Course B

Course B was developed with first graders in mind. Tailored to a novice reading level, this course also assumes limited knowledge of shapes and numbers.

At the moment, Course B closely parallels Course A, but provides more complex unplugged activities and more variety in puzzles. Students will learn the basics of programming, collaboration techniques, investigation and critical thinking skills, persistence in the face of difficulty, and internet safety. At the end of this course students will create their very own custom game from Play Lab that they can share with a link.

Persistence

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

Picture Book: Unspotted Bugs Slides | **Video**

Picture Book: Stevie and the Big Project Slides | **Video**

Optional Activity: Marble Run Teacher Guide

Journaling

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

Think Spot Journal Student Handout

Debugging

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

Debugging Guide Student Handout

Chapter 1: Digital Citizenship

Lesson 1: Your Digital Footprint

Unplugged | Online Safety

Learn about your digital footprint and how to stay safe when visiting websites.

Chapter Commentary

Chapter 2: Sequencing

Lesson 2: Move It, Move It

Unplugged | Sequencing

Program your classmates to step carefully from place to place.

Lesson 3: Sequencing with Angry Birds

Skill Building | Sequencing

Help Red the Angry Bird follow the path to the naughty pig.

Lesson 4: Programming with Angry Birds

Skill Building | Sequencing

Create programs to help the Angry Bird move through the maze.

Lesson 5: Programming with Harvester

Skill Building | Sequencing

Help the Harvester collect vegetables along a path.

Chapter Commentary

Chapter 3: Loops

Lesson 6: Getting Loopy

Unplugged | Loops

In this lesson, we'll have a dance party using repeat loops!

Lesson 7: Loops with Harvester

Skill Building | Loops

Help the Harvester collect even more, using loops!

Lesson 8: Loops with Laurel

Skill Building | Loops

Program Laurel the adventurer to collect treasure in an open cave.

Lesson 9: Drawing Gardens with Loops

Skill Building | Loops

Use patterns and loops to finish the images.

Chapter Commentary

Loops

Chapter 4: Impacts of Computing

Lesson 10: The Right App

Unplugged | Impacts of Computing

Sketch your own smartphone app.

Chapter Commentary

Impacts of Computing

Chapter 5: Events

Lesson 11: The Big Event Jr.

Unplugged | Events

Move or shout when your teacher presses buttons on a giant remote.

Lesson 12: A Royal Battle with Events

Application | Events

Use events to create a story or make an interactive game.

Chapter Commentary

Events

Lesson 1: Your Digital Footprint

Overview

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

Purpose

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

Agenda

- **Warm Up (20 min)**
  - **Vocabulary**
- **Main Activity (20 min)**
  - **Follow the Digital Trail**
- **Wrap Up (15 min)**
  - **Reflection**
- **Assessment (5 min)**
- **Extended Learning**

View on Code Studio

Objectives

Students will be able to:

- Understand that being safe when they visit websites is similar to staying safe in real life.
- Learn to recognize websites that are safe for them to visit.
- Recognize if they should ask an adult they trust before they visit a particular website.
- Explore what information is appropriate to be put online.

Preparation

- ☐ Print at least one copy of **Your Digital Footprint - Digital Trail Squares** and **Your Digital Footprint - Worksheet** per group of three or four.
- ☐ Print one copy of **Your Digital Footprint - Assessment** per student.
- ☐ Prepare to show **Your Digital Footprint - Lesson Video** to students.

Links

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

For the Teachers

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

For the Students

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

Make a Copy

Vocabulary

- **Digital Footprint** - The collected information about an individual across multiple websites on the Internet.

Teaching Guide

Warm Up (20 min)

Vocabulary

Say: This lesson has one new and important phrase:

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

"The information about someone on the internet."

Prompt: Engage students on the topic of internet safety. **What does it mean to be safe?** When you walk down the street or play in your neighborhood without a trusted adult there, how do you stay safe?

Say: Tell students that just as they should stay safe in the real world, they should stay safe when they go into the online world (visiting websites). Make parallels between the answers students gave you about their neighborhood and the online world.

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

Discuss: Introduce the idea that there are three different kinds of websites that students may have the opportunity to visit.

Lesson Tip

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

- Green: A "green" website...
  - ...is good for kids your age to visit.
  - ...fun, with things for you to do and see.
  - ...has appropriate words.
  - ...doesn't let you talk to people you don't know.
- Yellow: A "yellow" website...
  - ...is a site you are not sure is right for you.
  - ...asks for information such as who you are, where you live, your phone number, email address, etc.
  - ...a site where you are allowed to communicate freely with others
- Red: A "red" website...
  - ...is not right for kids your age to visit.
  - ...is a place you might have gone to by accident.
  - ...is filled with things that are for older kids or adults.

Where appropriate, discuss examples of each kind of website.

Transition: Tell students that they will now learn what they can do to keep themselves safe.

Main Activity (20 min)

common sense education®

For more in-depth modules, you can find additions to this curriculum at the Common Sense Media webpage on Scope and Sequence.

Follow the Digital Trail

Display: Place the **Your Digital Footprint - Digital Trail Squares** on the ground, face down, in two different trails, keeping Mizzle the Mouse and Electra the Elephant's trails separate from one another.

Say: Share the stories of Mizzle and Electra.

- These animals decided it would be fun to put some information about themselves online.
- They went onto **www.wildkingdom.com** and posted information.
- The only problem is that they forgot to ask their parents if it was okay first!
- You are from the "Things Big and Small" Detective Agency. A hunter has hired you to find out as much as possible about Mizzle the Mouse and Electra the Elephant. The more you learn, the better the agency's plan to take over the animal kingdom!

Lesson Tip

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

Group: Divide students into groups of three or four four. Tell them that each group should have a detective that will keep detailed notes.

Distribute: Pass out one copy of **Your Digital Footprint - Worksheet** to each group. Optionally, each student can have their own worksheet to take their own notes.

Activity: Invite students to go on a hunt for information. Let them know that the information that Mizzle and Electra post can be seen by anyone, including the detectives. Each group should follow the digital trail of both animals, starting with the mouse and then the elephant. Stagger the groups so they are on the trail at slightly different times. Students should fill out their worksheet as they go.

Wrap Up (15 min)

Reflection

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

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

Lesson Tip

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

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

SAFE	UNSAFE
Interests	Address
Hobbies	Full Name
First Name	Information that would hurt others

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

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw some things that you should never talk to a stranger about on the internet. For example, draw your house to represent your address, draw your school, or draw your family.

