

# Exploring Candy Sharing in NetLogo

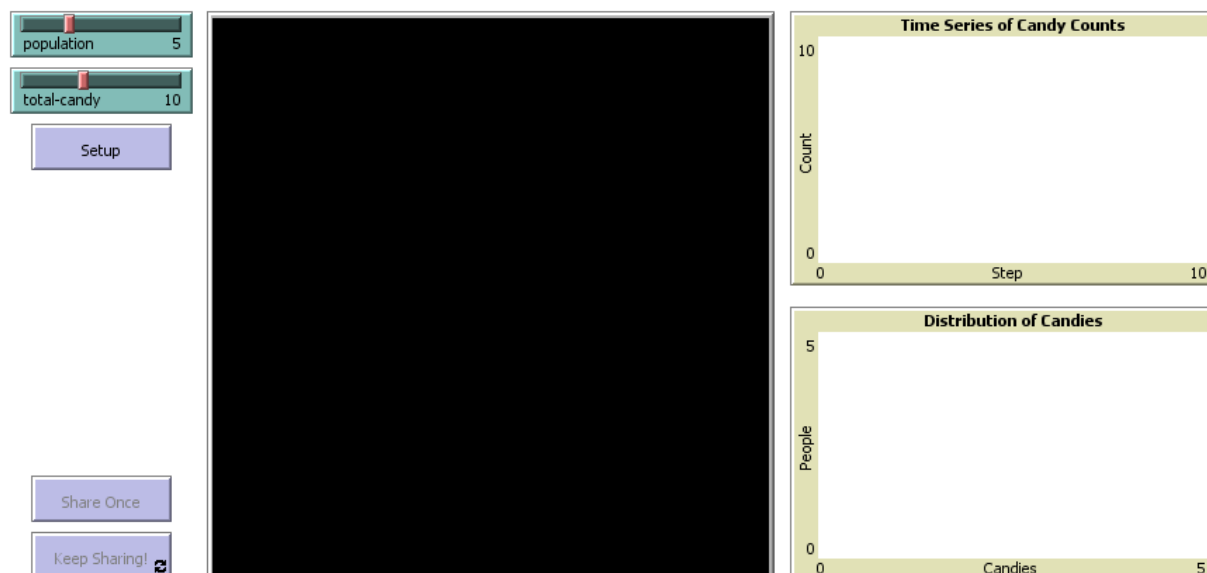
## Introduction

The Candy Sharing activity is one in which experiments are relatively easy to conduct. It's not too hard to assemble a group of participants to play the game, and each trial can be done in a short amount of time. However, there are practical limits on the experiments we can perform in this fashion. For example, what if we want to try the activity with 20 people, but we only have 10 available? Or even with 10 people, what if we want to test every possible initial distribution of 20 total pieces of candy among those 10? How feasible would such experiments be?

These are the sorts of questions that – along with other factors – sometimes lead us to build and use computational models. In this case, we'll use a models developed in NetLogo, an agent-based modeling and simulation tool, to conduct experiments that would be too difficult to do without such a tool.

## Interactive Investigation

Open the [Candy Sharing \(basic\).nlogo](#) model in NetLogo. Before you click any buttons, the model user interface will look something like this:



The **Setup** button initializes the model with the number of players specified in the **population** slider control, and the total amount of candy specified in **total-candy**. (Note that a change to either slider's position has no effect until **Setup** is pressed.)

Once **Setup** has been pressed, the sharing buttons are enabled. **Share Once** directs all players with at least 2 pieces of candy to share with their neighbors; **Keep Sharing!** does the same thing, but repeats this instruction indefinitely, until the button is pressed again to stop.

Take some time to familiarize yourself with the controls and plots in the user interface, to answer these questions:

1. Each player is represented by a colored circle, with the players themselves arranged around a circle, as shown in the Figure 1 example.

