

Lesson 5: Loops Explore

Overview

Students begin the lesson by discussing the purpose of loops before completing the unplugged activity. This activity involves moving a "robot" around a game board while practicing tracing blocks of code by hand. To conclude, the lesson is wrapped up with a vocabulary discussion and a video.

Purpose

Students are introduced to the concept of loops through a robot maze activity. This unplugged lessons provides students a physical mental model they will be able to use when they start programming with loops in the subsequent lessons.

Standards

Full Course Alignment

CSP Conceptual Framework

- **AAP-2** - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

Agenda

Lesson Modifications

Warm Up (5 minutes)

Set-up

Activity (30 minutes)

Loops

Wrap Up (10 minutes)

Introduction to Loops

Assessment: Check For Understanding

Objectives

Students will be able to:

- Identify the exit point of a loop.
- Trace a simple program with a loop
- Use appropriate vocabulary to describe loops.

Preparation

- 1 game board per pair of students
- 1 "robot" per pair of students
- Game pieces, markers, or tokens that can be used to represent barriers
- Review the Intro to Loops presentation and click through all animations

Links

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

For the teachers

- **CSP Unit 5 - Lists Loops, and Traversals** - Slides
- **Loops Game Board**

For the students

- **Using Loops** - Video (**Download**)

Teaching Guide

Lesson Modifications



Attention, teachers! If you are teaching virtually or in a socially-distanced classroom, please read the full lesson plan below, then click **here** to access the modifications.

Warm Up (5 minutes)

Set-up

💡 Teaching Tip ▲

Supplies Substitutions:

- The game pieces can be replaced with knick-knacks, tokens, markers, paperclips, or scraps of paper.
- The "robot" can be any small item that is clearly facing a direction. It could be as simple as a scrap of paper shaped like a triangle or a paperclip with googly eyes.

📋 Distribute: The activity today involves a lot of moving pieces. Take this time at the beginning of class to pass out supplies and pair students.

Per pair of students:

- 1 game board
- 1 "robot"
- Game pieces, markers, or tokens that can be used to represent barriers

Group: Put students in groups of two.

Activity (30 minutes)

Loops

💡 Teaching Tip ▲

Running the Activity: This activity asks students to follow along as a number of core concepts for programming are introduced. The model is typically that a term or concept is introduced and modeled and then afterwards students are encouraged to try it out on their own. Trying it out typically means they are writing information on a sticky note and sharing it with another group before discussing the results with the whole class.

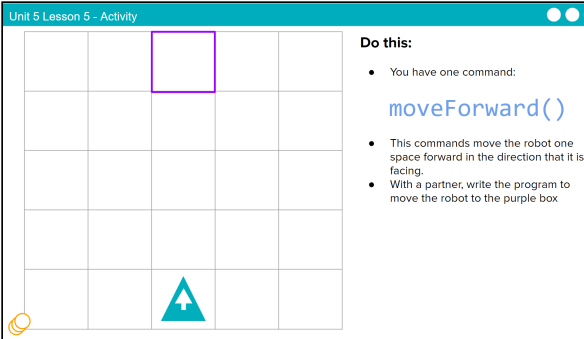

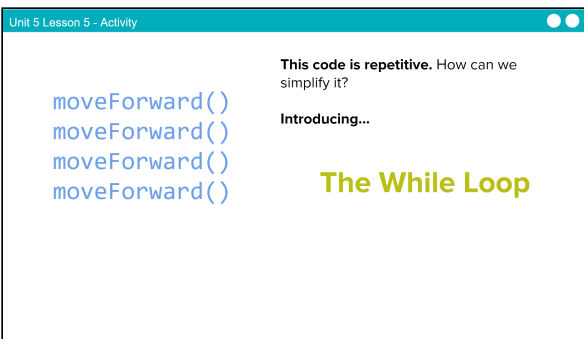
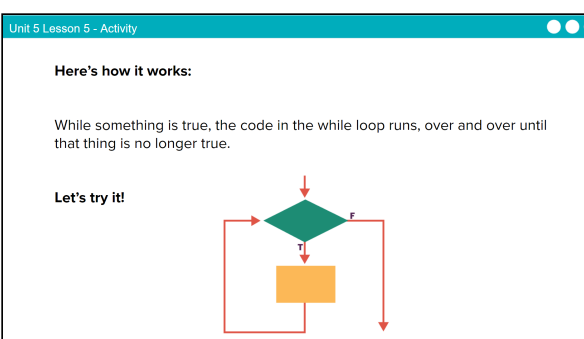
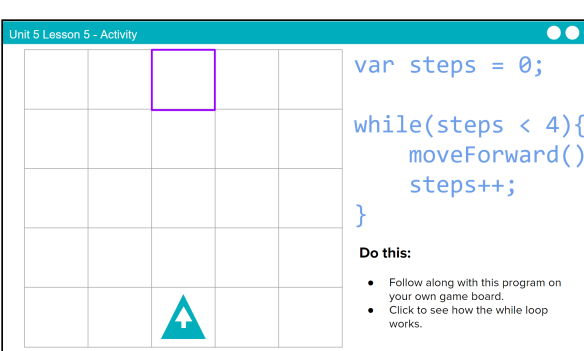
Slides with animations have an icon in the bottom left corner to let you know you need to click to reveal more of the slide's content.