Assessment (5 min)

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

Extended Learning

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

Common Sense Education

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

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▸ **IC** - Impacts of Computing
- ▸ **NI** - Networks & the Internet

Lesson 2: Move It, Move It

Overview

This lesson will work to prepare students mentally for the coding exercises that they will encounter over the length of this course. In small teams, students will use physical activity to program their classmates to step carefully from place to place until a goal is achieved.

Purpose

By using physical movement to program their classmates, students will run into issues and emotions similar to what they will feel when they begin coding on a computer. Encountering those stresses in a playful and open environment will help to alleviate intensity and allow students to practice necessary skills before they run into problems on their own.

Agenda

View on Code Studio

Objectives

Students will be able to:

- Define a list of steps (algorithm) to get a friend from their starting position to their goal
- Translate a list of steps into a series of physical actions
- Identify and fix errors in the execution of an algorithm

Preparation

- ☐ Watch the **Move It, Move It - Teacher Video**
- ☐ Print (or otherwise prepare to display) one **Move It, Move It - Worksheet Answer Key**
- ☐ Print one **Move It, Move It - Map Activity** and one **Move It, Move It - Worksheet** per group of 2-3 students
- ☐ Prepare blank papers to fill out the rest of the walking grid (4-7 needed per group)

Links

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

For the Teachers

- **Move It, Move It** - Worksheet Answer Key
  Make a Copy
- **Move it, Move it** - Teacher Video

For the Students

- **Move It, Move It** - Map Activity
  Make a Copy
- **Move It, Move It** - Worksheet
  Make a Copy

Vocabulary

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

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

Teaching Guide

Warm Up (20 min)

Where did I go wrong?

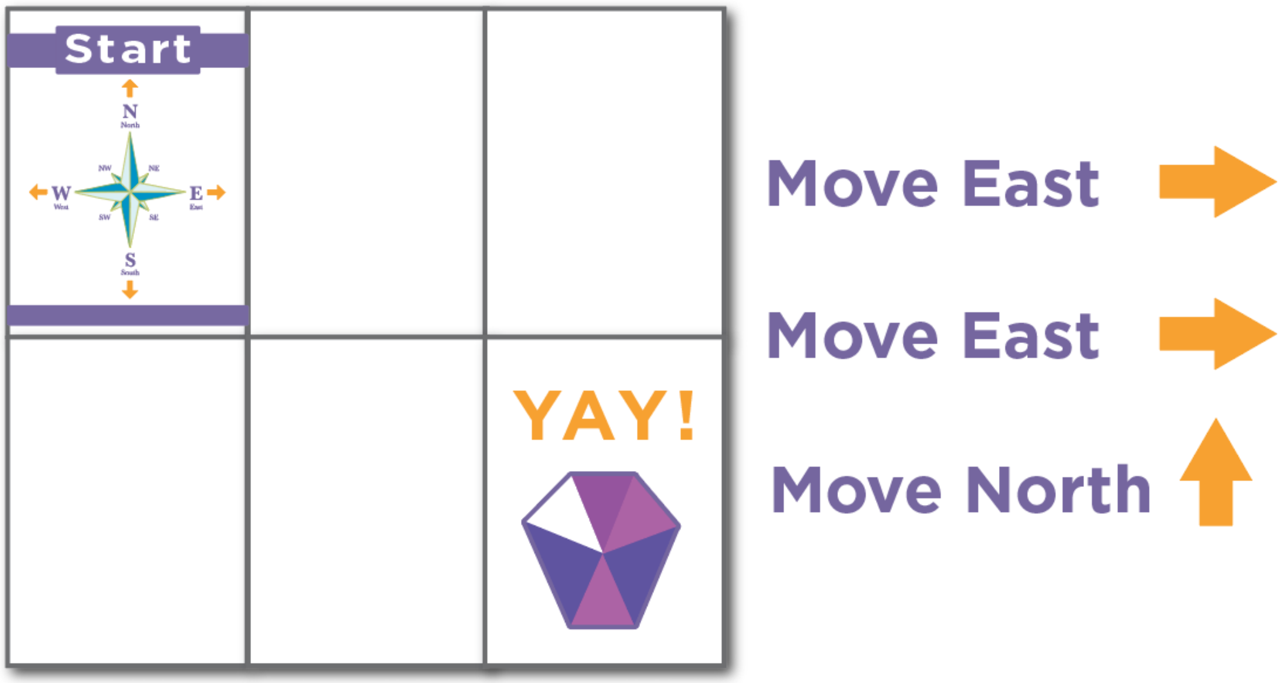
Goal: In this lesson, we want to help students learn to identify and fix bugs in their own programs. The easiest way to do that is to first present students with a program that contains bugs that are not their fault. Once they've helped you fix "your" program, share with them how frustrating it can be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

Display: Put an image of **Move It, Move It - Worksheet Answer Key** up on the screen where everyone can see it.

Teaching Tip

If your class has not already learned cardinal directions, it will be worth covering them before they begin Course B. This conversion will come in handy for nearly all of the online puzzles aimed at first grade, as well as several of the unplugged activities.

Let students know that they will continue to see those directions in the online programs next to the direction arrows.



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 compass rose
- Follow the instructions step-by-step
- End at the treasure

Optional: Walk through your program using your fingers on top of the map, under the document camera. Express frustration when your fingers end up off the map, instead of at the treasure.

Think: My program says "East, East, North". 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 **Move It, Move It - Worksheet** 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!

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

Activity: Move It, Move It (20 minutes)

Distribute: Hand each group of 2-3 students a **Move It, Move It - Map Activity**, as well as the blank papers for the grid on the ground. Allow students to either cut the halves of each map apart, or fold the sheets in half so that each map is clearly visible (without distraction.)

Set-Up: In each group, each player will get a task.

- Player 1: Choose/set-up the map to play
- Player 2: Programmer
- Player 3: Walking Machine

Directions for Class:

1) Decide who will take each job.
2) Have player 1 set a grid on the floor made up of pieces of paper (as shown on one of the Move It Maps) except with the gem paper facing the ground.
3) Player 3 will start by standing on the page with the compass rose.
4) Player 2 will then guide player 3 step-by-step through the paper maze using the provided arm signals.
5) When player 2 gives the signal to “STOP”, player 3 will flip over the page that they are on. If that page is a gem, then the maze was a success!
6) If there is time, let everyone rotate positions and go again!

Note that the rules are not the most important thing here. Feel free to clarify if the students have questions, but if the students are playing a bit differently than described, you don't need to hold them to the letter of the game. The crucial bit is that they are moving from immediate instructions to giving two or three instructions before the Walking Machine moves.

