



Music Blocks Lesson Plan

Circular Rhythm Maker

Age:

7-12 years

Lesson duration:

90 minutes

- Introduction: Back to basics: Rhythm (15m)
- Part 1: Rhythm Maker (10m)
- Break (5m)
- Part 2: Building a Rhythm Maker (20m)
- Performance/Critique (10m)

Number of students:

Up to 10.

Rationale:

Students will build upon the concept of events to build a tool for exploring rhythm..

Objectives:

Students will understand what is meant by broadcast in computation and how it can be used in music. Students will be able to utilize broadcast and events in programming interactive games.

LESSON

Introduction:

Begin by asking students to sit in a circle and explain that in today's lesson they are going to explore rhythm.

Go around the circle, assigning a number to each student, starting with 1, 2, 3...

Ask each student to come up with a percussion sound that they will make when their number is called. (A percussion sound is produced by striking an object against another object. A

percussion instrument is sounded by being struck or scraped by a beater or stick.) What are some examples of percussion instruments? A common percussion instrument is a drum.

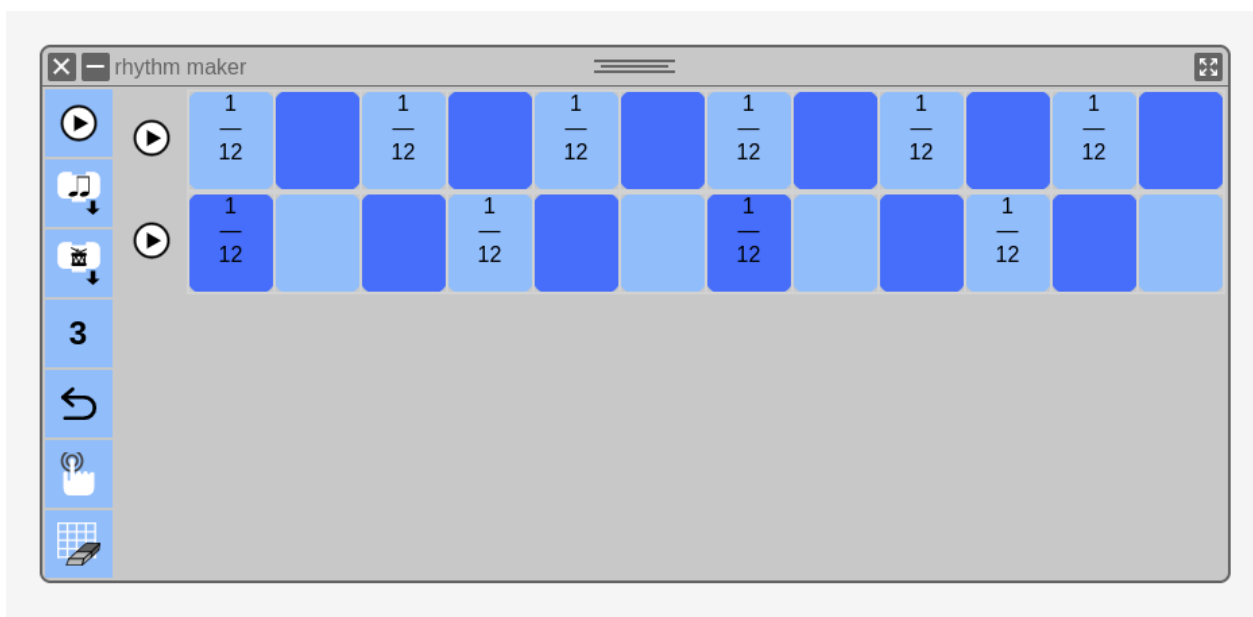
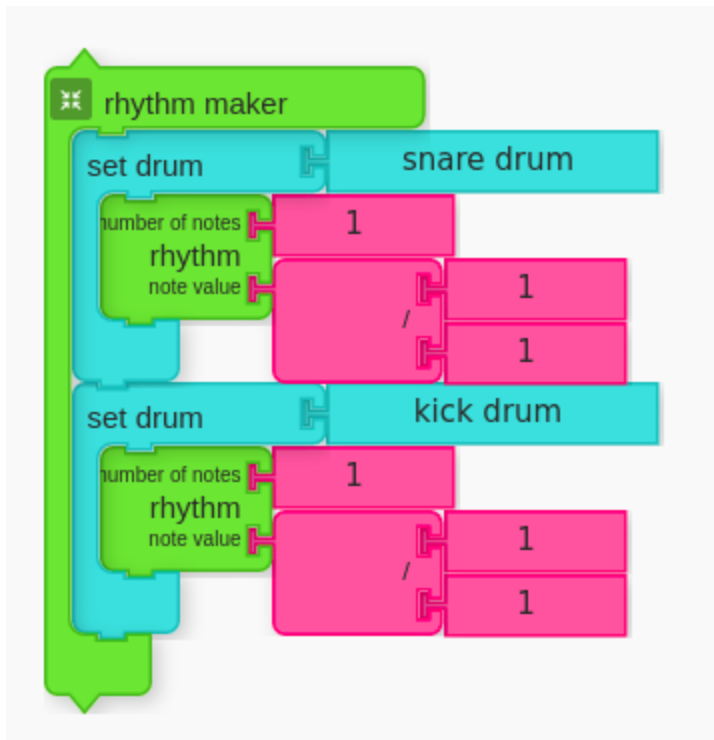
The leader ("conductor") will repeatedly count from 1 to n (the number of students in the class). When a student's number is called, they must make their percussion sound (or play their drum). (Repeat the exercise until the rhythm of sounds is consistent -- it takes some practice.)