To help you more easily prepare the activity and keep track of your instructions, detailed instructions have been included as speaker notes in the presentation. Here are some tips to help you throughout the presentation.

- There are opportunities throughout the presentation for students to actively engage. At these moments students should be making things with their manipulatives or using them to answer questions. Use these opportunities to check progress.

- There is a fair amount of new vocabulary introduced but it is introduced gradually and with intentional repetition. Make a point of actively modeling the use of new terms.
- The most important goal here is building a mental model. It is ok if students have some open questions that will get resolved over the subsequent conditional lessons.
- Both you and students can use the "Key Takeaways" to check your understanding at the end.

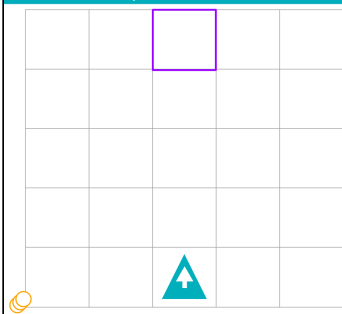
Display: Use the activity slides for this lesson to guide the unplugged activity on Loops.

Slides	Speaker Notes
	<p>Say: Today we are going to explore Loops. You will move a robot around a game board following commands. Here's the first one: <code>moveForward()</code> .</p> <p>Do This: Use this command to write a program to move the robot to the purple box.</p> <p> Click through animation to reveal the answer.</p>
	<p>Say: This code is repetitive. How can we simplify it? By using a While Loop.</p>
	<p>Say: Here's how a While Loop works. While something is true, the code in the while loop runs, over and over until that thing is no longer true. Let's try it together!</p>
	<p>Say: Let's run this program together. Set up your board with the robot in the same place as the robot on the screen. We are going to run each step of the loop together. First you will try it on your board, and then we will look at the answers on the screen.</p>

Slides

Speaker Notes

Unit 5 Lesson 5 - Activity



Round 1

```
var steps = 0;

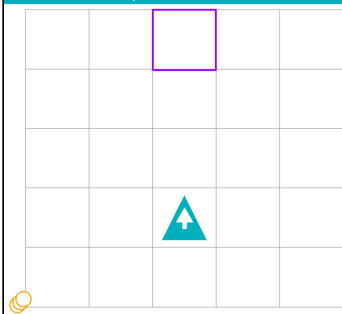
while(steps < 4){
  moveForward();
  steps++;
}
```

Say: Read the code with your partner and complete the first round of the loop. You may want a sticky note or a scrap piece of paper out to keep track of the value in the variable.



****Click through to view the answer after students have had a chance to follow the code on their own board. Discuss each part of the code as it is running.**

Unit 5 Lesson 5 - Activity



Round 2

```
var steps = 1;

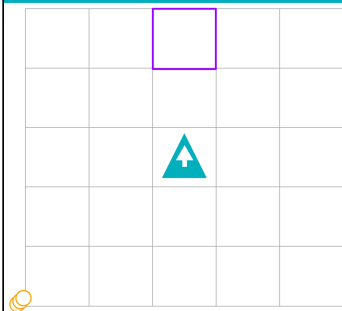
while(steps < 4){
  moveForward();
  steps++;
}
```

Say: Let's continue on with the second round. You do it first.



****Click through to view the answer after students have had a chance to follow the code on their own board. Discuss each part of the code as it is running.**

Unit 5 Lesson 5 - Activity



Round 3

```
var steps = 2;

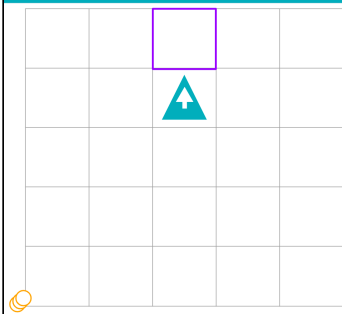
while(steps < 4){
  moveForward();
  steps++;
}
```

Say: Now on to the third round. You do it first.



