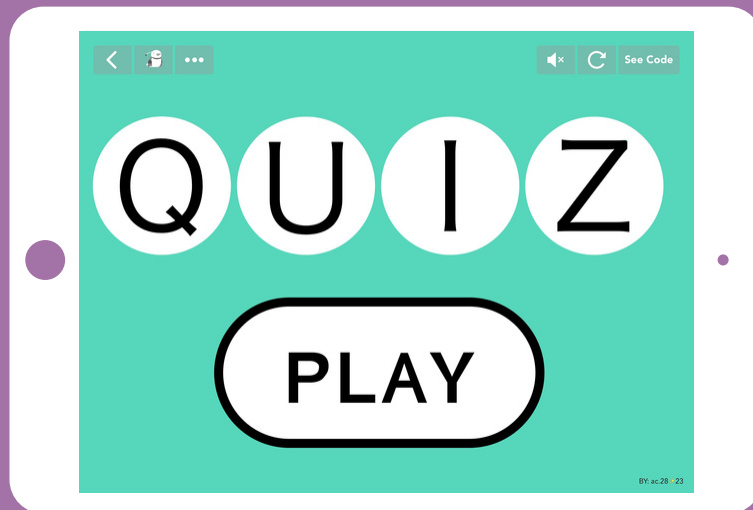


LESSON 3

WHICH EMOJI ARE YOU?



A personalized quiz that keeps track of your answers and calculates results.

TIME

45-60 minutes (+15 minutes of optional, free code time)

BIG IDEA

If you know how to use individual blocks like conditionals and variables, you can put them together in powerful ways to make what you want. Little blocks build big programs!

SKILL FOCUS

- Planning and Execution
- NGSS Practice 3 Planning and carrying out investigations
- NGSS Practice 4 Analyzing and interpreting data
- CCSS.MATH.PRACTICE.MP2 Reason abstractly and quantitatively
- CCSS.MATH.PRACTICE.MP8 Look for and express regularity in repeated reasoning

KEY VOCABULARY

Value/Variable: A number that can change

Conditional: Statement of the form “IF (something is true), THEN (do an action)”

TRANSFER GOALS

- Students will understand that a conditional checks if something is true and can identify the use of conditionals in their lives.
- Students will understand that a value or variable is a holder for a number and can identify the use of values in their lives.
- Students will become comfortable with keeping track of variables and doing operations.
- Students will make a plan and see it through until the end.
- Students will recognize patterns in their code and apply them to new situations.

MATERIALS

– 1 computer per student, or 1 device per 2

students, for pair programming

– Video available on YouTube:

<http://hop.sc/EmojiQuizVideo>

– Complete project available:

<http://hop.sc/emojiquizproject>

In this lesson, students will create an if-you-chose-mostly-A's type quiz, like you find in the back of a fashion magazine. In our example, it will help them answer an age-old question: "Which emoji are you?". With a little adjustment, it could be made into a factual quiz that checks the answers for correctness. It is very different from the last two games as it contains only text objects and revolves around logic and order, rather than animation and collisions. This lesson introduces two important computing concepts: **variables and conditional statements**.

In this lesson, the example quiz content is "Which emoji are you?". Students can stay close to the example quiz by choosing different emojis and questions. This is also an opportunity to tie in other content they are learning.




LESSON

0. Make a plan

The first step is to design the questions and answers for your quiz. Pass out copies of the blank matrix below, or put it on the projector and have students create their own on paper. Don't worry too much about making them the best questions ever; this is just your first quiz!

	Question 1	Question 2	Question 3
Result A	Answer 1A	Answer 2A	Answer 3A
Result B	Answer 1B	Answer 2B	Answer 3B
Result C	Answer 1C	Answer 2C	Answer 3C

Some ideas include: What job should you have? What fictional character are you most like? What book should you read next? If time is limited, or if a student has a hard time coming up with an idea, use the sample matrix below:

	What are you best at?	Where do you like to hang out?	Would you move to Mars if you could?
	Playing music	Library	No.
	Playing sports	Park	Yes!
	Telling jokes	Pool	Depends, do they have ice cream?

1. Setup and introduction to values (V)

Before coding, discuss how the quiz will work and the tools we will use to build it. The quiz has three main functions: First, it keeps track of what question you're on and advances the question number when the player chooses an answer. Second, it keeps track of how many of each answer you've chosen. Third, it compares the answers at the end to give the player a result. We will use **variables** to keep track of all this information.