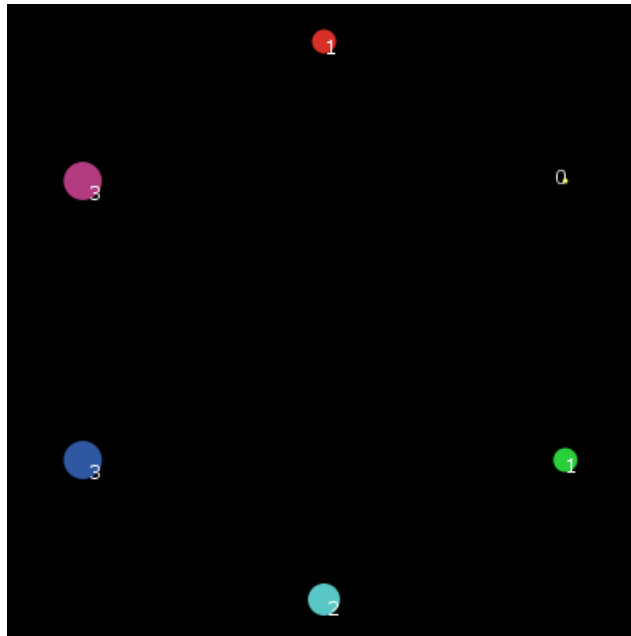


Figure 1: Example configuration after model initialization.

What do the sizes of the colored circles, and the numbers displayed near each, seem to represent?

2. How does it appear that the candy is being distributed initially? Does it seem to be done in a fair fashion? If not, how do you think the distribution should be done?
3. Does the sharing seem to follow the rules followed in the Math Circle activity?
4. What appears to be shown in the **Time Series of Candy Counts** and **Distribution of Candies** plots (e.g. Figure 2)?

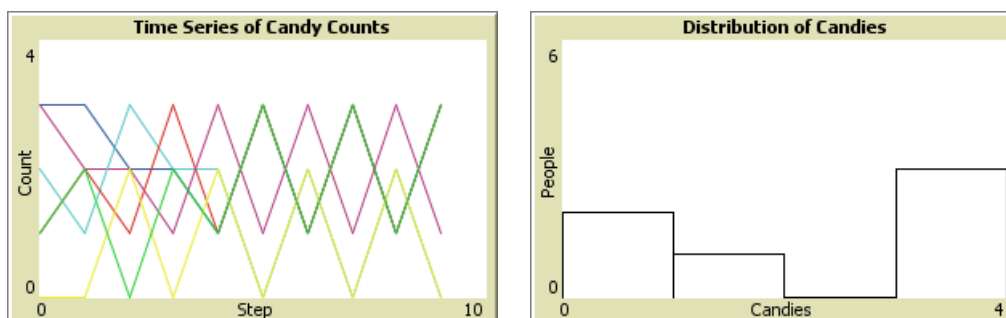


Figure 2: Example plots.

Now that you're familiar with the model controls and outputs, use them to investigate what happens with different numbers of players and different candy amounts.

1. Do you get the same results every time you setup and run the model with a given combination of **population** and **total-candy** values? If not, what do you think is causing the differences?
2. When you keep the **population** value constant, and initialize and run the model with varying values of **total-candy**, what do you observe about the change (if any) in the overall results?
3. Similarly, when you keep **total-candy** constant, and initialize and run the model with varying values of **population**, what do you observe about the overall results? Are your observations consistent with what you observed for question 2?
4. Are there any apparent relationships between **population**, **total-candy**, and the number of rounds it takes for the game to reach a fixed point or a cycle?

## Parameter sweeps with BehaviorSpace

In many cases, it's clearly easier to conduct experiments with the NetLogo model than it would be to do so with a group of people playing the Candy Sharing game. Nonetheless, tracking and tabulating the results can quickly become cumbersome – especially when varying one or both of **population** and **total-candy** across a range of values. Fortunately, NetLogo provides a tool to help with this: BehaviorSpace, which is accessed in the NetLogo **Tools** menu (Figure 3).

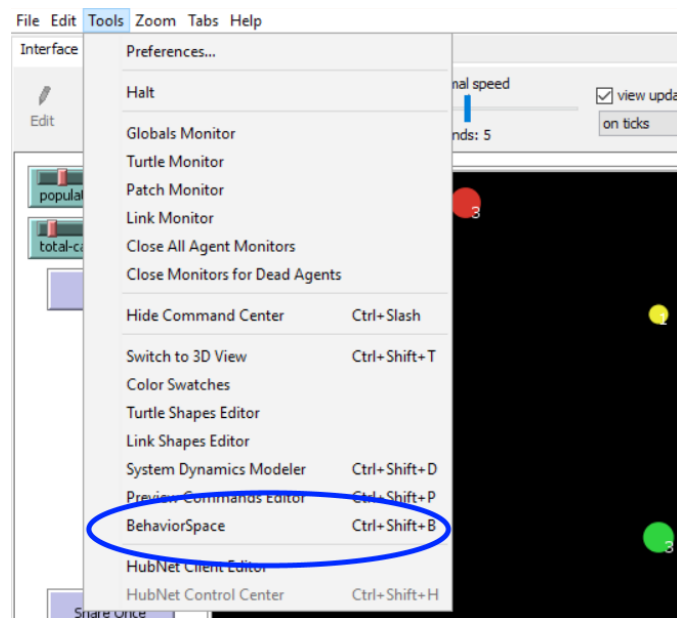


Figure 3: BehaviorSpace command in Tools menu.

