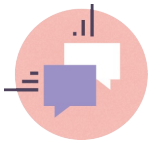
I am a Coder
Unplugged



OVERVIEW

In this lesson, students discover the importance of code in today's world. Additionally, students will repeatedly encounter failure in a positive way through an interactive problem-solving game. They should begin to view failure as a step towards succeeding.

AGENDA

- Do Now: Students write their names and their career (5 min)
- Attention Getting Signal: Teach or review your signal to move from small group to whole group (1-5 min)
- River Crossing Activity: Students repeatedly encounter failure and connect it to progress as they solve the river crossing challenge (25-30 min)
    - Introduce Challenge: Students learn what code is and understand the challenge they will be solving (5 min)
    - Small Groups: Students work in groups to solve the puzzle, returning whole class every few minutes to troubleshoot together and identify their progress through failed solutions.
- Norm Building: Students reflect on how it feels to fail and create a document that outlines how they will support themselves and each other when they encounter failure in the classroom (10 min)

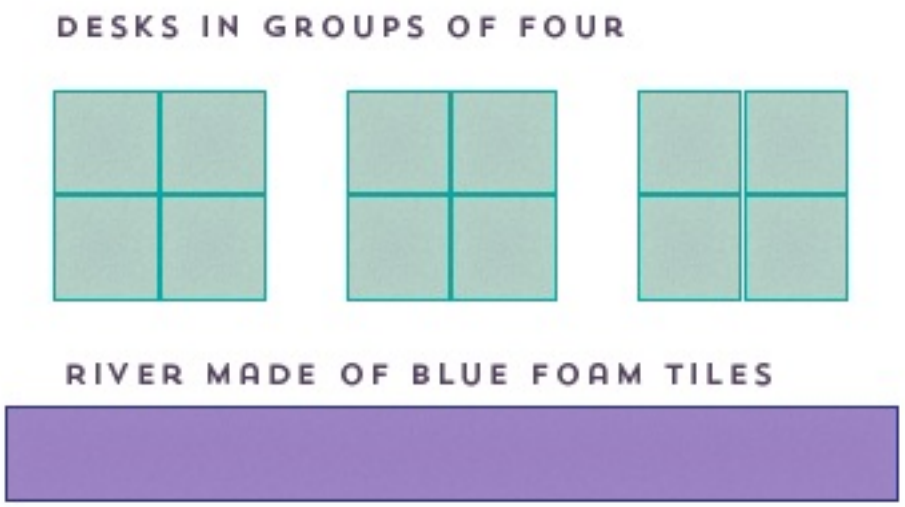
VOCAB

Code: A set of instructions designed to be carried out by a computer

MATERIALS

- Index Cards (class sets)
- Markers for index cards (class set)
- Code Cards (1 set for each group)
- Felt Strips (1 for each group)
- Paper River Crossing Worksheet (class set)
- Characters (1 set for each group)
- Classroom river (teal foam tiles)
- Teacher magnetic code cards & step arrow
- Teacher Magnetic Characters
- Chart paper (2 pieces)
- Sticky notes, 2 colors (class set of each color)
- Group Roles Sheet

Ideal Classroom Setup



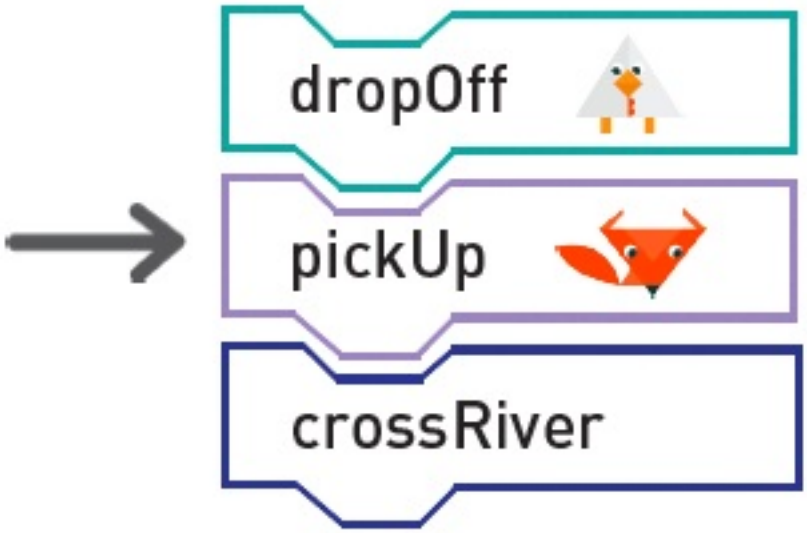
Ideal Board Setup

CHART FOR PROGRESS

Debugging Moments	Lines of Code

RIVER DRAWN ON BOARD

MAGNETIC CODE STRIPS, EXECUTION ARROW AND CLIPS FOR HANGING



Small Group Roles

When working in a group each student will have one of these roles assigned by the teacher. These roles will remain consistent throughout the school year, though students should be assigned different roles for different lessons.

Role	Description	Tools
Stepper	Keeps track of what the code has done	Execution Arrow
Computer	Executes the code from the beginning	Manipulatives
Writer	Writes the code.	Pencils or Blocks
Driver	Proposes ideas for new lines of code; group then discusses.	All

Resources

Attention getting signals:

- One
- Two
- Three
- Four
- Five

Working in groups:

- Teaching Channel

Choral Response:

- Video

DO NOW

Length: 5 minutes

Introductions, students make name cards you can use for cold calling.

