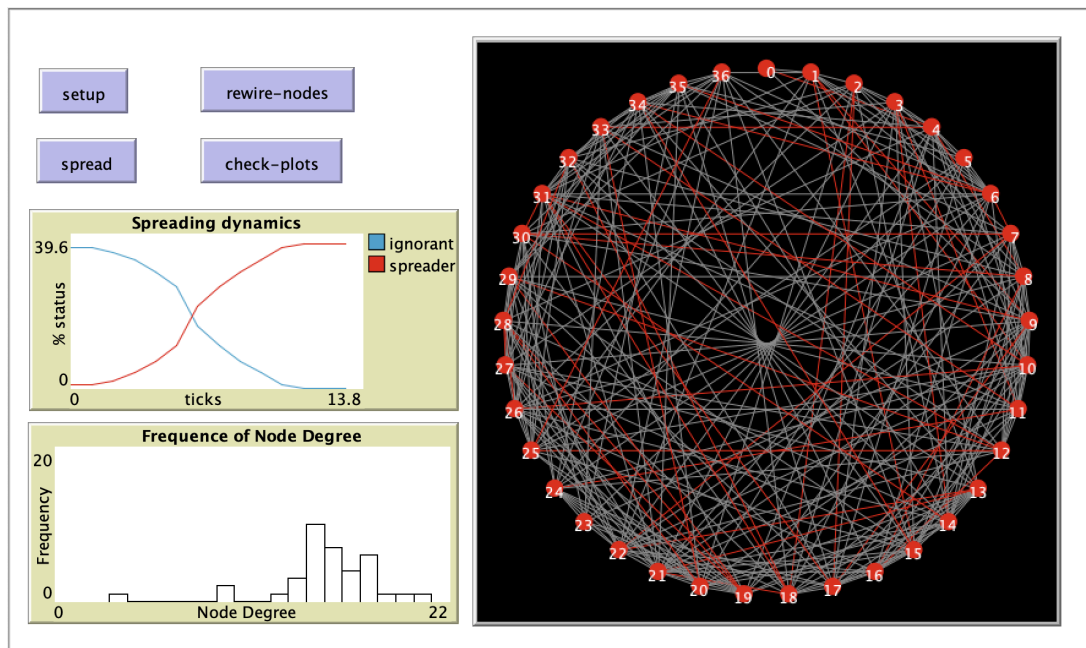# CS108L Computer Science for All
Module 8: Rubric Networks Milestone 2
How Rumors Spread

In this module we will be creating a small world network that shows how a rumor might spread amongst individuals of the network.. Make sure you turn wrapping off.

The GUI
- Add a setup, spread, rewire-nodes and check-plots button

- Add two plots
    - The first plot has a title Spreading Dynamics and plots the number of ignorant nodes in blue and spreader nodes in red vs ticks.
    - The second plot is a histogram of the degree of nodes.

Breeds, Globals and Agent Variables
- Create a breed nodes (plural) and node (singular)
- Create a nodes-own variable 'status'
- Create a global variable spread-chance

The setup button
- Clear the world.
- Create a local variable filename to hold the name of the file we are using 'networks2.txt'.

- Set the spread-chance to 0.8.
- Call an import-graph procedure that takes the filename as an argument.
- Make ticks available for the program.
- Use a circular layout with a radius of max-pxcor - 1
- The world looks similar (red node may be different) as to the below picture when setup is pressed.
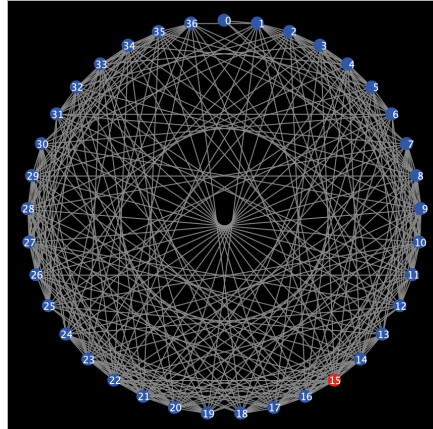
The import-graph procedure

NOTE: networks2.txt file structure: The first line is the number of nodes (turtles) the remaining lines are pairs of turtles that will be linked with a space between them (so 0 1 links turtle 0 to turtle 1.

