

Course D

Course D was created for students who read at roughly a third grade level. Angles and mathematical concepts are introduced with helpful videos and hints.

The course begins with a review of the concepts found in Courses A, B, and C. This review helps introduce or refresh basic ideas such as repeat loops and events. Students will develop their understanding of algorithms, nested loops, while loops, conditionals, and events. Lessons on digital citizenship are also included. This course is crafted to build a strong foundation of basic concepts before opening up to a wide range of new and exciting topics.

Teacher Links: [Teacher Videos Playlist](#)

Lesson 1: Graph Paper Programming

In this lesson, you will program your friend to draw pictures!

Lesson 2: Introduction to Online Puzzles

Sequencing | Debugging | Loops | Angry Bird | Collector | Artist | Harvester

This lesson will give you practice in the skills you will need for this course.

Lesson 3: Relay Programming

Unplugged | Relay Programming | Algorithms

Remember at the beginning of the course when you made drawings with code? In this lesson, you will be working with a team to do something very similar!

Lesson 4: Debugging with Laurel

Debugging | Bug | Collector | Laurel

Have you ever run into problems while coding? In this lesson, you will learn about the secrets of debugging. Debugging is the process of finding and fixing problems in your code.

Lesson 5: Events in Bounce

Event | Bounce

Ever wish you could play video games in school? In this lesson, you will get to make your own!

Lesson 6: Loops in Ice Age

Loop | Ice Age | Scrat

Lesson 7: Drawing Shapes with Loops

Loop | Artist

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

Lesson 8: Nested Loops in Maze

Nested Loops | Loops | Bee | Maze

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

Lesson 9: Fancy Shapes using Nested Loops

Nested Loops | Loops | Artist

More nested loops! This time, you get to make some AMAZING drawing with nested loops.

Lesson 10: Snowflakes with Anna and Elsa

Loop | Nested Loop | Frozen

Anna and Elsa have excellent ice-skating skills, but need your help to create patterns in the ice. Use nested loops to create something super COOL.

Lesson 11: While Loops in Farmer

While Loops | Loops | Farmer

Loops are so useful in coding. This lesson will teach you about a new kind of loop: while loops!

Lesson 12: Until Loops in Maze

Conditional | Loop | Maze | Angry Bird | Zombie

You can do some amazing things when you use `until` loops!

Lesson 13: Conditionals with Cards

Conditionals | Unplugged

It's time to play a game where you earn points only under certain conditions!

Lesson 14: If/Else with Bee

Conditional | Bee | Maze

Now that you understand conditionals, it's time to program Bee to use them when collecting honey and nectar.

Lesson 15: Harvesting with Conditionals

Conditional | Loop | Harvester

It's not always clear when to use each conditional. This lesson will help you get practice deciding what to do.

Lesson 16: Digital Citizenship

Common Sense Education | Unplugged

Some information is not safe to share online. This lesson will help you learn the difference between safe and private information.

Lesson 17: Ninjas vs. Pirates Game

Play Lab | Event

This lesson will guide you through making your very own video game.

Lesson 18: Binary Images

Binary | Unplugged

Learn how computers store pictures using simple ideas like on and off.

Lesson 19: Binary Images with Artist

Binary | Artist

In this lesson, you will learn how to make images using on and off.



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

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

Lesson 1: Graph Paper Programming

Overview

By "programming" one another to draw pictures, students get an opportunity to experience some of the core concepts of programming in a fun and accessible way. The class will start by having students use symbols to instruct each other to color squares on graph paper in an effort to reproduce an existing picture. If there's time, the lesson can conclude with images that the students create themselves.

Purpose

The goal of this activity is to build critical thinking skills and excitement for the course, while introducing some of the fundamental programming concepts that will be used throughout the course. By introducing basic concepts like sequencing and algorithms to the class in an unplugged activity, students who are intimidated by computers can still build a foundation of understanding on these topics. In this lesson, students will learn how to develop an algorithm and encode it into a program.

Agenda

Warm Up (10 min)

Introduction to Graph Paper Programming

Main Activity (30 min)

Practice Together
The Students' Turn

Wrap Up (15 min)

Journaling / Flash Chat

Optional Assessment (10 min)

Graph Paper Programming - Assessment

Extended Learning

Objectives

Students will be able to:

- Reframe a sequence of steps as an encoded program
- Explain constraints of translating problems from human language to machine language

Preparation

Watch the **Graph Paper Programming - Teacher Video**.

Watch the **Graph Paper Programming - Lesson in Action Video**.

Print out one **Graph Paper Programming - Worksheet** for each group.

Print one **Graph Paper Programming - Assessment** for each student.

Supply each group with several drawing grids, paper, and pens/pencils.

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 Teacher

- **Graph Paper Programming** - Teacher Video
- **Graph Paper Programming** - Lesson in Action Video
- **Graph Paper Programming** - Worksheet Answer Key
- **Graph Paper Programming** - Assessment Answer Key

For the Students

- **Graph Paper Programming** - Unplugged Video (**download**)
- **Graph Paper Programming** - Worksheet
- **Graph Paper Programming** - Assessment
- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Teaching Guide

Warm Up (10 min)

Introduction to Graph Paper Programming

In this activity, students will encode instructions to guide each other toward making drawings without letting the rest of their group see the original image. This warm-up frames the activity for the class.

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

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

🗣️ **Discuss:** How do you suppose that robots know how to do the things that they do? Do they have brains that work the same way that ours do?

Work this into a discussion on how people have to program robots to do specific things, using specific commands.

Discussion

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

Main Activity (30 min)

Practice Together

In this activity, students will act as both programmers and robots, coloring in squares according to programs that they have written for one another.

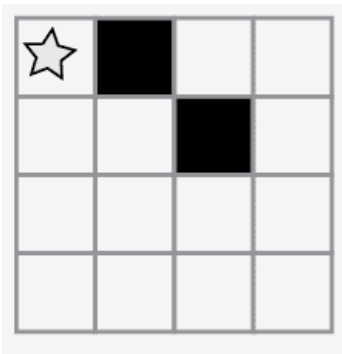
Distribute: Students will use 4x4 grids (or sheets of graph paper with 4x4 boxes sectioned off). They will also need the image worksheet.

Display: Project these commands, or write them on the board. They won't persist long, but they will help students make the transition from Algorithm to Program.

Move one square right
Move one square left
Move one square up
Move one square down
Fill in square with color

Say: Today, we all get to program robots...and they're already here in the room! It's you! We're going to write programs using symbols with special meanings to help each other recreate a picture. First, we'll practice together as if I am the robot and you are the programmers, then we can break up into groups so that everyone can get a turn.

Display: Display both the image that you are going to have the students walk you through, and a blank grid that you will fill-in with your ARM. Make sure that the instructions, grid, and image remain visible at the same time.



Remarks

Here is an image. Pretend that I am the robot with an Automatic Realization Machine (ARM). These are the only instructions that I understand.

Starting at the upper left-hand corner, guide my ARM out loud with your words.

Model: The class might give you instructions like these below. As you hear an instruction that you intend to follow, make sure to repeat it out loud so that the students can keep track of what you are doing.

Move One Square Right
 Fill In Square with Color
 Move One Square Right
 Move One Square Down
 Fill In Square with Color

Continue with the activity until you have completed your sample square.

Capture: Write each of the commands down so that students can see all of the steps that went into the one image.

Move One Square Right
 Fill In Square with Color
 Move One Square Right
 Move One Square Down
 Fill In Square with Color

Say: You just gave me a list of steps to finish a task. In programming, they call that an algorithm. Algorithms are great, because they are easy for you to understand as the programmer. BUT, what happens when we want to write down the algorithm for a drawing like this?

Display: Show the students a more complicated image, like the one below.



Next, begin writing down some of the instructions that it would take to replicate that image. Hopefully, students will see that writing everything out longhand would quickly become a bit of a nightmare.

Move One Square Right
Fill In Square with Color
Move One Square Right
Move One Square Right
Fill In Square with Color
Move One Square Down
Move One Square Left
Fill In Square with Color
Move One Square Left
Move One Square Left
Fill In Square with Color
PLUS 12 MORE INSTRUCTIONS!

Display: Show the students this list of symbols.



Discussion: How could we use these symbols to make our instructions easier?

Draw out ideas that relate to transitioning from the verbal instructions to the symbols. Once the students get to that place, point out that this text:

- “Move one square right, Move one square right, Fill-in square with color”

would now correspond to the program:



Model: Now, have the class help you draw the larger image using only symbols. Do not worry about unnecessary steps for now. If their final program works to create the image, consider it a win.

The classroom may be buzzing with suggestions by this point. If the class gets the gist of the exercise, this is a good place to discuss alternate ways of filling out the same grid. If there is still confusion, save that piece for another day and work with another example.

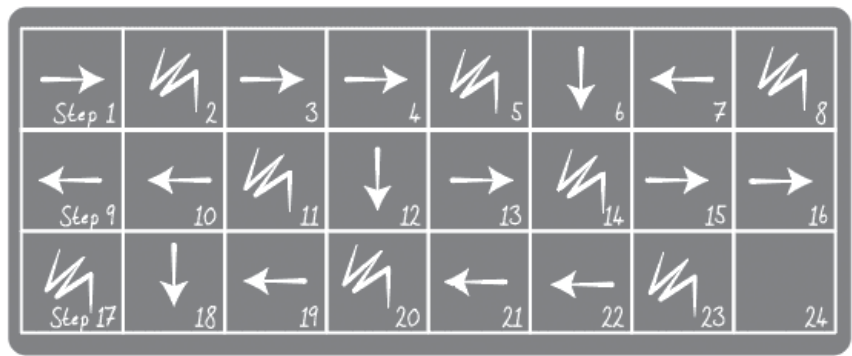
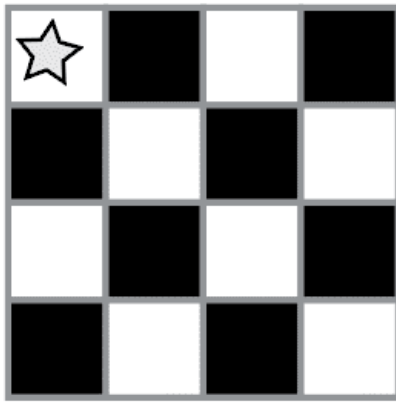
See a sample solution below:

Discussion

The goal for this discussion is to get at the idea that students can use symbols to stand for entire phrases. Once they understand that, share with them that making the switch from listing steps in detail to encoding them is called “programming.”