Prep:

- Index Cards
- Pen or Pencils

Teacher Actions	Student Actions
1 Circulate room as students answer the do now on their paper. (3 min)	1 Students write their name on one side of their tent and on the other side they draw or write the answer to this question: When I grow up I want to be a(n) **___**.
2 Whole group: (2 min) • Introduce yourself • Have one student at each table collect the name cards and hand them to you.	2 Students place index cards at the top of desk.

ATTENTION GETTING SIGNAL

Length: 2 minutes

Teach class your attention getting signal.

Teacher Actions	Student Actions
1 There are many times during this lesson where you will need to use an attention getting signal. Use a signal that you already have or grab some ideas from our Resources.	1 Students practice the attention getting signal.

CROSSING THE RIVER

Length: 30 minutes

River Crossing Activity: Students repeatedly encounter failure and connect it to progress as they solve the river crossing challenge.

Prep:

- Code Cards
- Felt Strips
- Paper River Crossing Worksheet
- Characters
- Classroom River
- Teacher Magnetic Code Cards & Step Arrow
- Teacher Characters

Teacher Actions	Student Actions
1 Introduce What is Code: (1 minute) • Over this school year we are going to meet as a class once a week to build our skills as computer coders. • Can anyone tell me what is code? • As a __ (profession a student wrote on their card) how could you use code?	1 Students raise their hands to share their ideas about what is code and what is a coder. • Answer: A set of instructions designed to be carried out by a computer. It is the instructions someone wrote to make your website, app, game, etc. run. • Answer: Design your own website, build your own app, etc.

```
It is important that students d
```

	on’t say that using a website o r computer is code. The distinc tion is that learning to code a llows them to produce, not just consume.
2 Introduce Productive Failure: (1 minute) • Today we’re going to become coders as we tackle a difficult challenge. • Our code will not be carried out by a computer today, but it is going to mimic the process of writing code for a computer. • We are probably not going to get it right the first time. It may take us getting it wrong a lot before we find our solu tion! That is what happens when you write code. • When coders get it wrong it is called a bug. • Debugging is at the heart of the practice of coding. So today it is ok when your solution is wrong. Nothing bad is going to happen.	2 Students choral respond keywords on teacher’s queue to increase engagement.
3 Reveal the Mission: (3 minutes) A farmer needs to cross a river with a chicken, a fox, and a bag of grain. However, his boat can only hold him and one other object. If left together, the fox will eat the chicken and the	3 Students follow along with mission on the board and answer CFUs. • Answer: The fox will eat the chicken.

chicken will eat the grain. You need to get the farmer across the river without losing any of them.

```
If a student has done this puzz
le before it is not likely they
 have the solution memorized. A
sk the student to write out the
 solution and if they have it t
hat student can become your hel
per for the day.
```

- What happens if the chicken and the fox are left alone while the farmer goes in the boat?
- If left alone, what will the chicken eat?
- If the farmer is with them, will the chicken still eat the grain?

- Answer: The chicken will eat the grain.
- Answer: No, the fox and chicken only act up if the farmer crosses the river.

4

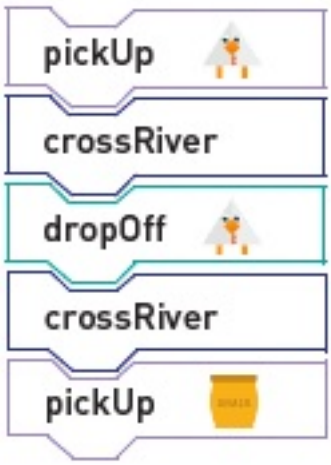
Demo a round of problem solving with roles as a whole class: (5 mins)

Driver: (teacher) Ask students to propose a solution. Writer: Move the code cards to match the proposed solution. Stepper: Move the execution arrow and read each line of code one at a time. Computer: Move the magnetic pictures on the board.

```
Suggested script when our solut
ion is wrong:
Our solution is wrong. We've go
t to start over. It's part of b
eing a programmer!
```

4

Student volunteers act as:
Writer: Move the code cards to match the proposed solution.
Stepper: Move the execution arrow and read each line of code one at a time. Computer: Move the magnetic pictures on the board.



5 Group Tackle: (15 minutes)

- Instruct students to draw their roles and connect them to their peer models in step 4
- Instruct students to pull materials from envelopes and work in their groups
- Pause every 3-5 minutes to troubleshoot as a whole class and reveal the following hints:
  1. Chicken has to cross the river first.
  2. A character can cross the river more than once.
  3. The farmer has to take either the chicken or the other item back across the river on his fourth trip.

5 Working in groups:

Students draw roles from the bags. Students follow their roles to assemble their code linearly on the felt. Groups that have the correct answer early: there are multiple solutions. Challenge them to find another solution.

6 Come to a solution: (5 min)

6

Call on a different group to share their solution. Follow the guidelines on whole group sharing. • Help students come to the solution. • Act out the solution as you step through each code card on the board.	One group brings solution to the board and perform it using their roles.
7 Direct students to return materials to the envelopes.	7 Students return materials to envelopes.

NORM SETTING

Length: 10 minutes

Students reflect on how it feels to be wrong as they set norms for how they will treat themselves and their peers when they encounter failure.