Wrap-up (10 min)

Journaling

Give the students a journal prompt to help them process some of the things that they encountered during the day. You can choose one of the prompts below, or make up your own.

Journal Prompts:

- Draw a feeling face in the corner of your journal page
- What were the four directions on the compass rose?
  - What tricks can we use to remember North, South, East and West?
- Draw another way we could we have given instructions without using our arms
- Draw your favorite part about that game

Extended Learning

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

X's and O's

- Draw a tic-tac-toe board for the class.
- Place a single X and a single O somewhere on the board.
- Ask the class if they can get the X to the O using arm gestures as a class.

X's, O's, and Arrows

- Similar to the activity above, but have the students write their programs in advance using arrows instead of hand gestures.
  - This can be done in groups.
  - Groups can share their solutions for the class.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 3: Sequencing with Angry Birds

Overview

Using characters from Angry Birds, students will develop sequential algorithms to move a bird from one side of a maze to the pig at the other side. To do this they will stack code blocks together in a linear sequence.

Purpose

In this lesson, students will develop programming and debugging skills on a computer platform. The block-based format of these puzzles help students learn about sequence and concepts, without having to worry about perfecting syntax.

Agenda

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

Bridging Activity - Drag and Drop (10 - 15 min)
- **1) Unplugged Activity Using Paper Blocks**
- **2) Online Activity Using Unplugged Arrows**

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

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

Extension Activities

View on Code Studio

Objectives

Students will be able to:

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

Preparation

- ☐ If your students are brand new to dragging and dropping, consider assigning them **Drag and Drop Practice** before starting this lesson.
- ☐ Watch the **How to Make a Class Section on Code.org - Teacher Video**. Create your own class section on Code.org and make sure every student has a card with their **passcode** on it.
- ☐ Have the school IT person add a quick link for your class section to the computer desktop.
- ☐ Make sure each student has a journal.

Links

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

For the Teachers

- **How to Make a Class Section on Code.org** - Teacher Video

For the Students

- **Drag and Drop Practice**
- **Unplugged Blockly Blocks (Grades K-1)** - Manipulatives
- **Move It, Move It** - Map Activity
  Make a Copy ⌄
- **Pair Programming** - Student Video

Vocabulary

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

Teaching Guide

Warm Up (10 min)

Behaving in the Computer Lab

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

Discuss:

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

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

Some possible things to cover:

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

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

To connect the unplugged lesson with the upcoming online lesson, choose **one** of the following activities to do with your class.

1) Unplugged Activity Using Paper Blocks

Model: Select a map from **Move It, Move It - Map Activity** from the Move It, Move It unplugged lesson. Using movement pieces from the **Unplugged Blockly Blocks (Grades K-1) - Manipulatives**, show students how you would code this structure in this new way.

Pair/Think: Next, choose another map and have the students program what blocks a "robot" would need to read to get to the goal.

Make sure that they understand that the blocks need to go from top to bottom and they all need to touch!

Share: Have the students check each other's answers and resolve any questions or bugs that may come up.

2) Online Activity Using Unplugged Arrows

Model: Pull a puzzle from the corresponding online levels. Show students how to get the Angry Bird to the pig using the symbols. It can be helpful to rename the arrows "North", "South, "East", and "West". Once you have a program, trace it with your finger (or a pointer) and show how the bird will travel when the program is run.

Pair/Think: Next, move back to an easier puzzle, and have students try writing programs (using arrows) on their own.

Share: Encourage students to share their programs with other groups and see if they came up with solutions that are the same or different. Can anyone come up with another way of solving the puzzle?

Main Activity (20 - 30 min)

Online Puzzles

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

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

Teacher Tip

Show the students the right way to help classmates:

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

Code Studio levels

Maze Intro: Programming with Blocks

1 *(click tabs to see student view)*

Practice

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

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

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

Wrap Up (5 - 10 min)

Journaling

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

Journal prompts could include:

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

Extension Activities

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

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

Lesson 4: Programming with Angry Birds

Overview

Using characters from the game Angry Birds, students will develop sequential algorithms to move a bird from one side of a maze to the pig at the other side. To do this they will stack code blocks together in a linear sequence.

Purpose

In this lesson, students will develop programming skills on a computer platform. The block-based format of these puzzles help students learn about sequence and concepts, without having to worry about perfecting syntax.

Agenda

View on Code Studio

Objectives

Students will be able to:

- Translate movements into a series of commands.

Preparation

☐ Play through the puzzles to find any potential problem areas for your class
☐ Locate or reprint supplies for Happy Maps
☐ Make sure each student has a **Think Spot Journal - Reflection Journal**
☐ Cut enough **Unplugged Blockly Blocks (Grades K-1) - Manipulatives** of each direction (N, S, E, W) to give two sets to every pair of students (if you choose Bridging Option #1)

Links

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

For the Students

- **Happy Map Game Pieces** - Manipulatives
  Make a Copy ▾
- **Pair Programming** - Student Video
- **Move It, Move It** - Map Activity
  Make a Copy ▾
- **Compass Rose** - Handout **Make a Copy ▾**
- **Unplugged Blockly Blocks (Grades K-1)** - Manipulatives

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.
- **Programming** - The art of creating a program.

Teaching Guide

Warm Up (5 min)

Review Unplugged Activity

This lesson relies on many of the unplugged ideas that students have learned in the weeks leading up to this first online activity. It is important that you bring those concepts (such as persistence, algorithms, and programs) around full-circle so that your class can benefit from them in their online work as well.

Display: Show students a picture from the "Move It, Move It" exercise that you completed in the lessons prior to this one.

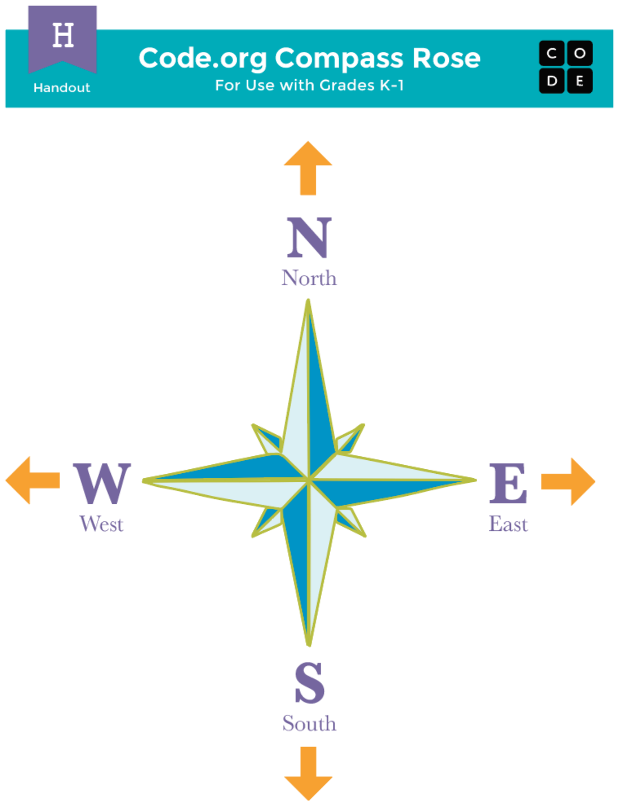
Discuss: Ask students to recall the symbols used in "Move It, Move It."

