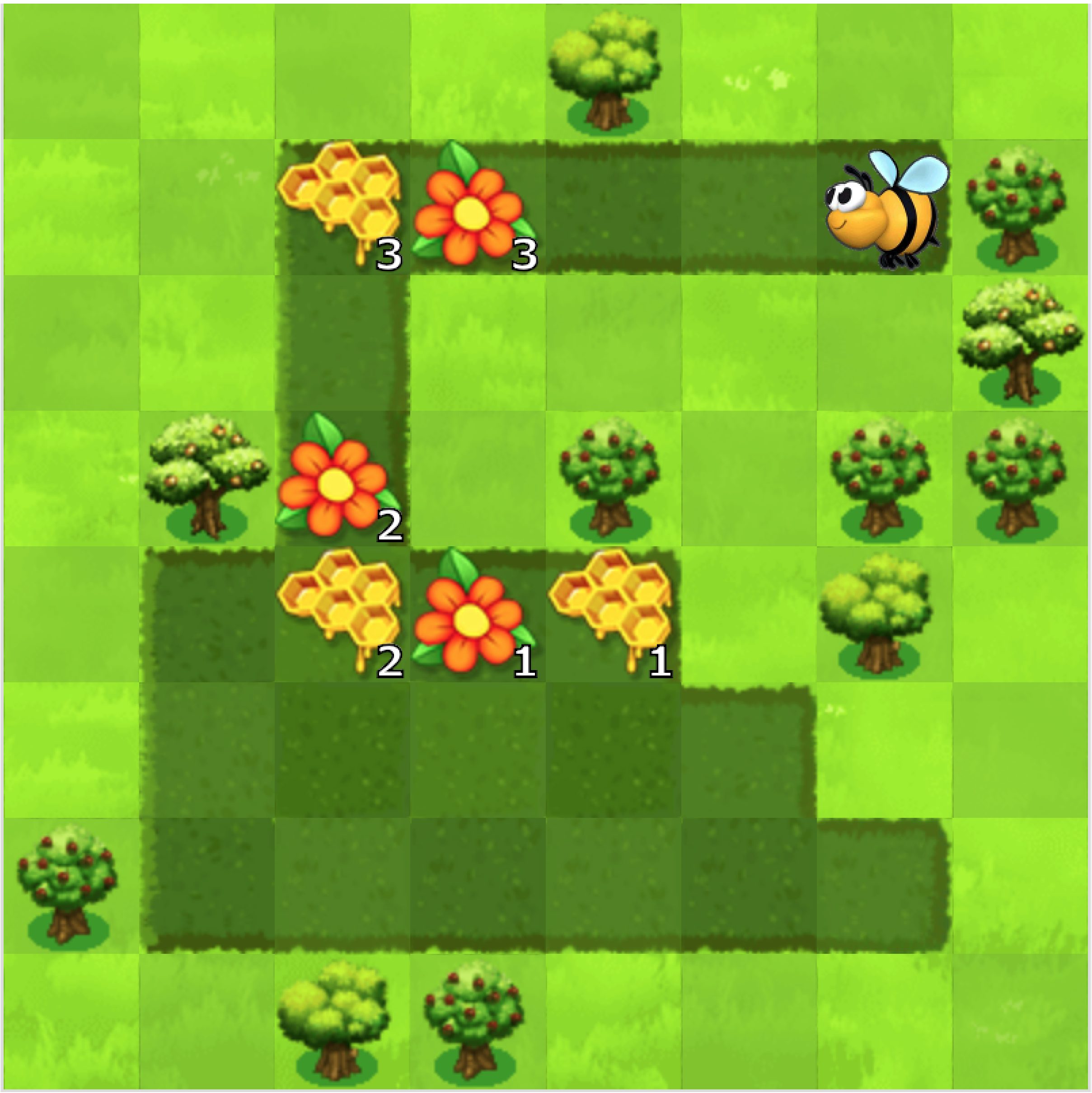
Teaching Guide

Warm Up (15 min)

Introduction

This series is a little different than what students have done in the past. Now, instead of simply assigning a value to a variable and running your code, you'll need to help students see how a variable can be modified during program runtime.

Display Show students the play area from one of the later puzzles.



There are several things to unpack here, so you might need to give your students a chance to look at it critically before you expect them to do anything with it.

Think/Pair: Suppose all of the flowers/honeycomb in this picture had the same amount of nectar/honey. How would you solve this puzzle?

Share: Let students share their ideas until you decide on one that the entire class is good with.

Discuss: Now, imagine that we didn't want to have to write a separate chunk of code for each set of flowers and honeycomb. How could we use a variable to have our loop do this for us?

(Eventually, you'll want to get to the place where you initialize a variable to the original value, then change it each time through the loop so that it's ready for the next time.)

Main Activity (30 min)

Online Puzzles

🖥 Code Studio levels

Practice 🖥 **1** 🖥 **2** 🖥 **3** 🖥 **4** 🖥 **5** ***(click tabs to see student view)***

Challenge 🖥 **6** ***(click tabs to see student view)***

Practice 🖥 **7** ***(click tabs to see student view)***

This set of puzzles takes some serious computational thinking skills. If you find that students are getting stuck, help them break down the puzzles into the individual pieces:

- What would it look like if the flowers/honeycomb all had the same amount of nectar/honey?
- What would it look like without the functions?
- Now how can you use a variable to get the quantities the way you want them?
- Now can you build it back up to use a function?

Hint: Puzzle 7 becomes much easier if students utilize the `while path ahead` loop instead of a variable.

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?
- What are some ways you have used variables so far?
- What else do you think you can do with variables?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 10: Changing Variables with Artist

Overview

In this lesson, students will explore the creation of repetitive designs using variables in the Artist environment. Students will learn how variables can be used to make code easier to write and easier to read. After guided puzzles, students will end in a freeplay level to show what they have learned and create their own designs.

Purpose

Variables are essentially placeholders for values that might be unknown at the time that you run your program or for values that can change during the execution of a program. These are vital to creating dynamic code because they allow your program to change and grow based on any number of potential modifications. This stage reinforces the use of variables, using the most basic capabilities of setting and using them.

Agenda

Warm Up (5 min)
- **Introduction**

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

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

View on Code Studio

Objectives

Students will be able to:

- Assign values to existing variables.
- Utilize variables in place of repetitive values inside of a program.
- Use variables to change values inside of a loop.

Preparation

□Play through the puzzles to find any potential problem areas for your class.
□Make sure every student has a journal.

Links

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

For the Students

- **Variables** - Student Video (**download**)
- **Unplugged Blocks (Courses C-F)** - Manipulatives

Vocabulary

- **Variable** - A label for a piece of information used in a program.

Teaching Guide

Warm Up (5 min)

Introduction

It might be helpful to remind students of "Variables that Change in Bee," since variables will be used in a similar way here.

- How can we change the value of a variable inside of a loop?
- Do we have to change the value of a variable only by one each time?

Main Activity (20 min)

Online Puzzles

The latter half of this series is made up of freeplay puzzles. Students will have the opportunity to play with variables, shape, and color to create something unique.

🖥 Code Studio levels

Practice 🖥 1 🖥 2 🖥 3 🖥 4 ***(click tabs to see student view)***

Free Play 🖥 5 ***(click tabs to see student view)***

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

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?
- Have you tried mixing multiple variables into one program? What might that look like? When would it be helpful?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 11: Simulating Experiments

Overview

By running a simple simulation in Sprite Lab, students will experience how computing can be used to collect data that identify trends or patterns. After running the simulation multiple times, students will have an opportunity to make a prediction about how changing a variable in the simulation might impact the outcome, and then test that hypothesis.

Purpose

Sciences in many disciplines use Computer Science to run models and simulations to run experiments, collect data, and analyze that data for insights. Though the simulation introduced in this lesson is quite simplistic, it can be used as a jumping off point for students to consider how more sophisticated computational models could be used to test hypotheses.