BehaviorSpace lets us specify ranges or sequences of input parameter values, along with output measures of interest, and then automate the execution of the model and capture of the data, for later analysis (e.g. in Excel).

In this case, there’s already one experiment (“Initial experiment”) defined in this model. As seen in Figure 4, this experiment will vary **population** over a range from 5 to 10, in increments of 1, and vary **total-candy** from 5 to 25, also in increments of 1 (A). For each combination of **population** and **total-candy** values, 5 trials will be performed (B). For each trial, the list of candy counts for all players will be captured after each step or round (C), and each game will be played for 50 rounds (D).

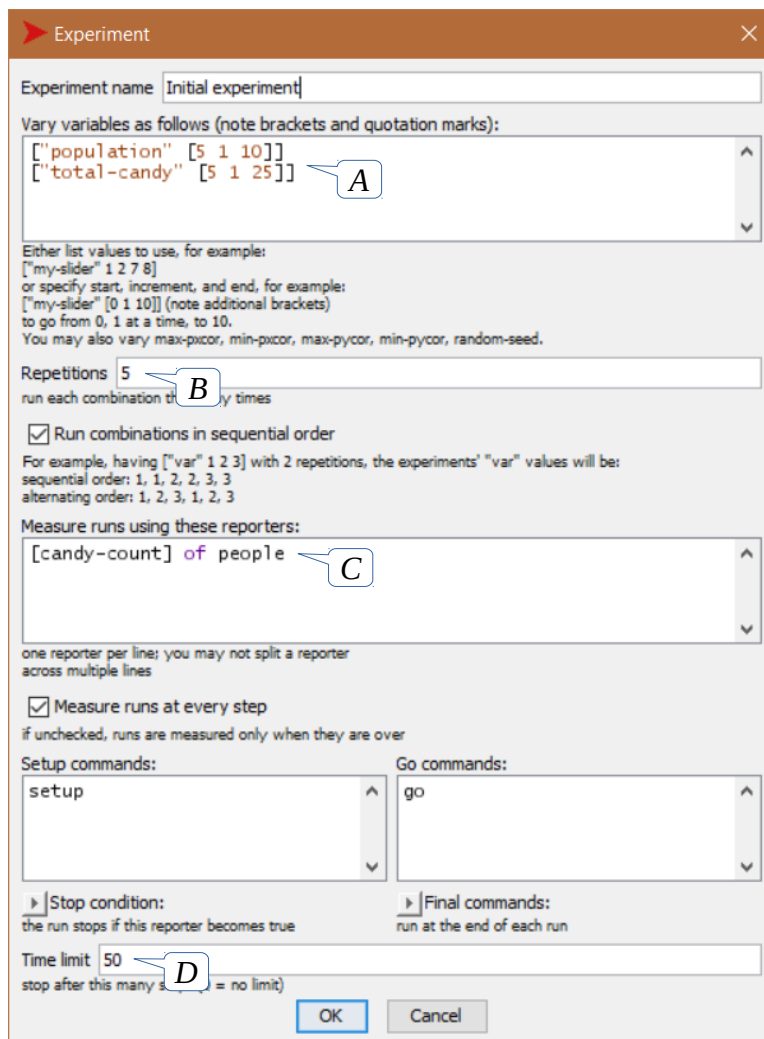


Figure 4: BehaviorSpace experiment specification.

The output of this experiment will be captured in a CSV (comma-separated values) file that can be opened by Excel or other spreadsheet programs. Note that while this is a convenient way to examine the output, it’s also very easy for BehaviorSpace to produce files that are too large for Excel to handle comfortably. (Theoretically, the maximum number of rows in the latest version

of Excel is over 1,000,000; however, the practical limit – based on available memory and processing power – is almost always much smaller than that.)

Use BehaviorSpace to conduct a few experiments – starting with the experiment already defined, but modifying or extending it as you see fit. (See the BehaviorSpace section of the NetLogo User Guide, at <https://ccl.northwestern.edu/netlogo/docs/>, for more information on how to configure and run BehaviorSpace experiments.) Based on what you observed in your interactive use of the model, try defining and conducting experiments that might help you answer some of the unanswered questions (if any) from the previous list (page 3).