- What would you do when when you saw the "North" arrow?
- How about the "East" arrow?

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

Teaching Tip

If your class has already learned cardinal directions, then changing "Up" and "Down" to "North" and "South" shouldn't be a problem. If they have not, we have provided a handy worksheet with the Code.org Compass Rose that you can use to get students onboard. This conversion will come in handy for nearly all of the online puzzles aimed at kindergarten and first grade.



Let students know that they will see those letters in their online programs next to the direction arrows.

Bridging Activity - Choose One (10 min)

This activity will help bring the unplugged concepts from Move It, Move It 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

Distribute: Give students **Unplugged Blockly Blocks (Grades K-1) - Manipulatives** with cardinal commands like E → (East) and W ← (West).

Display: Choose an empty map from the **Move It, Move It - Map Activity** and display it for the class to see.

Model: Beginning at the start location, use your finger to show students what each block does. Show them how the E → corresponds to the right arrow and moves the the "robot" one step to the right. Do the same for each of the other three.

Think: By this point, the class should know how to get goal using arrows. How would you get to the fruit using Blockly blocks?

Pair: Now, have the students program from their desks using the paper Blockly blocks. Make sure that they understand that the blocks need to go from top to bottom and they all need to be connected!

Share: Have pairs discuss their answers with elbow partners. Did everybody get the same thing?

Online Puzzles using Arrows

Display: Show students a the playspace from one of the puzzles corresponding to this lesson. We recommend puzzle 5.

Think: Ask students to imagine that this puzzle is just like the maps, but instead, it's a bird trying to get to the pig. How can they write a program to get the bird to the pig using arrows?

Pair: Using just the symbols from the **Happy Map Game Pieces - Manipulatives** , have students lay out a pattern that they think will get the bird to the pig.

Share: Ask the students to share their answers with the class. Did anyone else have the same answer?

Previewing Online Puzzles as a Class (3 min)

Students should now be ready to see a real puzzle in action!

Model: Pull up Puzzle 5 to do in front of the class. This will be the same puzzle that they just saw in the bridging activity. While working through this puzzle with the class, remind students that making mistakes is okay and remind them that the only way to be successful is to be persistent. Tie the the issues into ideas that they've seen in previous lessons, such as what to do when a program doesn't work (debug it!) or how to get through the frustration that can come with working on a computer.

Next, you'll need to describe how the blocks in the workspace move the bird toward the pig. Show students how to drag blocks from the toolbox and connect them beneath the `when run` block, but don't solve the puzzle.

Discuss: Think about how we would get the bird to the pig using arrows. How do we use these blocks instead?

Have students use their fingers to point the direction that the bird should go next. Once you feel like you have a classroom consensus, try to get students to put into words which block will make that action happen. Roll your mouse over different options and have them shout "Yes" or "No".

Drag blocks into place one at a time, then click "Run" after each one. This will not only let them see how far the bird has gone, but set good habits for when they start working to solve their own puzzles.

Continue this pattern, fixing bugs as they arise, until the bird successfully gets to the pig.

Transition: Now that students have seen an online puzzle in practice, they should be ready to start solving puzzles of their own. Continue to the lab or bring out their classroom machines.

Main Activity (30 min)

Online Puzzles

Additional Demonstration

We've included some multiple choice prediction levels that are difficult for non-readers. Alternatively, these could be used after finishing the stage as a review for the class.

Prediction Levels:

- **Programming in Maze #1**
- **Programming in Maze #2**

Code Studio levels

Pair Programming 1 ***(click tabs to see student view)***

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

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

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

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

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

- 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.
- 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 by:

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

Discuss: After providing students with end-of-class warnings, grab everyone's attention and get them to reflect on the experiences that they just had.

- Did anyone feel frustrated during any of the puzzles?
- Did anyone notice the need to be persistent?

Transition: Have students grab their Thinkspot Journals and take a moment to leave lessons for themselves.

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 something that you shouldn't do while helping a friend with a Code.org puzzle

Extended Learning

In small groups, let students design their own mazes on paper and challenge other students or groups to write programs to solve them. For added fun, make life-size mazes with students as the pig and bird.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming

Lesson 5: Programming with Harvester

Overview

Students will apply the programming concepts that they have learned to the Harvester environment. Now, instead of just getting the character to a goal, students have to collect corn using a new block. Students will continue to develop sequential algorithm skills and start using the debugging process.

Purpose

In this lesson, students will develop debugging skills and will continue developing their programming skills.

Agenda

Warm Up (5 min)

Vocabulary
Debugging, Persistence, and Frustration

Main Activity (30 min)

Online Puzzles

Wrap Up (5 - 10 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Translate movements into a series of commands.
- Identify and locate bugs in a program.

Preparation

☐ Play through the puzzles to find any potential problem areas for your class.
☐ Make sure each student has a **Think Spot Journal - Reflection Journal**.
☐ (Optional) Read **Stevie and the Big Project** and **Unspotted Bugs** storybooks with your class beforehand.

Vocabulary

- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Persistence** - Trying again and again, even when something is very hard.

Teaching Guide

Warm Up (5 min)

Vocabulary

This lesson has three new and important vocabulary words:

- **Bug** - Say it with me: Buhh-g.

Something that is going wrong. An error.

- **Debugging** - Say it with me: Dee-bug-ing.

To find and fix errors.

- **Persistence** - Say it with me: Purr-siss-tense.

Not giving up. Persistence works best when you try things many different ways, many different times.

Debugging, Persistence, and Frustration

Discuss: Prepare students for today's online exercises by asking them what they would do if they thought they had a bug in their code. More specifically, what would they ask themselves?

