ARTICLE
Cartoon Network: A tool for open-ended exploration of neural circuits

Robert J Calin-Jageman
Psychology Department, Dominican University, River Forest, IL 60305.

Cartoon Network is an open-source simulator for neural circuits. It was designed to provide a microworld for the playful exploration of neural networks (similar to the niche Logo/Scratch fills for computer programming). The simulator makes it easy to construct and experiment with closed-loop circuits, enabling students to explore how small sets of neurons can generate complex patterns of activity (oscillations, reverberation). Cartoon Network can be connected to the Finch robot from BirdBrain Technologies, a cheap USB robotics platform. This enables students to design a nervous system for a Finch, creating sensory neurons which read Finch inputs (touch, light, and temperature), motor neurons which control Finch outputs (wheels, lights, and sound), and interneurons to produce intrinsic activity and/or link together inputs and outputs in ways that can produce complex and surprising behaviors. Students use Cartoon Network by taking on structured challenges (For example, can you make the Finch follow a light? Can it turn and run when it bumps a wall?) that require deep engagement with important principles of neural circuit operation (e.g., lateral inhibition, parallel processing, positive- and negative-feedback). On successfully completing a challenge, students can systematically explore the neural properties that help control the behavior, compare their work to other successful designs, and/or reflect on how the circuit they designed might be modified via evolution to produce different behaviors. Cartoon Network has proven an engaging and effective activity for undergraduates and is accessible for students as young as late elementary school. Download it for free at: https://github.com/rcalinjageman/cartoon_network.

Key words: simulation, neural network, lab exercise, neural circuit, closed-loop, microworld, Logo.

The Astonishing Hypothesis is that "You", your joys and your sorrows, your memories and your ambitions, your sense of personal identity and free will, are in fact no more than the behavior of a vast assembly of nerve cells and their associated molecules.

(Crick, 1994)

If there is a central dogma in neuroscience, it is that nervous systems produce behavior. That is, neuroscientists work from the assumption that any behavior, no matter how complex, is generated by the nervous system through the operation of *neural circuits*.

Why is this a reasonable working assumption? Because neural circuits are remarkably powerful (Marder and Calabrese, 1996). The basic operating principles for individual neurons enables the construction of circuits of limitless complexity. Recent developments with artificial neural networks (e.g., Alex et al., 2012) suggest that neural circuits can categorize information with remarkable subtlety and flexibility. Moreover, neural circuits need not simply 'process information' by mediating between inputs and outputs (open-loop processing). Neural circuits generate their own intrinsic patterns of activity (Brown, 1914; Mnih et al., 2013) while simultaneously incorporating continuous feedback from the environment (closed-loop processing). Decades of exploration have still not plumbed the limits of what neural circuits can do, lending credence to the premise that they mediate the expression of all complex behaviors.

Given the central explanatory role of neural circuits in modern neuroscience, it is essential that neuroscience education provide students with the opportunity to develop a deep sense of the power and principles of neural circuit function. Surveying neuroscience textbooks, however, shows relatively little attention to the operation of neural circuits. Most focus on cellular/molecular properties of individual neurons and/or the anatomical/functional properties of entire nervous systems. For example, the popular intro to neuroscience textbook by Kolb et al. (2016) has limited emphasis on neural circuits. Across the entire book there is discussion of two basic reflex circuits: a Renshaw loop and a sensory input circuit. The only coverage of positive and negative feedback is in the context of hormone control. A similar gap exists in neuroscience education materials. Browsing ERIN (Olivo et al., 2015) reveals a dearth of circuit-level educational resources relative to the abundance available at the molecular, cellular and systems levels, though the SWIMMY simulations (Grisham et al., 2008) and NeuroBytes (Petto et al., 2017) stand out as notable exceptions.

While circuit-level teaching resources are rare, neuroscience educators can draw on a variety of neuronal simulators (e.g., Moore et al., 2004; Moore and Stuart, 2007). Simulations are ideal for inquiry-based learning, as they allow students to repeatedly iterate through the critical-thinking cycle (predict, observe, reflect). One difficulty, though, is that most simulators have steep learning curves and arcane interfaces. Moreover, most simulators for complex networks emphasize open-loop processing for pattern classification. These types of simulations don't allow exploration of the fascinating dynamics of closed-loop systems. In addition, they are often so complex that understanding how the network works can prove challenging even for experts.

