Lesson 4.4: Roll the Dice

Objectives

In this lesson, students will:

- Learn about broadcasting with wait and how to implement it in Scratch
- Recognize how to apply a concept learned to a programming exercise
- Be introduced to program timing

Agenda

1. How Sprites Communicate: The Broadcast	10 mins
2. Student Activity: Roll the Dice - Part 1 (and review)	20 mins
3. Student Activity: Roll the Dice - Part 1 (and review)	15 mins
4. Wrap Up and Reflections	5-10 mins

Preparation

- ❑ Remix and view the solution project to both demonstrate it and become familiar with it.
- ❑ Print student activity worksheet, one per student pair.

Resources & Links

- ❑ Starter project: https://scratch.mit.edu/projects/281515595
- ❑ Solution project: https://scratch.mit.edu/projects/281200465

© Code for fun
Released under the Creative Commons By Attribution license

1. How Sprites Communicate: The Broadcast and Wait

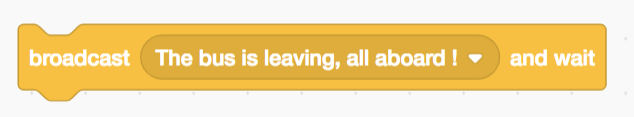
Engage students in an interactive discussion and instruction on broadcasting.

Ask students how sprites communicate with each other? Through a broadcast message. Review the meaning of a broadcast message if necessary: A message that is sent from one party for anybody to listen to.

An example of a broadcast message in our daily lives could be a performer shouting to a crowd: “Hello everybody!” . You can choose to say hello back or simply go on with what you were doing. Likewise, the performer goes on to start singing or get ready for their performance.

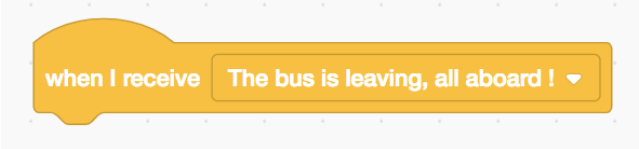
Another example could be a bus getting ready to leave to take kids to an outing. The driver yells: “The bus is leaving, all aboard”. In this case, the driver is not going to just take off after broadcasting the message and leave kids behind, right? The driver will wait until all kids are on the bus and then start driving.

In Scratch, we use the following broadcast block when we need to wait for other sprites to finish their code when they listen to our broadcast message.



How do we listen to the broadcast message?

In the same Event category we can find the following block to listen and react to the message broadcasted:



2. Student Activity: Roll the Dice - Part 1

It is recommended that students work in pairs for this activity. Students will create the first half, section 1 through 4, of the roll the dice project.

© Code for fun
Released under the Creative Commons By Attribution license

Explain the activity to students.

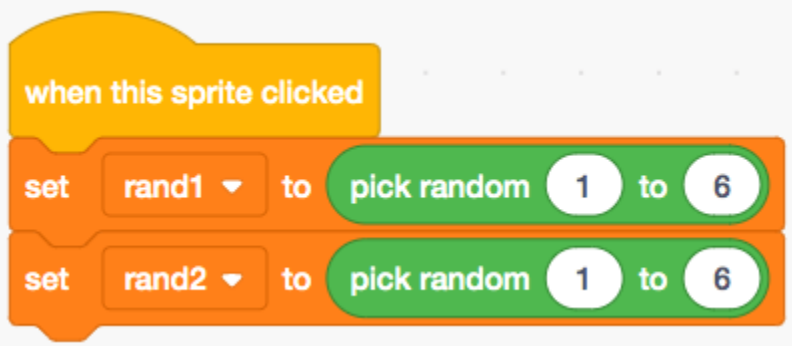
In this activity students are guided through the steps of what to do, but not specifically how to code it. Remind them of the rule of 3:
If you get stuck, use the rule of 3: 1) Discuss with your partner 2) Ask another group 3) Ask your teacher

Review Student Activity Part 1.

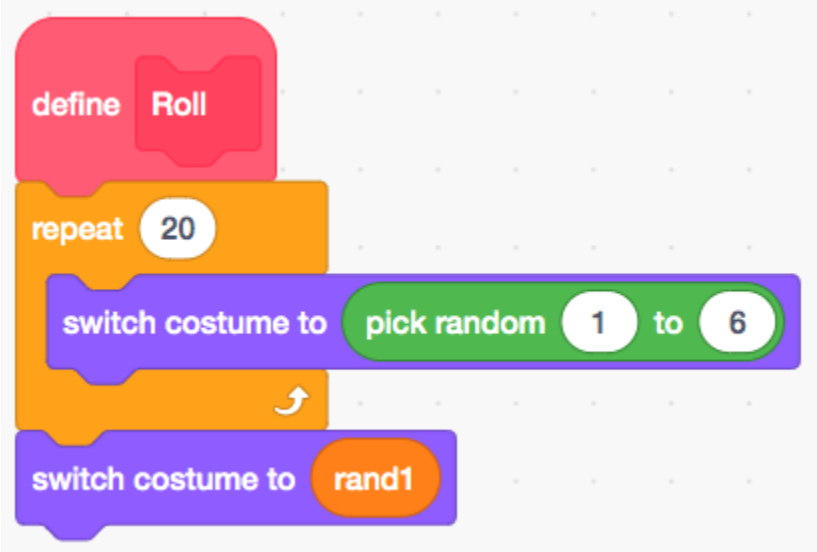
Regroup with everybody and optionally go over the solution for part 1. Ideally students can answer questions to help everyone catch up.

Solution for Part 1

ClickMe Sprite:



Die1 Sprite:



© Code for fun
Released under the Creative Commons By Attribution license

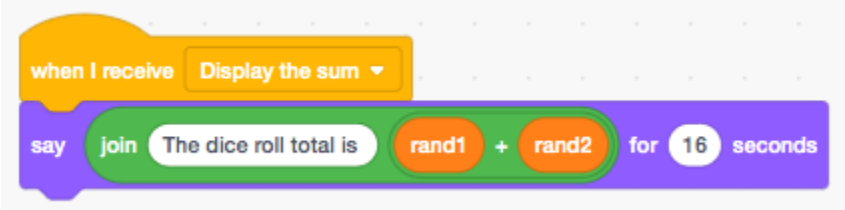
3. Student Activity: Roll the Dice - Part 2

Instruct students to work on Part 2. In part 2, students require the use of the broadcast blocks.

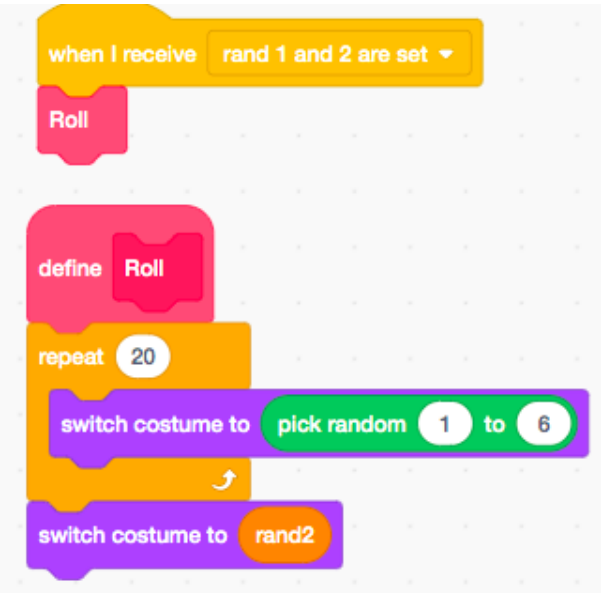
Following the activity, review the solution code with students. It is helpful to have students explain their code or their solutions first.

Solution for Part 2

Owl Script:



Die2 Script:



ClickMe Script:

© Code for fun
Released under the Creative Commons By Attribution license

4. Wrap Up and Reflections

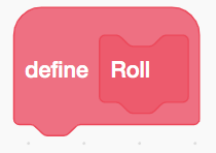
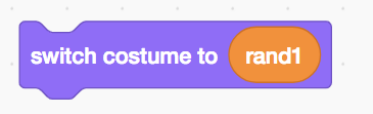
Reflection Points:

- What does the Scratch broadcast message do?
- How is it different from the broadcast message and wait ?
- How did you simulate the roll of the dice in your project?
- What did you use to assign a random number to the dice?

© Code for fun
Released under the Creative Commons By Attribution license

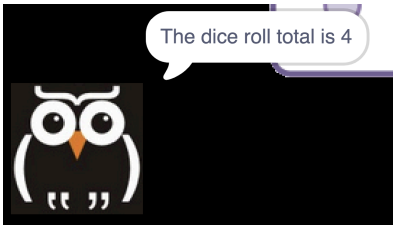
Student Activity Worksheet: Roll the Dice

Part 1:

What to do:	Using/Details:
Remix and save	281515595
Create 2 variables called ***rand1*** and ***rand2***. Uncheck them so that they don't show on the stage.	
Write a script that does the following: When the sprite is clicked, assign a random number between 1 and 6 to ***rand1*** and ***rand2***	
We want the dice to look like it is rolling when someone clicks on ***Roll the dice***. To do that we change the costume for each die 20 times. Check out the costumes for **Die1** and **Die2**. Write the script for **Die1**: Define a procedure called ***Roll***. The ***Roll*** procedure should: 1. Switch the Die1 costume to the next **random** costume 20 times. 2. Change the costume to the value of ***rand1*** value. Do the same for **Die2**, but change the costume to ***rand2***.	 
Test your code to make sure everything is working	

© Code for fun
Released under the Creative Commons By Attribution license

Part 2:

What to do:	Using/Details:
• When the **ClickMe** sprite is done setting the ***rand1*** and ***rand2*** values, it lets other sprites know the values are set and waits. • When the ***rand1*** and ***rand2*** values set, ***Die1*** and ***Die2*** roll the dice	  
The owl script will add the values of the 2 dice and display a message giving what the sum is. Here is the pseudocode: • When the owl sprite receives a broadcast message to display the sum • Display a message that says "The dice roll total is " • Display the sum of rand1 and rand2	
Extended Activity	
It would be nicer if the sum of the dice is displayed as a single message for a few seconds. Rewrite the owl script say block as shown to the right. **Hint**: Explore the Operator category for an operator that can combine 2 other operators, one for the text and one for the sum operator.	

© Code for fun
Released under the Creative Commons By Attribution license