Variables, also known as **values**, hold pieces of information. Variables can be set, changed, or checked. They can be used inside **events** and other blocks, and can stand in any place you can use a number. What makes values so powerful is that you can actually change them programmatically. For example, think of your score in a game like Angry Birds or the number of unread emails in your inbox. Both of these numbers are actually variables. As you score more points in Angry Birds your score goes up; the variable changes. As you read your email the number of unread emails you have goes down. Variables are used to represent some kind of information.

In Hopscotch, the yellow Variables tab pops up when it's time to input a number. In addition to the built-in variables like "device's width" or "Character's X position", you can add new values to your project to keep track of other things like a score or a state. To understand different types of variables, please visit: <https://help.gethopscotch.com/article/198-variables-in-hopscotch>

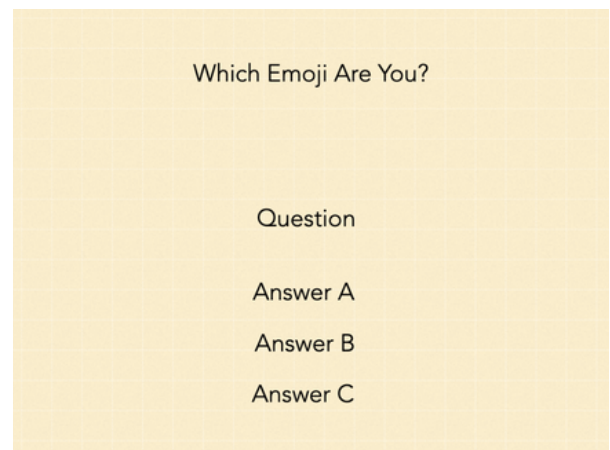
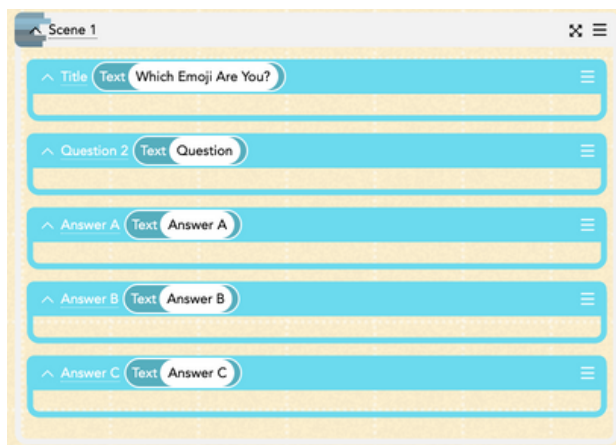
LESSON

It is super important to name your variables well! You'll need to be able to tell them apart later and remember what they're for! If you need to rename or delete a variable, press and hold down on it.

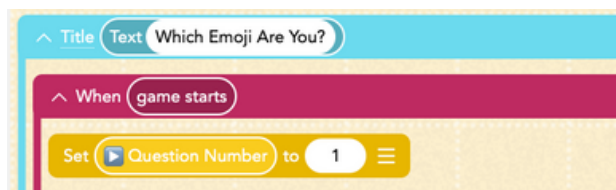
After discussing the concept of variables with your students, you might want to go through a list of objects that you will need and their respective initial rules (five text objects: a title, a question, and three answers.) The title (or any object, really) should initialize a value that will keep track of what question the player is on. In the example, we call it "Question Number". We create separate values that will keep track of each answer category. We initialize "Question Number" to 1 and the three "Answer" variables to 0.

1.1 Add text objects for the title, question, and three answers (5 objects) to Scene 1

Name each text object according to the role it will serve in the project, e.g. "Title" or "Answer A/B/C".



1.2 Add new code to the Title object: Initialize Question Number to 1



Make sure to create a new Game variable under "Game" tab since the question number applies to the entire game.

1.3 Edit the Title object's code: Initialize all three answer values to 0



Create a new game variable for each Answer (e.g. A Answers, B Answers, C Answers.)