Teaching Tip

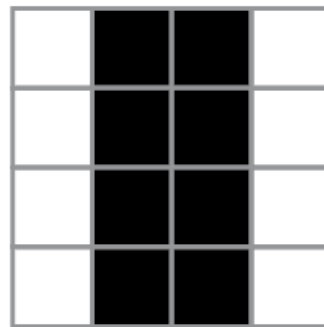
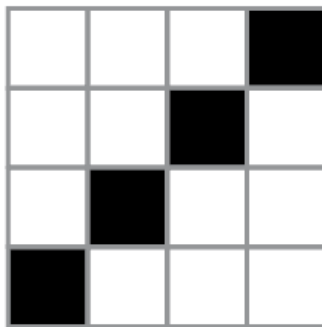
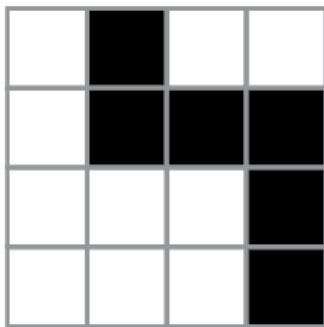
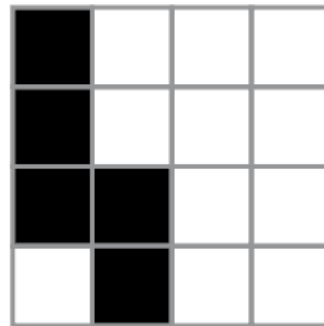
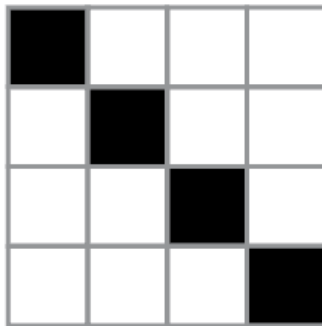
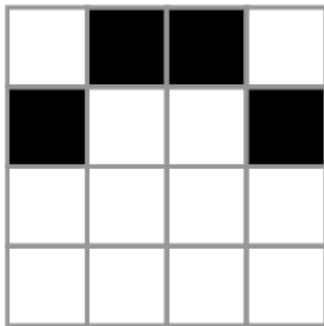
Notice that we have written our program from left to right like you would read a book in English. Some students prefer this method. Others like to start each line of the grid on a new line of paper. The way they write their program doesn't matter as much as whether the other people in their groups can follow along!



The Students' Turn

Group: Divide students into pairs or small groups.

- Have each pair/group choose an image from the worksheet.
- Discuss the algorithm to draw chosen image with partner(s).
- Convert algorithm into a program using symbols.
- Trade programs with another pair/group and draw one another's image.
- Choose another image and go again!



Wrap Up (15 min)

Journaling / Flash Chat

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 another image that you could code. Can you write the program that goes with this drawing?
- What other types of robots could we program if we changed what the arrows meant?

Optional Assessment (10 min)

Graph Paper Programming - 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.

Better and Better

- Have your class try making up their own images.
- Can they figure out how to program the images that they create?

Class Challenge

- As the teacher, draw an image on a 5x5 grid.
- Can the class code that up along with you?

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ AP - Algorithms & Programming



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

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

Lesson 2: Introduction to Online Puzzles

Sequencing | Debugging | Loops | Angry Bird | Collector | Artist | Harvester

Overview

In this set of puzzles, students will begin with an introduction (or review depending on the experience of your class) of Code.org's online workspace. There will be videos pointing out the basic functionality of the workspace including the Run, Reset, and Step buttons. Also discussed in these videos: dragging Blockly blocks, deleting Blockly blocks, and connecting Blockly blocks. Next, students will practice their **sequencing** and **debugging** skills in maze. From there, students will see new types of puzzles like Collector, Artist, and Harvester when they learn the very basics of **loops**.

Purpose

We recognize that every classroom has a spectrum of understanding for every subject. Some students in your class may be computer wizards, while others haven't had much experience at all. In order to create an equal playing (and learning) field, we have developed this "Ramp Up Stage" for Course D. This can be used as either an introduction or a review of how to use Code.org and basic computer science concepts. This stage covers all prerequisites needed to start Course D.

Agenda

Warm Up (10 min)

Introduction
Vocabulary

Bridging Activities - Programming (10 min)

Unplugged Activity Using Paper Blocks
Preview of Online Puzzles as a Class

Main Activity (30 min)

Course D Online Puzzles - Website

Wrap Up (10 min)

Journaling

Objectives

Students will be able to:

- Order movement commands as sequential steps in a program.
- Modify an existing program to solve errors.
- Break down a long sequence of instructions into the largest repeatable sequence.

Preparation

Play through the **Course D Online Puzzles - Website** to find any potential problem areas for your class.

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 Teacher

- **Course D Online Puzzles - Website**
- **CS Fundamentals Main Activity Tips - Lesson Recommendations**

For the Students

- **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives (download)**
- **Think Spot Journal - Reflection Journal**

Vocabulary

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

Teaching Guide

Warm Up (10 min)

Introduction

Students will either be learning a lot of new concepts or reviewing a lot of basic concepts. Based on your class's experience, you can cover the following vocabulary or move on to a bridging activity. We recommend using the following words in sentences if the definitions aren't explicitly covered.

Vocabulary

This lesson has four new and important vocabulary words:

- **Program** - Say it with me: Pro - Gram An algorithm that has been coded into something that can be run by a machine.
- **Programming** - Say it with me: Pro - Gramm - ing The art of creating a program.
- **Bug** - Say it with me: Bug An error in a program that prevents the program from running as expected.
- **Debugging** - Say it with me: De - Bugg - ing Finding and fixing errors in programs.
- **Loop** - Say it with me: Loo-p The action of doing something over and over again

Bridging Activities - Programming (10 min)

This activity will help bring the unplugged concepts from "Graph Paper Programming" 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

Similar to "Graph Paper Programming", have the students in your class pair up. Pass out multiple fill 1 and move ___ blocks from the **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives** to each pair. Have each pair of students draw a design on a four by four graph from **Graph Paper Programming - Worksheet**. Next, have the students work together to write the program needed to draw this design using the paper Blockly blocks. The students will need to write up, down, right, or left on the move ___ block. Make sure the students know that the program goes from top to bottom and the blocks need to touch!

Preview of Online Puzzles as a Class

Pull up a puzzle from **Course D Online Puzzles - Website**. We recommend puzzle 6 for this activity. Break up the students into groups of three or four. Have them "program" Red, the angry bird, to get to the pig using arrows from "Graph Paper Programming."



The class will not need to use the last arrow.

Once all the groups have an answer, discuss the path as a class.

Main Activity (30 min)

Course D Online Puzzles - Website

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

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

Teacher Tip:

Show the students the right way to help classmates:

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

Wrap Up (10 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel about today's lesson?
- List some of the bugs you found in your programs today.
- What was your favorite puzzle to complete? Draw your favorite character completing a puzzle.

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ **AP** - Algorithms & Programming



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

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

Lesson 3: Relay Programming

Unplugged | Relay Programming | Algorithms

Overview

This activity will begin with a short lesson on debugging and persistence, then will quickly move to a race against the clock as students break into teams and work together to write a program one instruction at a time.

Purpose

Teamwork is very important in computer science. Teams write and debug code with each other, instead of working as individuals. In this lesson, students will learn to work together while being as efficient as possible.

This activity also provides a sense of urgency that will teach students to balance their time carefully and avoid mistakes without falling too far behind. This experience can be stressful (which is expected!) Make sure you provide students with the tools to deal with potential frustration.

Agenda

Warm Up (15 min)

Where did I go wrong?

Main Activity (20 min)

Relay Programming

Wrap Up (15 min)

Journaling

Extended Learning

Objectives

Students will be able to:

- Define ideas using code and symbols.
- Verify work done by teammates.
- Identify signs of frustration

Preparation

Watch the **Relay Programming - Teacher Video**.

Locate a wide open space for this activity, such as the gym or outdoor field.

Print out one **Relay Programming Activity Packet - Activity Packet** for each group.

Supply each group with plenty of paper and pens/pencils.

Print one **Relay Programming - 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 Teacher

- **Relay Programming - Teacher Debugging** - Image
- **Relay Programming** - Unplugged Video (download)
- **Relay Programming** - Teacher Video

For the Students

- **Relay Programming Debugging Packet**
- **Relay Programming Activity Packet** - Activity Packet
- **Think Spot Journal** - Reflection Journal

Vocabulary

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

- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Frustrated** - Feeling annoyed or angry because something is not the way you want it.
- **Persistence** - Trying again and again, even when something is very hard.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Teaching Guide

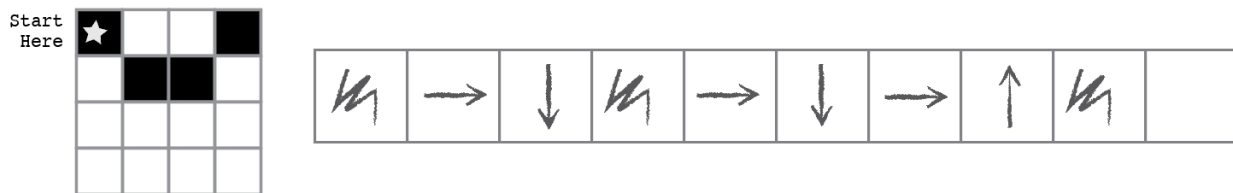
Warm Up (15 min)

Recall that in "Graph Paper Programming" we guided our teammate's Automatic Realization Machine (ARM) using arrows. This warm up will bring back these ideas, which will be needed in the main activity.