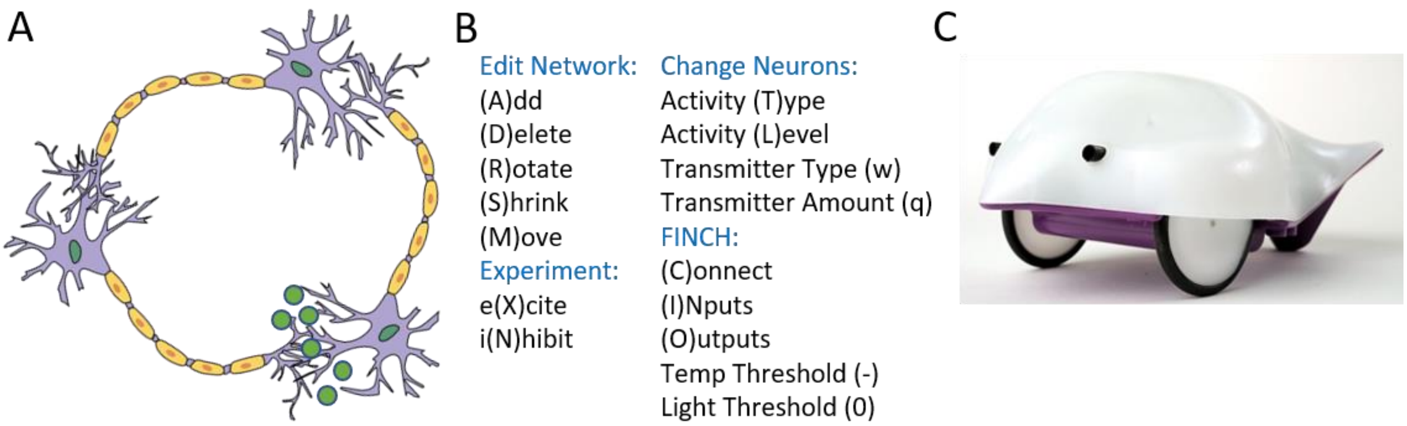

Figure 1. Cartoon Network. *A)* Screen-shot of a neural circuit in Cartoon Network. Neurons are represented with a cartoon-like style. The green dots are molecules of secreted neurotransmitter. Once stimulated sufficiently, this circuit remains constantly active. *B)* The menu system for Cartoon Network. Short-cut keys for each mode are in parenthesis (e.g., A enters Add Mode). *C)* The Finch Robot. Cartoon Network can interface with this USB robot, with sensory neurons active based on the light, touch, and temperature sensors on the Finch and motor neurons controlling the wheel motors, LED lights, and sound chip. Use of the robot is optional.

With these considerations in mind I set out to create a simulation that could help students develop a deep understanding of the power of neural circuits. In developing this project I was guided by Papert's concept of a micro-world (1980). A microworld is a "small playground of the mind" (Clements, 1989); an environment that learners explore to intuitively acquire important ideas or skills. Microworlds should be open-ended and complex enough to allow surprising/unexpected outcomes. Most of all, microworlds should be fun—they should call upon a learner's natural curiosity and desire for achievement to present challenges in which they *enjoy* mastering the desired skills and concepts. By making complex ideas an embedded part of the environment being explored, microworlds often open up deep learning to students at a wider range of interests and developmental stages than traditional instruction. Prominent examples of successful microworlds include the Logo (Papert, 1980) and Scratch programming languages (Maloney et al., 2010). Within neuroscience, the NeuroBytes platform provides a microworld for exploring neural networks with real-life sensors and actuators (Petto et al., 2017).

Cartoon Network represents my attempt to develop a microworld for exploring neural network circuits. It enables students to playfully discover some deep principles of circuit function, and can also enable them to build the brain for a mobile robot. The simulation and its source code are available for free non-commercial use via GitHub: https://github.com/rcalinjageman/cartoon_network. A video showing how to install and use the simulation is available at https://youtu.be/_M4_F9M2ODc.