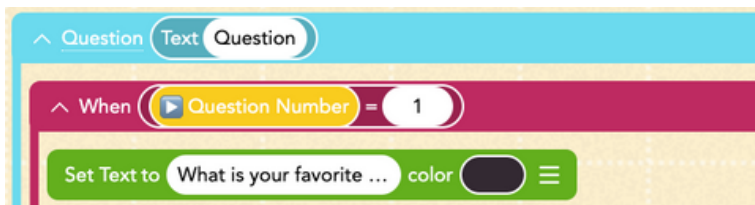
LESSON

2. Variable Events (EV)

The Conditionals at the very bottom of the Rules menu deal with comparing variables and values. They track whether a value is equal, not equal, bigger than, or smaller than another value or number. When you select the “_ equals _” event, the calculator menu opens. On the right of “Calculator” are variables.

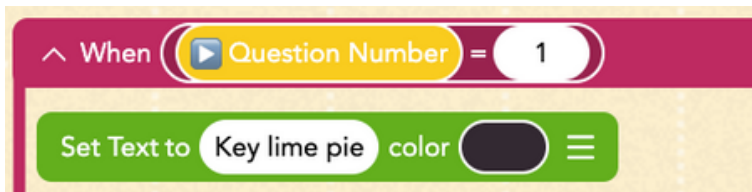
The quiz should display the first question and its answers when the “Question Number” value equals 1. The player chooses an answer and the game should keep track of how many of each kind of answer is chosen. We do that by increasing the appropriate answer value when each answer is tapped. For example, if an “A Answer” is chosen, the variable “A Answers” should increase. Show the students these new events, and then discuss the rules that they will need to code before having them start programming. Make sure they choose the right object-value combination!

2.1 Add new code to the question object



Scroll down on the panel to find the game variable "Question Number." Type the first question in the "Set Text" window.

2.2 Add new code to each of the answer objects



Refer to your question matrix to enter each answer in its respective rule. We will do questions 2 and 3 later.

2.3 Add new code to each of the answer objects: Increment the right score when tapped



Make sure to select the right text object in the rule's header for each answers tapped.

LESSON

3. Show Values (LV)

It's hard to know whether our values are working properly, so programmers design tests to display them. In Hopscotch, you can do this by displaying the variables on the screen. This is an advanced debugging strategy, and the variables should be deleted from the finished version of the project once you know everything works. Discuss this technique with students and see if they can figure out how to display a variable when their project is played.

We recommend using “Set Text” and putting the appropriate variable inside a forever loop to display it when the project is played. You should see the variables change as the player progresses through the quiz.

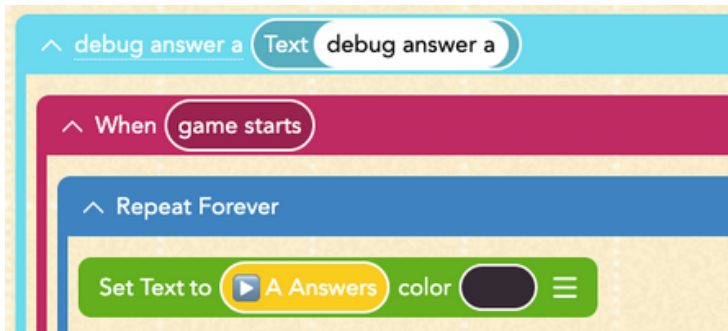
3.1 Add four new blank text objects. Place them next to the question and each of the answers.

3.2 Add new code to the question test object

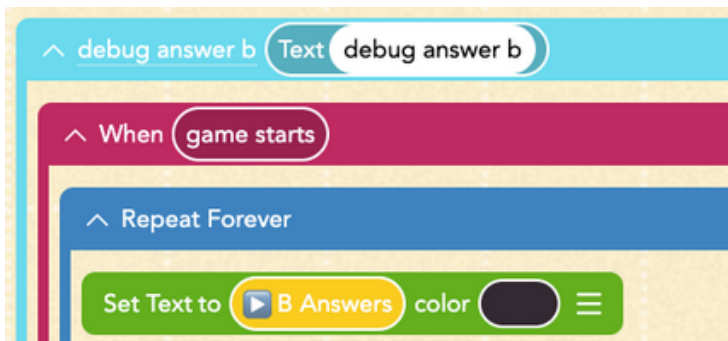


To set text to the variable Question Number, scroll down to the game section of the variables panel.