Where did I go wrong?

Goal: In this lesson, we want to help students learn to identify and fix bugs in their own programs. The easiest way to do that is to first present students with a program that contains bugs that are not their fault. Once they've helped you fix "your" program, share with them how frustrating it can be to make mistakes, and help them see that those feelings are completely normal and they shouldn't feel embarrassed by them.

Display: Show your students the provided **Relay Programming - Teacher Debugging - Image** .



Discuss: Get the attention of the class and let them know that you are stuck! You have this challenge, and you thought you had solved it, but it doesn't seem to be working. Your program has a bug, can they help you fix it?

Take a moment to walk them through the rules: - Start at the star - Follow the instructions step-by-step - End when all of the right squares are filled in

Optional: Follow along by filling in a blank grid. Express frustration when the picture doesn't turn out the way that you wanted it to.

Think: Can you figure out why my program doesn't work?

Pair: Let students work together to see if they can figure out what the program is supposed to say.

Share: Ask students if anyone was able to figure out a way to solve the problem. When you get a correct answer, let the students know that they are great at "debugging"!

Discuss: Ask the students if they could tell how you were feeling when you couldn't figure out the answer. They might suggest that you were "mad" or "sad". Instead of telling them "no", describe that you were feeling a little bit mad, a little bit sad, and a little bit confused. When you put all of those emotions together, it makes a feeling called "frustration". When you are "frustrated" you might think you are mad, sad, or confused -- and you might be tempted to give up -- but frustration is a natural feeling and it's a big hint that you are about to learn something! Instead of quitting, practice persistence. Keep trying over and over again. After a few times, you will start to understand how to debug your problems!

Distribute: To make sure that students understand the idea of finding and fixing errors (debugging) pass out the **Relay Programming Debugging Packet** and have students complete the task in pairs.

Optional: If you want to move the activity along more quickly, feel free to complete these as a class, instead.

Transition: Now it's time to play the game!

Main Activity (20 min)

Relay Programming

With Graph Paper Programming in mind, it's time to split up into teams and prepare to run the activity as a relay!

Set-Up: Prepare the **Relay Programming Activity Packet - Activity Packet** by printing out one copy for each team of 4-5 students. Cut or fold each page along the center dotted line.

Go over the rules of the game with your class:

- Divide students into groups of 3-5.
- Have each group queue up relay-style.
- Place an identical image at the other side of the room/gym/field from each team.
- Have the first student in line dash over to the image, review it, and write down the first symbol in the program to reproduce that image.
- The first student then runs back and tags the next person in line, then goes to the back of the queue.
- The next person in line dashes to the image, reviews the image, reviews the program that has already been written, then either debugs the program by crossing out an incorrect symbol, or adds a new one. That student then dashes back to tag the next person, and the process continues until one group has finished their program.

First group to finish with a program that matches the image is the winner! Play through this several times, with images of increasing difficulty.

Go through the game as many times as you can before time runs out or your students begin feeling exhausted.

Transition: Once the game is over, circle everyone up to share lessons learned.

Discuss: What did we learn today?

- What if each person on a team were allowed to do five arrows at a time?
 - How important would it be to debug our own work and the work of the programmer before us?
 - How about with 10 arrows?
 - 10,000? Would it be more or less important?
- Do you think a program is better or worse when more than one person has worked on it?
- Do you think people make more or fewer mistakes when they're in a hurry?
- If you find a mistake, do you have to throw out the entire program and start over?

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:

Content Corner

For more on persistence and frustration, try reading **Stevie and the Big Project** to your students. It will help them spot moments of frustration. It will also help give them the tools to deal with it.

If you do not read the book, take a moment to cover tips on frustration and persistence as a class:

Tips to Help With Frustration

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

Tips for Being Persistent

- Keep track of what you have already tried
- Describe what is happening
- Describe what is supposed to happen
- What does that tell you?
- Make a change and try again

Clarifications

Here are some clarifications that need to be shared from time to time:

- Only one person from each group can be at the image at one time.
- It is okay to discuss algorithms with the rest of the group in line, even up to the point of planning who is going to write what when they get to the image.
- When a student debugs a program by crossing out an incorrect instruction (or a grouping of incorrect instructions) this counts as their entire turn. The next player will need to figure out how to correct the removed item.

- What was today's lesson about?
- How did you feel during today's lesson?
- How did teamwork play a role in the success of writing today's program?
- Did you start to get frustrated at any point? What did you do about it?

Extended Learning

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

Pass the paper

- If you don't have the time or room for a relay, you can have students pass the paper around their desk grouping, each writing one arrow before they move the paper along.

Fill It, Move It

- As the teacher, draw an image with as many filled squares as children in each group.
- Have the students write as many arrows in the program as it takes to get to a filled-in square (including actually filling that square in) before passing to the next person.

Debugging Together

Draw an image on the board. Have each student create a program for the image. Ask students to trade with their elbow partner and debug each other's code.

- Circle the first incorrect step, then pass it back.
- Give the students another chance to review and debug their own work.
- Ask for a volunteer to share their program.

Ask the class:

- How many students had the same program?
- Anyone have something different?

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ AP - Algorithms & Programming



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

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

Lesson 4: Debugging with Laurel

Debugging | Bug | Collector | Laurel

Overview

In this online activity, students will practice debugging in the "collector" environment. Students will get to practice reading and editing code to fix puzzles with simple algorithms, loops and nested loops.

Purpose

The purpose of this lesson is to teach students that failure is normal when learning a new skill. Students will be given pre-written programs that do NOT work. They will be asked to fix these programs. This process, called "debugging", teaches students essential problem solving and critical thinking skills. These skills transfer over as students proceed to harder and harder programming projects.

Agenda

Warm Up (15 min)

Introduction

Bridging Activities - Debugging (15 min)

Unplugged Activity with Paper Blocks

Preview of Online Puzzles as a Class

Main Activity (30 min)

Course D Online Puzzles - Website

Wrap Up (15 min)

Journaling

Objectives

Students will be able to:

- Read and comprehend given code.
- Identify a bug and the problems it causes in a program.
- Describe and implement a plan to debug a program.

Preparation

Play through **Course D Online Puzzles - Website** to find any potential problem areas for your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Course D Online Puzzles - Website**
- **CS Fundamentals Main Activity Tips - Lesson Recommendations**

For the Students

- **Relay Programming Activity Packet - Activity Packet**
- **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives (download)**
- **Think Spot Journal - Reflection Journal**

Vocabulary

- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.

Teaching Guide

Warm Up (15 min)

Introduction

One of the most important parts of learning to program is learning to debug. Ask the class if they have ever learned a new skill and faced failure.

For example:

- Learning to ride a bike and falling down
- Learning to bake and burning the food
- Learning to play a sport and not winning a game

Facing failure is very common when learning new things. Have students discuss past failures and how they overcame them.

In programming, computer scientists often run into "bugs" in their code.

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

A bug can really mess up the program, so it's important to learn to "debug" your code.

- **Debug:** Finding and fixing problems in your algorithm or program.

Continue the conversation if you think your class needs more of an introduction, but leave time for one of the bridging activities.

Bridging Activities - Debugging (15 min)

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

Unplugged Activity with Paper Blocks

Break your class up into teams of 3-5 and go to a large space. This space could be in a gym or outside. Queue up these teams like in "Relay Programming". Pick a semi-difficult design from **Relay Programming Activity Packet - Activity Packet**. Display this design at the end of a long distance between each team. Along with the display, provide each team with enough paper blocks from **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives**. Each team will need plenty of fill 1 and move ___ blocks. The move ___ blocks can either be filled in ahead of time, or filled in during the game. Either way, make sure these blocks are properly defined when the game is playing out.

Once all the teams are lined up, display or read out the following rules:

- The first student in line gets to run over to the image, review it, and place the first code block in the program to reproduce that image.
- The first student then runs back and tags the next person in line, then goes to the back of the queue.
- The next person in line dashes to the image, reviews the image, reviews the program that has already been written, then either debugs the program by taking out an incorrect code block, or adds a new block in. That student then dashes back to tag the next person, and the process continues until one group has finished their program.

Make sure the students are only using the fill 1 or move ___ blocks, and only placing one down per turn. The first team to **correctly** write out the code for their image wins.

Preview of Online Puzzles as a Class

Group the students into teams of 3. Choose a puzzle from the **Course D Online Puzzles - Website** associated with this lesson. We recommend the third puzzle. Have the students in each team sit at a computer with the puzzle displayed. Each team only gets one computer and only one student can be looking at the screen. Display or read the following rules:

- Only one student in each team can look at the screen.
- This person can only delete or add one block at a time. Once that person has added or removed a block, they can tap the shoulder of the next person.
- The next person can proceed to play out their turn.
- No turns can be skipped or repeated, everyone must play an equal amount.

The first team to finish the puzzle **correctly** wins!

Main Activity (30 min)

Course D Online Puzzles - Website

It might be helpful for students to sit with their teams from the bridging activities. Every student should work on these puzzles individually or in pairs, but having a closely knit group to ask and answer questions with can help develop confidence and understanding with the subject matter.

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 bug? How do you know there is a bug in your program?
- What does it mean to debug code? How do you debug a program?

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ **AP** - Algorithms & Programming



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

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

Lesson 5: Events in Bounce

Event | Bounce

Overview

In this online activity, students will learn what events are, and how computers use them in programs like video games. Students will work through puzzles making the program react to events (like arrow buttons being pressed.) At the end of the puzzle, students will have the opportunity to customize their game with different speeds and sounds.

Purpose

Events are very common in computer programs, especially in video games.

In this lesson, students will develop their understanding of events by making a sports-based game. Students will learn to make their paddle move according to arrow keys, and make noises when objects collide. At the very end, they will get to customize their game to make it more unique!