Agenda

Warm Up (5 min)

Why Simulate an Experiment

Activity (35 min)

Simulating Experiments

Wrap Up (5 min)

Reflection

View on Code Studio

Objectives

Students will be able to:

- Use a computer simulation to collect data about a model
- Create double line graph to compare data about two different sources
- Make and test a prediction by modifying simulation variables

Preparation

□Determine whether students will run simulations on their own computer, or if you will be running them as a whole class
□If necessary, prepare to project the two simulations in this lesson.

Links

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

For the Students

- **Running Simulations** - Worksheet
  Make a Copy ▾

Vocabulary

- **Models and Simulations** - a program which replicates or mimics key features of a real world event in order to investigate its behavior without the cost, time, or danger of running an experiment in real life.

Teaching Guide

Warm Up (5 min)

Why Simulate an Experiment

Prompt: If you were a scientist, when might you want to simulate an experiment on a computer instead of in real life? Encourage students to consider experiments that might be dangerous or difficult to reproduce in real life.

Say: Today we're going to run a simulation to gather data about an experiment, just like scientists do.

Activity (35 min)

Simulating Experiments

Distribute: Pass out the **Running Simulations - Worksheet** along with pencils and coloring instruments of some kind. Direct students to Code Studio or project the level on the board.

Running Simulations

> **Step 1 Gather Data**: The first bubble is a simple simulation of an elephant and a hippo collecting apples. Students will run the simulation at least 5 times, recording the number of apples collected by each animal and the total time it took to run the simulation.
>
> **Step 2 Line Graph**: Using the data collected in the previous step, students will plot a double line graph. Encourage students to share their graphs with neighbors and discuss the following questions:
>
> - Are the graphs the same or different? Why do you think that is?
> - Do you see any patterns?
> - What do you think you'd see if you ran this simulation 5 more times?
>
> **Step 3 Modify and Predict**: In the second bubble students will be able to modify some of the variables that control this simulation. Each student will choose a variable to modify, predict how that change will impact the simulation. Potential variables to modify include:
>
> - Number of elephants
> - Number of hippos
> - Number of apples
> - Speed of elephants
> - Speed of hippos
>
> **Step 4 Collecting More Data**: With the modifications students decided on in the last step, they'll run the simulation five more times and collect the data.
>
> **Step 5 Visualize Your Data**: This step is purposefully left open to allow students to explore any visual representation of their data that they'd like. The goal of this visual should be to prove or disprove their prediction.

Share: After finishing, students can share their results with a neighbor. When the whole class is ready, bring everyone together.

Wrap Up (5 min)

Reflection

Discuss: Ask students to discuss how their modification and prediction came out, first with a neighbor and then as a whole class. Encourage students to consider *why* they made the modifications they did and how that change connected to their predictions.

Journal: What is one interesting thing that you might simulate using a computer? What kinds of variables would you want to control in that experiment?

Discussion Goal

This simulation used in this lesson is purposefully broad and kind of silly. Through this discussion, students should start to connect the predictions and variable changes with actual scientific hypotheses. From there you can encourage students to think about computational modeling or more authentic scenarios or experiments.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **DA** - Data & Analysis

Lesson 12: AI For Oceans

Overview

Tutorial Summary: First students classify objects as either "fish" or "not fish" to attempt to remove trash from the ocean. Then, students will need to expand their training data set to include other sea creatures that belong in the water. In the second part of the activity, students will choose their own labels to apply to images of randomly generated fish. This training data is used for a machine learning model that should then be able to label new images on its own.

Checking Correctness: This tutorial will not tell students whether they completed the level correctly. It is possible to skip through the different parts of the activity quickly. Encourage students to watch the videos, read the instructions, and try different things along the way. At any time, they can share their findings with you or a classmate.

Purpose

This tutorial is designed to quickly introduce students to **machine learning**, a type of **artificial intelligence**. Students will explore how **training data** is used to enable a machine learning model to classify new data. Students should have a positive experience during the tutorial and more importantly should be motivated to keep learning computer science.

Agenda

Warm Up

Build Excitement!

Activity

General Support
Level 1 - Machine Learning
Levels 2-4 - Train AI to Clean the Ocean
Level 5 - Training Data & Bias
Levels 6 - Using Training Data
Level 7 - Impacts on Society
Level 8 - Teach A.I. a new word

Extended Learning

Help Classify Animals at Mountain Zebra National Park

Wrap Up

Celebrate and Keep Going!

View on Code Studio

Objectives

Students will be able to:

- Discuss the role artificial intelligence plays in their lives.
- Train and test a machine learning model.
- Reason about how human bias plays a role in machine learning.

Links

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

For the Students

- **AI: Machine Learning** - Video (**download**)
- **AI: Training Data & Bias** - Video (**download**)
- **AI: Impact on Society** - Video (**download**)

Vocabulary

- **Machine Learning** - How computers recognize patterns and make decisions without being explicitly programmed

Teaching Guide

Warm Up

Build Excitement!

Motivate: Explain to students the goals of today's activity. They are going to start using a new tool that will let them train a real machine learning model, a form of artificial intelligence.

Video: The first level of this activity is a video that gives important context around artificial intelligence and machine learning. Watch it as a class and debrief afterwards to help students build connections to the content.

Activity

General Support

General Support: As a teacher your role is primarily to support students as they make their way through the tutorial. Here are a few tips that should help students regardless of the level they're working on

- **Collaborate with Neighbors:** Encourage students to check in with a neighbor to discuss what they are experiencing. Since this tutorial includes videos and students may be wearing headphones it can get easy to "go into a bubble". Help break those barriers by actively pairing students.
- **Read the Instructions:** The instructions usually provide helpful information about what is happening behind the scenes.
- **Go back and try different things:** If students finish quickly, encourage them to go back to "Train More". In the last part of the activity, students can also go back and choose a "New Word". More training data tends to make the machine learning model more accurate and consistent. Students can also learn by purposefully training their model incorrectly, or not training it at all.

Level 1 - **Machine Learning**

Video: **AI: Machine Learning - Video**

Quick Share-out: Where have you seen or experienced artificial intelligence in your lives? Examples from the video include:

- email filters
- auto-complete text
- video recommendation systems
- voice recognition
- translation apps
- digital assistants
- image recognition

Prompt: Based on what you saw in the video, what is machine learning?

Discuss: Beginning in small groups then moving to whole class, students share their responses.

Teaching Tip

You can share these stories with your class to help them see how AI will impact the future.

- **Food Waste Is a Serious Problem. AI Is Trying to Solve It**
- **AI tech can identify genetic disorders from a person's face**
- **How an AI Startup Designed a Drug Candidate in Just 46 Days**
- **MIT AI tool can predict breast cancer up to 5 years early**
- **The Army steps up its pace on self-driving cars**
- **San Francisco says it will use AI to reduce bias when charging people with crimes**
- **AI is helping scholars restore ancient Greek texts on stone tablets**

Say: Machine learning refers to a computer that can recognize patterns and make decisions on its own based on data. In this activity you’re going to give the computer data to train it. Imagine an ocean that contains creatures like fish, but also contains trash dumped by humans. What if we could train a computer to tell the difference and then use that technology to help clean the ocean?

Levels 2-4 - **Train AI to Clean the Ocean**

Students can work through the first three levels on their own or with a partner. To program A.I., use the buttons to label an image as either "fish" or "not fish". Each image and label becomes part of the data used to train A.I. to do it on its own. Once trained, A.I. will attempt to label 100 new images on its own, then present a selection that it determined have the highest probability of being "fish" based on its training. Students who consistently label things correctly should see an ocean full of different types of sea creatures, without much (or any) other objects.

Discussion Goal

Goal: Get students familiar with the world of artificial intelligence.