3.3 Add new code to Answer A test object

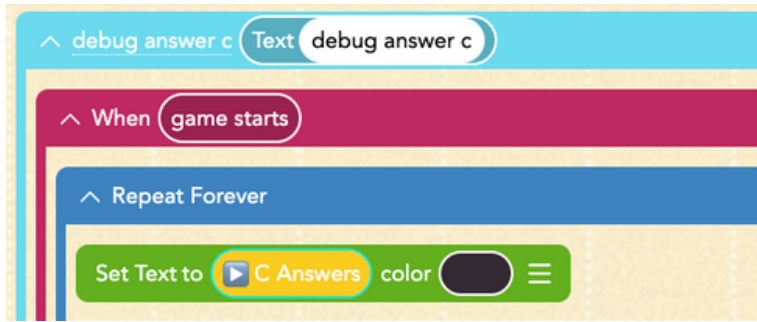


3.4 Add new code to Answer B test object



LESSON

3.5 Add new code to Answer C object

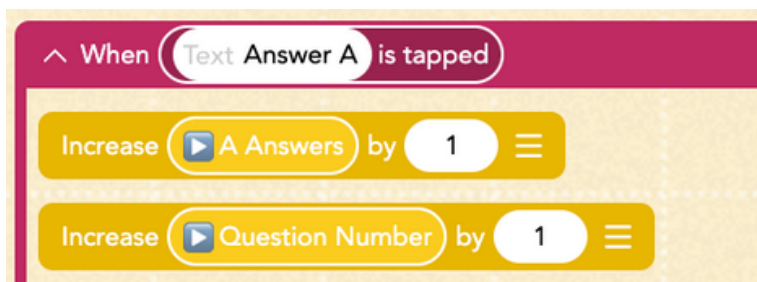


4. Next Question (EV)

The score for each of the answers advances as you tap them, but the question does not change (neither does the “Question Number” variable). Decide as a class, what event(s) should trigger the quiz to move on to the next questions. If we make “Question Number” increase, we also need to make the question (and answers) display the correct text for when the “Question Number” value equals 2 or 3. How should this work?

Hint: You advance “Question Number” when any of the answers is tapped! We have already made rules for this event, so we just need to add a block to increase the “Question Number” variable. You can do this step as a class and then together or in pairs test your programs to make sure the “Question Number” is advancing. Allow your students to implement the answer code in their own time and to help their neighbors if they finish early.

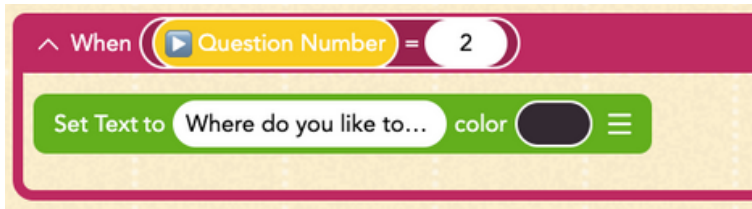
4.1 Edit code for all 3 answer objects to change “Question Number”



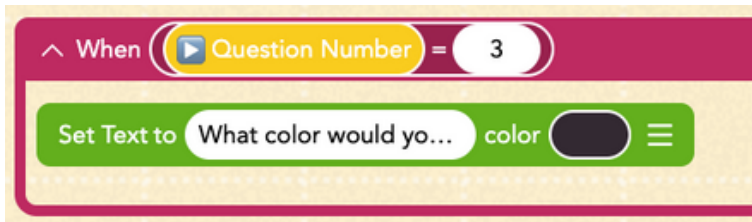
Add the same Increase block to Answers B and C.

LESSON

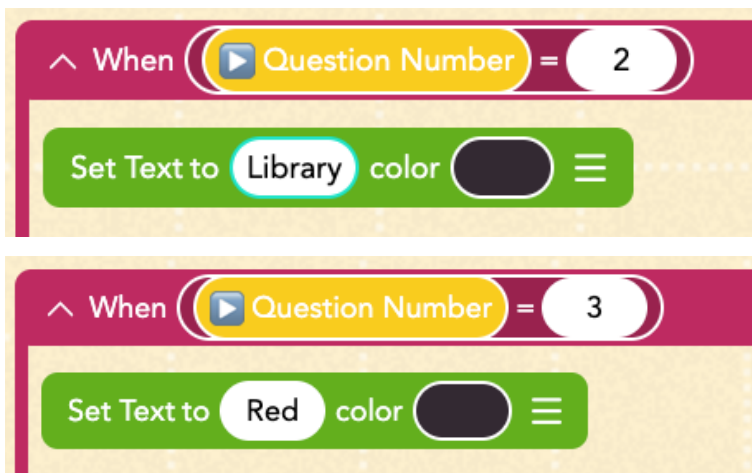
4.2 Add new code to Question object to display second question



4.3 Add new code to Question object to display third question



4.4 Add new code to each Answer object to display second and third answers



For Answer objects A, B, and C, add these same two rules with their respective answers. Refer to your matrix for the text of the answers. In this step, you're creating a total of 6 new rules!