Agenda

Warm Up (10 min)

Introduction

Main Activity (30 min)

Course D Online Puzzles - Website

Wrap Up (10 min)

Journaling

Extended Learning

Objectives

Students will be able to:

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

Preparation

Make sure every student has a **Think Spot Journal - Reflection Journal**.

Play through **Course D Online Puzzles - Website** in stage 3 to find any potential problem areas for your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Links

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

For the Teacher

- **Course D Online Puzzles - Website**
- **CS Fundamentals Main Activity Tips - Lesson Recommendations**

For the Students

- **Think Spot Journal - Reflection Journal**

Vocabulary

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

Teaching Guide

Warm Up (10 min)

Introduction

Ask the students to come sit down near you. Now tell them to all stand up!

Tell the students what you just did was declare an event and an action. When you say to sit down, it is an event. The action responding to this event is the class sitting down. This is the same when you ask the class to stand up. Events and actions are easily identifiable in our lives.

Some other events and actions include:

- Feeling hungry and eating food
- Stubbing your toe and yelling "Ouch!"
- Getting the basketball in the basket and scoring a point for your team!

Ask the class to come up with a couple of more events. Tell them that they will be making a game where the program will have actions associated to events that they code!

Main Activity (30 min)

Course D Online Puzzles - Website

At the end of the set of puzzles, students will have the opportunity to make their game unique. Have the students try new ways to make the game more challenging. For example, try playing with many balls at once, or each time the ball bounces off a wall, launch more balls.

Teacher Tip

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

This is what **Justin Trudeau, the Prime Minister of Canada, did when he completed an Hour of Code in 2016** .

□

Wrap Up (10 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- What did you do to make your game super cool?
- What kind of game do you want to code in the future?

Extended Learning

Take Me Out to the Ball Game

Take the students outside to play some sort of ball game. Keep track of events and actions. For example, not dribbling in basketball results in a traveling foul and the other team gets the ball. In soccer, kicking the ball out of bounds results in the

other team kicking the ball in. Getting the ball to the goal results in a point! Make up more events if your students are into it. Have the all of the students yell "Yippee" when the captain of one team scores a point. Have everyone fall to the ground and roll around if a student makes two goals in a row!

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ **AP** - Algorithms & Programming



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

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

Lesson 6: Loops in Ice Age

Loop | Ice Age | Scrat

Overview

As a quick update (or introduction) to using loops, this stage will have students using the `repeat` block to get Scrat to the acorn more efficiently.

Purpose

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

Agenda

Warm Up - The Unplugged Foundation (10 min)

Repeat After Me

Online Foundation: Preview Loops in Ice Age

Main Activity (30 min)

Course B Online Puzzles - Website

Wrap Up (5 - 10 min)

Journaling

Extended Learning

Objectives

Students will be able to:

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

Preparation

(Optional) Pick a couple of puzzles to do as a group with your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Make sure each student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Course D Online Puzzles - Website**
- **CS Fundamentals Main Activity Tips - Lesson Recommendations**

For the Students

- **Feeling Faces - Emotion Images**
- **Think Spot Journal - Reflection Journal**

Vocabulary

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

Teaching Guide

Warm Up - The Unplugged Foundation (10 min)

Repeat After Me

Model: Ask for a volunteer and have them stand.

- Instruct your volunteer to walk around the table (or their chair, or a friend).
- When they finish, instruct them to do it again, using the exact same words you did before.
- When they finish, instruct again.
- Then again.

Prompt: Would it have been easier for me to just ask you to go around the table four times?

Think: What if I wanted you to do it ten times? How would you reword my instructions so that they were more efficient and I didn't have to repeat myself so much? Feel free to write your instructions down on a piece of scrap paper.

Share: Ask a few students to share their instructions with the class, pointing out how each approach has simplified the overall approach to giving instructions.

Say: Today we're going to work on finding ways to make giving lots of instructions easier, especially when those instructions repeat themselves a lot.

Online Foundation: Preview Loops in Ice Age

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

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

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

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

Main Activity (30 min)

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

Course B Online Puzzles - Website

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

- Utilize **Pair Programming - Student Video** whenever possible
- Encourage students with questions/challenges to start by asking their partner
- Unanswered questions can be escalated to a nearby group, who might already know the solution

💡 Teacher Tip:

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

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

- Remind students to use the debugging process before you approach
- Have students describe the problem that they're seeing. What is it supposed to do? What does it do? What does that tell you?
- Remind frustrated students that frustration is a step on the path to learning, and that persistence will pay off.
- If a student is still stuck after all of this, ask leading questions to get the student to spot an error on their own.

Wrap Up (5 - 10 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw your own maze with Scrat trying to get to an Acorn. Will loops help you solve it?
- Draw yourself using a loop to do an everyday activity, like brushing your teeth.

Extended Learning

So Moving

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

Connect It Back

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

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ **AP** - Algorithms & Programming



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

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

Lesson 7: Drawing Shapes with Loops

Loop | Artist

Overview

This lesson builds on the understanding of loops from previous lessons and gives students a chance to be truly creative. This activity doubles as a debugging exercise for extra problem-solving practice.

Purpose

This series highlights the power of loops with an array of puzzles meant to get students thinking about why repeat loops are superior to longhand.

Agenda

Warm Up (15 min)

Introduction

Main Activity (30 min)

Course C Online Puzzles - Website

Wrap Up (15 min)

Journaling

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 **Course D Online Puzzles - Website** corresponding to this course to find any potential problem areas for your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Course D Online Puzzles - Website**
- **CS Fundamentals Main Activity Tips - Lesson Recommendations**

For the Students

- **Think Spot Journal - Reflection Journal**

Vocabulary

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

Teaching Guide

Warm Up (15 min)

Introduction

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

- Creating a new dance with loops just like in **"Getting Loopy"**
- Reviewing a puzzle from the last lesson
- Previewing a puzzle from this lesson

Any of these should help prepare your class for fun with the online puzzles!

Main Activity (30 min)

Course C Online Puzzles - Website

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 as they program online.

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 could program using loops? Can you come up with the code to create it?

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ AP - Algorithms & Programming



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

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

Lesson 8: Nested Loops in Maze

Nested Loops | Loops | Bee | 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)

Course D Online Puzzles - Website

Wrap Up (15 min)

Journaling

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 **Course D Online Puzzles - Website** to find any potential problem areas for your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Course D Online Puzzles - Website**
- **CS Fundamentals Main Activity Tips - Lesson Recommendations**

For the Students

- **Think Spot Journal - Reflection Journal**

Vocabulary

- **Command** - An instruction for the computer. Many commands put together make up algorithms and computer programs.
- **Loop** - The action of doing something over and over again.
- **Repeat** - 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)

Course D Online Puzzles - Website

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.

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

- ▶ AP - Algorithms & Programming



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

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

Lesson 9: Fancy Shapes using Nested Loops

Nested Loops | Loops | Artist

Overview

Students will create intricate designs using Artist in today's set of puzzles. By continuing to practice nested loops with new goals, students will see more uses of loops in general. This set of puzzles also offers a lot more potential for creativity with an opportunity for students to create their own design at the end of the stage.

Purpose

In this online activity, students will create designs in Artist that they can proudly share with their loved ones.

The purpose of this activity is to utilize nested loops as a way to inspire students with artistic minds to see coding as another creative outlet. This set of puzzles was built to develop critical thinking skills, an understanding of elementary geometry, and creativity -- all within the scope of nested loops!

Agenda

Warm Up (10 min)

Introduction

Main Activity (30 min)

Course D Online Puzzles - Website

Wrap Up (15 min)

Flash Chat: What did you make today? Journaling

Extended Learning

Objectives

Students will be able to:

- Combine simple shapes into complex designs with nested loops.
- Count the number of times an action should be repeated and represent it as a loop.
- Break complex tasks into smaller repeatable sections.

Preparation

Play through **Course D Online Puzzles - Website** to find any potential problem areas for your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Make sure every student has a **Think Spot Journal - Reflection Journal**.

Either display or print out copies of **Turns & Angles - Student Handout** for students to reference while working through the online puzzles.

Links

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

For the Teacher

- **Course D Online Puzzles - Website**
- **CS Fundamentals Main Activity Tips - Lesson Recommendations**

For the Students

- **Turns & Angles - Student Handout**
- **Turns & Angles - Student Video**
- **Think Spot Journal - Reflection Journal**

Vocabulary

- **Loop** - The action of doing something over

and over again.

- **Repeat** - Do something again

Teaching Guide

Warm Up (10 min)

Introduction

Review using nested loops in Maze.

Ask the students how they felt about nested loops.

- What did they like and dislike about them?
- What are some advantages of using nested loops?

Tell the students that they will be using nested loops again, but in Artist this time. They will be making amazing projects today!

Main Activity (30 min)

Course D Online Puzzles - Website

Students might benefit from having a puzzle done as a class. If you believe your class could benefit from that, we recommend puzzle 2 of stage 5.

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.

Be sure to 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, so paper can be helpful if students like to write out the long answer before searching for repeating patterns.

💡 Teacher Tip

Students will have the opportunity to share their own work at the end of this stage. These pieces of artwork can be shared virtually or printed out. We recommend printing out the class's work and displaying it for the students' loved ones to see.

Wrap Up (15 min)

Flash Chat: What did you make today?

Get the class together and allow time for students to show off their Artist drawings! Make sure everyone feels included by checking that every student is done with their Artist drawing before starting the presentations. Discuss how each drawing was made and what was in the student's nested loop.

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:

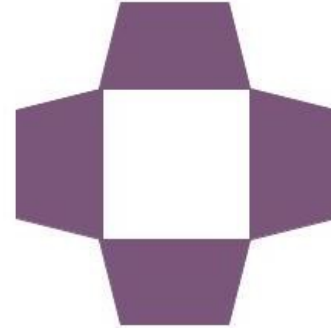
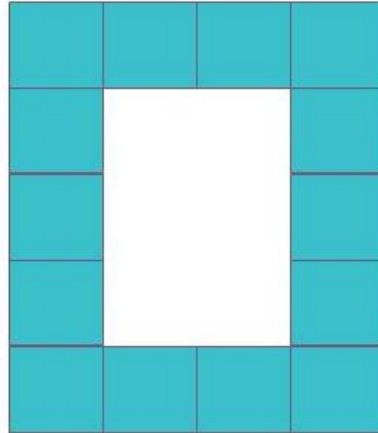
- What was today's lesson about?
- How did you feel during today's lesson?
- Draw something you used nested loops to make.
- How do nested loops help you code complex images?