Prep:

- Sticky Notes (2 colors, class set of each color)

Teacher Actions	Student Actions
1 Whole class have students identify how it felt to get a solution wrong. • How does it feel when you are wrong? • As coders, we are going to be wrong. It is part of the process of creating good code. • It is about how we act and what we do when we fail that defines the kind of coder we are.	1 Students raise their hands to share their answers to the question.
2 Answer questions on stickies: • We are going to define how we can support each other when we have a bug.	2 Individually students answer the two questions on their sticky notes.

- Students may need more prompting with "how does it look like/sound like when..." as follow up questions to oversimplified answers.

1. How do you want to be treated when you're wrong?
2. How can you manage your fear when you're wrong?

3 By the end of this activity you should have a succinct set of norms for how the class wants to be treated when they fail.

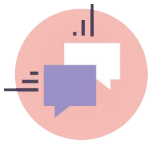
- Collect stickies on board
- Group together similar ideas and read aloud.
- Students give thumbs up when they agree

3 One student in each group collects the stickies and brings them up to the appropriate place on the board. Students respond with a silent thumbs up if they agree with a statement.

4 Close out the lesson by acknowledging their great ideas and work they have accomplished.

```
Suggested script:
You have such great ideas! I am
 going to write these up so tha
t everyone can see your work an
d bring it to our next class. T
his is going to help guide us t
his year as we become coders in
 how we are going to treat ours
elves and eachother when we enc
ounter bugs in our code or an i
dea of ours fails. The lesson i
s that you have to fail before
you succeed. Thank you so much
for sharing your time and ideas
 with us today. I can't wait to
 tackle our next coding problem
 next week.
```

Getting Started on Scratch
Plugged



OVERVIEW

In this lesson, students will learn to login to Scratch and begin to use the editor, block palette, and stage to sequence a program.

OBJECTIVES

- I can login, save my work, title my work using naming conventions, and organize my projects in Scratch.
- I can use the editor, block palette, and stage in Scratch to code my program.
- I can decompose a maze game into its fundamental requirements: hero, enemy, obstacles, goal.

AGENDA

- Do Now (5 min) - Students see language written in block code and interpret its meaning, identifying that programming language is not the same as how we speak and write.
- Logging-in (15 min): Students login to their Scratch accounts.
- Saving Work (10 min): Students learn how to save and title their projects.
- Starter Maze (10 min): Students are introduced to Escape the Maze and practice reading and predicting how blocks of code will perform in the Scratch stage.
- Brainstorm (5 min): Students brainstorm ideas for building out the Escape the Maze game.
- OPTIONAL: Exit Ticket (15 min): Students predict how simple sequences of blocks will perform in the Scratch stage.

VOCAB

- Editor: A program designed for editing computer code by coders.

MATERIALS

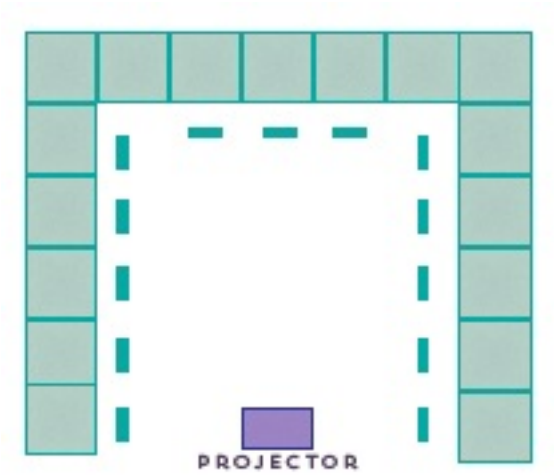
- Projector
- Do Now (class set)
- Computers (class set)
- Scratch Log-In Stickers
- Ideas Journals (class set)
- Exit Ticket (class set if printing, otherwise on Socrative)
- Chart Paper (1 piece)

Notes

Prior to this lesson pass out the Scratch log-in stickers and have students place them on the inside cover of their idea journals.

This lesson includes several sets of steps that walk students through navigating the Scratch page. While the lesson includes suggestions for how to chunk them, you should modify the number of steps they are to follow at one time based on your students' familiarity with navigating websites and their ability to follow a sequence of instructions independently.

Ideal Desk Setup



Lesson Resources

If coding is new to your students we suggest showing them the Code.org short Code Stars

Pass out idea journals and have students place login stickers on the inside cover of their journals. Journals should be passed out and collected at the beginning and ending of each class.

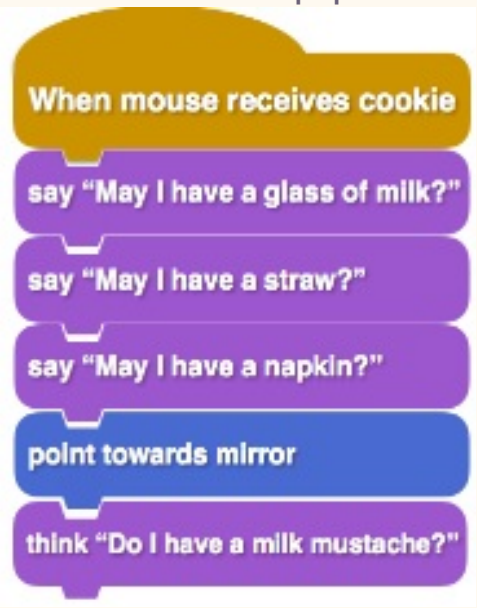
DO NOW

Length: 5 minutes

Students translate a block code version of "If you give a mouse a cookie" into a narrative and identify that computer code is different than the narrative language we use when telling a story or talking to each other.