Part 1:

Exploring the Rhythm Maker

Ask the students if they recall a way to make rhythmic patterns in Music Blocks. Some of them may recall the Rhythm Maker widget.

1. Drag a Rhythm Maker block from the Widget palette.
2. Explore designing some Rhythms.
3. Once everyone has a Rhythm that they like, have them share their creations and talk about why they chose their particular patterns.



Part 2:

Building a Rhythm Maker

We can make our own rhythm-maker in Music Blocks.

One way to approach it is to arrange some mice in a circle, where each mouse represents one beat in the rhythm. We will refer to these mice as the “drum” mice.

We need another mouse to travel around the circle and ask each mouse what beat to play. We will refer to that mouse as the “conductor” mouse.

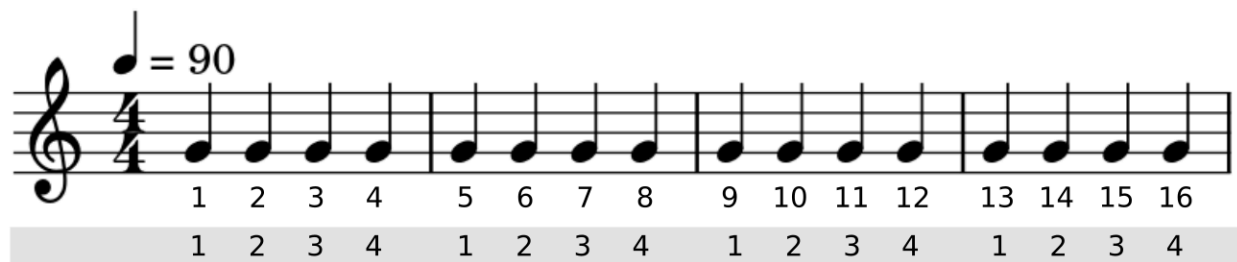
(This is similar to how we made our circle beat at the beginning of the lesson.)

To do this, we have several steps we need to take:

- (1) We need to decide how many drums to have in our circle (to start, we will use 12);
- (2) We need a way to position the drums in a circle (we will use the setxy block to do this);
- (3) We need a way to keep track of what drum sound each drum makes (we will use a box for this).