Extended Learning

Together We Draw

Have the students pair up with two pieces of paper. Partners should individually draw a shape or simple pattern. Once the simple pattern has been drawn, have the partners switch papers. Now each partner must repeat that pattern how ever many times they want. For example, if one partner draws a square, the other partner can make a rectangle made up of squares! If one partner draws a staircase pattern, the other student can fill the page with staircases! Each pair will have a set of unique drawings. If there's time, have students discuss how they might code their drawings.

Here are some examples:



Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ AP - Algorithms & Programming



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

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

Lesson 10: Snowflakes with Anna and Elsa

Loop | Nested Loop | Frozen

Overview

Now that students know how to layer their loops, they can create so many beautiful things. This lesson will take students through a series of exercises to help them create their own portfolio-ready images using Anna and Elsa's excellent ice-skating skills!

Purpose

In this series, students will get practice nesting loops while creating images that they will be excited to share.

Beginning with a handful of instructions, students will make their own decisions when it comes to creating designs for repetition. They will then spin those around a variety of ways to end up with a work of art that is truly unique.

Agenda

Warm Up (15)

Introduction

Main Activity (30)

Course E Online Puzzles - Website

Wrap Up (15)

Journaling

Objectives

Students will be able to:

- Break apart code into the largest repeatable sequences using both loops and nested loops.
- Recognize the difference between using a loop and a nested loop.
- Describe when a loop, nested loop, or no loop is needed.

Preparation

Play through the **Course E Online Puzzles - Website** corresponding to this lesson to find and potential problem areas for your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Course D Online Puzzles - Website**
- **CS Fundamentals Main Activity Tips - Lesson Recommendations**

For the Students

- **Think Spot Journal - Reflection Journal**

Vocabulary

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

Teaching Guide

Warm Up (15)

Introduction

Ask the class to discuss the last set of puzzles.

- What did they like/dislike?
- Which puzzles were hard? Why?
- Which puzzles were easy? Why?
- If you were to teach nested loops to a friend, what would you say to help them understand?

If there's time, give an introduction to the main characters of today's puzzles, Anna and Elsa from Frozen. Give the class the sister's back story if the class doesn't already know. To build excitement, tell the class they will be using nested loops to make some fantastic drawings with Anna and Elsa's ice skates!

Main Activity (30)

Course E Online Puzzles - Website

This set of puzzles is set up as a progression. This means every puzzle builds a foundation for the next puzzle. Students will enjoy making more and more interesting designs by making small and simple changes to code they have already written.

Wrap Up (15)

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?
- When do you use a loop? When do you use a nested loop?
- Thought exercise: Can you make everything a nested loop can with just a normal loop? Can you draw out an example?
 - Answer: Yes, you can, but it is a lot more difficult. Nested loops make programs simpler.

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ AP - Algorithms & Programming



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

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

Lesson 11: While Loops in Farmer

While Loops | Loops | Farmer

Overview

By the time students reach this lesson, they should already have plenty of practice using repeat loops, so now it's time to mix things up.

While loops are loops that continue to repeat commands while a condition is met. While loops are used when the programmer doesn't know the exact number of times commands need to be repeated, but does know what condition needs to be true in order for the loop to continue repeating. For example, students will be working to fill holes and dig dirt in Farmer. They will not know the size of the holes or the height of the mountains of dirt, but the students will know they need to keep filling the holes and digging the dirt as long as the ground is not flat.

Purpose

As your students continue to deepen their knowledge of loops, they will come across problems where a command needs to be repeated, but it is unknown how many times it needs to be repeated. This is where while loops come in. In today's lesson, students will develop a beginner's understanding of condition-based loops and also expand their knowledge of loops in general.

Agenda

Warm Up (10 min)

Introduction

Main Activity (30 min)

Course D Online Puzzles - Website

Wrap Up (15 min)

Journaling

Extended Learning

Bridging Activity (15 min)

Unplugged Activity Using Paper Blocks
Preview of Online Puzzles

Objectives

Students will be able to:

- Distinguish between loops that repeat a fixed number of times and loops that repeat as long as a condition is true.
- Use a while loop to create programs that can solve problems with unknown values.

Preparation

Play through **Course D Online Puzzles - Website** to find any potential problem areas for your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Conditionals with Cards Sample Program** - Teacher Prep Guide
- **Course D Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives ([download](#))
- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Condition** - A statement that a program checks to see if it is true or false. If true, an action is taken. Otherwise, the action is ignored.
- **Loop** - The action of doing something over and over again.

- **Repeat** - Do something again
- **While Loop** - A loop that continues to repeat while a condition is true.

Teaching Guide

Warm Up (10 min)

Introduction

Use "while" in a sentence in front of the students. Ask the students what the word "while" means. If you were to say "while there is a hole, fill it with dirt" what would they do? How long would they do that?

When you use a word like "while", you are relying on a condition to tell the computer how long the loop should run. A condition is a statement that is tested and found to be true or false. In the case above, the condition is if there is a hole. It's only possible for there to be a hole or for there not to be a hole, thus the statement is only ever true or false.

Tell the students they will be learning about a new kind of loop. Previously, students only used loops to repeat a command a certain number of times. Here, they won't always know how many times to repeat the command, however, they will know when to stop or when to keep going. While loops allow the programmer to repeat a command as long as a condition is still true. In the previous example, the condition is the existence of a hole.

If there's time, have the students discuss other times using a while loop would be useful. Examples include:

- Running toward a ball **while** it is in front of you.
- Filling a glass **while** it has space for more liquid.
- Walk forward **while** there is a path ahead.

Main Activity (30 min)

Course D Online Puzzles - Website

While loops are not always a difficult concept for students to understand, but if you think your class might struggle with these puzzles, we recommend **Pair Programming - Student Video**. This will allow students to bounce ideas of each other before implementing the code. Pair programming works to increase confidence and understanding with topics like while loops.

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 is the difference between a while loop and a normal repeat loop?
- Give an example of a puzzle where you would use a while loop, but not use a repeat loop. Can you give an example of a puzzle where you would use a repeat loop, but not a while loop?

Extended Learning

While Simon Says

Go out to a large playing field and have the students stand in a line facing you. Make sure every student can see you. Declare a couple of while loops such as:

- While my right hand is up, you can walk toward me
- While I cover my eyes, you can skip toward me
- While my head is turned to the right, you have to walk backwards

The first student to get to you wins. If there's time, let other students be "Simon" at the front of the class.

Bridging Activity (15 min)

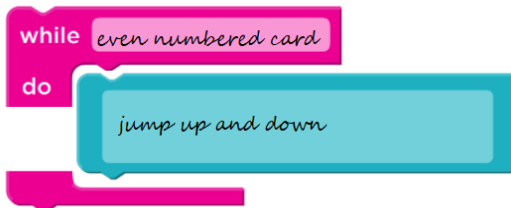
This activity will help bring the unplugged concepts from "Conditionals With Cards" into the online world that the students are moving into. It will also help bridge the concept of conditionals to a new type of loop- while loops. Choose one of the following to do with your class:

Unplugged Activity Using Paper Blocks

Print and cut out 2-3 while / do blocks and blank action blocks from **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives** and pull out a deck of cards. Ask the class to come up with a couple of conditionals to use with the deck of cards like they did in "Conditionals with Cards." This time, instead of using the conditional in an if / else statement, they will be using it in a while loop.

When the conditionals have been decided on as a class, fill in the blank part of the while block with the various card groups that the kids came up with. Examples include "Even Numbered", "Red card", or "Diamonds". Fill in the action blocks with the actions the students came up with. Make sure the students know the action blocks need to be directly inside the while block. Line the blocks up sequentially so students can see how it runs as a program.

Below is an example.



Now shuffle the deck of cards and play "Conditionals with Cards" again. Flip through the deck card-by-card, reacting to cards if a conditional has been made for it. To maintain authenticity for this while loop experience, make sure you hold up the selected card for a good long while to give students time to react over several beats. Feel free to jump back to previous cards to get the point across that once you have left a while loop, you continue moving forward and do not return to the previous loop just because the conditions match again.

Preview of Online Puzzles

Pull up a puzzle from **CSF Express Course - Website**, we recommend Puzzle 6.

- Ask the class what the farmer should do when she gets to the pile of dirt.
 - The farmer should use a conditional in a while loop to check if there is a pile.
- Use the while there is a pile / do block. Ask the class what the farmer should do if there is no pile. When should the farmer stop?
 - The farmer should remove 1 if there is a pile. The while loop will continue until there is no pile, so then your code is finished!

Fill in the rest of the code and press Run. Discuss with the class why this worked.

Standards Alignment

CSTA K-12 Computer Science Standards



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

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

Lesson 12: Until Loops in Maze

Conditional | Loop | Maze | Angry Bird | Zombie

Overview

In this lesson, students will learn about until loops. Students will build programs that have the main character repeat actions until they reach their desired stopping point.

Purpose

This set of puzzles will work to solidify and build on the knowledge of loops by adding the until conditional. By pairing these concepts together, students will be able to explore the potential for creating complex and innovative programs.

Agenda

Warm Up (10 min)

Introduction

Bridging Activity (15 min)

Unplugged Activity Using Paper Blocks
Preview of Online Puzzles

Main Activity (30 min)

Course D Online Puzzles - Website

Wrap Up (15 min)

Journaling

Extended Learning

Objectives

Students will be able to:

- Build programs with the understanding of multiple strategies to implement conditionals.
- Translate spoken language conditional statements and loops into a program.