MATERIALS AND METHODS

Implementation Details. Cartoon Network is written in Java using the Greenfoot framework (Kölling, 2010). Greenfoot is an agent-based gaming platform which enables complex simulations to be developed by simply specifying the look and behavior of different agents. The platform then handles iteratively applying each set of behavioral rules to each agent. Use of the Greenfoot platform enabled Cartoon Network to be developed with a surprisingly compact code-base (~1000 lines of code and comments). Only two primary agents had to be coded: neurons and neurotransmitter (one additional agent handles the menu system and user interface). In addition, Greenfoot enabled the use of cartoon-like animated neurons (Figure 1A) and an interactive interface (Figure 1B) that hides all the simulation details from end users.

Neurons are simulated in Cartoon Network using a hybrid integrate-and-fire scheme (Getting, 1983). In this scheme each neuron is a single isopotential compartment. Membrane potential varies in proportion to total input current divided by the neuron's capacitance. Rather than simulate detailed voltage-gated channels, however, spikes are fired whenever the membrane potential reaches above a specified threshold. To mimic action potential undershoot, the threshold is dynamic: each spike sets threshold to a high level which then decays back to baseline. Intrinsically active neurons are simulated by setting the resting membrane potential higher than the spike threshold. Bursting neurons are simulated in the same way, but with a spike-activated hyperpolarization (a simplified calcium-activated potassium current). This modeling scheme is abstract enough to enable high-speed computation but detailed enough to enable realistic firing patterns that can be very closely fit to real neurons (Calin-Jageman and Katz, 2006; Calin-Jageman et al., 2007).

Each neuron in Cartoon Network has a pool of available neurotransmitter. After each action potential, there is a short delay and then a fixed fraction of available neurotransmitter is released at the axon. Released transmitter then undergoes Brownian motion, producing either an EPSP or IPSP when contacting *any part* of a postsynaptic neuron. At each time-point, each molecule of transmitter has a fixed probability of being taken back up by the emitting neuron, replenishing that neuron's pool of available transmitter. This simulation scheme means that synapses undergo use-dependent depletion (Tsodyks et al., 1998; Sakurai et al., 2007). Each contact between transmitter and neuron produces a small discrete change in membrane potential (not scaled by a driving force).

Connection to a Finch robot is handled via a free API provided by BirdBrain technologies (http://www.birdbraintechnologies.com/; Figure 1C shows an image of a Finch robot). Neurons can be designated to receive input from Finch sensory (sensory neuron), or to have their outputs activate Finch actuators (motor neuron), or both. Each sensory input has an adjustable threshold; readings over that threshold produce depolarization in designated sensory neurons. Each actuator has a resting state of no activation. Each molecule of NT released by a designated motor neuron produces a brief activation of a FINCH output (a short turn of a wheel, brief flash of light, or brief activation of its sound module).

Installation. Running Cartoon Network requires a desktop computer with an updated installation of the Java Runtime Environment (version 8 or higher, available for free at http://java.com). The simulation is packaged into a single jar file; just download the file from GitHub (https://github.com/rcalinjageman/cartoon_network/raw/master/cartoon_network.jar) and then run it (in Windows, just double-click the jar file). GitHub is a version control system—as updates and bug fixes are made to the simulation the posted jar file is updated. One can even use the GitHub client to check out the jar file and then pull updates automatically.

Cartoon Network can also run within a browser. This is not ideal though, because a) it cannot control a Finch robot from the browser, and b) most modern browsers now prevent running Java from within a webpage due to security issues. If desired, though, a browser-enabled version is available on the Greenfoot community showcase (http://www.greenfoot.org/scenarios/19282). To run this applet users will need to add greenfoot.org as an acceptable source under their Java security settings.

The source code for Cartoon Network is also available via GitHub. Once downloaded or checked out, it can be opened in the Greenfoot editor. From there, users can make customizations to the code and/or re-export as a jar file. Those interested in contributing patches or extensions can request via GitHub to obtain permissions to push code back to the main repository.

Operation. Opening Cartoon Network shows a blank screen. The simplest way to get started is to type 'a' to enter add mode, and then to click different places on the screen. Each click will add a new neuron. Typing 'x' then enters excite mode, and clicking on a neuron will stimulate it, potentially to threshold. When threshold is reached, an action potential is animated racing down the neuron's axon and releasing transmitter that jiggles around the simulation space for a few moments until re-uptake occurs. Typing 'm' enters move mode, allowing neurons to be placed in proximity to each other—synapses are represented simply by placing the dendrites of one neuron so that they will come into contact with the released neurotransmitter of another neuron. After demonstrating an initial two-layer feed forward network, students can typically figure out the rest of the simulation controls on their own.