- Was everything in the first step right?
- How about the second? Third?
- Where did my program go wrong?
- What does that tell me?

Prompt: Debugging can be frustrating! But persistence can help us get through it. What are some things we can do to persist through frustration?

Example responses:

- Count to 10
- Take deep breaths
- Journal about bugs we've found
- Talk to a partner about bugs
- Ask for help

Transition: Let your students know that frustration is part of coding. Everyone gets bugs! It's okay if they think they have a bug but they can't find or fix it right away. Using strategies like those listed above can help them persist through frustration and solve any problem!

Main Activity (30 min)

Online Puzzles

At this point, students should already be familiar with the programming environment. Some new things to look out for in this lesson are confusion about the debugging process or not remembering to use the `pick corn` block when the harvester reaches corn.

Code Studio levels

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

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

Debugging with the Step Button 5 *(click tabs to see student view)*

Practice 6 7 8 9 10 11 *(click tabs to see student view)*

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

Practice 13 14 15 *(click tabs to see student view)*

Circulate: During online activities, the role of the teacher is primarily one of encouragement and support. In addition to the ideas listed in the last lesson, some more ideas on how to do this are:

- 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 by:

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

Transition: Have students grab their Thinkspot Journals and take a moment to leave lessons for themselves.

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 a time you found a bug in your code.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 6: Getting Loopy

Overview

As we start to write longer and more interesting programs, our code often contains a lot of repetition. In this lesson, students will learn about how loops can be used to more easily communicate instructions that have a lot of repetition by looking at the repeated patterns of movement in a dance.

Purpose

At this point in the course, students should have developed comfort with programming a set of linear instructions. Frequently the linear set of instructions includes patterns that are repeated multiple times and as students want to write more complex and interesting programs, manually duplicating that code becomes cumbersome and inefficient. To enable students to write more powerful programs, we'll need to rely on structures that break out of the that single linear list. **Loops** allow for students to structure their code in a way that repeats. In this lesson, we will focus on identifying patterns in physical movement before moving back onto the computer to look for patterns in our code.

Agenda

Warm Up (5 min)

- **Repeat After Me**

Main Activity (15 min)

- **Dance Party**

Assessment (10 min)

Wrap-Up (15 min)

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

Extended Learning

- **So Moving**
- **Connect It Back**

View on Code Studio

Objectives

Students will be able to:

- Repeat actions initiated by the instructor.
- Translate a picture program into a real-world dance.
- Convert a series of multiple actions into a single loop.

Preparation

- ☐ Prepare to display the **Getting Loopy - Worksheet** to students.
- ☐ Print one **Getting Loopy - Assessment** per student.
- ☐ Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teachers

- **Getting Loopy** - Assessment Answer Key
  Make a Copy

For the Students

- **Getting Loopy** - Unplugged Video (**download**)
- **Getting Loopy** - Worksheet
  Make a Copy
- **Getting Loopy** - Assessment
  Make a Copy

Vocabulary

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

Teaching Guide

Warm Up (5 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?

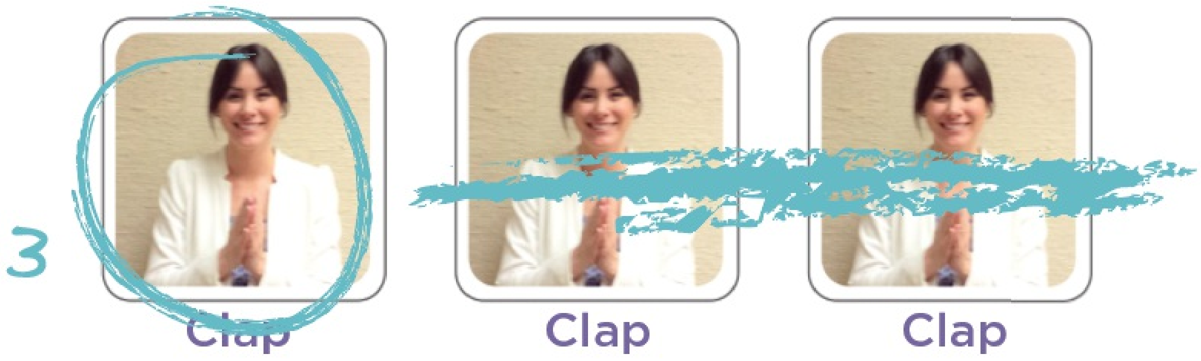
Think: What if I wanted you to do it ten times? How would you reword my instructions so that they were more efficient and I didn't have to repeat myself so much? Feel free to write your instructions down on a piece of scrap paper.

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.

Remarks

> Today we're going to work on finding ways to make giving lots of instructions easier, especially when those instructions repeat themselves a lot. This will be really useful when we go back to the computers and have to write lots of instructions in our programs.

Main Activity (15 min)



Dance Party

Say: Introduce the main activity by letting the class know that we will be having a dance party. In order to have that party, we'll need to know what all of the steps in the dance are, and how many times we should do them.

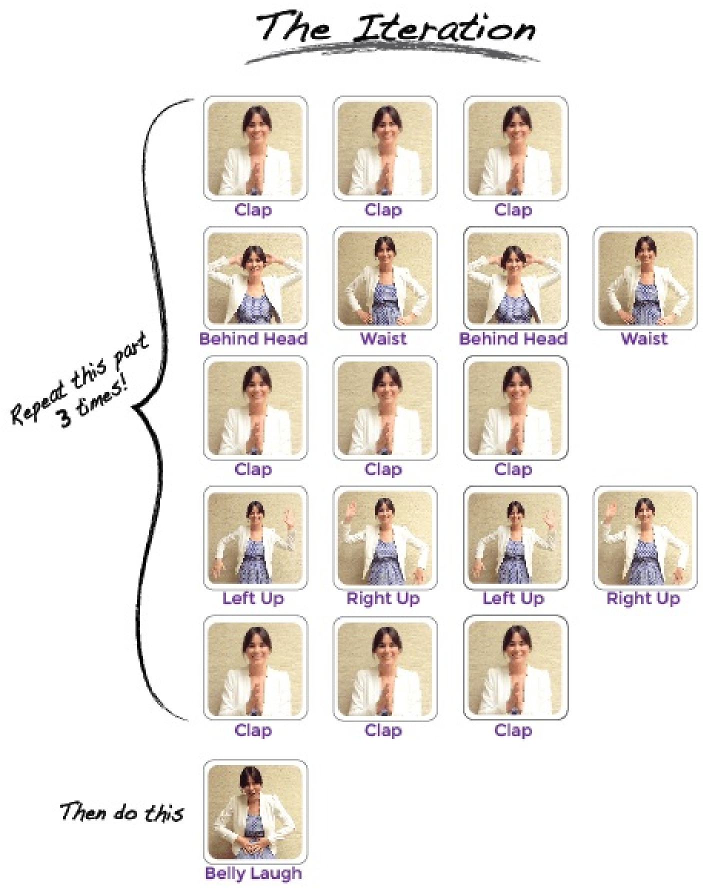
Display: Show the **Getting Loopy - Worksheet** so that all students can see it. Talk through the different sections of the dance as a class. Point out the section that repeats, in particular.