Preparation

Play through **Course D Online Puzzles - Website** to find any potential problem areas for your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Course D Online Puzzles - Website**
- **CS Fundamentals Main Activity Tips - Lesson Recommendations**

For the Students

- **Think Spot Journal - Reflection Journal**

Vocabulary

- **Condition** - A statement that a program checks to see if it is true or false. If true, an action is taken. Otherwise, the action is ignored.
- **Conditionals** - Statements that only run under certain conditions.
- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again
- **Until** - A command that tells you to do something only up to the point that something

becomes true.

Teaching Guide

Warm Up (10 min)

Introduction

In this lesson, students will be creating loops that only run until a condition is true. Help the students understand how this works by leading them in group activities and having them do an action until some condition is true. **For example: Have students touch their nose until you tell them to stop.**

Bridging Activity (15 min)

This activity will help bring the unplugged concepts from "Conditionals With Cards" into the online world that the students are continuing to move into. It will also help bridge the concept of conditionals to repeat until loops. Choose one of the following to do with your class:

Unplugged Activity Using Paper Blocks

Print and cut out 2-3 repeat until loops and some blank action blocks from **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives**. Pull out a deck of cards. Ask the class to come up with a couple of conditionals to use with the deck of cards like they did in "Conditionals with Cards." When the conditionals have been decided on as a class, fill in the blank part of the repeat until loop with the various card groups that the kids came up with. Examples include "Even Numbered", "Red card", or "Diamonds". Fill in the action blocks with the actions the students came up with. Make sure the students know the action blocks need to be directly inside the repeat until block. Line the blocks up sequentially so students can see how it runs as a program. Below is an example.



Now shuffle the deck of cards and play "Conditionals with Cards" again. Flip through the deck card-by-card, reacting to cards if a conditional has been made for it. Make sure students know that once you have left the until loop, you continue moving forward and do not return to the previous loop just because the conditions match again.

Preview of Online Puzzles

Pull up a puzzle from **Course D Online Puzzles - Website**, we recommend Puzzle 4.

- Ask the class what the bird should repeat to get to the pig.
 - The bird should repeat move forward, turn right, move forward, and then turn left.
- Ask the class what they can use to repeat this code.
 - The bird should repeat this pattern **until** it reaches the pig.

Fill in the rest of the code using the repeat until loop and press Run. Discuss with the class why this worked.

Main Activity (30 min)

Course D Online Puzzles - Website

Bringing together concepts is not easy, but this set of lessons is meant to help students see the endless possibilities of coding when using conditions. If students struggle at all with understanding the similarities or differences between `while` loops and `until` loops, have them try to think of how they would use similar statements in their real lives.

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's the difference between an `until` loop and a `while` loop?

Extended Learning

Until Simon Says

Go out to a large playing field and have the students stand in a line facing you. Make sure every student can see you. Declare a couple of "until loops" such as:

- Until my right hand is raised up, you can walk toward me
- Until I say "eggplant", you can walk backwards in my direction
- Until I turn my head to the right, you have to walk like a crab

The first student to get to you wins. If there's time, let other students be "Simon" at the front of the class.

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ AP - Algorithms & Programming



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

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

Lesson 13: Conditionals with Cards

Conditionals | Unplugged

Overview

This lesson demonstrates how conditionals can be used to tailor a program to specific information. We don't always have all of the information we need when writing a program. Sometimes you will want to do something different in one situation than in another, even if you don't know what situation will be true when your code runs. That is where conditionals come in. Conditionals allow a computer to make a decision, based on the information that is true any time your code is run.

Purpose

One of the best parts of teaching **conditionals** is that students already understand the concept from their everyday lives.

This lesson merges computer science into the real world by building off of their ability to tell if a condition is true or false. Students will learn to use `if` statements to declare when a certain command should be run, as well as `if / else` statements to declare when a command should be run and what do run otherwise. Students may not recognize the word **conditionals**, but most students will understand the idea of using "if" to make sure that some action only occurs when it is supposed to.

Agenda

Warm Up (20 min)

Vocabulary

Introduction

Main Activity (20 min)

Conditionals with Cards Sample Program - Teacher Prep Guide

Wrap Up (15 min)

Flash Chat: What did we learn?

Journaling

Assessment (5 min)

Conditionals with Cards - Assessment

Extended Learning

Objectives

Students will be able to:

- Define circumstances when certain parts of a program should run and when they shouldn't.
- Determine whether a conditional is met based on criteria.
- Traverse a program and predict the outcome, given a set of input.

Preparation

Watch the **Conditionals with Cards - Teacher Video**.

Watch the **Conditionals with Cards - Lesson in Action Video**.

Gather decks of cards or something similar.

One **Conditionals with Cards Sample Program - Teacher Prep Guide** for the class to look at.

Print one **Conditionals with Cards - 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 Teacher

- **Conditionals with Cards** - Unplugged Video ([download](#))
- **Conditionals with Cards** - Teacher Video
- **Conditionals with Cards** - Lesson in Action Video
- **Conditionals with Cards Sample Program** - Teacher Prep Guide
- **Conditionals with Cards** - Assessment
- **Conditionals with Cards** - Assessment Video
- **Conditionals with Cards** - Assessment

Answer Key

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Conditionals** - Statements that only run under certain conditions.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

Conditionals - Say it with me: Con-di-shun-uls

Statements that only run under certain conditions.

Introduction

- We can start this lesson off right away
 - Let the class know that if they can be completely quiet for thirty seconds, you will do something like:
 - Sing an opera song
 - Give five more minutes of recess
 - Do a handstand
 - Start counting right away.
 - If the students succeed, point out that they succeeded, so they get the reward.
 - Otherwise, point out that they were not completely quiet for a full thirty seconds, so they do not get the reward.
- Ask the class "What was the condition of the reward?"
 - The condition was IF you were quiet for 30 seconds
 - If you were, the condition would be true, and you would get the reward.
 - If you weren't, the condition would be false, so the reward would not apply.
 - Can we come up with another conditional?
 - If you can guess my age correctly, the class can give you applause.
 - If I know an answer, I can raise my hand.
 - What examples can you come up with?
- Sometimes, we want to have an extra condition, in case the "IF" statement is not true.
 - This extra condition is called an "ELSE" statement
 - When the "IF" condition isn't met, we can look at the "ELSE" for what to do
 - Example: IF I draw a king from this deck of cards, everybody claps. Or ELSE, everyone says "Awwwwwwwwe."
 - Let's try it. (Draw a card and see if your class reacts appropriately.)
 - Ask the class to analyze what just happened.
 - What was the IF?
 - What was the ELSE?
 - Which condition was met?
 - Believe it or not, we have even one more option.
 - What if I wanted you to clap if I draw a 7, or else if I draw something less than seven you say "YAY," or else you say "Awwwwwwwwe"?
 - This is why we have the terms If, Else-If, and Else.
 - If is the first condition
 - Else-If gets looked at only if the "If" isn't true.
 - Else gets looked at only if nothing before it is true.

Now let's play a game.

Main Activity (20 min)

Conditionals with Cards Sample Program - Teacher Prep Guide

Directions:

- Create a few programs with your class that depend on things like a card's suit, color, or value to award or subtract points. You can write the program as an algorithm, pseudocode, or actual code.

Here is a sample algorithm:

```
if (CARD is RED)
  Award YOUR team 1 point

Else
  Award OTHER team 1 point
```

Here is a sample of the same program in pseudocode:

```
If (card.color == RED){
  points.yours = points.yours + 1;
}

Else {
  points.other = points.other + 1;
}
```

- Decide how you want to split your class into teams.
- Each team should have a pile of cards (at least as many cards as team members) nearby.
- Put one of your “Programs” up on the board for all to see.
- Have the teams take turns drawing cards and following the program to see how many points they score in each round.
- Play several times with several different programs to help the students really understand conditionals.

Once the class has had some practice, you can encourage students to nest conditionals inside one another. Make sure they understand that if the card is red, YOUR team is awarded 1 point, and then **nothing else happens**, since the condition was met:

```
If (CARD is RED)
  Award YOUR team 1 point

Else
  If (CARD is higher than 9)
    Award OTHER team 1 point
  Else
    Award YOUR team the same number of points on the card
```

Here is the same program in pseudocode:

```
If (card.color == RED){
  points.yours = points.yours + 1;
}

Else {
  if (card.value > 9){
    points.other = points.other + 1;
  }
  Else {
    points.yours = points.yours + card.value;
  }
}
```

Wrap Up (15 min)

Flash Chat: What did we learn?

- If you were going to code this up in Blockly, what would you need to add around your conditionals to let the code run more than one time? (A loop)

- What other things do you do during the day under certain conditions?
- If you are supposed to do something when the value of a card is more than 5, and you draw a 5, do you meet that condition?
- Notice that conditions are either "True" or "False."
There is no assessment of a condition that evaluates to "Banana."
- When you need to meet several combinations of conditions, we can use something called "nested conditionals."
 - What do you think that means?
 - Can you give an example of where we saw that during the game?
- What part of that game did you like the best?

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 conditional? How did you use a conditional today?
- What are some of the conditionals you used today? Can you come up with some more that you would use with a deck of cards?

Assessment (5 min)

Conditionals with Cards - 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. Here's a **Conditionals with Cards - Assessment Video** to watch as a guide.

Extended Learning

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

True/False Tag

- Line students up as if to play **Red Light / Green Light**.
- Select one person to stand in front as the Caller.
- The Caller chooses a condition and asks everyone who meets that condition to take a step forward.
 - If you have a red belt, step forward.
 - If you are wearing sandals, take a step forward.
- Try switching it up by saying things like "If you are **not** blonde, step forward."

Nesting

- Break students up into pairs or small groups.
- Have them write if statements for playing cards on strips of paper, such as:
 - the suit is clubs
 - the color is red
- Have students create similar strips for outcomes.
 - Add one point
 - Subtract one point

- Once that's done, have students choose three of each type of strip and three playing cards, paying attention to the order selected.
- Using three pieces of paper, have students write three different programs using only the sets of strips that they selected, in any order.
 - Encourage students to put some if statements inside other if statements.
- Now, students should run through all three programs using the cards that they drew, in the same order for each program.
 - Did any two programs return the same answer?
 - Did any return something different?