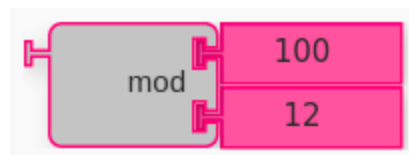
There is a tricky bit of math we need to do called modular arithmetic. It is a way of counting where at a certain point, you return back to one. For example, in mod 4, we'd only use four numbers: 0 1 2 3, 0 1 2 3, 0 1 2 3... Note that whenever we get to 4, we go back to 0. How would mod 10 work? Mod 2?

You can think about mod as being the same as counting beats in a measure. Even though we might play 16 notes in the example below, the beat count goes 1 2 3 4, 1 2 3 4...



There is one difference between modular arithmetic and counting beats. When we count beats we start counting at 1. When we count with modular arithmetic, we start at 0.

There is a block on the Numbers palette (only in advanced mode) that lets you calculate Mod.



In this example, we are using 12 to calculate mod, so the block will always return a number between 0 and 11.

Another “trick” we will be using also involves a block from the Numbers palette. You’ve probably used the plus (+) block to add numbers. But did you know that you can also use it to add words?

When we “add” words, we stick them together (or “concatenate” them).

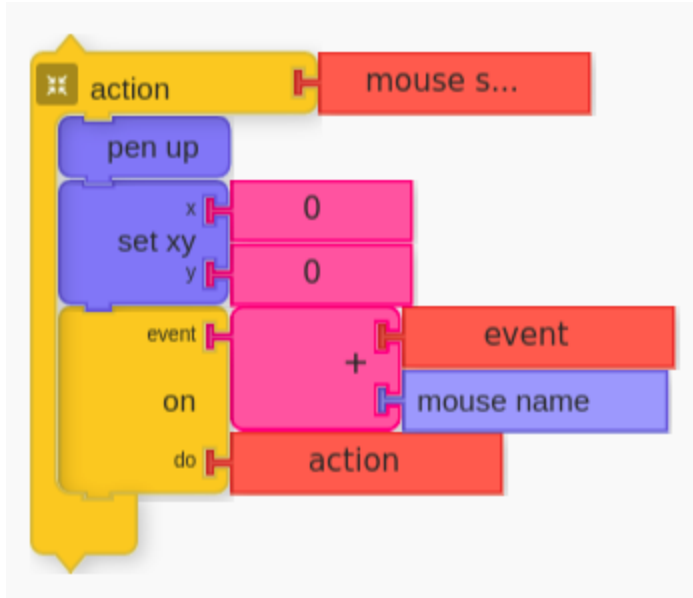
“Blue” + “Fish” = “BlueFish”

“Event” + “5” = “Event5”

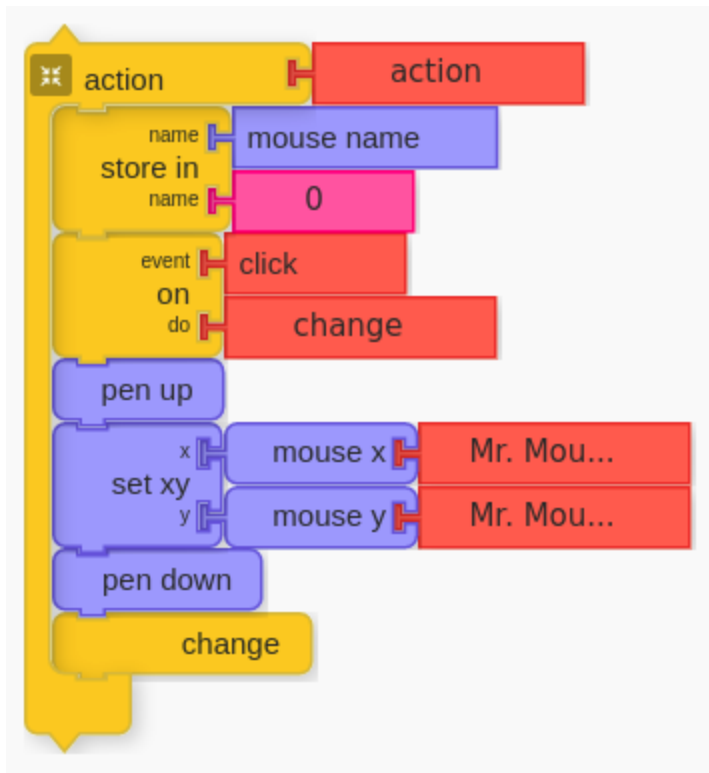
We will use this trick to create an event name unique to each drum mouse.