5. Results and introduction to Conditionals (VC)

Now that all the questions are answered, we have to check our results! The correct result to show is the one for which the “Answers value” is the greatest. We can use **conditionals** to compare variables.

Conditionals are statements of the form “IF (something is true) THEN (do an action)”. It executes code only under the *condition* that you specify, like IF the score is greater than 10 or IF a character is invisible. We use conditionals in our lives all the time. Stoplights are a great example of conditions: “If the light is green, then go.” They are also extremely important in programming since computers need explicit directions.

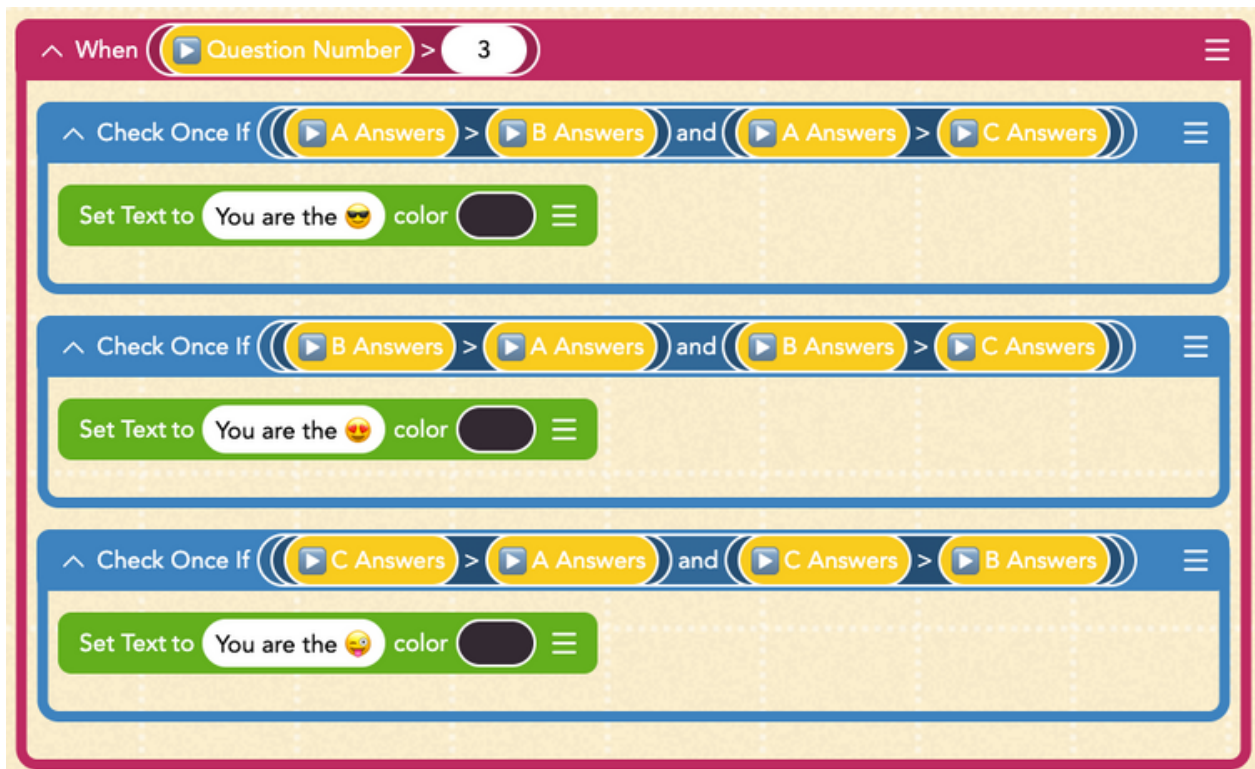
There are two types of conditionals in Hopscotch: “Check Once If” and “Check If... Else”.