Content Corner

Every image in this part of the tutorial is fed into a neural network that has been pre-trained on a huge set of data called **ImageNet**. The database contains over 14 million hand-annotated images. ImageNet contains more than 20,000 categories with a typical category, such as "balloon" or "strawberry", consisting of several hundred images. When A.I. is scanning new images and making its own predictions in the tutorial, it is actually comparing the possible categories for the new image with the patterns it found in the training dataset.

Level 5 - **Training Data & Bias**

Quick Share-out: How well did A.I. do? How do you think it decided what to include in the ocean?

Video: **AI: Training Data & Bias - Video**

Prompt: How do you think your training data influence the results that A.I. produced?

Discuss: In small groups, students share their responses. Circulate the room and listen to student ideas. This can be followed with full class discussion, or students can jump right back into the tutorial.

Discussion Goal

Goal: Get students to reflect on their experience so far. It is important at this point that they realize the labeling they are doing is actually programming the computer. The examples they show A.I. are the "training data".

Levels 6 - **Using Training Data**

In the second half of the activity, students will teach A.I. about a word of their choosing by showing it examples of that type of fish. As before, A.I. doesn't start with any training data about these labels. Even though the words in this level are fairly objective, it's possible that students will end up with different results based on their training data. Some students may even intentionally train A.I. incorrectly to see what happens. If students are reflecting on how machine learning works, it should be encouraged!

Content Corner

The fish in this tutorial are randomly generated based on some pre-defined components, including mouths, tails, eyes, scales, and fins, with a randomly chosen body color, shape, and size. Rather than looking at the actual image data, A.I. is now looking for patterns in these components based on how the student classifies each fish. It will be more likely to label a fish the same way the student would have if it has matching traits.

Level 7 - **Impacts on Society**

Video: **AI: Impact on Society - Video**

Say: Artificial intelligence systems learn from the data we give it, but sometimes we might not give it enough data or we might give it data that makes it act strangely.

Say: Think back to the examples of artificial intelligence we discussed at the beginning. Think of a time where machine learning might have got something wrong in the real world? *(For example, voice recognition fails to understand you.)*

Prompt: Could training data actually create problems? How/

Discussion Goal

Goal: The goal of this discussion is to bring students back to the context of artificial intelligence in the real world.

Discuss: Beginning in small groups then moving to whole class, students share their responses.

Say: Some ways to fix this are by using a lot of training data, and making sure we understand the problem well ourselves so we give the right kinds of data. In the final part of the activity you’re going to teach A.I. a word that could be interpreted in different ways.

Level 8 - **Teach A.I. a new word**

Here, as before, students will use training data to teach A.I. to recognize different types of fish. The words in this list are intentionally more subjective than what students will have seen so far. Encourage students to decide for themselves what makes a fish look "angry" or "fun". Two students may choose the same label and get a very different set of results based on which fish traits were their focus. Encourage students to discuss their findings with each other or go back and choose new words. Each student will rely on their own opinions to train A.I. which means that A.I. will learn with the same biases held by the students. As students begin to see the role their opinion is playing, ask them to reflect on whether this is good or bad, and how it might be addressed.

Open question: How could artificial intelligence be used to solve a problem in the world?

Extended Learning

Help Classify Animals at Mountain Zebra National Park

Snapshot Safari has placed hundreds of hidden cameras across southern Africa, capturing millions of images of beautiful and rare animals. Students can help protect the endangered Cape Mountain Zebra by classifying the different animals in these images. You can read about the project **here** or click below to give it a try!

Snapshot Mountain Zebra - Zooniverse

Wrap Up

Celebrate and Keep Going!

Print Certificates: Print certificates for students to celebrate their achievement.

code.org/learn: At the end of the lesson encourage students to head to Code.org/learn where they can get ideas for more ways to keep their adventure with computer science going.

Lesson 13: The Internet

Overview

Even though many people use the internet daily, not very many know how it works. In this lesson, students will pretend to flow through the internet, all the while learning about connections, URLs, IP Addresses, and the DNS.

Purpose

If you have been doing every lesson in this course, then each student in your classroom has used the internet...but how many know how it works? Learning more about the internet will help students develop a better understanding of its endless possibilities.

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Getting the Message**

Main Activity (20 min)

- **The Internet**

Wrap Up (15 min)

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

Assessment (5 min)

- **Internet - Assessment**

View on Code Studio

Objectives

Students will be able to:

- Learn about the complexity of sending messages over the internet.
- Translate URLs into IP Addresses.

Preparation

- □Watch the **Internet - Teacher Video**.
- □Print enough **IP Address Cards and Delivery Type Cards - Manipulatives** for each group.
- □Print one **Internet - Assessment** for each student.
- □Access to the internet (such as **get-site-ip.com**).
- □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

- **The Internet** - Unplugged Video (**download**)
- **IP Address Cards and Delivery Type Cards** - Manipulatives Make a Copy ▾
- **Internet** - Assessment Make a Copy ▾
- **Internet** - Assessment Answer Key Make a Copy ▾

Vocabulary

- **DNS** - short for Domain Name System, this system translates domain names (like example.com) to IP addresses (like 93.184.216.34)
- **DSL/Cable** - A method of sending information using telephone or television cables.
- **Fiber Optic Cable** - A connection that uses light to transmit information
- **Internet** - A group of computers and servers that are connected to each other.

- **IP Address** - A number assigned to any item that is connected to the Internet.
- **Packets** - Small chunks of information that have been carefully formed from larger chunks of information.
- **Servers** - Computers that exist only to provide things to others.
- **URL** - An easy-to-remember address for calling a web page (like www.code.org).
- **Wi-Fi** - A wireless method of sending information using radio waves.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has several new and important words:

- **IP Address** - Say it with me: I-P Add-ress

A number assigned to any item that is connected to the Internet

- **DNS (Domain Name Service)** - Say it with me: D-N-S

The service that translates URLs to IP addresses

- **URL (Universal Resource Locator)** - Say it with me: U-R-L

An easy-to-remember address for calling a web page (like www.code.org)

- **Internet** - Say it with me: In-ter-net

A group of computers and servers that are connected to each other

- **Servers** - Say it with me: Ser-vers

Computers that exist only to provide things to others

- **Fiber Optic Cable** - Say it with me: Fye-ber Op-tic Cay-bl

A connection that uses light to transmit information

- **Wi-Fi** - Say it with me: Wye-Fye

A wireless method of sending information using radio waves

- **DSL/Cable** - Say it with me: D-S-L / Cay-bl

A method of sending information using telephone or television cables

- **Packets** - Say it with me: Pack-ets

Small chunks of information that have been carefully formed from larger chunks of information

Lesson Tip

A quick preview is all you need here. These words will all be explained as part of the lesson, so it would be far less confusing to do a brief intro to the words as a "see if you can spot these during the day" type of heads-up.

Getting the Message

- It's quite likely that your students are aware of what the internet is, but they may not really understand what the internet does.
  - Ask "What is the internet?"
  - Is the internet a public place or a private place?
  - (Truthfully, many people think it can be both, but it should be viewed as a public space no matter what settings you think you've mastered.)
  - How does information get from place to place?
- Let's say that I want to look at the webpage for Code.org. What do you suppose the process would be like for me to send a message to request that page?
  - What do I do as a user?
  - What do you think happens inside the internet?

Sending a message over the internet is a lot like sending a message through the mail...if every letter we sent required thousands of envelopes!

Lesson Tip

There are some great YouTube videos on this subject that can make this lesson a little easier to understand. You can show them to the class in advance, or just watch them yourself. **Here is one of the videos in the Code.org video series on "How the Internet Works"**. (We recommend watching from 1:44 - 5:13, if possible.) The rest of the playlist is available **here**.

