Module 5: The Menstrual Cycle Using Boolean Logic

Platform: NetLogo

Following this lesson students should be able to:

1) Describe the hormones that contribute to the menstrual cycle and how they function
2) Apply Boolean logic to biological phenomena
3) Identify stability in a biological process and computational system
4) Evaluate model outputs to determine emergent phenomena

Purpose: This module will guide students through a tutorial to code a NetLogo model from scratch that uses Boolean Logic to create a simulation of the menstrual cycle. Biologically, students will be exposed to **the hormones involved in the menstrual cycle**. Computationally, students will **define rules using Boolean logic** and **analyze model outputs to identify a system's emergent phenomena**.

Biological Terms:

1) Follicular Phase: the period of the menstrual cycle from the first day of the period until ovulation. Starts with the release of FSH and continues as estrogen slowly builds up
2) Luteal Phase: the period of the menstrual cycle from the start of ovulation until the period starts. Progesterone peaks and then drops
3) Follicle Stimulating Hormone (FSH): released from the pituitary gland and stimulates the maturation of an egg in the ovaries
4) Progesterone: released by the corpus luteum after ovulation. If fertilization of an egg does not occur, the corpus luteum dies and levels drop
5) Estrogen: released by the ovaries. Levels rise and fall twice within the menstrual cycle
6) Luteinizing Hormone (LH): released from the pituitary gland due to high estrogen levels
7) Ovulation: the release of an egg. Triggered by a spike in LH

Computational Terms:

1) Boolean Logic: a problem which reports only a True or False value
2) If: used to carry out a function only when a Boolean problem is True
3) If Else: used to carry out two functions, one when a Boolean problem is True and one when it is False
4) Emergent phenomena: property that arises from the collective behavior of a dynamic system
5) Stability: the ability to converge to an equilibrium across a range of inputs or perturbations (unstable systems do not move toward an equilibrium)

Time Estimation:

1) In-Class Activity: 30 minutes
2) Model Tutorial: 1 hour
3) Model Testing and Advancement: 2 hours 30 minutes

Total: 4 hours

Part One: In Class Activity

Materials: none

Rules:

1) Assign each student a component/hormone in the menstrual cycle (5 FSH, 5 LH, 5 progesterone, 5 estrogens, 1 egg (ovulation))
2) Have students stand to represent an active state and sit to represent an inactive state
3) Have students decide what must happen for each component to reach an active state (i.e. all LH stand up if 5 estrogen and 3 FSH are standing) and what will need to happen to reach an inactive state (i.e. all LH sit if egg is standing)
4) Start with everyone sitting
5) Play a game of Simon Says If and If Else to determine their activation (i.e. If you are wearing a red shirt stand, if else remain seated)
6) After each If/If Else command allow the students to determine if who is standing would be part of the menstrual cycle (i.e. if it follows one of the rules outlined in step 3). If so have them cycle
7) Keep introducing commands until they have cycled at least three times

Suggested Discussion Questions:

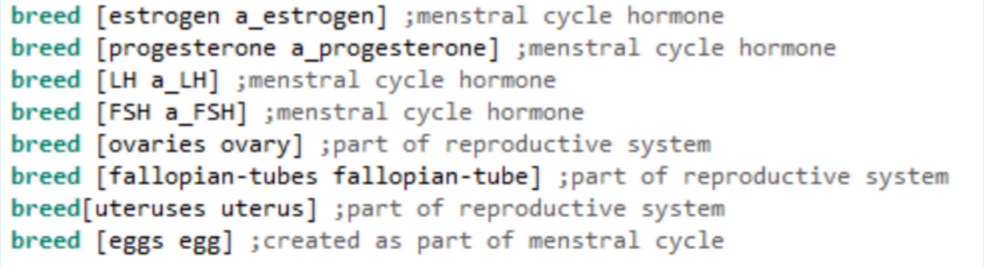
1) How is sitting or standing a good representation of Boolean logic?
2) What components of the menstrual cycle does Boolean logic does not fully cover?