LESSON

“Check Once If” is used when there are only two possibilities: 1. If the condition is true, do something, and 2. If the condition is not true, move on. “Check Once...Else”, by contrast, lets you check two things before moving on: 1. If something is true, do something, and 2. If this thing is NOT true, do something else, then move on. For example, say you can go in the ocean only if you’re wearing sunscreen or else you will have to put some on first. If you are wearing sunscreen (the condition is true), then you may go into the ocean. If you are not wearing sunscreen (the condition is false), then you must put some on first.

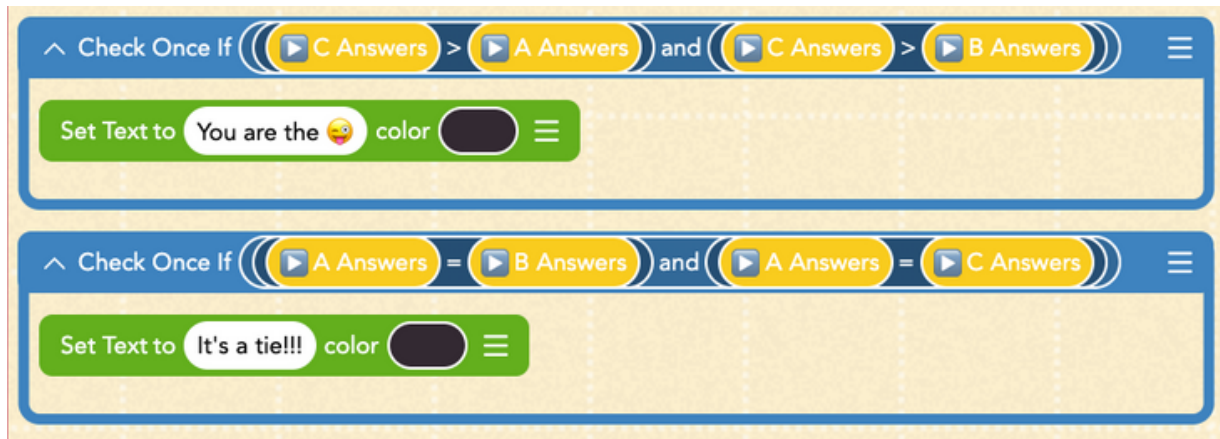
There are several ways to determine the answer to the quiz. The safest way is the one that won’t have to be changed if more questions are added: A is the winner if the “A answers” value is greater than *both* the “B answers” and the “C answers” values (and likewise for B and C.) In Hopscotch, we check if two things are true at the same time by using the “and” expression under Conditionals after choosing Check Once If.

5.1 Add new code to the Title object



LESSON

5.2 Edit the Title object's code in case there's a tie (optional)



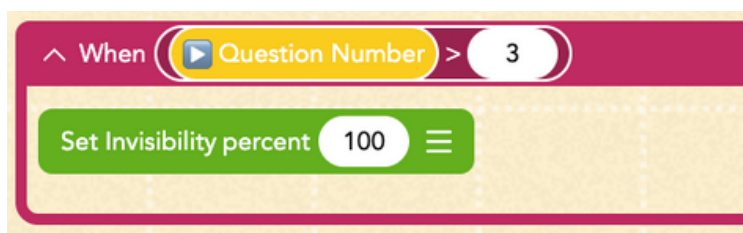
6. Tidying up (ES)

The hard part is done, and there's just a bit more work to do to make the ending look snazzy. One option is to center the results and hide the question and answers at the end. Your students may come up with better ideas!

6.1 Edit Title's code and add a Set Position rule



6.2 Add new code to the question and all three answers



Don't forget to delete your test objects!

DIFFERENTIATION

(15 minutes, optional)

- How would you change the quiz to have more answers or be longer?
- How could you change it to be a factual quiz—to change up whether the correct answer is A, B, or C?
- Turn the quiz into a survey that keeps track of how lots of people have answered—use this data to make a histogram.

REFLECTION

(5 minutes, optional)

- What is a variable?
- What is a conditional?
- How are variables and conditionals related or used together?
- How are events related to conditionals? How do you make a custom event? (When play button is tapped, repeat forever, check once if ...)
- What happens if there's a tie? How do we use conditionals to detect a tie? (if $a=b=c$)