Every message we send through the internet gets chopped up and each piece is wrapped in its own version of an envelope. We call those "packets." Packets are specially formed chunks of information that are able to easily flow through any of the internet's channels.

Sometimes, a few of those packets will get lost, because the internet is a crazy place. In that case, the packets need to be resent, and the whole message has to get put on hold until they arrive.

Where do you think those packets are headed?

- Even if you're sending messages to another person, they first have to go to at least one "server."
    - A server is a special computer that is supposed to be always on and ready to send and receive information.
    - Every website has a server.
    - Even email goes through servers.

Servers don't have names like you and I do. They're actually addressed using numbers. These numbers are called IP addresses, and they look a little strange.

- For example: One of Code.org's IP addresses used to be 54.243.71.82
    - (Please be sure to check this out in advance. Most IP addresses change from time to time and they are then reused for other sites.)

There are many ways to reach the internet from your house, school, or place of business.

- You can connect directly using a cable (that might be DSL, Cable, or Fiber Optic)
- Or you can connect using radio waves over the air through Wi-Fi

Direct connections are most reliable, but they can be inconvenient.

- Can you figure out why?
    - (You have to be attached to a cable!)

Wi-Fi connections are super convenient, but the aren't always reliable.

- Can you figure out why not?
    - (Radio waves bounce all over the place and can get lost.)

So, if you're used to sending information to URLs (like **www.code.org**) and the servers actually have IP addresses for names (like 54.243.71.82) how does the Internet change from one to the other? That's what the DNS is for. The DNS (Domain Name Server) has tables that allow the system to go back and forth between URLs and IP addresses. If the Domain Name Servers ever stopped working, it would shut down the internet as we know it!

Lesson Tip

If you're thinking that this is a lot of text and it would be extremely boring to try to lecture this to a class full of elementary school kids, you're absolutely right! If you're unable to show a YouTube video in class to help explain it all, I highly recommend drawing pictures to explain each idea above, or choosing students as volunteers to act out what you describe while you're explaining. They're not expected to get every detail and definition at this point, only to gain exposure.

With that said, let's try to understand what the DNS does by making a little DNS table ourselves.

Pull out a piece of paper and draw a grid similar to that in the internet activity:

Sample of DNS Table:

#	URL	IP Address
1	code.org	54.243.71.82
2		
3		
4		

#	URL	IP Address
5		

First, we need to fill in this table.

- Survey the class for their favorite websites and write the URLs in the left column
- Use a site like **get-site-ip.com** to find the IP addresses for those sites and write them in the corresponding rows of the right column.

Now let's take this DNS Table and pretend to send messages through the internet!

Main Activity (20 min)

The Internet

Directions:

- Create your own DNS table, similar to what is shown above.
- Have the class help you fill in the blank spots in the table. Pick your favorite URLs and find their IP addresses using a site like **www.get-site-ip.com**.
- Divide into groups of 3 to 5.
- Assign each group an IP address from the newly created table, and assign each person in the group a position:
    - The Message Writer
    - The Internet
    - The Server (carries the IP address)
    - The Return Internet (optional)
    - The Message Receiver (optional)
- Each group will draw an **IP Address Cards and Delivery Type Cards - Manipulatives** to find out where their message is going and what their method of message delivery (Wi-Fi, Cable/DSL, or Fiber Optic Cable) will be.
- The Message Writer will craft a note to send to the server.
- The Internet will rip the message up into 4 small pieces called packets, then deliver each packet one at a time to the Server with the IP address that was drawn from the IP Address Card stack.
- The Server will make sure that the message arrives in order, then will send each packet off one at a time with the Return Internet (can be the same person or different person than the original Internet).
- The Return Internet will deliver each piece back to the Message Receiver (can be the same person or different person than the Message Writer) and put it back together.
- The Message Receiver will wait for all of the pieces to arrive, then read the message to be sure it arrived correctly!

Rules:

- The Internet must rip the message into exactly four packets.
- If the Internet drops a packet, they have to pick it up and go back to the start to deliver it again.
- The server has to wait for all of the message pieces to arrive before it can begin to send the message along.

Info:

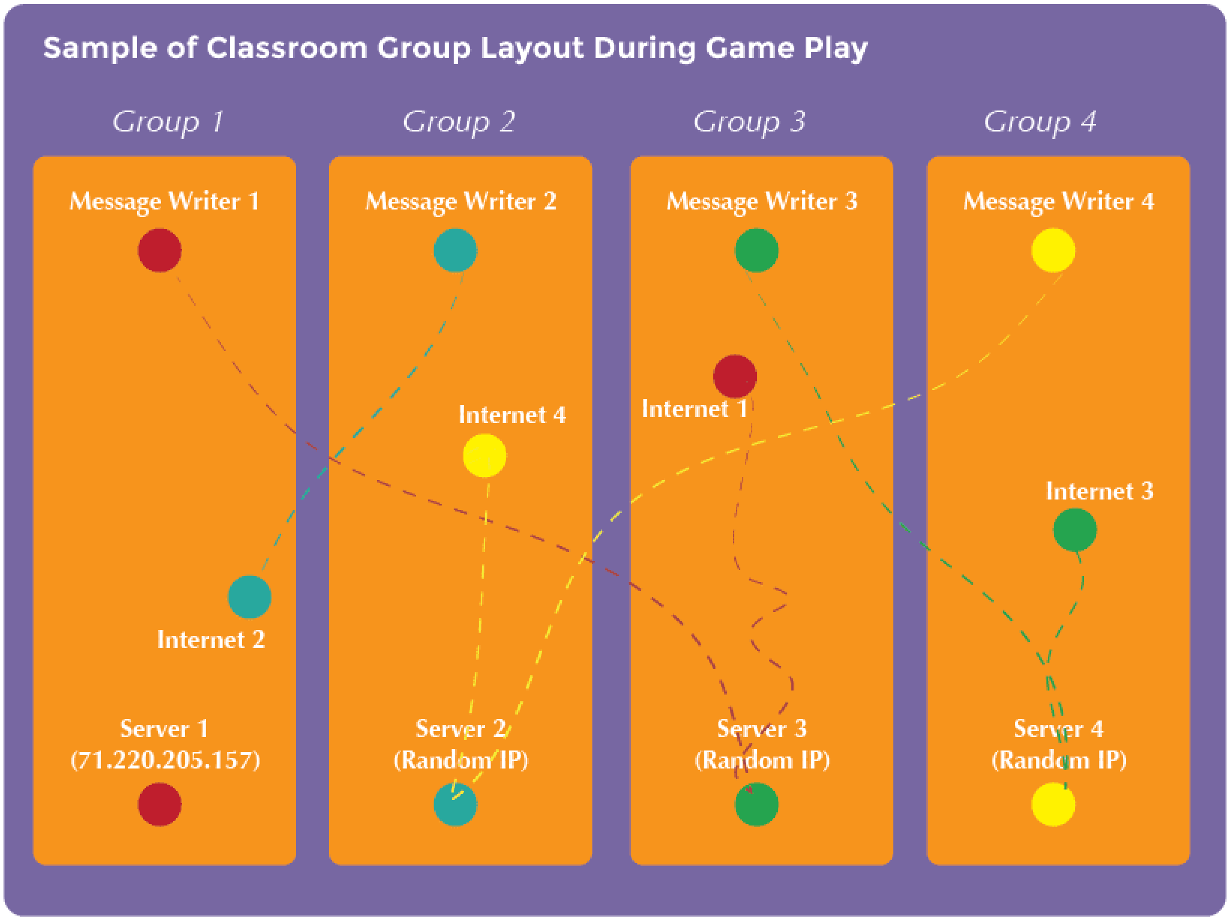
- Wi-Fi: Convenient, but spotty. Wi-Fi doesn’t require cables, but since the signal bounces all over the place, packets can get lost pretty easily.
    - Simulation: Internet must carry each packet on their shoulder (no hands).
- Cable/DSL: Fairly good at delivering messages, but you must be connected to a wire.
    - Simulation: Internet must carry each packet on the back of one hand and must keep the other hand touching a wall, desk, chair or the floor at all times.
- Fiber Optic Cable: The best at delivering messages, but you must be connected to a wire.
    - Simulation: Internet can carry packets in hand, but must keep the other hand touching a wall, desk, chair or the floor at all times.