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ **AP** - Algorithms & Programming



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

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

Lesson 14: If/Else with Bee

Conditional | Bee | Maze

Overview

Up until this point students have been writing code that executes exactly the same way each time it is run - reliable, but not very flexible. In this lesson, your class will begin to code with conditionals, allowing them to write code that functions differently depending on the specific conditions the program encounters.

Purpose

After being introduced to conditionals in "Conditionals with Cards," students will now practice using them in their programs. The `if / else` blocks will allow for a more flexible program. The bee will only collect nectar **if** there is a flower or make honey **if** there is a honeycomb. Students will also practice and recognize a connection between `if / else` blocks and `while` loops in this set of puzzles.

Agenda

Warm Up (10 min)

Introduction

Bridging Activity - Conditionals (15 min)

Unplugged Activity Using Paper Blocks
Preview of Online Puzzles

Main Activity (30 min)

Course D Online Puzzles - Website

Wrap Up (15 min)

Journaling

Extended Learning

Objectives

Students will be able to:

- Translate spoken language conditional statements into a program.
- Solve puzzles using a combination of looped sequences and conditionals.

Preparation

Play through **Course D Online Puzzles - Website** to find any potential problem areas for your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Course D Online Puzzles - Website**
- **CS Fundamentals Main Activity Tips - Lesson Recommendations**

For the Students

- **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives (download)**
- **Think Spot Journal - Reflection Journal**

Vocabulary

- **Conditionals** - Statements that only run under certain conditions.

Teaching Guide

Warm Up (10 min)

Introduction

Review the "Conditionals with Cards" activity with your students.

- What is a conditional statement?
- When is a conditional useful?
- What are some of the conditions you used in the last activity?

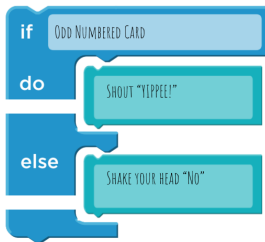
Now we're going to use conditionals with the Code.org bee to help him deal with some mysterious clouds. We don't know if his flowers have nectar or not, so we'll need to use conditionals to make sure that we collect nectar if it's there, but that we don't try to collect nectar from a flower that doesn't have any.

Bridging Activity - Conditionals (15 min)

This activity will help bring the unplugged concepts from "Conditionals With Cards" 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

Print and cut out 2-3 `if / else` and blank action blocks from **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives** and pull out a deck of cards. Ask the class to come up with a couple of conditionals to use with the deck of cards like they did in "Conditionals with Cards." When the conditionals have been decided on as a class, fill in the blank part of the `if` block with the various card values that the kids came up with. Examples include "King of Hearts", "Even Numbered", or "Diamonds". Fill in the action blocks with the actions the students came up with. Make sure the students know the action blocks need to be directly under the `if` or `else` block. Below is an example.



Now shuffle the deck of cards and play "Conditionals with Cards" again. Flip through the deck card-by-card, reacting to cards if a conditional has been made for it.

Preview of Online Puzzles

Pull up a puzzle from Course D, we recommend puzzle 9.

- Ask the class what the bee should do when it gets to the cloud.
 - The bee should use a conditional to check for a flower or a honeycomb.
- Use the `if at flower / else` block. Ask the class what the bee should do if there's a flower. If there's not a flower, there will be a honeycomb. What should the bee do then?
 - The bee should get nectar `if` there is a flower and make honey `if` there is a honeycomb.

Fill in the rest of the code and press Run . Discuss with the class why this worked.

Main Activity (30 min)

Course D Online Puzzles - Website

These puzzles might sprout some questions, so have the students work in pairs or implement the "Ask three before you ask me" rule (have the students ask three other peers for help before they go to the teacher.) This will spark discussions that will develop each student's understanding.

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 today's lesson make you feel?
- What conditionals did you use in your code today?
- What are some other conditionals a bee might use? Examples include:
 - if there is a tree in front of me, buzz out of the way
 - if my wing is hurt, rest on the ground
 - if I see another bee, say "Hello!"

Extended Learning

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

True/False Tag

- Line students up as if to play **Red Light / Green Light**.
- Select one person to stand in front as the Caller.
- The Caller chooses a condition and asks everyone who meets that condition to take a step forward.
 - If you have a red belt, step forward.
 - If you are wearing sandals, take a step forward.
- Try switching it up by saying things like "If you are not blonde, step forward."

Nesting

- Break students up into pairs or small groups.
- Have them write if statements for playing cards on strips of paper, such as:
 - If the suit is clubs
 - If the color is red
- Have students create similar strips for outcomes.
 - Add one point
 - Subtract one point
- Once that's done, have students choose three of each type of strip and three playing cards, paying attention to the order selected.
- Using three pieces of paper, have students write three different programs using only the sets of strips that they selected, in any order.
 - Encourage students to put some if statements inside other if statements.
- Now, students should run through all three programs using the cards that they drew, in the same order for each program.
 - Did any two programs return the same answer?
 - Did any return something different?

Standards Alignment

CSTA K-12 Computer Science Standards

► **AP** - Algorithms & Programming



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

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

Lesson 15: Harvesting with Conditionals

Conditional | Loop | Harvester

Overview

Students will practice while loops, until loops, and if / else statements. All of these blocks use conditionals. By practicing all three, students will learn to write complex and flexible code.

Purpose

Practicing the use of conditionals in different scenarios helps to develop a student's understanding of what conditionals can do. In the previous lesson, students only used conditionals to move around a maze. In this lesson, students will use conditionals to help the farmer know when to harvest crops. New patterns will emerge and students will use creativity and logical thinking to determine the conditions where code should be run and repeated.

Agenda

Warm Up (5 min)

Introduction

Main Activity (30 min)

Course D Online Puzzles - Website

Wrap Up (15 min)

Journaling

Objectives

Students will be able to:

- Nest conditionals to analyze multiple value conditions using if, else if, else logic.
- Pair a loop and conditional statement together.

Preparation

Play through **Course D Online Puzzles - Website** to find any potential problem areas for your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Course D Online Puzzles - Website**
- **CS Fundamentals Main Activity Tips - Lesson Recommendations**

For the Students

- **Think Spot Journal - Reflection Journal**

Vocabulary

- **Condition** - A statement that a program checks to see if it is true or false. If true, an action is taken. Otherwise, the action is ignored.
- **Conditionals** - Statements that only run under certain conditions.
- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again
- **While Loop** - A loop that continues to repeat while a condition is true.

Teaching Guide

Warm Up (5 min)

Introduction

Students shouldn't need as much of an introduction to concepts today because they have had practice with them in the previous lesson. Instead, you can share the story of the harvester.

The harvester is trying to pick crops like pumpkins, lettuce, and corn. However, the farmer has forgotten where she planted these crops, so she needs to check each plant before harvesting.

Main Activity (30 min)

Course D Online Puzzles - Website

Students will continue to work with if / else statements, while loops, and until loops. These puzzles are a bit more challenging, though, so encourage students to stick with them until they can describe what needs to happen for each program.

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 can you see conditionals being useful in programs?
- What if people only spoke in if/else statements? What would be some advantages and disadvantages of this?

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ AP - Algorithms & Programming



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

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

Lesson 16: Digital Citizenship

Common Sense Education | Unplugged

Overview

In collaboration with Common Sense Education, this lesson helps students learn to think critically about the user information that some websites request or require. Students learn the difference between private information and personal information, distinguishing what is safe and unsafe to share online.

Students will also explore what it means to be responsible and respectful to their offline and online communities as a step toward learning how to be good digital citizens.

Purpose

As students spend more time on computers, they should be aware that the internet is not always a safe space. In this lesson, students are taught what information is safe to share and what information should remain private. Students will create "superheros" and learn what it means to be a Digital Citizen on the internet.

Agenda

Warm Up (15 min)

Vocabulary

Personal vs. Private Online

Main Activity (35 - 40 min)

Cubecraft Superhero Templates - Manipulatives

Wrap Up (15 min)

Flash Chat: What did we learn?

Journaling

Assessment (5 min)

Digital Citizenship - Assessment

Extended Learning

Objectives

Students will be able to:

- Compare and contrast their responsibilities to their online and offline communities.
- Understand what type of information can put them at risk for identity theft and other scams.
- Reflect on the characteristics that make someone an upstanding citizen.
- Devise resolutions to digital dilemmas.

Preparation

Watch the **Digital Citizenship - Teacher Video**.

Print out a good selection of male and female **Cubecraft Superhero Templates - Manipulatives** sheets for the whole class.

Print one **Digital Citizenship - Assessment** for each student.

Review **CSF Digital Citizenship - Resource List** for more online safety content.

Links

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

For the Teacher

- **Digital Citizenship** - Teacher Video
- **Cubecraft Superhero Templates** - Manipulatives
- **Digital Citizenship** - Assessment
- **Digital Citizenship** - Assessment Answer Key
- **Common Sense Education** - Website
- **CSF Digital Citizenship** - Resource List

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

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

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has one new and important phrase:

Digital Citizen - Say it with me: Dih-jih-tal Sit-i-zen

Someone who acts safely, responsibly, and respectfully online

Personal vs. Private Online

- Ask "What types of information do you think are okay to share publicly online or on a profile that others will see?"
- What are some examples of websites where you must register in order to participate?
 - Write the names of the websites on the board.
- What information is required and why do you think it is required?
 - Information may be required to help distinguish one person from another.
 - The website may keep a record of who uses it.
- Explain that it's important to know that sharing some kinds of user information can put you and your family's privacy at risk.
- Point out that you do not have to fill out fields on websites if they are not required.
 - Required fields are usually marked by an asterisk (*) or are highlighted in red.
- Elementary school students should never register for sites that require private information without the approval and guidance of a parent or guardian.
- Here is an example of public versus private information:

💡 Lesson Tip

If you have access to a computer, feel free to navigate to a site that might require this type of information, such as Gmail or Facebook.