Lesson Tip:

Looking for some good music? Here are some great places to find some:

- **Radio Disney**
- **Nick Radio**
- **Kidz Bop Radio**

Please be advised that some of these stations may display ads with third-party content. If you find that displayed ads are inappropriate, you may want to direct students to a different site, or research ad-blockers that can prevent this content.

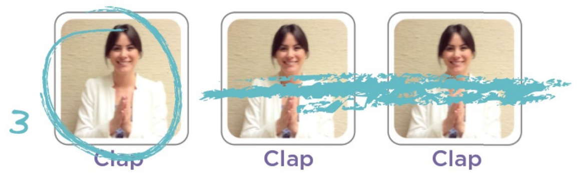


Model: Show the class what the entire dance looks like done at full-speed. Then run through the dance slowly, asking a different student to call out each line of instructions. Next, have the students perform the dance along with you, saying the instructions aloud as they get to each move.

Prompt: Ask students to work with a neighbor to find all of the sections of the dance that repeat.

Share: Ask a few students to share the repeating patterns that they found. As a class, talk through how you might rework the instructions to be even shorter by repeating those patterns.

Finally, help them understand a symbology for capturing these loops on their picture program, since the assessment will utilize this same method. Here is an example:



Assessment (10 min)

Ending with an assessment sheet will help solidify this lesson for your students.

Distribute: Hand out the **Getting Loopy - Assessment** to each student. Allow students to complete the activity independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities.

Wrap-Up (15 min)

Vocabulary

Display: Present the vocab for this lesson, loop. Ask the class to point out the main loop that was in the dance. Why do you think we call it a loop?

Flash Chat: What did we learn?

- Do you think it is easier to add more pictures to the screen or change the number of times we loop?
  - Would your answer be the same if we wanted to loop 100 times?
- Could we use these same loops with different dance moves?
- Do you know any dances that are done inside a loop?
- What was your favorite part about that activity?

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Draw a picture of you dancing today. Draw the loops that you did, like clapping three times.
- What else can you use a loop for?

Extended Learning

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

So Moving

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

Connect It Back

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

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 7: Loops with Harvester

Overview

Building on the concept of repeating instructions from "Getting Loopy," this stage will have students using loops to pick corn more efficiently on Code.org.

Purpose

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

Agenda

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

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

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

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

Extended Learning

View on Code Studio

Objectives

Students will be able to:

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

Preparation

- ☐ Play through the puzzles to find any potential problem areas for your class.
- ☐ Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.
- ☐ Make sure each student has a **Think Spot Journal - Reflection Journal**.
- ☐ (Optional) Pick a couple of puzzles to do as a group with your class.
- ☐ (Optional) If you haven't already, print and cut out blocks from **Unplugged Blockly Blocks (Grades K-1) - Manipulatives** to prepare for one of the bridging activities in this lesson.

Links

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

For the Teachers

- **CS Fundamentals Main Activity Tips** - Lesson Recommendations **Make a Copy**

For the Students

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

Vocabulary

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

Teaching Guide

Warm Up (10 min)

Introduction

Review the Getting Loopy activity with your students:

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

Bridging Activity - Loops (10 min)

This activity will help bring the unplugged concepts from "Getting Loopy" 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

Revisit the dance from "Getting Loopy." This time, work with the class to "code" it out using **Unplugged Blockly Blocks (Grades K-1) - Manipulatives** instead of writing the dance instructions on paper. Make sure the students know that the blocks need to go from top to bottom and they all need to touch!

Previewing Online Puzzles as a Class

Pull up the online puzzles and choose a puzzle to do in front of the class. We recommend puzzle 7. Ask the students to write a program to solve the puzzle on paper. Have the students circle repeated chunks and label with the number of repeats, the same way they did in "Getting Loopy."

Main Activity (30 min)

Online Puzzles

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

Code Studio levels

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

Repeat Blocks 3 ***(click tabs to see student view)***

Practice 4 5 6 7 8 9 10 ***(click tabs to see student view)***

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

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

Wrap Up (5 - 10 min)

Journaling

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

Journal Prompts:

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

Extended Learning

So Moving

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

Connect It Back

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

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 8: Loops with Laurel

Overview

In this lesson, students continue learning the concept of loops. Here, Laurel the Adventurer uses loops to collect treasure in open cave spaces. A new `get treasure` block is introduced to help her on her journey.

Purpose

This lesson gives students more practice with loops and encourages them to put multiple blocks inside of a `repeat` as they try to collect as much treasure as possible.

Agenda

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

Online Foundation: Preview Loops in Collector
Main Activity (30 min)
- **Online Puzzles**

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

View on Code Studio

Objectives

Students will be able to:

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

Preparation

☐ Play through the puzzles to find any potential problem areas for your class.
☐ (Optional) Pick a couple of puzzles to do as a group with your class.
☐ Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**
☐ Make sure each student has a **Think Spot Journal - Reflection Journal**.

Vocabulary

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

Teaching Guide

Warm Up (10 min)

Introduction

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

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

Online Foundation: Preview Loops in Collector

To introduce Laurel the Collector, preview an online puzzle (or two) as a class.

Model: Reveal an entire online puzzle from the progression to come. We recommend Puzzle 8. Do students see any similarities to the last set of exercises that they did? What are the big differences? When should the `get treasure` block be used?

Work with your class to drag code into the workspace in such a way that Laurel (eventually) collects all of the treasure.

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

Main Activity (30 min)

Online Puzzles

Teacher Demonstration

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

Prediction Levels:

- **Course B, Loops in Collector**

🖥 Code Studio levels

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

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

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

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

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

Practice

11 | 12 | 13 *(click tabs to see student view)*

Levels

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

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

Wrap Up (5 - 10 min)

Journaling

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

Journal Prompts:

- What was today’s lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw a line of treasure that Laurel could collect.
- Draw something that uses loops.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 9: Drawing Gardens with Loops

Overview

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

Purpose

This lesson gives a different perspective on how loops can create things in programming. Students will test their critical thinking skills by evaluating given code and determining what needs to be added in order to solve the puzzle. Students can also reflect on the inefficiency of programming without loops here because of how many blocks the program would require without the help of `repeat` loops.