We need 12 drum mice, with names 0 through 11.



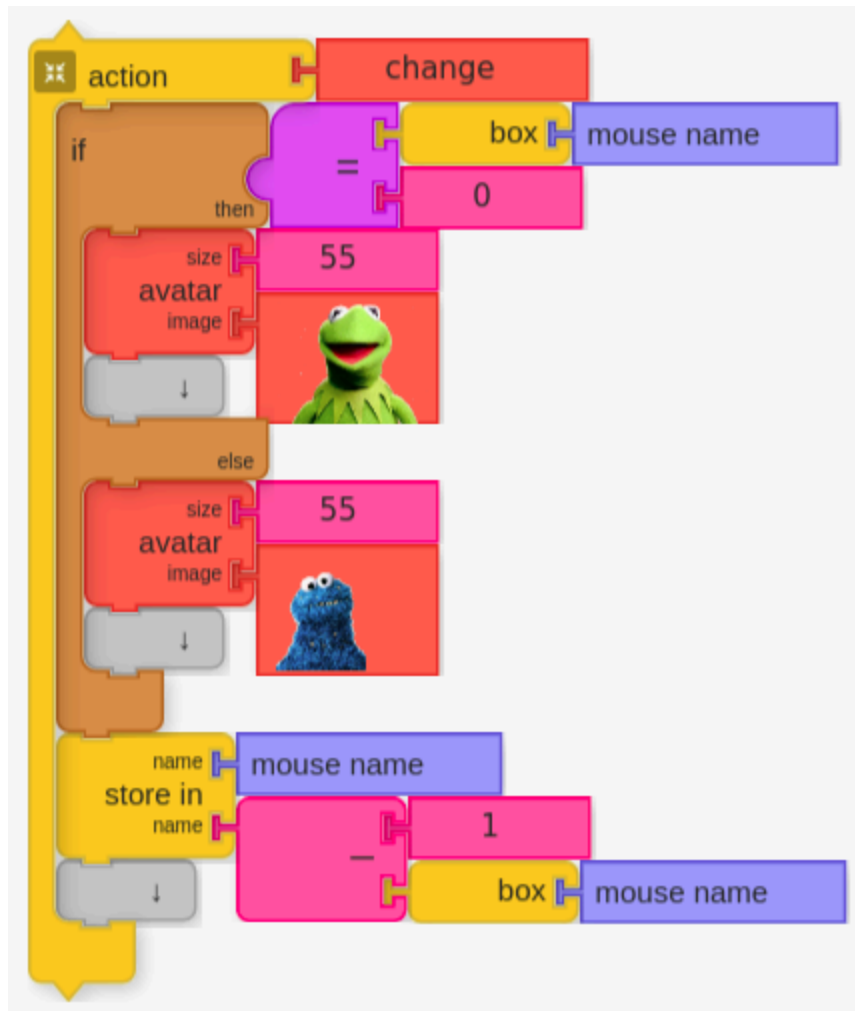
In the mouse setup action, we place each mouse at the center of the screen using the setxy block and then use plus (+) to create an event name for the mouse based on its name: event0, event1, ... event11.