SAFE - Personal Information	UNSAFE - Private Information
Your favorite food Your opinion (though it should be done respectfully) First name (with permission)	Mother's maiden name Social Security number Your date of birth Parents' credit card information Phone number

- Explain that some people will actively try to get you to share this kind of information so that they can use it to take over your identity. Once a thief has taken someone's identity, he or she can use that person's name to get a driver's license or buy things, even if the person whose identity they stole isn't old enough to do these things!
 - It's often not until much later that people realize that their identity has been stolen. Identity thieves may also apply for credit cards in other people's names and run up big bills that they don't pay off. Let students know that identity thieves often target children and teens because they have a clean credit history and their parents are unlikely to be aware that someone is taking on their child's identity.

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

Main Activity (35 - 40 min)

Cubecraft Superhero Templates - Manipulatives

- Spiderman says "With great power comes great responsibility." This is also true when working or playing on the Internet.
- The things we read, see, and hear online can lead people to have all sorts of feelings (e.g., happy, hurt, excited, angry, curious).
 - What we do and say online can be powerful.
- The Internet allows us to learn about anything, talk to people at any time (no matter where they are in the world), and share our knowledge and creative projects with other people.
 - This also means that negative comments can spread very quickly to friends of all ages.
- CREATE a three-column chart with the terms "Safe," "Responsible," and "Respectful" written at the top of each column. Invite students to shout out words or phrases that describe how people can act safely, responsibly, and respectfully online, and then write them in the appropriate column.

Safe	Responsible	Respectful

Now, let's really make sure we understand how to be a Super Digital Citizen!

Directions:

- Have each student grab a small selection of papercraft sheets and encourage them to blend the pieces to make their very own superhero.
- Allow plenty of time for students to cut, glue, and color.
- Give students a 5 minute warning to wrap up.
- Separate students into groups of 2-4 and tell them to use their superheroes and leftover supplies to stage a scene in which one superhero sees an act of poor digital citizenship. Then have the superhero fix the problem ... and save the day!
- Go around the room, having each student explain their scene to the class.

💡 Lesson Tip

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

Wrap Up (15 min)

Flash Chat: What did we learn?

- What is a good way to act responsibly online?
- What kinds of personal information could you share about yourself without showing your identity?
- What kinds of superpowers or qualities did your digital superheroes have in common?
- What does Spider-Man's motto "With great power comes great responsibility" mean to you, as someone who uses the internet?

💡 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 Digital Citizen?
- What do you need to do to be a Digital Citizen?

Assessment (5 min)

Digital Citizenship - Assessment

- Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Extended Learning

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

Common Sense Education

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

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ **NI** - Networks & the Internet



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

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

Lesson 17: Ninjas vs. Pirates Game

Play Lab | Event

Overview

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

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 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 (10 min)

Introduction

Review of "Digital Citizenship"

Main Activity (30 min)

Course D Online Puzzles - Website

Wrap Up (15 min)

Journaling

Extended Learning

Objectives

Students will be able to:

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

Preparation

Play through **Course D Online Puzzles - Website** to find any potential problem areas for your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Course D Online Puzzles - Website**
- **CS Fundamentals Main Activity Tips - Lesson Recommendations**

For the Students

- **Think Spot Journal - Reflection Journal**

Vocabulary

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

Teaching Guide

Warm Up (10 min)

Introduction

Today students will be introduced to **events** in programming.

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

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 "when I say 'raise your hands' you will raise your hands".

You can also think of cities also declaring events. There are laws that say "when there is a green light, the cars can move through the intersection". Ask the students why they think this is an event.

Today, students will play in Play 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 pressing the up arrow or two characters running into each other.

Review of "Digital Citizenship"

Remind students of information that is safe to share online and information that is strictly private.

SAFE - Personal Information	UNSAFE - Private Information
Your favorite food Your opinion (though it should be done respectfully) First name (with permission)	Mother's maiden name Social Security number Your date of birth Parents' credit card information Phone number

Discuss other examples of the two categories above.

Main Activity (30 min)

Course D Online Puzzles - Website

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

Lesson Tip

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

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

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 is an event your program used today?
- Is there an event that would you like to have used in your game that you did not get to use in Play Lab?

Extended Learning

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

Look Under the Hood

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

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

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ **AP** - Algorithms & Programming



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

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

Lesson 18: Binary Images

Binary | Unplugged

Overview

Though many people think of binary as strictly zeros and ones, students will be introduced to the idea that information can be represented in a variety of binary options. This lesson takes that concept one step further as it illustrates how a computer can store even more complex information (such as images and colors) in binary, as well.

Purpose

In this lesson students will learn how information is represented in a way such that a computer can interpret and store it. When learning **binary**, students will have the opportunity to write code and share it with peers to view as images. This can then be related back to how computers read a program, translate it to binary, use the information in some way, then reply back in a way humans can understand. For example, when we type a sentence into a document then press "save", a computer translates the sentence into binary, stores the information, then posts a message indicating the document has been stored.

Agenda

Warm Up (10 min)

Vocabulary
Introduction to Binary

Main Activity (20 min)

Binary Images - Worksheet

Wrap Up (10 min)

Flash Chat: What did we learn?
Journaling

Assessment (10 min)

Binary Image - Assessment

Extended Learning

Objectives

Students will be able to:

- Identify methods for encoding images into binary.
- Relate images to a peer using binary encoding.
- Reproduce an image, based on binary code.

Preparation

Watch the **Binary Images - Teacher Video**.

Print one **Binary Images - Worksheet** per pair.

Print one **Binary Image - Assessment** per student.

Gather groupings of items that can show opposites for students to use when coming up with their own binary encodings (Optional).

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 Teacher

- **Binary Images** - Unplugged Video ([download](#))
- **Binary Images** - Teacher Video
- **Binary Images** - Worksheet
- **Binary Image** - Assessment
- **Binary Images** - Assessment Answer Key
- **Binary Baubles** - Thinkersmith Lesson

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Binary** - A way of representing information

using only two options.

- **Binary Alphabet** - The two options used in your binary code.

Teaching Guide

Warm Up (10 min)

Vocabulary

This lesson has two new terms:

- **Binary** - Say it with me: Bi-nare-ee

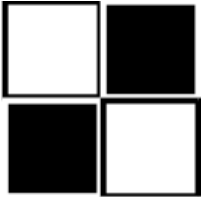
A way of representing information using only two options.

- **Binary Alphabet** - Say it with me: Bi-nare-ee Al-fa-bet

The two options used in your binary code.

Introduction to Binary

What if we had a picture like this, where there's only two color options for each square, black or white.



How might we encode this so that someone else could recreate the picture without seeing it?

- Some students might think back to the Graph Paper Programming lesson. While there could be a lot of similarities, let them know that this is different enough that they should not use that lesson to guide this one

You may hear suggestions like: "Say 'white, black, white, black'."

- "That's a great suggestion! Now I'm going to break you up into pairs. Work with your teammate to decide on a binary alphabet."

Decide whether you want your pairs to share their encodings with the other groups ahead of time, and tell them if they will be creating a key, or keeping their methods secret.

- "Now, let's encode some images, just like a computer would!"

Main Activity (20 min)

Binary Images - Worksheet

Now it's the students' turn!

Activity Directions:

1. Divide students into pairs.
2. Have them choose an image with their partner.
3. Encourage them to figure out what their binary alphabet is going to be.
4. Have them encode their image using their new binary alphabet.
5. Instruct students to trade encodings with another team and see if they can figure out which picture the other worked on.
6. Choose a Level
 - Easy: Let the other team know what your encoding method was
 - Tough: Have the other team guess your encoding method.

Wrap Up (10 min)

Flash Chat: What did we learn?

- What did we learn today?
- What kind of binary alphabet did you create?
- Can you think of how you could encode an image using only your fingers?
- Do you think you could create a binary alphabet out of sounds?

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 binary alphabet?
- What kind of information can you share using binary?

Assessment (10 min)

Binary Image - Assessment

Pass out this assessment for students to do individually. Try to save time at the end to go over answers.

Extended Learning

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

Storing Color Images

- If your class really gets the idea behind storing binary images, they may want to know how to do color images.
 - First, you'll need to discuss how color works using binary (as in **Binary Baubles - Thinkersmith Lesson**, page 21).
 - Then, introduce some images that use combinations of those colors
- Encourage your students to come up with ways to code these color images.



Hexadecimal

- Take the idea of color one step further to introduce **hexadecimal color codes**.

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ AP - Algorithms & Programming



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

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

Lesson 19: Binary Images with Artist

Binary | Artist

Overview

This series of online lessons will have students learning to make images using on and off.

Purpose

This will help reinforce the fact that computers can do a multitude of things with 0s and 1s.

Agenda

Warm Up (15)

Main Activity (30)

Course D Online Puzzles - Website

Wrap Up (15)

Journaling

Objectives

Students will be able to:

- Create pictures using unique combinations of on and off
- Identify repeated sequences and break long codes up into smaller chunks that can be looped
- Utilize loops and binary code to recreate provided images

Preparation

Play through the **Course D Online Puzzles - Website** corresponding to this course to find any potential problem areas for your class.

Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Course D Online Puzzles - Website**

Vocabulary

- **Binary** - A way of representing information using only two options.

Teaching Guide

Warm Up (15)

To begin, it can be helpful to review the previous lesson, specifically different ways of using binary to indicate how to create an image on a grid. This stage will translate the unplugged activity into a simple, independent online lesson.

Main Activity (30)

Course D Online Puzzles - Website

Not all of these images are intuitive. Encourage students to click "Run" to see what happens, even if their code isn't "finished" yet.

Wrap Up (15)

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?
- Did you like drawing on the 8x8 grid or the 16x16 grid better? Why?
- Computers also store sounds using binary. Use your imagination to write down a possible way for that to work.

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ AP - Algorithms & Programming



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

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