To play this game, you can have your groups cluster anywhere, but for the first time it can be less confusing to have groups play in a line.

- Line up the "Servers" on one end of the room (holding their IP addresses). The Return Internet players can be over there as well (if you have that many people in each group).
- Have the everyone else line up across from their server at the other side of the room.
- The Message Senders will likely be sending their messages to a server other than their own, so the Internet players will likely cross over from group to group. It may look something like the diagram below (in English):

💡 Lesson Tip

If it feels like there are too many rules to explain outright, feel free to post them on the board and just explain the game as you go. You can play multiple rounds until the class really understands.



Wrap Up (15 min)

Flash Chat: What did we learn?

- What kind of connection would you rather have (Wi-Fi, DSL/Cable, or Fiber Optic)? Why?
- Why might it take your message a long time to get somewhere?

💡 Lesson Tip

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

Journaling

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 do you feel about today's lesson?

- What's something you learned about the internet today?
- Why is learning about the internet important?

Assessment (5 min)

Internet - 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.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **NI** - Networks & the Internet

Lesson 14: For Loop Fun

Overview

We know that loops allow us to do things over and over again, but now we're going to learn how to use loops that have extra structures built right in. These new structures will allow students to create code that is more powerful and dynamic.

Purpose

At this point, students have become masters of loops. Today, they will learn about another loop commonly used in programming. The for loop repeats commands a certain number of times, but also keeps track of the values it is iterating over. For example, a for loop that begins at 4, ends with 8, and has a step value of 1 will repeat 4 times, but the values 4, 5, 6, and 7 will also be captured for use elsewhere. Using this structure with variables can create some pretty fantastic programs. Today, students will simply be learning the basics of a for loop before diving into programming with them next time!

Agenda

Warm Up (20 min)

- **Vocabulary**
- **For One and All**

Main Activity (20 min)

- **For Loop Fun - Worksheet**

Wrap Up (15 min)

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

Assessment (5 min)

- **For Loop Fun - Assessment**

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Determine starting value, stopping value, and stepping value for a `for` loop.
- Illustrate the counter values hit each time through a for loop during runtime.

Preparation

☐Watch the **For Loop Fun - Lesson in Action Video**.

☐Print one **For Loop Fun - Worksheet** per group.

☐Print one **For Loop Fun - Assessment** for each student.

☐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

- **For Loop Fun** - Unplugged Video (**download**)
- **For Loop Fun - Assessment Key** - Teacher Key Make a Copy ▾
- **For Loop Fun** - Lesson in Action Video
- **For Loop Fun** - Worksheet Answer Key Make a Copy ▾

For the Students

- **For Loop Fun** - Worksheet Make a Copy ▾
- **For Loop Fun** - Assessment Make a Copy ▾

Vocabulary

- **For Loop** - Loops that have a predetermined beginning, end, and increment (step interval).

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

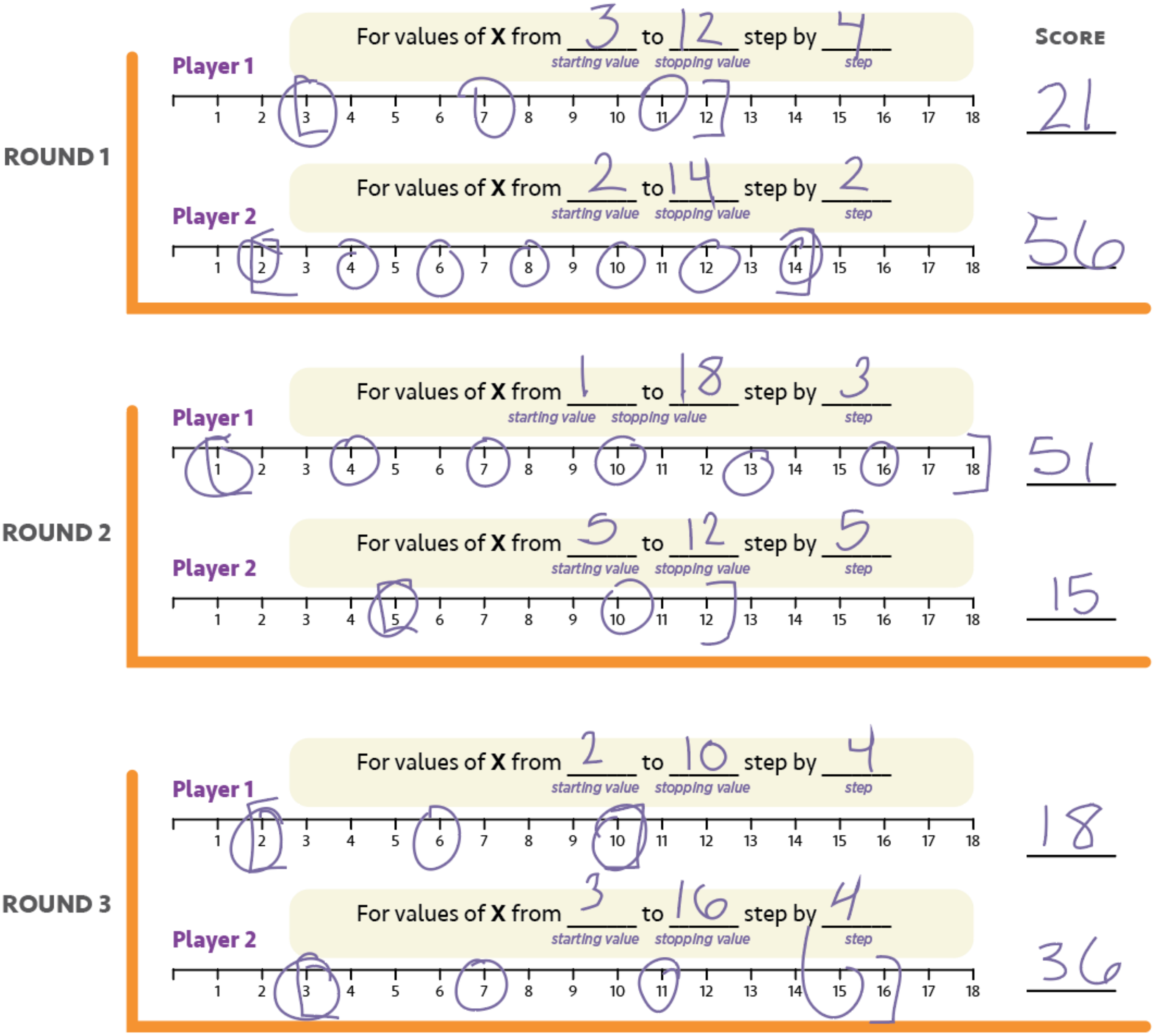
- **For Loop** - Say it with me: For-Loop

Loops that have a predetermined start, stop, and step value.

For One and All

- Point out that there are certain loops that happen very frequently, for example, loops where you need to keep track of how many times you have been through
    - Sometimes, you don't want to start with one
    - Sometimes, you don't want to count by ones
    - "for" loops give you a powerful way to keep a counter that starts when you want, ends when you want, and increases by whatever size step that you want

Here, you can jump right into a sample of the game (example in English)



SCORE

ROUND 1

Player 1

For values of **X** from 3 to 12 step by 4

starting value stopping value step

21

Player 2

For values of **X** from 2 to 14 step by 2

starting value stopping value step

56

ROUND 2

Player 1

For values of **X** from 1 to 18 step by 3

starting value stopping value step

51

Player 2

For values of **X** from 5 to 12 step by 5

starting value stopping value step

15

ROUND 3

Player 1

For values of **X** from 2 to 10 step by 4

starting value stopping value step

18

Player 2

For values of **X** from 3 to 16 step by 4

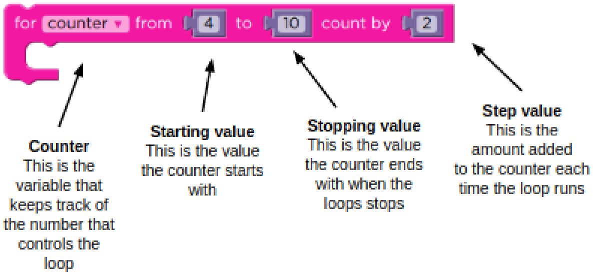
starting value stopping value step

36

Main Activity (20 min)

For Loop Fun - Worksheet

Sometimes we want to repeat things a certain number of times, but we want to keep track of values as we do. This is where a "for" loop comes in handy. When you use a "for" loop, you know right from the start what your beginning value is, what your ending value is, and how much the value changes each time through the loop.



for counter from 4 to 10 count by 2

Counter
This is the variable that keeps track of the number that controls the loop

Starting value
This is the value the counter starts with

Stopping value
This is the value the counter ends with when the loops stops

Step value
This is the amount added to the counter each time the loop runs

"for" Loop block (in English)

Directions:

It may be difficult for young students to understand this written in pseudocode, but it may be helpful to have you explain out loud (and perhaps with a diagram) what they will be using as the content of a "for" loop.

- Divide students into pairs
- To start the round, each student rolls three times:
  - One die to determine the starting value of X
  - Three dice to determine the stopping value for X
  - One die to determine the stepping value of X each time through
- Use one of the provided number lines to trace the "for" loop that they’ve made
  - Start at the starting value of X
  - Count down the number line, circling the numbers at the rolled interval
  - Stop when you get to the predetermined stopping value
- Add all of the circled values to your score, then let the other player take a turn
- Best 2 out of 3 wins

Lesson Tip

When you play this game, it's as if you're running through a loop like this

```
for (x=startValue; x <= stopValue; x = x + step){
circle currentXvalue;
add currentXvalue to roundScore;
}
```

Lesson Tip

If any of the values rolled are outside the bounds of the game (like rolling a start value of 6 but then rolling 2, 1, 2 for the end value), have the student re-roll everything.

Wrap Up (15 min)

Flash Chat: What did we learn?

- What would your interval need to be if you wanted to count from 4 to 13 by threes?
- What kinds of things do you think you could do with a for loop?
- Can you reproduce a normal loop using a for loop?
- What would you need to do?

Lesson Tip

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

Journaling

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 do you feel about today's lesson?
- What is a "for" loop?
- Why would you use a "for" loop instead of a "repeat" loop or a "while" loop?

Assessment (5 min)

For Loop Fun - 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.

Run it Backward

- Try this activity again, but this time have the start number be selected using three dice, and the stop number with only one. Make sure to have a negative increment!

Hop Scotch

- Using chalk, draw a hop scotch diagram outside on the blacktop
  - Number the squares from bottom to top
  - Have students give each other a start square, stop square, and how many at a time they need to jump
  - When the jumper is done, have them write down the loop they just performed
  - Start adding additional activities to be done at each square, this will add complexity to the written portion, as well

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 15: For Loops with Bee

Overview

Featuring Bee, this lesson focuses on `for` loops and using an incrementing variable to solve more complicated puzzles. Students will begin by reviewing loops from previous lessons, then they'll walk through an introduction to `for` loops so they can more effectively solve complicated problems.

Purpose

Today's concept, `for` loops, are a very important topic in computer science. Not only are they widely used, the process of learning `for` loops enhances the learning of other important concepts (such as variables and parameters.) Students will have plenty of practice critically thinking through problems by determining the starting, ending, and stepping values for each `for` loop. This concept uses plenty of math as well, so feel free to pair it with a math lesson for an even deeper learning experience.

Agenda

Warm Up (15 min)

- **Introduction**

Bridging Activity - For Loops (15 min)

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

Main Activity (30 min)

- **Online Puzzles**

Wrap Up (15 min)

- **Journaling**

View on Code Studio

Objectives

Students will be able to:

- Determine starting value, stopping value, and stepping value for a \`for\` loop.
- Recognize when to use a \`for\` loop and when to use other loops such as \`repeat\` and \`while\` loops.

Preparation

□Play through the puzzles to find any potential problem areas for your class.
□Make sure every student has a journal.

Links

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

For the Students

- **Unplugged Blocks (Courses C-F)** - Manipulatives

Vocabulary

- **For Loop** - Loops that have a predetermined beginning, end, and increment (step interval).

Teaching Guide

Warm Up (15 min)

Introduction

Remind students of the work they did in "For Loop Fun". Open a discussion about what they learned, why they think it might be useful, and if they had any fun. Here are some discussion starters.

- What did you learn in "For Loop Fun"?
- What are the three main components of a `for` loop?
  - starting value, step interval, ending value
- Why do you think a `for` loop might be helpful in programming?
  - Many students might not know an answer to this. Let them hypothesize, but don't dwell on this question for too long.
- Did you have fun learning about `for` loops? Why or why not?
- Are you excited to use `for` loops in online puzzles?

Bridging Activity - For Loops (15 min)

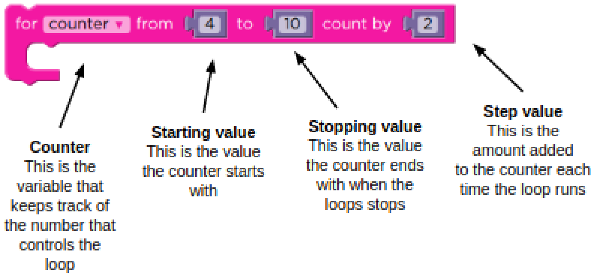
This activity will help bring the unplugged concepts from "For Loop Fun" into the online world that the students are moving into. Choose *one* of the following to do with your class:

Unplugged Activity Using Paper Blocks

Split up the students of your class into pairs. Ideally have the pairs be the same from when your class did "For Loop Fun". Print out a `for` loop from **Unplugged Blocks (Courses C-F) - Manipulatives** for each pair of students. Pass out one die to each pair. Have the partners take turns rolling the die to obtain the following values:

- One roll for the starting value
- Four rolls for the ending value
- One roll for the step value

Have each pair fill in the `for` loop with the appropriate values in the correct spot.


for counter from 4 to 10 count by 2
Counter
This is the variable that keeps track of the number that controls the loop
Starting value
This is the value the counter starts with
Stopping value
This is the value the counter ends with when the loops stops
Step value
This is the amount added to the counter each time the loop runs

Using a basic number line, like the one used in "For Loop Fun", have the students mark the beginning, ending, and middle values that this `for` loop will touch. When everyone is done, see who got the most points by totaling the starting, middle, and ending numbers of each pair.

Previewing Online Puzzles as a Class

Display a puzzle from the lesson. We recommend puzzle #4 because it displays a potential solution and asks the user to evaluate it.

Using a number line similar to the ones used in "For Loop Fun", mark the start and ending values of the given `for` loop (if you aren't using puzzle #4, you will need to come up with a potential solution first). With the class's help, circle the values between the start and end that the `for` loop will touch. If you are working on puzzle #4, ask the class what they think the answer is to the question, given what they found with the number line.

Main Activity (30 min)

Online Puzzles

Some students may have a hard time differentiating between `repeat` loops and `for` loops. We recommend having scratch paper out for students to make guesses on values like the start, stop, and step. Implementing pair programming amongst the class might also be helpful for your students.