Exploration will show that the simulation works by switching between different editing modes. Some modes enable subsequent mouse clicks to alter the network of neurons: add neurons, delete neurons, rotate neurons, shrik/grow neurons. Other modes enable clicks to change individual neuron properties: transmitter type (glutamate or GABA), amount of released transmitter, activity level (maximal firing rate; equivalent to adjusting duration of the undershoot phase), or activity type (silent at rest, spontaneously active, or bursting).

The current mode is displayed in the top-left corner. Modes can be selected by typing an appropriate hot key. In addition, clicking the current mode will open a menu which allows selection of the next mode via mouse (Figure 1B).

When the simulation starts it makes no attempt to connect to a Finch robot, and options related to the Finch are grayed out. Pressing 'c' attempts to connect to a Finch, and the status will change to "Finch!" if the connection is successful. Once connected, students can enter input or output modes. Subsequent clicks on neurons bring up a dialog to create a sensory neuron (fires based on Finch's temperature, light, or touch sensory) or motor neuron (activity activates the Finch's lights, wheels, or buzzers). The thresholds at which light and temperature begin activating sensory neurons can also be set (e.g. to make the neurons silent under ambient light). Neurons can be multi-functional (both sensory and motor).

RESULTS

Suggested Uses. Cartoon network can be used for classroom demonstrations. For example, when teaching basic neural functions in Intro Biology or Psychology an instructor can open Cartoon Network to demonstrate basic principles of neural network function (threshold, release and re-uptake, inhibition vs. excitation, feed-forward and feed-back, reverberating circuits). Relative to other simulations, a benefit of Cartoon Network is that students can see the transmitter being released after each action potential. Also, it is easy to demonstrate intrinsic activity (e.g., bursting neurons), a topic that is often neglected in introductory coverage of neurons.

The guiding principle for student usage is to explore and play. For laboratory use I first familiarize the students with the simulation and then give them a set of open-ended challenges to work on. When a Finch is not available, these include: making a circuit that can stay active once stimulated, finding a way to shut off an active circuit, and making a circuit that oscillates. When a Finch is available, challenges typically include following a light, chirping when hitting a wall, integrating two behaviors so that only one happens at a time, and oscillating the light display between the three colors. Students are bound to hit roadblocks, but encouraging them to cycle through prediction, observation, and reflection usually helps them sharpen their understanding and inch closer to a solution.

For undergraduate students, meeting a challenge is only the beginning of the process. Three directions for reflection/extension are particularly fruitful. First, students can be asked to systematically explore how the parameters of their circuit influence its behavior (e.g., how does the amount of transmitter released influence the rate of oscillation? Why? Is it different for different neurons in the circuit?). Second, students can be asked to compare/contrast their solution with those produced by others (e.g., is this essentially the same circuit or a different one? Which is more efficient? Which is more flexible?). Finally, students can think about how circuits might evolve or change over development (e.g., What changes would be required to produce the opposite behavior?). With the Finch available, I have also found that students respond well to the open-ended challenge of specifying the type of creature they want (e.g., scared of light but fond of contact) and then trying to develop the nervous system needed for their desired creature. A lab handout is available for download in the Supplementary Material and in GitHub (https://raw.githubusercontent.com/rcalinjageman/cartoon_network/master/cartoon%20network%20handout%20-%20d2.docx).

While Cartoon Network was developed with undergraduate students in mind, I have found that it is enjoyable and engaging for students as young as 8. For young students, the distinction between sensory and motor neuron is easy to grasp, so students can quickly build simple reflex circuits (e.g., connecting a temperature sensory neuron with a sound motor neuron makes a circuit that chirps when the student breathes on the Finch; we jokingly label this a 'bad breath detector'). From there, simple challenges with the Finch can be posed (e.g., "make a Finch brain so that it chirps if it is touched").

Regardless of age, students often invent their own challenges and end up playing with the simulator in ways that are completely unexpected (producing large reverberation loops seems to be particularly satisfying and exciting). Through play with the system, students will bump up against many important ideas in network analysis: lateral inhibition (such as to keep two things from happening at once), positive-feedback, and oscillation. The goal is to build an intuitive foundation from which formal treatment of these concepts later in the semester can build. In addition, students should hopefully develop a tangible sense of how easily even small sets of neurons can produce interesting behaviors of surprising complexity, giving them insight into this key pillar in the modern approach to neuroscience.