Prep:

- Do Now worksheet

Teacher Actions	Student Actions
1 Circulate room as students answer the do now on their paper.	1 Students independently answer the do now on their paper 
2	2

After 3 minutes ask volunteers to read their interpretations. Read through the blocks so students can hear the sequence flow. Follow up questions for the class: • Does anyone recognize what book this is from? • Is this how we talk or tell a story? • When have you seen a sequence that looks like this before? • Is the mouse ever going to ask for a straw before it asks for a cookie?	Students raise their hands to volunteer their interpretations of the sequence and answer the follow up questions. • Answer: If You Give a Mouse a Cookie • Answer: No • Answer: we used block code in last week's lesson to write the instructions for crossing the river • Answer: No, because the code order determines the order of the story.
3 Connect the Do Now to the previous lesson in which students used paper code blocks to transport animals across a river. Key points: • Code is a set of instructions designed to be carried out by a computer. • In the previous lesson we used paper code to transport animals across a river. We acted as the computer in that lesson. • In this Do Now the instructions are also in the form of blocks. • Today we get to explore coding blocks on a computer. Key points to hit:	3 Students are facing teacher, backs to computers, listening to the teacher's explanation.

- The language we use when we program a computer is not the same as the way we speak or write.
- Computers have their own languages and as we become coders we learn to write in the computer's language.

LOGGING-IN AND EXPLORING

Length: 10 minutes

Students log-in and explore the Scratch studio.

Prep: Idea Journals and Log-in Stickers

Teacher Actions	Student Actions
1 Introduce Scratch as the program we will be using to code. • Has anyone used Scratch before? `Acknowledge students who h ave. They will be a great asset to their peers as we all get comfortable navig ating the program.`	1 Students are faced away from their computers, towards the teacher.
2 Show the [Getting Started with Scratch Video]. (https://scratch.mit.edu/help/videos/#)	2 Students watch video on projector.
3 Point students to their log-in stickers inside their journals.	3 Students hold up their idea journals and point to their stickers.

• Hold up the idea journal to model where they put their stickers. • If students haven't put stickers in journals yet, do this now.	
4 Remind students of the computer contracts they signed and expectations while we are on computers: 1. Raise your hand silently if you have a question. 2. When you complete a step give 2 silent thumbs up. 3. Close your laptops all the way when instructed.	4 Students practice the 3 expectations.
5 Step through the 5 steps they will follow on their computer. • If students are independently computer literate: • Give the go signal for students to turn and face their com puters. • Circulate the room helping students who are stuck. If students are not independently computer literate: • You may need to teach students how to type an underscore (_)	5 Students walk through the steps to log-in to their Scratch accounts and begin to explore what they can do in the Scratch editor. Step 1: Go to scratch.mit.edu Step 2: Click the  button in the upper right Step 3: Enter your username and password from your sticker and answer the questions when prompted. Step 4: Click on the "Exploring Scratch" studio and open "Make Something" Step 5: Create! Try making the sprite do something.

• Go slide by slide through each step having students complete the slide with you	
6 Show the "Make Your Sprite Move Forward" video and give students time to explore the Scratch editor. • Practice dragging blocks in the script area, connect them, and click the green flag to run their program. • If a student is more advanced and experienced with Scratch, point them to the challenges in the studio. Hints in Scratch to share if students are struggling: • If you don't know what a block of code does click the ? in the menu bar. • Click on the top block to drag without separating blocks. • Clicking the bottom block will detach that piece. Attention getting signal • What did you learn about Scratch? • What did you observe in your exploration?	6 Students watch the video explore the Scratch editor.

SAVING WORK ON SCRATCH

Length: 10 minutes

Students learn to save and title their work. If there is time, share volunteers' programs.

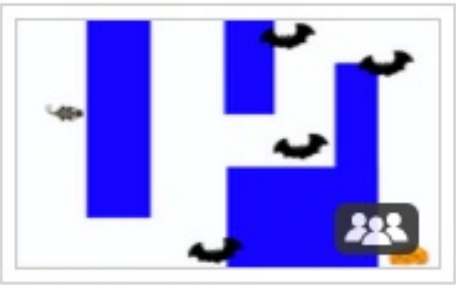
Teacher Actions	Student Actions
1 Step through the steps students will follow to save their work. If students are independently computer literate: • Give the go signal for students to turn and face their computers. • Circulate the room helping students who are stuck. If students are not independently computer literate: • Go slide by slide through each step having students complete the slide with you	1 Students are facing teacher, backs to computers, listening for instructions. Step 1: Click  in the top right of the editor. Step 2: Click on the title to rename your project. Step 3: Delete "remix" from the title and write "Challenge #_username" Step 4: Click "Share"
2 When 100% of the class has their programs named and remixed, if there is time: • Who created something in Scratch and will let us view their code?	2 Students raise their hands to volunteer to show their code.

3 Select a student volunteer from your class on Scratch and project their program on the big screen. • This is an opportunity to reference the norms the class drafted in lesson 1 of how they want to be treated when they fail. Recognize the student's bravery for showing off their code.	3 Students watch the screen at the front of the room and positively support their peers who demo their work.