Part Two: Model Tutorial

1) Open **NetLogo**

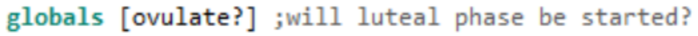
2) Instead of using one of the built-in codes, you will be making one from scratch today. Click the **Code** tab to get started

3) Define the breeds of variables that will be part of the model using **breed [plural_form singular_form]**. The components of this model should include estrogen, progesterone, LH, FSH, ovaries, fallopian tubes, uterus, and eggs



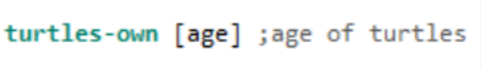
```
breed [estrogen a_estrogen] ;menstral cycle hormone
breed [progesterone a_progesterone] ;menstral cycle hormone
breed [LH a_LH] ;menstral cycle hormone
breed [FSH a_FSH] ;menstral cycle hormone
breed [ovaries ovary] ;part of reproductive system
breed [fallopian-tubes fallopian-tube] ;part of reproductive system
breed[uteruses uterus] ;part of reproductive system
breed [eggs egg] ;created as part of menstral cycle
```

4) Define the global variable **ovulate?** with **globals [transcribe?]**. This variable will use Boolean logic to determine if the conditions are met to turn on the lac operon



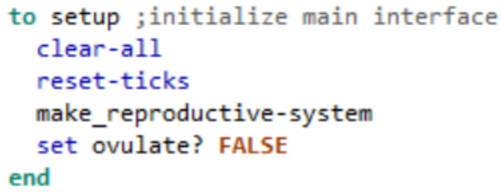
```
globals [ovulate?] ;will luteal phase be started?
```

5) Define the turtle owned global variable **age** with **turtles-own [age]**. This variable will keep track of the age of the turtles



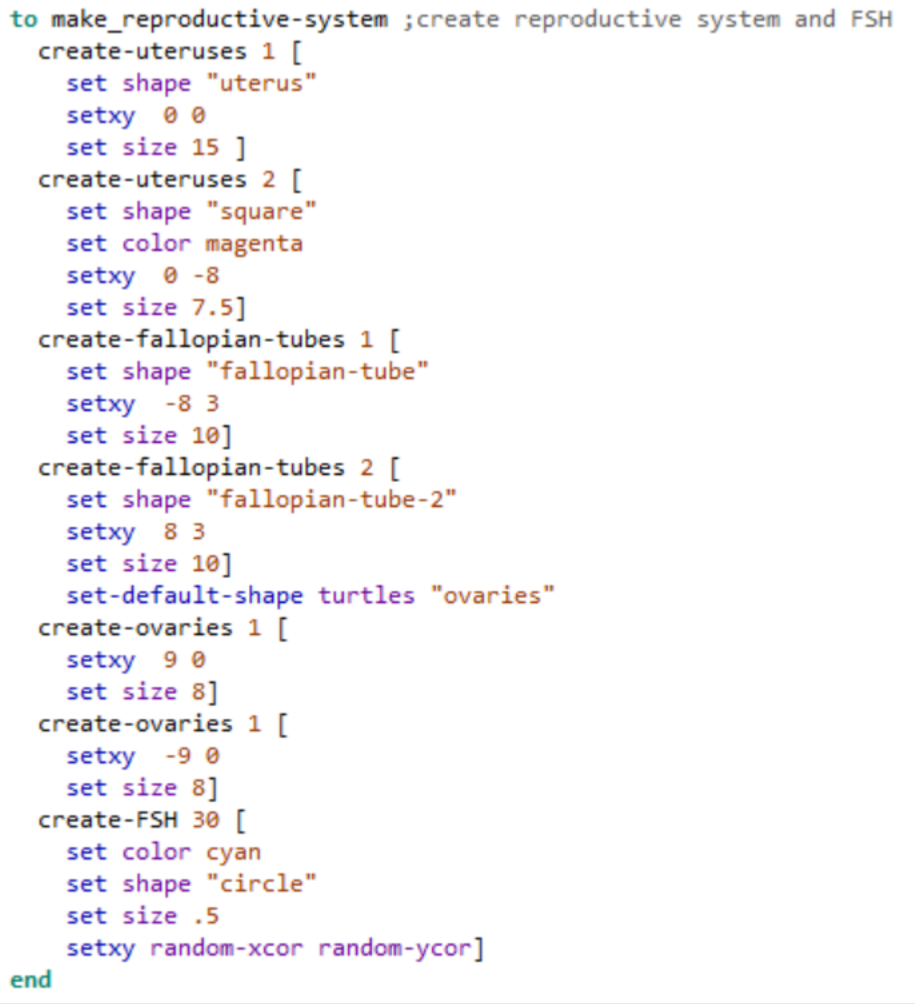
```
turtles-own [age] ;age of turtles
```