****Click through to view the answer after students have had a chance to follow the code on their own board. Discuss each part of the code as it is running.**

Unit 5 Lesson 5 - Activity



Round 4

```
var steps = 3;

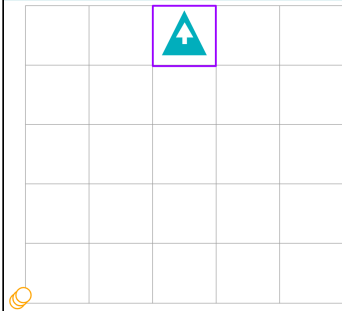
while(steps < 4){
  moveForward();
  steps++;
}
```

Say: Time for the fourth round. You do it first.



****Click through to view the answer after students have had a chance to follow the code on their own board. Discuss each part of the code as it is running.**

Unit 5 Lesson 5 - Activity



Round 5

```
var steps = 4;

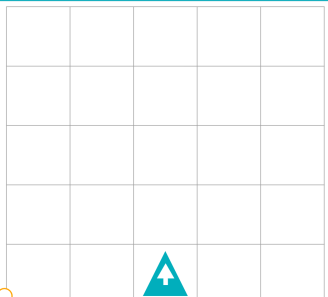
while(steps < 4){
  moveForward();
  steps++;
}
```

Say: What happens here? Discuss with your partner.

****Click through to view the answer after students have had a chance to follow the code on their own board. Discuss each part of the code as it is running. The main thing to point out here is that the loop ends because the Boolean expression evaluates to false.**

Slides

Unit 5 Lesson 5 - Activity



```
var steps = 0;

while(steps < 4){
  moveForward();
}
```

Do this:

- Run this program on your board.
- What happens?

Speaker Notes

Do This: ** With your partner, run this program on your board. What happens?

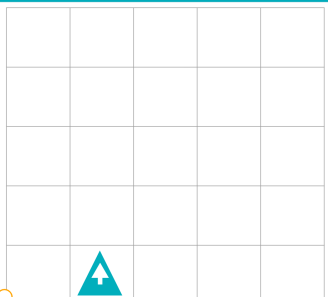


****Click for animation**

Discuss: Why does the robot run off the board?

- this is an example of an infinite loop. The program never ends, because steps will always be less than 4.

Unit 5 Lesson 5 - Activity



Do this:

- You have a new command:

```
turnRight();
```

- This command turns the robot 90 degrees to the right.
- Click to see a new program, Run the program on your board. Where will the robot end up?

Say: You have a new command `turnRight()`

Do This: Run this program on your board. Where will the robot end up?



Click through animation to see the answer

Unit 5 Lesson 5 - Activity

While Loop: 3 Parts

```
var steps = 0;

while(steps < 4){
  moveForward();
  steps++;
};
```

- A counting variable set to an initial value

Say: A While Loop has three distinct parts.



Click for animation: Read each step as the animation highlights the program code.

Unit 5 Lesson 5 - Activity

A for loop combines these three parts into one statement

```
var steps = 0;

while(steps < 4){
  moveForward();
  steps++;
};
```

```
for (var i=0; i<4; i++){
  moveForward();
};
```

A loop is an example of **iteration**: a repetitive portion of an algorithm which repeats a specified number of times or until a given condition is met.

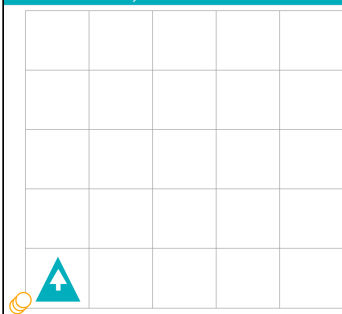
Say: A For Loop combines all three of these parts into one statement.

Click for animation: Read each step as the animation highlights the program code.

Say: A loop is an example of iteration: a repetitive portion of an algorithm which repeats a specified number of times or until a given condition is met.

Slides

Unit 5 Lesson 5 - Activity



Do this:

- You have another new command:

```
turnLeft()
```

- This command turns the robot to the left 90 degrees.
- Click to see a new program. Run the program on your board. Where will the robot end up?

Speaker Notes

Say: You have another new command
`turnLeft()`



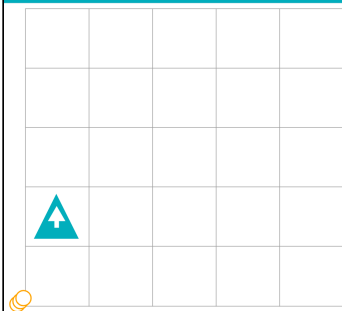
Click to reveal the program code

Do This: With your partner, follow the code with your robot.



Click through the animation to see the answer

Unit 5 Lesson 5 - Activity



Do this:

- Click to see a new program. Run the program on your board. Where will the robot end up?



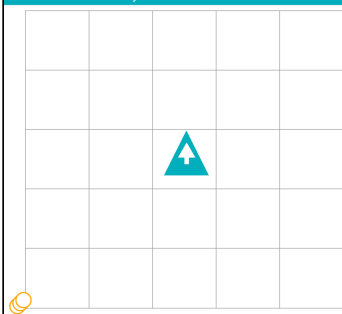
Click to reveal the program code

Do This: Run the program on your board. Where will the robot end up?



Click through the animation to see the answer

Unit 5 Lesson 5 - Activity



Do this:

- Click to see a new program. Run the program on your board. Where will the robot end up?



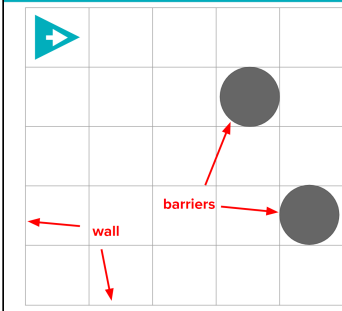
Click to reveal the program code

Do This: Run the program on your board. Where will the robot end up?



Click through the animation to see the answer

Unit 5 Lesson 5 - Activity



Let's make it a little more challenging.

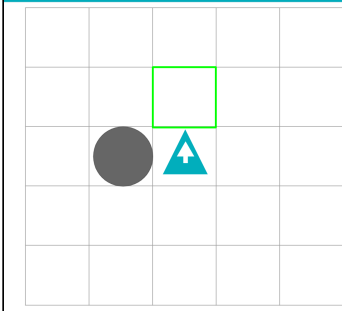
New commands:

```
canMove(left);  
canMove(right);  
canMove(forward);  
canMove(backward);
```

These evaluate to **True** or **False**. They are dependent on the direction that the robot is facing. If a barrier or a wall is in the way, the Boolean expression evaluates to False.

Say: We're going to add some new commands to make things more challenging: `canMove(left)`, `canMove(right)`, `canMove(forward)`, `canMove(backward)`. These evaluate to true or false and are dependent on the direction that the robot is facing. If a barrier or a wall is in the way, the Boolean expression evaluates to False.

Unit 5 Lesson 5 - Activity



```
canMove(forward);
```

True



Click through the animation to see the answer

Slides

Speaker Notes

Unit 5 Lesson 5 - Activity

`canMove(forward);`
True

Click through the animation to see the answer

Unit 5 Lesson 5 - Activity

Do this:

- Run the following program. Where will the robot end up?

`i = 0`

Click to reveal the program code

Do This: Run the program on your board. Where will the robot end up?

Click through the animation to see the answer

Unit 5 Lesson 5 - Activity

Do this:

- Run the following program. Where will the robot end up?

Click to reveal the program code

Do This: Run the program on your board. Where will the robot end up?

Click through the animation to see the answer

Unit 5 Lesson 5 - Activity

Challenge!

- Set up a game board. Add as many barriers as you'd like.
- Write a program using a for loop to navigate the board. Figure out the starting and ending points of the robot.
- Share your board and code with another group. See if you agree on the ending point of the robot!

For loop:

```
for(var i=0; i<3; i++){
  ... do something
}
```

Can be any number. Your choice!

Commands:

```
canMove(left)
canMove(right)
canMove(forward)
canMove(backward)
moveForward()
turnRight()
turnLeft()
```

Say: Now it's your turn to write your own programs!

Do This: With a partner, set up a game board and add as many barriers as you'd like. Write a program using a for loop to navigate the board. Figure out the starting and ending points of the robot. Share your board and code with another group. See if you agree on the ending point of the robot!

Unit 5 Lesson 5 - Activity

Key Takeaways

- While Loop**
Uses a boolean condition to repeatedly run a block of code. If it is true it runs the block of code contained within it. This process of checking the condition and running the block of code is repeated as long as the Boolean condition remains true. Once the Boolean expression becomes false it will stop.
- For Loop**
Condenses the parts of a while loop into a shorter statement. Similar to the while loop, once the Boolean expression becomes false, the loop ends.

```
var count = 0;
while(count < 3){
  ... do something
  count++;
}

for(var i=0; i<3; i++){
  ... do something
}
```

Do This: Review the key takeaways with students.

Wrap Up (10 minutes)

Introduction to Loops

 **Video:** As a class watch the video on Loops.

 **Journal:** Have students add to their journals: iteration and infinite loop.

 1

Introduction to Loops

Assessment: Check For Understanding

Check For Understanding Question(s) and solutions can be found in each lesson on Code Studio. These questions can be used for an exit ticket.

Question: When breaking a problem down, you often encounter elements that you want to use repeatedly in your code. Sometimes it's appropriate to write a new function; at other times it's appropriate to write a loop.

There is no hard-and-fast rule as to which is better, but what do you think? What kinds of circumstances would lead you to writing a function versus using a loop?

 2

 Check For Understanding