STARTER MAZE

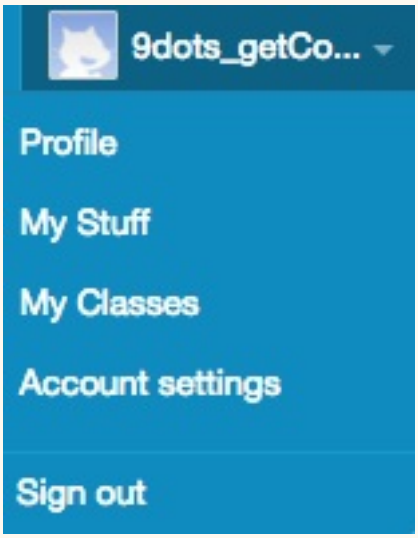
Length: 15 minutes

Students explore the maze project and predict how the code performs in the game

Teacher Actions	Student Actions
1 Introduce the maze project! • Who here has done a maze before? • Who can tell me what is a maze? Dramatically reveal Escape the maze: • Part videogame, part maze • Students are going to learn to code like app developers! Suggested script: Now that you played around in the coding editor of Scratch, I'm going to let you in on the big surprise. This year, you're not just becoming coders, you're going to learn to code like a game developer! You're going to build your very own video game: Escape the Maze! I think you are ready to see some of the code right now.	1 Students raise their hands to answer question. Answers should include: • A path you need to navigate to the end • A start and an end/goal • Sometimes there are dead ends or obstacles  Escape the Maze

2 Step through navigating to the starter maze and give them 2 minutes to play the game.

2 Step 1: Click on your username in the upper right and select "My Class"



Step 2: Click on the class studio "Escape the maze" Step 3: Open the starter maze, read the instructions, and click

to play the game.

3 Attention getting signal

- What is happening in the game?
- What happens if the ball touches the blue?
- What does the blue represent?
- What is the green block?

3 Students share out what happened in the game.

- Answer: They are moving the ball through the maze.
- Answer: The ball bounces backwards
- Answer: The walls of the maze

• What happens when the ball gets to the green block?	• Answer: The end, finish, goal • Answer: "You win!"
4 Click to look at the code inside. Go through each block of code and ask students to connect it to the observations they made above about how the game is performing. • What is each sequence of blocks telling the program to do? • Give students 1 minute to discuss with their partners what the code is doing before sharing out.	4 Students Think Pair Share to predict how the shown block of code is performing in the stage.

BRAINSTORM

Length: 5 minutes

Students brainstorm ideas for building out their maze in a list that we will reference as they develop their projects.

Prep: Chart Paper

Teacher Actions	Student Actions
1 Brainstorm improving the game and record answers on chart paper or in a document: • Think about some of the other games that you play. What could we add to Escape the Maze to make it more fun to play? ◦ Call on 2 students to share. ◦ Give table partners 2 minutes to discuss and write ideas in their idea journals. ◦ If a student says they like a particular game (e.g. Pokemon) ask them what features they like about the game (e.g. you can earn points).	1 Students brainstorm ideas with their table partners and write in their idea journals.

2 Attention getting signal Call on students to share out their ideas: • Continue to record their ideas • It's totally okay to write down an idea if your friends come up with it.	**2** Students share out ideas and record ones they like in their idea journals.
3 Close out this section of the lesson by letting students know you are excited about their ideas and the games they will code. Suggested script: These are all great ideas. We w ill add them to our game next time. We will start to bui ld onto the Escape the Maze game to make it more fun. As yo u become master coders, you are going to take over and design your own game. Keep brainstorming new ideas to put into your game. I can't wait to see the games you develop!	

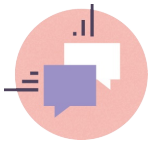
EXIT TICKET

Length: 15 minutes

Prep: Socrative Quiz SOC-23584751

Teacher Actions	Student Actions
1 Introduce the exit ticket: • This is an opportunity for me to see where we stand as a class. • It is important for you to try your best, but understand that some of these questions have been designed to be challenging and we are going to have time in upcoming lessons to follow up on questions you don't know yet.	1 Students independently complete the exit ticket on Socrative.
2 Launch the exit ticket at t.socrative.com. Walk students through what Socrative looks like and how to answer the questions on m.socrative.com. You will need to give them the room name. SOC-23584751	

Maze Scavenger Hunt
Unplugged



OVERVIEW

In this lesson, students learn the importance of giving clear and explicit directions. They practice sequencing code, writing code, and navigating their robot on a paper grid.

AGENDA

1. Warm Up: Students are introduced to the importance of writing clear instructions by writing a set of instructions for their teacher to make a peanut butter and jelly sandwich. (10 min)
2. Navigate the Maze: The class is introduced to sequencing code by navigating their teacher to a spot on a grid. (10 min.)
3. Maze Scavenger Hunt: Students practice sequencing code in groups by writing code to collect prizes on a grid. (20 min.)
4. Collect the Dots Extension: Students continue practicing in groups by writing code to collect dots on a grid. (15 min.)
5. Exit Ticket: Students write code to navigate a robot to a spot on a paper grid. (5 min)

VOCAB

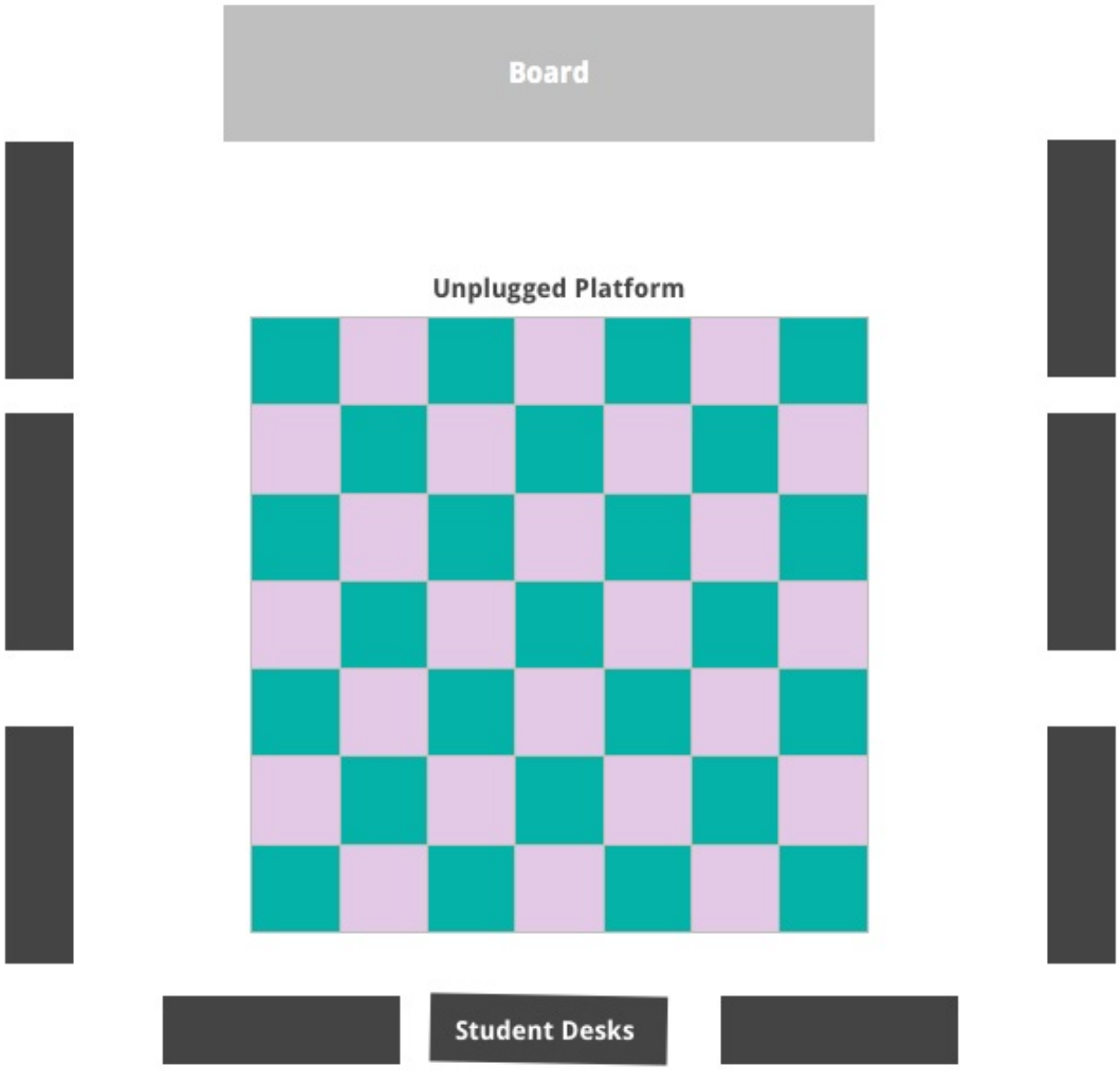
- Code: A set of instructions designed to be carried out by a computer.
- Validate: To check if something is correct or does what it is intended to do.

MATERIALS

- Peanut butter

- Plastic butter knife
- Jelly
- Bread
- Paper plate
- Warm up handout
- Paper maze worksheet
- Scavenger hunt maze worksheet
- Role cards
- Rover cutouts
- Brown paper bag
- Large maze cutouts
- Collect the Dots worksheet
- Exit ticket
- Unplugged platform
- Rover code blocks

Classroom Setup


Board
Unplugged Platform
Student Desks

WARM-UP

Length: 10 minutes

Students learn about the importance of giving clear and precise instructions by writing step-by-step instructions to make a peanut butter and jelly sandwich.

Materials:

- Warm Up Handout
- Peanut Butter
- Jelly
- Bread
- Paper Plate
- Plastic Butter Knife

Participation: Individual

Teacher Actions	Student Actions
1 Have students respond to the prompt below in 5 minutes. • Using the materials on the front table, write a set of step-by-step instructions for your teacher to make a peanut butter and jelly sandwich.	1 Students write instructions for their teacher to make a peanut butter and jelly sandwich.
2 Ask for volunteers to share out their instructions in a step-by-step manner. Follow each step as the	2 Volunteers share their instructions with the class as the teacher attempts to follow

student reads it. If a student says to put the peanut butter on the bread, put the jar of peanut butter on top of the bag of bread. The point is to show the importance of giving clear and precise instructions. Allow at least two different students to share.	along.
3 Key Points • Giving clear and precise instructions is important. • Humans can often interpret the meaning and intentions behind a set instructions in ways computers can't. `Example: A person can say "you open door" to another person and that person would be able to determine that they are being instructed to open a door.` ◦ In this class we will be writing clear and precise instructions that a computer can follow.	

NAVIGATE THE MAZE

Length: 10 minutes

Students are introduced to sequencing by coding instructions to navigate their teacher through a maze.

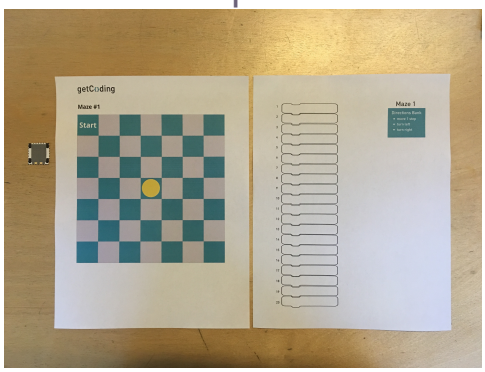
Prep:

- Paper maze worksheet
- Rover cutout
- Unplugged platform
- Rover code blocks

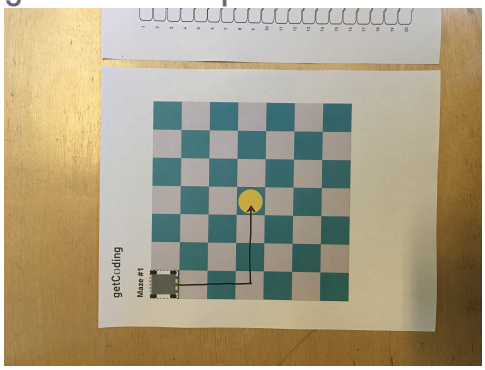
Participation Whole Group and Individual

Teacher Actions	Student Actions
1 Introduce the class to Rover the robot. Just like other computers, Rover can only follow clear and precise instructions in the form of code Code is a set of instructions designed to be carried out by a computer. Rover the robot can only follow three instructions. • move 1 step • turn left • turn right	1 Students listen to the introduction to Rover the robot.
2 Rover's Instructions	2

	Explain Rover's three instructions. • move 1 step: makes Rover move one square in the direction it is facing (direction its headlights are pointing). • turn left: makes Rover turn to its left Have students point to their left. Then have them point to the direction Rover would turn if it was instructed to turn left. Explain that Rover's left is different than ours when it is facing in a different direction than we are. • turn right: makes Rover turn to its right Have students point to their right. Then have them point to the direction Rover would turn if it was instructed to turn right. Emphasize that our left and right is different than Rover's when it it facing in a different direction than we are.	Students follow along by responding to teacher's prompts.
3	Check for Understanding Perform a check for understanding. Show the first Check for Understanding slide and ask the class to point to the direction Rover would turn if it was asked to turn left. Show the next slide and repeat the same prompt.	3 Students respond to the teacher prompts by pointing in the direction Rover would turn.

4 Have two student volunteers pass out robot cutouts and paper maze worksheets. Students should have their materials laid out in front of them like in the picture below.  	**4** Volunteers pass out materials and students lay out their materials in front of them.
5 Introduce the Maze Challenge Explain the goal and rules of the first challenge. • We will try to navigate Rover to the yellow spot on the grid by working together as a class. • We can navigate Rover by using the only three instructions it can follow. • We will sequence the code by writing the directions in the blank code blocks. Describe the roles that students will play during the activity. 1. Driver: Proposes idea for new lines of code; group discusses 2. Writer: Writes the code.	**5** Students listen to the introduction of the maze challenge.

3. Stepper: Keeps track of what the code has done 4. Computer: Executes the code from the beginning. Explain who will play each of the roles during the activity. • The driver will be played by volunteers in the class. • The writer will be a volunteer that will stick code blocks on the board. • The stepper will be a volunteer that moves Rover on the unplugged platform. • The computer will be played by the whole class when we read the code out loud.	
6 Ask the class for volunteers to play the role of the writer and stepper.	6 Students volunteer to play the role of the stepper and writer during the maze challenge.
7 Complete the Challenge: Have students place their Rover cutout at the starting square of their paper maze. Explain that the first step in completing this challenge is planning the path the Rover will take to get to the yellow spot. Show that there are many paths that can be taken by displaying three different choices in the slides.	7 Students place their Rover cutout on the starting spot on the grid.

8 Have the class vote on the path they will take and have them trace the path on their paper grid like in the picture below: 	8 Students vote on the path to take to get to the yellow spot on the grid and trace the path on their paper mazes.
9 Ask the class what the first instruction should be. `Highlight that this is a an exa mple of the class playing the r ole of the driver.`	9 Students share out possible instructions.
10 Once the class has settled on an instruction, have the writer place a code block on the board and have the stepper move Rover in response to the instruction.	10 The writer places a code block on the board and the stepper moves Rover on the unplugged platform.
11 Instruct students to write the instruction into the first blank code block on their worksheet and to move their Rover cutout on their paper maze. Continue this process each time an instruction is chosen by the class.	11 Students write the code into their code block and move Rover on their paper maze to match the position of Rover on the unplugged platform.

• Inform the class that you will be cold calling students for instructions after a volunteer shares the second instruction for Rover.	
12 After the volunteer shares the second instruction for Rover, have the writer place the second code block on the board and have the stepper move Rover in response to the instruction.	12 A volunteer shares out the next line of code. The stepper updates the code on the board and the stepper moves Rover on the unplugged platform. All students update their code and move Rover on their paper mazes.
13 Cold call different students for the rest of the sequence.	13 Students called on share out the next line of code.
14 When the sequence is complete, have the stepper place Rover back at the starting square. Have the class read the code in unison and have the stepper move Rover as each block of code is read to check if the code is correct. `Highlight that this is an examp le of the class playing the rol e of the computer.`	14 The stepper moves Rover back to the starting spot. The class reads the code in unison as the stepper moves Rover in response to the lines of code that are read.
15 Plan, Code, Validate Process	15

Describe the process the class just used to plan, write, and check the code. 1. Plan: Explain that the class first planned how to complete the challenge by choosing a path and tracing it out on the paper grid. 2. Code: Explain that the class coded the path one step at a time. 3. Validate: Explain that the word validate means to check if something is correct or does what it is intended to do. In this process the class validated the code by reading the code starting at the beginning to see if it did what it was intended to do (get Rover to the yellow spot).	Students listen to the process they just went through together.
16 Maze 2 Change the position of Rover and the yellow spot on the grid to match maze two. Explain that in this challenge students will work independently to get Rover to the yellow spot by writing their own code using the paper maze and Rover cutout. • Display the plan, code, validate process on the board and remind students	**16** Students listen to the instructions for the next maze challenge.

that they should use this process to complete the challenge. • Inform students that they will have 5 minutes to complete the challenge and that you will call on volunteers to share their code.	
17 Allow students to begin planning and writing their code.	
18 After 5 minutes have passed, have volunteers share out their code and act it out as they read it.	18 Volunteers share out their code.

MAZE SCAVENGER HUNT

Length: 20 minutes

Students practice sequencing instructions by navigating a classmate to collect a series of prizes in a maze.

Prep:

- 1 Scavenger hunt maze worksheet per group
- 1 Rover cutout per group
- Brown paper bag
- 1 set of Role cards per group
- Prizes
- Unplugged platform

Participation: Groupe of 4

Teacher Actions	Student Actions
1 Introduce the Scavenger Hunt Introduce the class to the scavenger hunt challenge. 1. Divide the class into groups of 4. 2. Pass out 1 scavenger hunt worksheet and Rover cutout per group. 3. Each group will have 7 minutes to write code to get Rover to collect prizes on the unplugged platform.	1 Students move into groups of

4. Each person in the group will play one of the roles described earlier in the lesson.

2 Have a student reach into the bag and pass out the role cards to the rest of his/her groupmates.

2 Students receive the role they will play in the group from a group mate.

3 Explain the roles each person will play in the group.

- Driver: proposes lines of code and asks if the group agrees
- Writer: writes the code on the worksheet
- Stepper: moves the Rover cutout on the paper maze and plays the role of Rover when the team presents its code
- Computer: checks another group's code for by walking through their code from the beginning and presents his/her team's code by reading it out loud as the stepper acts out Rover's movements.

- Emphasize that this is not a competition.

- Project the scavenger hunt maze on the board and explain that each group will

3 Students listen to the roles they will play.

have 1 minute to trace the path they will try to code to collect prizes on the platform.	
4 Start the 1 minute timer and allow teams to plan and trace their path on the scavenger hunt paper maze. • After 1 minute has passed check in with each group to ensure that they have a path traced out.	4 Groups plan and trace the path they will take to complete the challenge.
5 After 3 minutes have passed, have all of the groups put down their materials and have the computers rotate to the next group to walk through the code. • After 1 minute has passed, have the computers go back to their original group and allow each group to work for 3 more minutes.	5 The person playing the role of the computer rotates and checks the code of a neighboring group.
6 After 3 minutes have passed, have groups put down their materials and call on groups to present their code.	6 Groups present and act out their code on the unplugged platform.

EXTENSION ACTIVITY

Length: 15 minutes

The class practices sequencing by writing code to collect dots on a grid without barriers.

Prep:

- 1 Collect the Dots paper worksheet per group
- 1 Rover cutout per group
- Unplugged platform

Participation: Groups of 4

Teacher Actions	Student Actions
1 Explain that in this activity students are collecting dots by navigating a classmate (Rover) around a grid that is free of obstacles. Inform the class that they will be working with the same teams, but they will be choosing new roles.	1 Students listen to directions.
2 Instruct students to place their role cards back in the paper bag. Have them re-select cards from the bag in order to choose their new role.	2 Student place their role cards back in the brown paper bag and re-select roles

3 Present the new grid layout and explain that there are many ways to complete the task. Inform students that they will only have 5 minutes to plan and code their route.	3 Students listen to instructions.
4 Set the timer for 5 minutes and allow groups to begin.	4 Groups plan and code their routes.
5 After five minutes have passed, have groups present their code in the same manner that the scavenger hunt code was presented.	5 Groups present their code.

EXIT TICKET

Length: 5 minutes

The class completes an exit ticket.

Prep: Exit ticket

Participation: Individual

Teacher Actions	Student Actions
1 Say: Complete the exit ticket independently at your desk. This is your opportunity to show off what you know. Start 5 minute timer Collect exit tickets.	1 Students complete the exit ticket independently.