Assessment.

With Undergraduate Students. I assessed the impact of Cartoon Network in an upper level Behavioral Neuroscience laboratory. The lab experience was scheduled at the beginning of the semester, for the second week of lab.

First, understanding of simple network function was assessed using a pre-test/post-test design. During the lab pre-test, students were presented with a diagram of a classic two-neuron half-center oscillating network. They were asked to predict the behavior of the network (the diagram was not labelled as an oscillator). Only 4 of 17 students correctly predicted the network would oscillate (*p*roportion correct = 0.24, 95% CI [0.10, 0.47]). Most predicted the neurons would silence each other and remain permanently inhibited, showing a lack of understanding of intrinsic activity. After three hours working with Cartoon Network and a Finch, students completed a post-lab assessment. This assessment had a similar diagram but with a 3-neuron configuration (which is arguably more difficult to interpret). On the post-test 12 of 17 students correctly predicted circuit oscillation (p = 0.71, 95% CI [0.47, 0.87]). Thus, there was a substantial increase in the proportion of students who could mentally work through the operation of an oscillating neural circuit (increase in proportion correct= 0.47, 95% CI [0.14, 0.68]; this comparison of two independent proportions is similar to conducting a Chi Square test; because the 95% CI does not include the null hypothesis of no improvement, the analysis is is statistically significant at $p < 0.05$).

As a qualitative assessment, students also completed a lab writeup in teams, explaining their solution to developing a circuit for enabling the Finch to follow a light source. Subjective review of the write-ups showed that students were able to develop suitable circuits and to understand some of their deep operating principles. For example, one team wrote:

> *"The Finch circuit our group created that enabled it to track a light source, wasn't just one circuit, it was two separate neural circuits. These two separate neural circuits, were created to represent each of the sensory neurons or "eyes" of the Finch robot. These separate circuits then branched out to control the motor neurons (left and right wheels) of the Finch allowing it to rotate and move according to the changes in light. Changing the amount of neurotransmitter released by an interneuron on one wheel caused the Finch to move in the same direction as the visual stimulus. The interneuron served as a regulator by changing the amount of NT released to the wheel opposite to the side of the visual stimulus. This would cause a greater motor output by the wheel and turn the finch in the direction of the light. The right circuit had a stronger connection to the left wheel and the left circuit had a stronger connection to the right."*

This answer shows the students developing an intuitive sense of parallel processing, correctly diagnosing the most sensitive parameters for controlling network output, and beginning to formulate some ideas of how neural plasticity could relate to changes in behavior.

With Neuroscience Faculty. Cartoon network was presented at the 2017 Faculty for Undergraduate Neuroscience workshop during the pre-workshop intensive laboratory experience. During this workshop participants worked in groups of 2 to complete most of the lab handout provided for undergraduate students. At the end of each session, faculty participants provided feedback on Cartoon Network using a standardized form developed by the workshop organizers. One item asked participants to rate their agreement with the statement "The exercise presented in this workshop is a good vehicle for teaching undergraduates principles of neuroscience" using a scale from 1 (*strongly agree*) to 5 (*strongly disagree*). Of the 32 participants, 28 indicated strongly agree (88%); the overall mean rating for this item was 1.1 (*SD* = 0.33). In addition, 22/32 (73%) indicated that they strongly agree with the statement "I am likely to incorporate the material from this workshop as a classroom activity" (*M* = 1.5, *SD* = 0.73). Responses to a prompt for general comments were very positive; some representative examples are: "I really like that the software is ready to use with or without the robot. That will allow me to incorporate the activity immediately and obtain the robots upon the next budget cycle… Fun and creative!"; "Very engaging lab!"; "Inspiring and creative"; "This was so fun and interesting." Overall, the consensus among faculty participants was that this is a useful and fun lab that would make a valuable addition to the undergraduate curriculum.

DISCUSSION

Cartoon Network is a promising tool for exploring neural circuits, a critical but often neglected level of analysis for neuroscience students. Work with Cartoon Network can help students begin to appreciate the remarkably complex patterns of behavior that emerge from the simple operating principles of individual neurons. This understanding may evolve in a large part because the simulation is fun; students genuinely enjoy using the simulator and often find unique/unexpected ways of configuring networks.