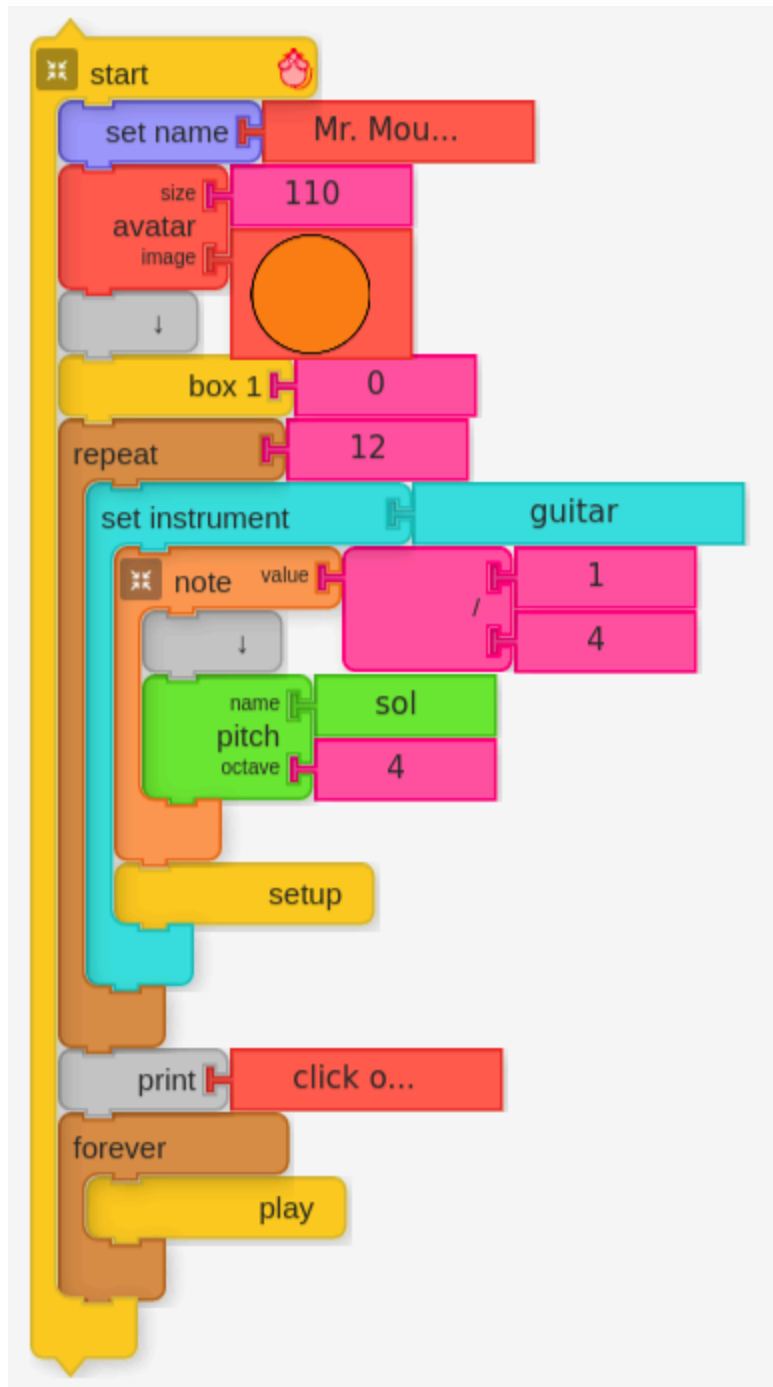
In the action named “action”, we do some more sneaky things:

(1) We create a box with the name of the mouse where we’ll store which drum this mouse plays.

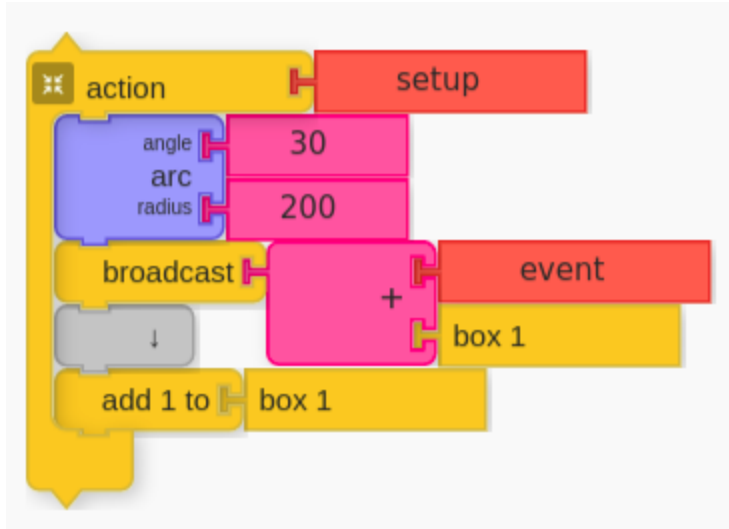
- (2) We also set up an event handler for “click”. (Whenever the mouse is clicked, we call the “change” action. The change action is where we decide which drum to play.)
- (3) We move the mouse using the setxy block to the current position of our conductor, “Mr. Mouse”. (You can find the “mouse x” and “mouse y” blocks on the Ensemble palette.) We’ll explain this in more detail later.
- (4) And we call the “change” action to set the initial value for the drum we’ll play.



The “change” action is used to change the drum value. We are storing either a 1 (for Kermit) or a 0 (for Cookie Monster) in the box named for each mouse. Whenever the mouse is clicked, we reset the avatar and change the value in the box (using a store-in block). We use another math trick here: We always put in the box 1 minus the number in the box. If there was a 0 on the box, $1 - 0 = 1$. So we end up with a 1 in the box. If there was a 1 in the box, then $1 - 1 = 0$. This is a simple way to switch back and forth between 0 and 1. Later on, the “conductor” mouse will look in the box to determine which drum sound to make: 0 for clap and 1 for ride bell.



The "conductor" has two roles to play: first to position each of the "drum" mice in a circle (in the setup action); then to walk around the circle to trigger the drum events (in the play action).

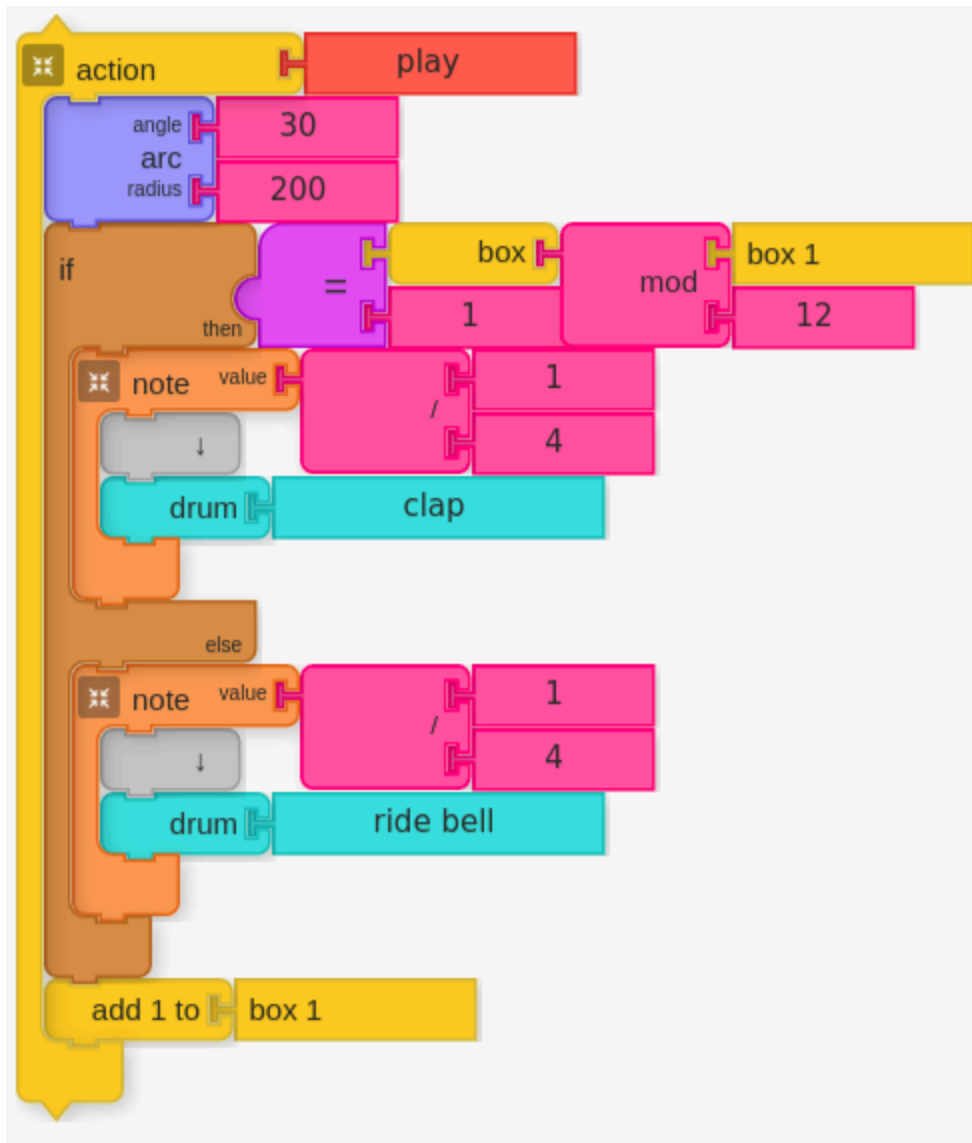


The setup action is pretty simple, but also sneaky. It is called 12 times (in a repeat 12 block). Each time the “conductor” mouse completes a journey 12th of the way around a circle (using the arc block). Remember that $30 \times 12 = 360$, so repeating arc 30 twelve times will go all of the way around the circle once.

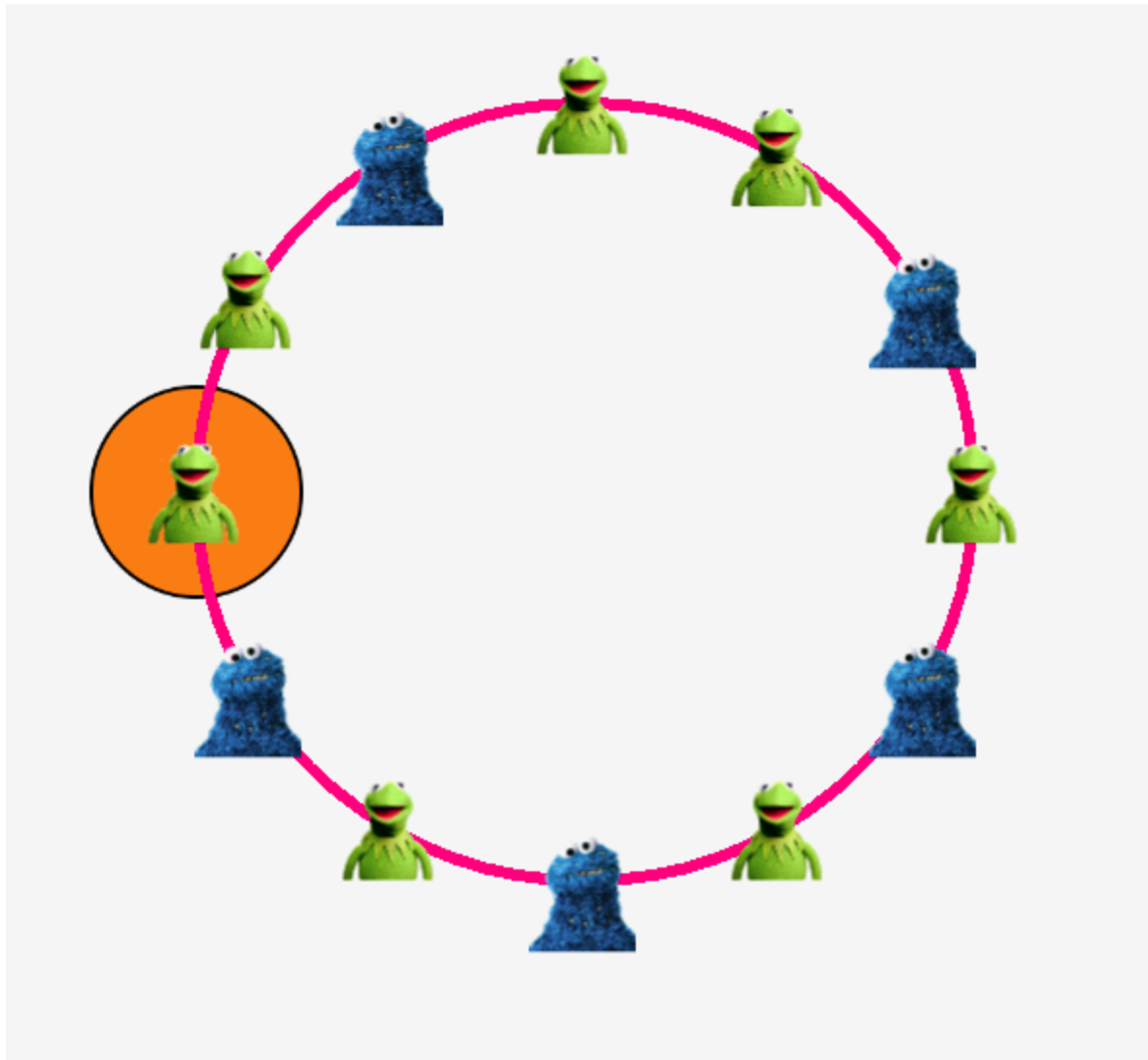
We use box 1 to count. We start at 0 and count up to 11, adding one each time. Using the plus (+) block, we create an event name to broadcast after each step: “event0”, “event1”, “event2”,...

Remember that the “drum” mice are listening for the event that corresponds to their name. Drum mouse 0 is listening for “event0”. Drum mouse 1 is listening for “event1” etc.

Here is the sneaky part: the drum mouse will move to wherever the “conductor” mouse is when it hears its event. So as the “conductor” mouse goes around the circle, the “drum” mice jump into position.



The “play” action is pretty simple. The “conductor” mouse continues to go around the circle and continues to add 1 to the box each time. This is where we use the modular function. We want to trigger one “drum” mouse at a time. So we take the value in box 1, mod 12, and use that as the name of the “drum” mouse we are interested in. We look in that mouse’s box to see which drum to play. Since we are in a forever loop we keep adding 1 and moving to the next mouse.



Variations

Can you come up with dances for the drum mice?

<https://musicblocks.sugarlabs.org/index.html?id=1591015346593413> is a version of the rhythm circle that uses an Input block to set the number of drums.

Performance/Critique:

1. Have each student perform their variation of the rhythm circle.
2. Engage in a discussion about more things they could do with broadcast and events.

Key events:

- Introduction of key concept: broadcast
- The students create their own programs using events.

Materials:

- Music Blocks software

Assessment:

- Observe participation.
- Do the programs include creative use of events? Of rhythms?



© Copyright Walter Bender, 2020