- Open the networks2.txt file
- Create the number of nodes as listed on line 1 from the file
  - Set each node to a circle shape
  - Set their color blue
  - Label each node by their turtle ID (Hint: who).
  - Set their status to "ignorant"
  - Set one of the nodes at random to status "spreader" and color red.
- Call an import-links procedure
- Close the file (note this is done after the import-links procedure is called).

The import-links procedure

- Setup a while loop to read the remainder of the file.
  - Create a local variable 'items' that calls a procedure make-number-list with the line from the file as an argument.
  - From the items list, pull the first item of the list and ask the node associated with that number to create a link with the node associated with the second item.

The make-number-list procedure

- As an argument this takes a bunch of words (text) separated by spaces.
- Enclose the words in [] by concatenation.

- Change the words within the brackets to numbers.
- Report the list of numbers.

The spreader procedure (linked to spreader button, this runs the rumor portion of the program instead of 'go').
- Run a while loop until there are no "ignorant" nodes.
  o If a node status is "spreader" select one of its linked neighbors
    ▪ If that neighbors status is "ignorant" and a random-float 1 (this just returns a value between 0 and 1, a decimal) is less than or equal to the spread-chance then set its status to "spreader" and color to red
  o tick (last thing within the loop).
- Stop the program (outside of the loop…needed for BehaviorSpace or it never stops).

The rewire-nodes procedure.
- Ask all the nodes
  o If the degree is greater than 0
    ▪ If a random-float 1 < 0.4 (40% chance)
      • Ask one of the nodes links to die
      • Create a link with one of the other nodes in and make that link red in color.

The check-plots procedure
- Update the plots.

Experiments
- Run two experiments using BehaviorSpace
  o For both experiments
    ▪ Use the variable 'spreader-chance' starting at 0.1, incrementing by 0.1 and stopping at 1.
    ▪ Repeating 1.
    ▪ 'ticks' as the output.
    ▪ All boxes unchecked.
    ▪ Setup command: setup
    ▪ Go command: spreader
    ▪ Stop condition: stop when the number of nodes with the status ignorant is 0 (write this as code in the stop condition box).
  o For the first experiment, just run from the default setting, there should be no red links. Save the output as M8.MS2.EX1.<first name>.<last name>
  o For the second experiment, hit the re-wire button a couple of times and run. Save the output as M8.MS2.EX2.<first name>.<last name>
  o In the info tab, under things to notice: Comment on the differences between the two data sheets. Also note why we select the maximum spreader-chance to be 1.
  o Make sure to submit both sheets.

Module 8: Networks Milestone 2 (20 Points Total)		
Done	Points	Task
	1	**A:** • Submit a NetLogo source code with the file name: M8.MS2.*firstname.lastname*.nlogo. • The first few lines of your Code tab are comments including your name, the date, your school, and the assignment name (Lab 1: Spirograph).
	1	**B:** • Include appropriate in-line comments.
	1	**C:** • Include a detailed Info tab.
	1	**D:** • All breeds, global and local variables exist and are correctly identified.
	2	**E:** • Setup button contains the specified function and makes the picture as shown.
	2	**F:** • The import-graph procedure creates the nodes as specified in the networks2.txt file, calls the import-links procedure and closes the file.
	2	**G:** • The import-links procedure calls the make-number-list procedure, associates the return list to a local variable and sets up links between the nodes as specified in the networks2.txt file.
	2	**H:** • The make-numbers-list takes a line from the networks2.txt file as an argument and turns that line into a list of numbers (ex: 0 1 → [0 1] and reports the list.
	3	**I:**

		• The spreader procedure will propagate 'rumors' between linked nodes and change them from "ignorant" nodes to "spreader" nodes causing the colors to change at the probability described
	2	J: • The rewire nodes procedure will break a link if its degree is greater than 0 at a 40% probability using random-float. It will also create a new link with the color of red.
	3	K: • Behavior space is correctly setup, the two data sheets have been submitted named (M8.MS2.EX1.<first name>.<last name>) and (M8.MS2.EX2.<first name>.<last name>) and the info tab has comments as described above.