Development of Cartoon Network is ongoing. Participants at the FUN 2017 provided very useful feedback on additional features that would make Cartoon Network more useful. Based on their suggestions, I will soon add functions for saving/loading circuits that have been created, for interfacing with other robotics platforms, and for enabling different forms of synaptic plasticity. Additional features will be released via GitHub.

A key drawback of Cartoon Network is that the fun of exploring it is greatly enhanced by connecting it to a Finch, something which currently requires an investment of about $100/robot. One possible extension would be writing a software-based robot that could be controlled via Cartoon Network. I welcome feedback and collaborations for further development; bugs and requests for new features can be submitted via the project's issues page on GitHub.

REFERENCES

Alex K, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Neural Inf Process Syst 1097–1105.

Brown TG (1914) On the nature of the fundamental activity of the nervous centres; together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system. J Physiol 48:18–46.

Calin-Jageman RJ, Katz PS (2006) A distributed computing tool for generating neural simulation databases. Neural Comput 18:2923–2927.

Calin-Jageman RJ, Tunstall MJ, Mensh BD, Katz PS, Frost WN (2007) Parameter space analysis suggests multi-site plasticity contributes to motor pattern initiation in Tritonia. J Neurophysiol 98:2382–2398.

Clements D (1989) Computers in elementary mathematics education. Englewood Cliffs, NH: Prentice Hall.

Crick F (1994) The astonishing hypothesis. New York, NY: Simon and Schuster.

Getting PA (1983) Mechanisms of pattern generation underlying swimming in Tritonia. II. Network reconstruction. J Neurophysiol 49:1017–1035.

Grisham W, Schottler NA, Krasne FB (2008) SWIMMY : free software for teaching neurophysiology of neuronal circuits. J Undergrad Neurosci Educ 7:A1–A8.

Kolb B, Whishaw IQ, Teskey GC (2016) An introduction to brain and behavior. New York, NY: MacMillan.

Kölling M (2010) The Greenfoot programming environment. ACM Trans Comput Educ 10:1–21.

Maloney J, Resnick M, Rusk N, Silverman B, Eastmond E (2010) The scratch programming language and environment. ACM Trans Comput Educ 10:1–15.

Marder E, Calabrese RL (1996) Principles of rhythmic motor pattern generation. Physiol Rev 76:687–717.

Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing Atari with deep reinforcement learning. arXiv preprint: 1312.5602.

Moore BJW, Stuart AE, Barry SR (2004) Neurons in action : computer simulations with NeuroLab. J Undergrad Neurosci Educ 2:R6–R7.

Moore JW, Stuart AE (2007) Neurons in action 2: tutorials and simulations using NEURON. 2nd ed. Sinauer Associates, Inc.

Olivo RF, Burdo JR, Calin-Jageman R, Grisham WE, Linden ML, Rosenberg RL, Symonds LL, Thornton JE (2015) ERIN: a portal to resources for higher education in neuroscience. J Undergrad Neurosci Educ 13:A126-130.

Papert S (1980) Mindstorms: children, computers, and powerful ideas. 2nd ed. New York, NY: Basic Books.

Petto A, Fredin Z, Burdo J (2017) The use of modular, electronic neuron simulators for neural circuit construction produces learning gains in an undergraduate anatomy and physiology course. J Undergrad Neurosci Educ 15:A151–A156.

Sakurai A, Calin-Jageman RJ, Katz PS (2007) Potentiation phase of spike timing-dependent neuromodulation by a serotonergic interneuron involves an increase in the fraction of transmitter release. J Neurophysiol 98:1975–1987.

Tsodyks M, Pawelzik K, Markram H (1998) Neural networks with dynamic synapses. Neural Comput 10:821–835.

Received May 05, 2017; revised September 06, 2017; accepted September 07, 2017.

Thank you to my students in Behavioral Neuroscience for their enthusiasm and their thoughtful feedback on the simulation and lab exercises. Thank you also to the participants in the pre-lab portion of the FUN 2017 workshop who helped road-test this simulation and who provided many excellent suggestions for improvement.

Address correspondence to: Dr. Robert J. Calin-Jageman, Psychology Department, 7900 West Division, River Forest, IL 60305. Email: rcalinjageman@dom.edu.