🖥 Code Studio levels

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

For Loops and Repeat Blocks 📹 3 ***(click tabs to see student view)***

Prediction 🖥 4 ***(click tabs to see student view)***

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

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

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

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

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?
- How is a `for` loop different from a `repeat` loop?
- Why do you think `for` loops could be useful?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 16: For Loops with Artist

Overview

In this lesson, students continue to practice `for` loops, but this time with Artist. Students will complete puzzles combining the ideas of variables, loops, and `for` loops to create complex designs. At the end, they will have a chance to create their own art in a freeplay level.

Purpose

Creativity and critical thinking come together beautifully in this lesson. Students will continue their practice with `for` loops and variables while they create jaw-dropping images. This lesson inspires a creative mind while teaching core concepts to computer science.

Agenda

Warm Up (15 min)

  Introduction

Main Activity (30 min)

  Online Puzzles

Wrap Up (15 min)

  Journaling

View on Code Studio

Objectives

Students will be able to:

- Use `for` loops to change loop several times with different values.
- Recognize when to use a `for` loop and when to use other loops such as `repeat` and `while` loops.

Preparation

□Play through the puzzles to find any potential problem areas for your class.
□Make sure every student has a journal.

Vocabulary

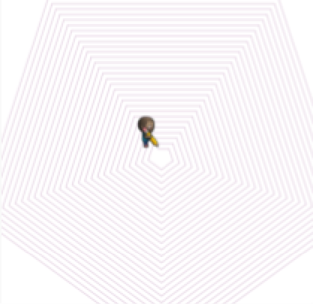
- **For Loop** - Loops that have a predetermined beginning, end, and increment (step interval).

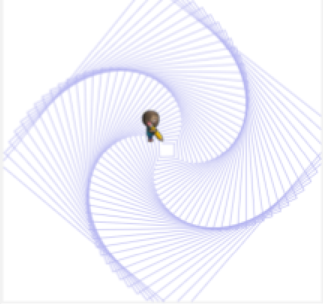
Teaching Guide

Warm Up (15 min)

Introduction

On a board displayed to the entire class, draw (or display via projector) one of the final projects from the **Course F Online Puzzles - 2018 - Website** associated with this lesson. We recommend one of the following:





Ask the class how a computer might draw the drawing you displayed.

After a few predictions have been said, reply with `for` loops of course!

Tell the students they will soon be learning how to create these fine drawings using `for` loops and variables.

Main Activity (30 min)

Online Puzzles

🖥 Code Studio levels

Intro to For Loops 🎥 1 ***(click tabs to see student view)***

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

Practice 🖥 3 🖥 4 🖥 5 🖥 6 🖥 7 🖥 8 🖥 9 ***(click tabs to see student view)***

Free Play 🖥 10 ***(click tabs to see student view)***

Prediction 🖥 11 ***(click tabs to see student view)***

Free Play 🖥 12 ***(click tabs to see student view)***

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

These puzzles are super fun, but it may be helpful for students to have protractors and scratch paper to see these designs made in the physical form. If that isn't an option in your class, try to get the students to trace on the computer screen with their fingers.

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 one of the designs you made today. What was the code needed to create it?
- What are some designs you would like to create? How do you think `for` loops or variables could help create those?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 17: Behaviors in Sprite Lab

Overview

Here, students will use Sprite Lab to create their own customized behaviors.

Purpose

Students will use events to make characters move around the screen, change size, and change colors based on user input. This lesson offers a great introduction to events in programming and even gives a chance to show creativity!

Agenda

Warm Up (15 min)

Introduction

Main Activity (30 min)

Online Puzzles

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Identify actions that correlate to input events.
- Create an animated, interactive game using sequence and events.

Preparation

□Play through the puzzles to find any potential problem areas for your class.
□Make sure every student has a journal.

Introduced Code

- `move in direction`
- `turn`
- `change sprite property`

Teaching Guide

Warm Up (15 min)

Introduction

Today students will revisit *sprite behaviors*, this time learning how to edit behaviors directly and even make new ones.

Review: Ask students questions about the Swimming Fish and Alien Dance Party lessons.

- What are some of the behaviors we can assign to our sprites?
- What do you imagine the code *inside* a behavior might look like?

Display: Begin by showing Puzzle 1 to your students.

Think/Pair: Ask them to predict what will happen when the code is run, and to discuss with their neighbors. Be sure to open the behavior editor by clicking "edit" on the `mystery behavior` block. Run the code, and discuss the outcome.

Discuss Ask students questions about how they might change this behavior's code to create a different effect.

- What would happen if you change the -1 to another number? -5? Positive 1? 0?
- What other properties might we be able to change about a sprite besides its size?

Main Activity (30 min)

Goal: Today, students will be editing and creating their own behaviors! They'll begin by making small changes to some familiar but new behaviors and gradually move towards writing their own behaviors from scratch.

Online Puzzles

Transition: Move students to their machines. Encourage students to follow the instructions for each puzzle. Help them realize that this is a creative activity, intended to help them learn Sprite Lab. It is not an assessment activity of any sort.

Reminder: If puzzles are sharable, 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**.

Teaching Tip

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. Have students describe the problem that they're seeing:

- What is it supposed to do?
- What does it do?
- What does that tell you?

Code Studio levels

Prediction 1 ***(click tabs to see student view)***

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

Free Play 7 ***(click tabs to see student view)***

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

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 do you feel about today's lesson?
- What other options would you like to be able to have your pet do?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 18: Virtual Pet with Sprite Lab

Overview

In this lesson, students will create an interactive Virtual Pet that looks and behaves how they wish. Students will use Sprite Lab's "Costumes" tool to customize their pet's appearance. They will then use events, behaviors, and other concepts they have learned to give their pet a life of its own!

Purpose

This lesson allows students to apply programming concepts from prior lessons in a more creative context. For example, students will use a variable to store their pet's "happiness" as an integer, which should help them understand how variables can be used in other applications. Last, completing this lesson should prepare students to engage with the open-ended final project.

Agenda

Warm Up (15 min)

Introduction

Main Activity (30 min)

Online Puzzles

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Create an interactive virtual pet using events, behaviors, variables, and custom art.
- Program solutions to problems that arise when designing a virtual pet, like feeding it or monitoring its energy.

Preparation

□ Play through the puzzles to find any potential problem areas for your class.
□ Make sure every student has a journal.

Introduced Code

- `print`
- `jump to`
- `key pressed`

Teaching Guide

Warm Up (15 min)

Introduction

Revisit *events* and *behaviors* in programming. Additionally, introduce the Sprite Lab "Costumes" tool that allows students to draw their own costumes.

Review: Ask students questions about events and behaviors. Show them the Alien Dance Party mini-project (from the *Events with Sprite Lab* lesson) as an example.

- Do you remember what an event is?
- Can you name any of the events that you used to make the aliens dance? What do they do?
  - `sprite clicked` event
  - `sprite touches sprite` event
  - `key pressed` event
- Do you remember what a behavior is?
- Can you remember some of the behaviors you used to make the aliens dance? What do they do?
  - `patrolling`
  - `jittering`
  - `spinning right/left`

Display: Begin by showing Puzzle 1 of today's lesson to your students.

Think/Pair: Ask students to predict what will happen when the code is run, and to discuss with their neighbors. Run the code, and discuss the outcome.

Display: Show Puzzle 2. Briefly demonstrate how to do the following:

- Navigate between the *Code* and *Costumes* tabs.
- Draw a costume.
- Choose a costume from the costume library.
- Change the virtual pet's sprite's costume to a custom one.

Main Activity (30 min)

Goal: Today, students will be creating their own virtual pet! They will begin by drawing or selecting a new costume for a sprite. Then they will create events that cause actions and behaviors upon interaction.

Online Puzzles

Transition: Move students to their machines. Encourage students to follow the instructions for each puzzle. Help them realize that this is a creative activity, intended to help them learn Sprite Lab. It is not an assessment activity of any sort.

Reminder: If puzzles are sharable, 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**.

Teaching Tip

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. Have students describe the problem that they’re seeing:

- What is it supposed to do?
- What does it do?
- What does that tell you?

Code Studio levels

Free Play 1 ***(click tabs to see student view)***

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 do you feel about today's lesson?
- What other options would you like to be able to have your pet do?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Lesson 19: End of Course Project

Overview

The next five lessons provide an opportunity for students to put their coding skills to use in a capstone project. This project will help individuals gain experience with coding and produce an exemplar to share with peers and loved ones. This is intended to be a multi-lesson or multi-week project where students spend time brainstorming, learning about the design process, building, and then presenting their final work.

In the "Explore" stage, students will play around with pre-built Artist and Sprite Lab programs for inspiration. Next, students will learn about the design process and how to implement it in their own projects. They will then be given the space to create their own project in Artist, Sprite Lab, or any other interface that you are comfortable providing. (This is likely the longest stage of the project.) Students will then revise their code after testing and peer review. Finally, students will be able to present their finished work to their classmates.

Purpose

Students may be ready to jump straight into building their projects, but this lesson will help shape their ideas into plans. This structure will keep the dreamers grounded and illuminate a path for those feeling left in the dark. Provide students with ample time to build and revise their projects. The trial and error inevitably involved in this lesson will teach problem solving and persistence.

Agenda

Day 1 - Explore Project Ideas (45 min)

Example Projects

Day 2 - The Design Process (45 min)

Define and Prepare

Day 3 - Build Your Project (45 min)

Try

Day 4 (Recommended for 5th Grade) - Revise Your Project (45 min)

Reflect and Try Again

Day 5 & 6 - Present Your Project (45 min each)

Presentations

Extension Activity

View on Code Studio

Objectives

Students will be able to:

- Learn to plan in advance for an ongoing assignment.
- Be able to explain how system limitations can affect project design.
- Describe how compromise can help keep a project on track and inspire creativity.
- Draft and implement plans to resolve any issues in their code.
- Articulate the design process and how it helped shape the finished culminating project.

Preparation

□Spend time making your own project with both the Artist and Sprite Lab. Familiarize yourself with the capabilities and limitations of each tool.
□Modify the rubric to fit your class goals and print out a copy for each student.
□Modify the project design worksheet to fit your class and print one packet for each student.

Links

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

For the Teachers

- **Design Process** - Teacher Prep Guide
  Make a Copy ▾
- **Final Project Design** - Worksheet
  Make a Copy ▾
- **CS Fundamentals Final Project** - Rubric

Vocabulary

- **Define** - Figure out the details of the problems that you are trying to solve
- **Prepare** - Research, plan, and acquire materials for the activity you are about to do

- **Reflect** - Carefully think back on something with the intention of improving the outcome in the future
- **Try** - Attempt to do something

Teaching Guide

Day 1 - Explore Project Ideas (45 min)

Example Projects

Goal: This part of the process is an exploration. Students will sit down with a stage full of example projects to remix and learn. Not only will this give students an idea of what is possible, it will also help them see the limitations of the tool.

Give students a day to play with and remix the projects found on Code Studio. Have them use their journals (or notebook paper) to keep track of thoughts and ideas as they go.

This activity should be done in the same pairs/groups that will be working on projects together over the next several lessons.

Make sure your class understands that they will be spending the next several weeks working with projects of their own, so they should pay close attention to how these programs were written, as well as the concepts that they use.

Code Studio levels

Example projects 1 ***(click tabs to see student view)***

End of Course Project 2 | 2a | 2b

Day 2 - The Design Process (45 min)

Define and Prepare

Goal: Students will come up with a project and plan their strategy for programming that project in a single day. Students should have a project sketch and a description by the time the day is done.

Preparing Students for the Process:

The most important responsibility you have in kicking off this segment is to help your class understand the scope of this project. Students should be clear about the various expectations over the coming weeks so that they can prepare for their review and presentations appropriately.

To help your class manage this multi-stage undertaking, they should be given both the **Final Project Design - Worksheet** and the **CS Fundamentals Final Project - Rubric** on the first day of planning. Students will then be able to follow the rubric each step of the way to predict what their project grade will be in the end.

The Final Project Design Worksheet will provide a place for students to capture relevant thoughts and processes as they go, so they are more prepared for their reviews and presentations in the end.

As the teacher, you should download a copy of the documents and decide which elements are important to you. Be sure to edit or remove anything that you do not intend to draw student focus.

Define and Prepare:

Now that the class has their Final Project Design Worksheet in hand, they should start filling out the questions under **Day 1**.

Lesson Tip:

Save 5 minutes or so at the end of the day to have students trade Final Project Design Worksheets to look at each other's work. This will help make sure that nothing is omitted or overlooked.

Students will likely need to refer back to their notes from playing with the example projects, especially if they don't have access to online Artist or Play Lab project levels while they plan.

Students should focus on defining and planning their project during Day 1, and not cross over into building until their ideas have been written up and/or drawn out.

If students get stuck, help them work through ideas by asking questions and recalling examples, rather than offering solutions.

Day 3 - Build Your Project (45 min)

Try

Goal: Students will use this day to build an initial version of their project.

Equipped with their Final Project Design Worksheet, students should head to the computers to start bringing their projects to life.

This process will come complete with plenty of trial and error. Projects are likely to become truncated versions of the original scope (if not morphed altogether). Remind students that this kind of compromise is common in software design, but they need to be sure to document the reasons for the changes in their product.

Don't let the class forget to fill out their Final Project Design Worksheets as they go. It might be helpful to suggest that pairs/groups take a worksheet break to begin discussing these questions about halfway through their lab time. Alternatively, the navigator can keep their eyes open for pertinent answers while the driver codes.

Be sure that each team member has their own Final Project Design Worksheet, as there are questions about each student's own individual thoughts and behaviors that need to get captured along the way.

Day 4 (Recommended for 5th Grade) - Revise Your Project (45 min)

Reflect and Try Again

Goal: Students will work with another group to give and receive feedback in an effort to make each other's projects stronger.

Reflect:

For reflections, have each group pair up with another group to try each other's projects. After about 10 minutes, have the groups discuss the questions in the Final Project Design Worksheet.

Encourage students to ask the questions on the FPDW and write down feedback provided by their reviewing teams so that they can refer back to it later. This portion should take approximately 15 more minutes.

Try Again:

With their new reflections in hand, students can head back to their machines to make a handful of edits. With just 10 minutes left, they will likely have to select only the most important feedback to incorporate.

Lesson Tip:

Teachers should avoid assigning the final bit of project work as homework unless they are certain that students both live within a close proximity to one another **and** have internet access at home.

Day 5 & 6 - Present Your Project (45 min each)

Presentations

Goal: Students will create and present their projects in an approved manner (written, oral, or using multimedia).

Create:

Ideally, you will have class time available to give students to work on their presentations. This will allow them to incorporate rich multimedia components, like **Google Slides**. For other presentation ideas, visit **72 Creative Ways for Your Students to Show What They Know - Website**.

Encourage students to include all of the information from Section J of the Final Project Design Worksheet into their presentation, as well as two or more questions from Section K.

Present

Students should showcase their apps first, then they can discuss the questions that they covered in their presentations.

Lesson Tip:

If you are looking for a section of this series to assign as homework, this is it! Projects do not have to be presented in electronic form, so this is a great offline option.

It can be very helpful to have students sign up for a specific order in which to give their presentations, so that they are able to enjoy the demonstrations of their classmates without worrying about whether they will be called on next.

Extension Activity

If your students are already comfortable with coding concepts, try having them create their projects in another platform, like **Scratch** or **Alice**.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming