

Lesson 2.5: Fraction Slider - Part 2

Objectives

In this lesson, students will:

- ❖ Gain experience incrementally developing a program.
- ❖ Practice translating english instructions into code and incorporating math into a program
- ❖ Practice using events, operators, conditionals, variables.

Agenda

- | | |
|--|---------|
| 1. Student Activity: Coding the Fraction and Tally Marks | 20 mins |
| 2. Student Activity: Coding the Dragonfly | 20 mins |
| 3. Wrap Up and Reflections | 10 mins |

Preparation

- Projector for demonstration
- Print student activity worksheets (one student pair)

Resources & Links

- Solution project:
<https://scratch.mit.edu/projects/544000050>
- Starter Project:
<https://scratch.mit.edu/projects/288557906>

1. Student Activity: Creating the Fraction and Tally Marks



Demonstrate the Fraction Slider again to remind students what they are creating.

Point out to students that every time we run the project, we see a new fraction appear and the tally marks are drawn. How many tally marks are drawn? Connect the denominator number to the number of tally marks.

Ask students what we need in order to code the fraction. We are looking for an answer that involves variables.

We need a variable for the denominator and one for the numerator. Each is assigned a random number when the program runs. The denominator should not be more than 10 and we want our fractions to be between 0 and 1 (hence the numerator should never be greater than the denominator).



Students work with the same partner from the previous activity (Planning).

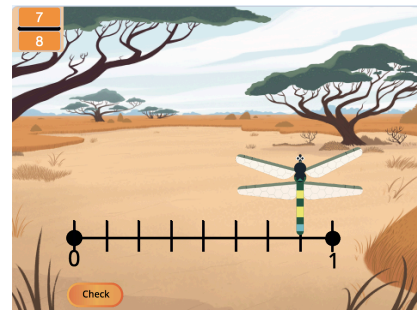
Instruct students to take turns being the navigator and driver during coding and testing.

Distribute the activity worksheet and instruct students to create the fraction and tally marks.

2. Student Activity: Coding the Dragonfly



Remind students that the goal is to move the dragonfly to the correct position to represent the fraction generated randomly by the program



When you run the project, demonstrate moving the dragonfly in both directions. Point out that you can not move it past the 1 line when moving right. Then point out that you cannot move left of the 0 line when moving to the left.


Distribute the student activity worksheet. Students continue to work with their same partner.

Solution:

Review the code with students if they had difficulties during the exercise. The comments explain the significant code sections (also available in the solution project).


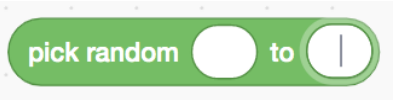



2. Wrap Up and Reflections


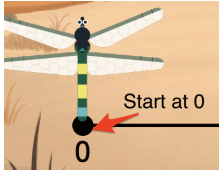
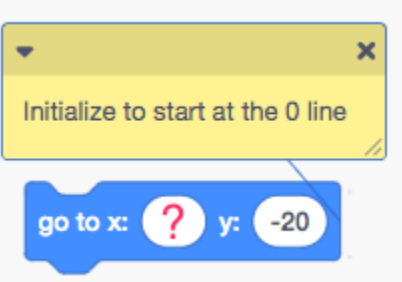
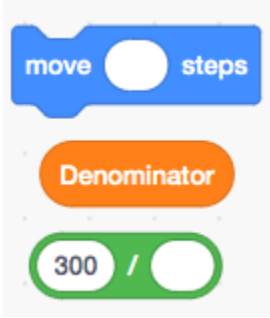
 **Reflection Points:**

- What did you learn today?
- What was challenging about today's activities?
- How has your understanding of fractions changed by coding this project?

Student Activity: Creating the Fraction and Tally Marks

What to do:	Using/Details:
<p>Remix and save</p>	<p>288557906</p>
<p>Initialize the fraction variables:</p> <p>Set the denominator to a random number between 1 and 10</p> <p>Set the numerator to a random number between 1 and ?</p> <p>What should the upper limit be? We only want fractions between 0 and 1 (not including 0), so the numerator should not be bigger than the denominator.</p>	 
<p>Test</p> <p>When the green flag is clicked, your code should create a new fraction.</p>	
<p>Draw the tally marks</p> <p>The fraction line is 300 steps long and remember the denominator number equals how many tally marks we need..</p>	<p>For every denominator number, repeat</p> <p>Stamp (draw a tally mark)</p> <p>Move (300/denominator) steps</p>
<p>Don't forget to snap all the code together. Lose code does not run when the green flag is clicked.</p> <p>Test and try until the code is working.</p>	

Student Activity Worksheet: Coding the Dragonfly

What to do:	Using/Details:
<p>In your fraction slider project create a Dragonfly sprite.</p>	
<p>When the program starts, initialize the position of the dragonfly so the tail tip is positioned at the 0 point.</p> <p>What is the value for x?</p>	 
<p>Each time the right arrow is clicked, the dragonfly moves 1 tally mark on the number line to the right.</p> <p>Snap these blocks together for the move script and add them to your event block.</p> <p>Hint: the move distance is the same as the distance between each tally mark.</p>	

Read the comment for this code.

Only move if not at the end, so put the move code from the previous step inside the **if-then** block.

We want to keep track of the dragonfly's **position**. This will come in handy when we want to know where the dragonfly is on the number line.

Each time the dragonfly moves right, increase the **position** variable by 1.

```

when right arrow key pressed
  if x position < 132 then
    change position by 1
  
```

When moving right, only move when the dragonfly's x position is less than the 1 line.
132 is the x position of 1 line

Now code the dragonfly moving left when the **left arrow is clicked**.

The code is similar to the right arrow script, however:

- remember to move left,
- the x position gets smaller
- we don't want to go beyond the 0 position which is at **-168**.

Each time the dragonfly moves left, we also want to keep track of it's position. When moving left, the **position** should be **decreased** by 1.

```

when left arrow key pressed
  
```