Agenda

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

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

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

View on Code Studio

Objectives

Students will be able to:

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

Preparation

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

Vocabulary

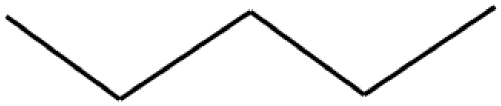
- **Loop** - The action of doing something over and over again.

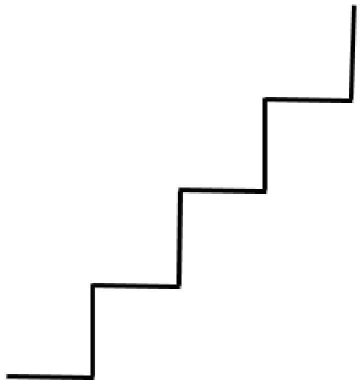
Teaching Guide

Warm Up (10 min)

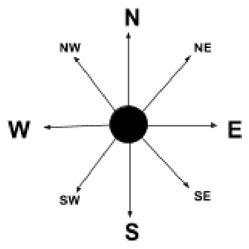
Introduction

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





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



Main Activity (30 min)

Online Puzzles

Teacher Demonstration

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

Teacher Tip

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

Prediction Levels:

- **Course B, Loops in Artist**

Code Studio levels

The Artist in Code Studio 1 ***(click tabs to see student view)***

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

Repeat Blocks with the Artist

3 *(click tabs to see student view)*

Practice

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

Challenge

9 *(click tabs to see student view)*

Practice

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

Free Play

12 *(click tabs to see student view)*

Levels

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

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 the patterns you made with a loop.
- Draw a pattern that you would like to make with a loop.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 10: The Right App

Overview

This lesson has students recognize that computer science can help people in real life. First, students empathize with several fictional smartphone users in order to help them find the "right app" that addresses their needs. Then, students exercise empathy and creativity to sketch their own smartphone app that addresses the needs of one additional user.

Purpose

Today, computing is more accessible than ever before. People from all walks of life use software on many different devices, particularly smartphones, for information, communication, and entertainment. Because people have such diverse experiences and needs, it is important for budding computer scientists to empathize with people and identify solutions with them in mind.

Agenda

Warm Up (5 - 10 min)

- **What's an app?**

Activity (35 min)

- **"The Right App" Scenarios (15 min)**
- **App Design Activity (20 min)**

Wrap Up (5 min)

View on Code Studio

Objectives

Students will be able to:

- List several different examples of smartphone apps.
- Recommend technology to others based on their unique needs.
- Apply empathy and creativity to design technology for others.

Preparation

☐ Read through the speaker notes in The Right App Scenarios slide deck.
☐ Prepare enough sketching/drawing supplies for all students.
☐ Make sure each student has a Think Spot Journal.

Links

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

For the Students

- **The Right App Scenarios** - Slide Deck

Teaching Guide

Warm Up (5 - 10 min)

What's an app?

Discuss: Ask students if they have ever played a game on a smartphone. If you have a smartphone of your own, show a few examples, or show screenshots of smartphone games on the web.

Discuss: Ask students what else they can do on a smartphone.

Examples to add to students' responses. Provide screenshots or live examples if available:

- search the web for information
- do math with a calculator
- talk to friends and family
- listen to music
- watch videos
- get directions using a map

Say:

- You might notice that many of the things you can do on a smartphone you can also do on a computer. This is because smartphones are actually computers, just smaller than those we use in computer science class.
- Smartphone programs are usually called "apps". Apps allow us to do all the things we just mentioned, along with many others.
- Smartphones are small and easy to take with us wherever we go, making it convenient to do things that would be difficult otherwise.

Discuss: Ask students if they have ever gotten frustrated while using an app. For example, have they ever played a game that was too hard, or the buttons didn't work, or other problems prevented them from having fun? Have they ever tried going to a website and couldn't because it was too hard to figure out how? Allow students to present an anecdote or two of their own about apps that caused them problems.

Say:

- Apps are made for people to use, just as food is made for people to eat.
- So, just like cooking, it's important to know what people want when making an app.
- An app that is made with no one in mind is probably going to have lots of problems!

Remark that there are many questions we can ask when suggesting or making an app for someone. Ideally, we want to ask questions that can help us "walk in a user's shoes":

- What do they want to do?
- What do they like?
- What is their personality?
- What kind of computer do they have?

Transition: Tell students that there are a few code.org friends who are struggling to find the apps that are "just right" for them. Ask if they are ready to put their expert skills to use and help their friends.

Activity (35 min)

"The Right App" Scenarios (15 min)

Display: Display each scenario from**The Right App Scenarios - Slide Deck** . While displaying a scenario image, read its accompanying script to the class (see below, or the notes beneath each slide in the deck). After each scenario, ask students to vote on which app they would choose for their friend, then discuss their reasons why.

The Right App Scenarios

Scenario 1: Daisy the Dragon

Slide 1: Help! Daisy the Dragon wants to play with her penguin classmates. However, when Daisy talks, her hot breath warms up anyone she is near, and penguins don't enjoy that AT ALL. So Daisy wants an app that can help her with this problem.

Slide 2: Which is the right app for Daisy?

- **Dance Master**: an app that teaches you cool dance moves.
- **Am I Too Close?!**: an app that shouts "too close!" if you are too close to someone.

Slide 3: Daisy decides to use "Am I Too Close?!", which makes it easier for her to play with her penguin friends! When they get too close, the app warns her with a friendly shout, "Too close!" Daisy and the penguins are happy! "Am I Too Close?!" is the right app for Daisy!

Scenario 2: Sam the Bat

Slide 4: Help! Sam the Bat wants to listen to his favorite music on his phone while he flies through the night sky. He's considering two different music apps, but he doesn't know which one is best for him. As a bat, Sam has poor eyesight but excellent hearing. He also happens to sing very well!

Slide 5: Which is the right app for Sam?

- **Eyes 'n Ears**: a music app that has a lot of tiny buttons.
- **Sing 'n Play**: a music app that lets you press a BIG green button to record your voice. You sing a part of any song you want to hear, and it'll play it for you.

Slide 6: Sam decides to use "Sing 'n Play". He simply presses the BIG green button, sings the first part of a song he likes, then the app plays the entire song for him! It's that easy! And that's why "Sing 'n Play" is the right app for Sam!

Scenario 3: Codella the Witch

Slide 7: Help! Codella the Witch is a serious gamer, and she's looking for a new game to play. Codella has many different kinds of computers, including a desktop, a laptop, a tablet, and a smartphone. Her favorite games are those she can play on all of her computers, including her smartphone.

Slide 8: Which is the right app for Codella?

- **Sort Knight**: a game where you play as a knight who travels around sorting everything. Also has cool dance moves. You can play it on many different kinds of computers!
- **Mobo-Robo**: a game that lets you build cool robots and make them do whatever you want. You can only play it on smartphones and tablets, though.

Slide 9: Codella decides to play "Sort Knight". At first she's not sure if she'll like it (she doesn't really like sorting things), but she loves that she can play it on any of her computers, and that's really important to her as a serious gamer. The more Codella plays "Sort Knight", the more she loves it. "Sort Knight" is the right app for her!

App Design Activity (20 min)

The final scenario allows students to sketch their own app design.

The Right App Design

Scenario 4: Anton the Alien

Slide 10: Help! Anton the Alien has crash-landed on Earth and is waiting for his fellow aliens to come rescue him. Until then, he has nothing to do, so he heads to a local library to find a book to read (Anton LOVES reading books). Unfortunately, all the books at the library are written in Earthling languages like English, Chinese, Spanish, and many others. So Anton is out of luck! Or is he?

> **Slide 11:** As an alien, Anton is very good with technology! He immediately opens his super-powered smartphone to look for an app that can help him solve his problem at the library, but he can’t find anything he likes! So let’s each design our own app for Anton!

Distribute: Pass out crayons/pencils/etc to each student as they open their**Thinkspot Journals**. In their Thinkspot Journals, students design the home screen of the app they believe best suits Anton’s needs. They can refer to the previous activity for inspiration.

Share: After designing, students share their app design with a neighbor, or if there is time, some students can share with the whole class. Each student should answer how their app design is "right" for Anton, given his situation.

Wrap Up (5 min)

Discuss: Discuss real-world examples of how apps can help people solve problems, much like they just experienced with our code.org friends.

Reiterate the importance of empathizing with users when recommending or designing apps.

- Who is this app made for?
- What does the user like?
- What problem do they have, or what do they want?
- Is this "the right app" for them? Why or why not?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **CS** - Computing Systems
- ▶ **IC** - Impacts of Computing

Lesson 11: The Big Event Jr.

Overview

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

Purpose

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

Agenda

Warm Up (15 min)
- **Vocabulary**
- **A Series of Events**

Main Activity (15 min)
- **The Big Event**

Wrap Up (10 min)
- **Reflection**

Assessment (10 min)

Extended Learning

View on Code Studio

Objectives

Students will be able to:

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

Preparation

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

Links

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

For the Teachers

- **The Big Event** - Unplugged Video (**download**)
- **The Big Event** - Assessment Answer Key **Make a Copy**

For the Students

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

Vocabulary

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

Teaching Guide

Warm Up (15 min)

Vocabulary

Say: This lesson has one new and important vocabulary word:

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

An action that causes something to happen.

A Series of Events

Discuss: Prep your class to answer a question:

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

Main Activity (15 min)

The Big Event

Discuss: Prepare students for today's activity by recalling previous work they have done.

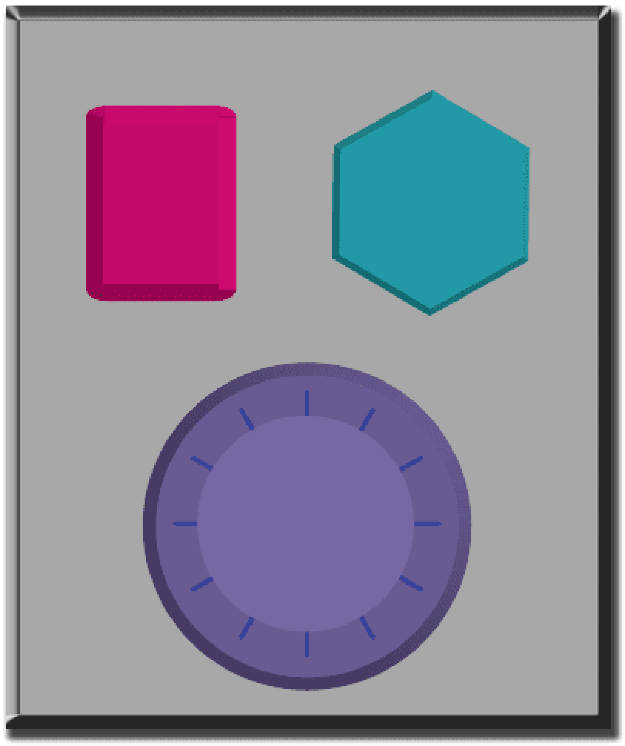
- Do you remember helping the Angry Bird find the pig?
  - In that exercise, you knew in advance exactly where you wanted Red to end up, so you could make a program that took the bird from start to finish without any interruptions.
  - In most real programs, we can't do that because we want to have options,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

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

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

Directions:

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



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

Wrap Up (10 min)

Reflection

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

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

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

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

Assessment (10 min)

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

Extended Learning

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

One Person's Event is Another One's Reaction

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

Eventopalooza

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

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▸ **AP** - Algorithms & Programming

Lesson 12: A Royal Battle with Events

Overview

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

Purpose

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

Agenda

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

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

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

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

Extended Learning

View on Code Studio

Objectives

Students will be able to:

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

Preparation

- ☐ Play through the puzzles to find any potential problem areas for your class.
- ☐ (Optional) Pick a couple of puzzles to do as a group with your class.
- ☐ Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.
- ☐ Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teachers

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

For the Students

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

Vocabulary

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

Teaching Guide

Warm Up (10 min)

Introduction

Review "The Big Event" activity with students:

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

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

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

Bridging Activity - Events (10 min)

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

Unplugged Activity Using Paper Blocks

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

Lesson Tip

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

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

Previewing Online Puzzles as a Class

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

Main Activity (30 min)

Online Puzzles and Free Play

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

Code Studio levels

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

Free Play

2 *(click tabs to see student view)*

Mini-Project: Royal Battle

3 4 5 6 7 8

(click tabs to see student view)

Free Play

9 *(click tabs to see student view)*

Levels

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

Wrap Up (5 - 10 min)

Journaling

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

Journal Prompts:

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

Extended Learning

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

Look Under the Hood

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

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

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming