TOKYMAKER LESSON

LOGIC

LESSON OVERVIEW

Students will complete the logic code tutorias in the Tokymaker Workbook to learn how to use conditional statements like [IF/DO] and [IF/DO/ELSE] to program rules and choices into their games.
Booleans are a data type that students will learn about in this course. In this stage, students will learn about Boolean (true/false) values, and explore how they can be used to evaluate logical questions.

Essential Question: How would you use logic to add rules and choices to your game?

LESSON TAGS

GRADE LEVEL	SUBJECTS	DIFFICULTY	DURATION
middle school	STEM/STEAM, engineering, computer science	beginner	70 minutes
PREREQUISITE KNOWLEDGE	SUPPLIES	ASSESSMENT	RESOURCES
Hello World: Knows how to use Tokymaker, code blocks in sequence and upload code from the website. Tutorial: Inputs & Outputs	Starter kit (1 kit per group of 2-3 students) MacBook OSX, or Chromebook, or Android Tablet or Android Smartphone with version 6.0 up. Internet	Use this guide to assess students' understanding of the concepts taught in the tutorials.	Tokymaker Workbook Debugging Checklist Logic Answer Key (included at the end)

STANDARDS

CSTA

1B-A-3-7: Construct and execute an algorithm (set of step-by-step instructions) which includes sequencing, loops, and conditionals to accomplish a task, both independently and collaboratively, with or without a computing device.

1B-D-4-14: Use numeric values to represent non-numeric ideas in the computer (binary, ASCII, pixel attributes such as RGB).

1B-A-5-4: Construct programs, in order to solve a problem or for creative expression, that include sequencing, events, loops, conditionals, parallelism, and variables, using a block-based visual programming language or text-based language, both independently and collaboratively (e.g., pair programming).

2-A-5-6: Develop programs, both independently and collaboratively, that include sequences with nested loops and multiple branches.

Common Core Math Standards

5.OA.1 - Use parentheses, brackets, or braces in numerical expressions, and evaluate expressions with these symbols.

5.OA.2 - Write simple expressions that record calculations with numbers, and interpret numerical expressions without evaluating them. For example, express the calculation “add 8 and 7, then multiply by 2” as 2 × (8 + 7). Recognize that 3 × (18932 + 921) is three times as large as 18932 + 921, without having to calculate the indicated sum or product.

6.NS.8 - Solve real-world and mathematical problems by graphing points in all four quadrants of the coordinate plane. Include use of coordinates and absolute value to find distances between points with the same first coordinate or the same second coordinate.

6.EE.9 - Use variables to represent two quantities in a real-world problem that change in relationship to one another; write an equation to express one quantity, thought of as the dependent variable, in terms of the other quantity, thought of as the independent variable. Analyze the relationship between the dependent and independent variables using graphs and tables, and relate these to the equation. For example, in a problem involving motion at constant speed, list and graph ordered pairs of distances and times, and write the equation $d = 65t$ to represent the relationship between distance and time.

7.EE.4 - Use variables to represent quantities in a real-world or mathematical problem, and construct simple equations and inequalities to solve problems by reasoning about the quantities.

8.F.1 - Understand that a function is a rule that assigns to each input exactly one output. The graph of a function is the set of ordered pairs consisting of an input and the corresponding output.1

8.F.2 - Compare properties of two functions each represented in a different way (algebraically, graphically, numerically in tables, or by verbal descriptions). For example, given a linear function represented by a table of values and a linear function represented by an algebraic expression, determine which function has the greater rate of change.

A.REI.3 - Solve linear equations and inequalities in one variable, including equations with coefficients represented by letters.

A.REI.10 - Understand that the graph of an equation in two variables is the set of all its solutions plotted in the coordinate plane, often forming a curve (which could be a line).

Common Core Math Practices

MP.1 - Make sense of problems and persevere in solving them.

MP.2 - Reason abstractly and quantitatively.

MP.3 - Construct viable arguments and critique the reasoning of others.

MP.4 - Model with mathematics.

MP.5 - Use appropriate tools strategically.

MP.6 - Attend to precision.

MP.7 - Look for and make use of structure.

MP.8 - Look for and express regularity in repeated reasoning.

VOCABULARY

logic
IF/DO/ELSE
AND/OR
signal

INSTRUCTIONAL STEPS

STEP 1: SETUP - Prior to Class

Duration: 5 minutes

This lesson can be done individually or in small groups (2-3 students).

The tutorials are well explained and do not require teacher guidance to complete, so teachers can focus on supporting students when they need extra help.

Each group will need at least one Tokymaker, a compatible computer, plus one Tokymaker Workbook per student. Craft materials like cardboard and straws are recommended but not essential.

NOTES

- Set up a screen or a projector to watch the loops video as a class (Step 2: Introduce). Note that this video has sound. If you don't have access to a projector students can view on their laptops.
- [Optional] Print out the Logic Answer Key (see Appendix below). The answer key provides the end code for each tutorial and an explanation of how it works, block by block.
- [Optional] If you are new to code do at least one tutorial beforehand.

STEP 2: INTRODUCE - What is Logic?

Duration: 15 minutes

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we would have had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

Suggested group activity after the video ends:

- Write a sentence on your board that looks like this: IF______ DO_____ ELSE ________.
  Ask one student at a time to come up with a short sentence to fill in the blanks. For example, IF (student 1: it rains today) DO (students 2: take an umbrella) ELSE (student 3: if it doesn't rain, stay home). You may want to provide them with an example to get started.
- Discuss how making choices and explaining rules in real life is like using logic in code.

STEP 3: TUTORIALS – Let's program some rules!

Duration: 30 minutes

Students head to the tutorials on the Tokymaker Workbook and start completing the tutorials in the logic category in this order:

3.0 Choose Your Own Colour

Students will make a multiple choice game using colours. They will learn how to use an RGB LED, use IF/DO statements with a single input, as well as use math to add multiple choices to a game.
To complement the learning. Show the next video to explain what a pixel is and how it works. Find it in the website, in the Downloads section. http://tokylabs.com/project/tutorial-lesson-3-logic/

3.1 Retrieve The Hidden Treasure

Students will make a lock combination that retrieves a Treasure. They will learn how to use multiple sensors, the IF/DO/ELSE as well as the AND/OR conditional statements to create different lock combinations

STEP 4: SHARE - What Did You Learn? What Do You Need Help With?

Duration: 5 minutes

Ask students to complete the Logic section of their Tokymaker Workbook.

[Optional] Ask students to share on a post-it: One thing they learned and one thing they need help with. Post their reflections on a designated wall or board. Encourage them to read the Post-Its and use them as a tool to help each other during the next class.

STEP 5: CLOSE

Duration: 5 minutes

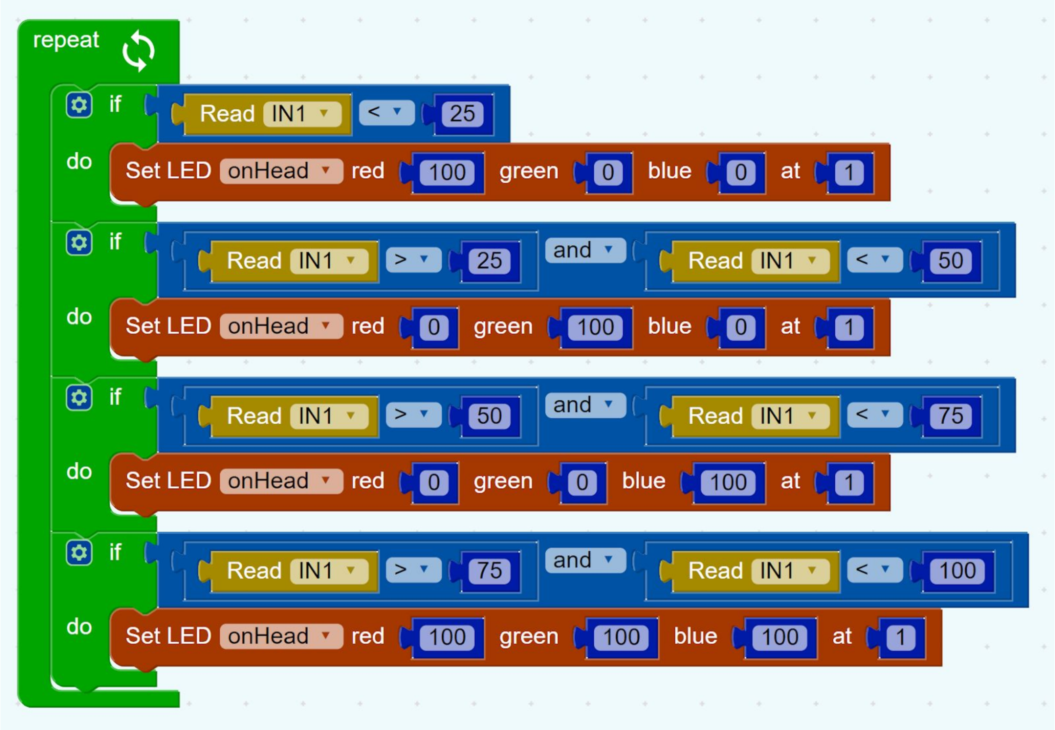
Students should take apart their circuits and put away the Bits according to the diagram on the back of the Bit Index. Students should clean up their workspace and close out their device (smartphone, tablet or laptop).

APPENDIX

CODE TUTORIALS | **LOGIC ANSWER KEY**

3.0 CHOOSE YOUR OWN COLOUR

FINAL CODE



```
repeat
  if  Read IN1 < 25
  do  Set LED onHead red 100 green 0 blue 0 at 1
  if  Read IN1 > 25 and Read IN1 < 50
  do  Set LED onHead red 0 green 100 blue 0 at 1
  if  Read IN1 > 50 and Read IN1 < 75
  do  Set LED onHead red 0 green 0 blue 100 at 1
  if  Read IN1 > 75 and Read IN1 < 100
  do  Set LED onHead red 100 green 100 blue 100 at 1
```

HOW IT WORKS

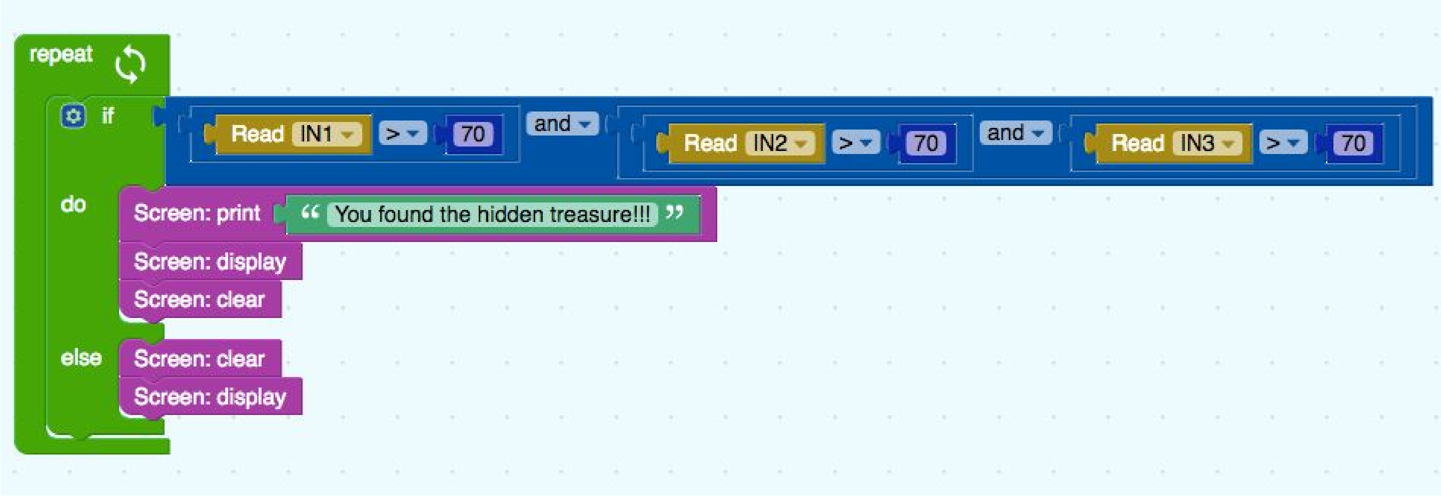
The green [REPEAT] loop block tells all the code inside it to keep running forever. Inside, there are 4 groups of similar code. Each of these groups of code are wrapped inside a green [IF/DO] logic block. This is the code that tells our circuit to show a different Led colour when the rotation sensor is in one of four positions.

Let’s take a closer look. As a condition of the green [IF/DO] logic block, the blue block [READ IN1 <25]. It basically selects the signal (which normally ranges from 0-100) that is lower than 25. Then, the code inside the green [IF/DO] logic block is executed.
We replicate the same action, dividing the range of the signal into 0-25, 25-50,50-75 and 75-100.

Make sure that the rotation sensor is connected into the IN 1 to modulate the Input Signal.

3.0 RETRIEVE THE HIDDEN TREASURE

FINAL CODE



```
repeat
  if  Read IN1 > 70  and  Read IN2 > 70  and  Read IN3 > 70
  do    Screen: print " You found the hidden treasure!!! "
        Screen: display
        Screen: clear
  else  Screen: clear
        Screen: display
```

HOW IT WORKS

The green [REPEAT] loop block tells all the code inside it to keep running forever.

ATTENTION: In order to make the green block ELSE appear, we must click the little blue wheel next to the “if” and drag the ELSE block under the IF block on the right, and then we obtain the whole block shown in the image.

ATTENTION: In order to be able to attach 3 sensors, we must use cables to connect the sensors to each input. We have to be careful to set the cables in the right position!

The green[IF/DO/ELSE] logic block tells our code to check IF signal from all three inputs is ON (we consider >70 out of 100 is ON, but you can change it based on your reasoning). If that is true, then the DO part tells our code to show the text in the Screen, via the [SCREEN PRINT + TEXT], [SCREEN DISPLAY] and [SCREEN CLEAR] blocks. If the inputs are not ON, then the ELSE part of the code makes all the pixels on the screen black, with the [SCREEN CLEAR] and [SCREEN DISPLAY] block.