6) To set up the simulation, make a command called **to setup**. Under the command, type 4 actions (1) **clear-all**, which will clear all variables each time the code is run, (2) **reset-ticks**, which will initialize the time to be 0, (3) **make_reproductive-system**, a command which will create the lac operon, and (4) **set ovulate? false**, a command that will set the Boolean logic gate as false at beginning of the simulation. Finish the command by typing **end**



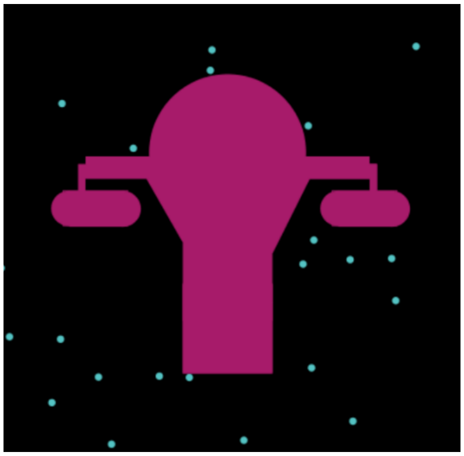
```
to setup ;initialize main interface
  clear-all
  reset-ticks
  make_reproductive-system
  set ovulate? FALSE
end
```

7) To create the lac operon, make a command called **to make_reproductive-system**. Under the command, create the 7 turtles that comprise the beginning of the model: uterus 1, uterus 2, fallopian tube 1, fallopian tube 2, ovaries x2, and FSH using the format **create-breed 1 [set color ____ set shape "____" setxy ____ ____ set size____]**. You can customize with your own preferences in the ____s and finish the command with **end**

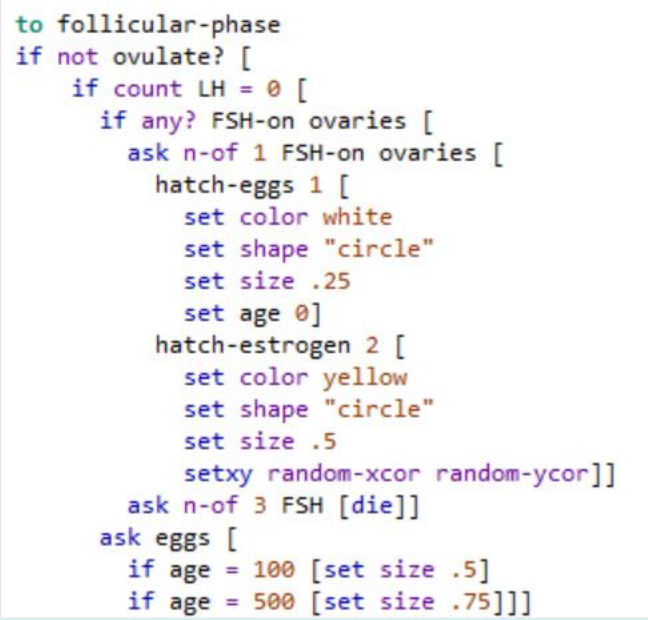


```
to make_reproductive-system ;create reproductive system and FSH
  create-uteruses 1 [
    set shape "uterus"
    setxy  0 0
    set size 15 ]
  create-uteruses 2 [
    set shape "square"
    set color magenta
    setxy  0 -8
    set size 7.5]
  create-fallopian-tubes 1 [
    set shape "fallopian-tube"
    setxy  -8 3
    set size 10]
  create-fallopian-tubes 2 [
    set shape "fallopian-tube-2"
    setxy  8 3
    set size 10]
    set-default-shape turtles "ovaries"
  create-ovaries 1 [
    setxy  9 0
    set size 8]
  create-ovaries 1 [
    setxy  -9 0
    set size 8]
  create-FSH 30 [
    set color cyan
    set shape "circle"
    set size .5
    setxy random-xcor random-ycor]
end
```

8) Navigate to the main interface by clicking **Interface** and check to see if your reproductive system was built correctly. Under **Button** click **Button**, place it near the top of the white part of the main interface, name it **setup**, and click **Ok**. Click the **setup** button and your lac operon should appear on the screen. If you used the above commands, it will give you the reproductive system and FSH pictured below. [Hint: uterus, fallopian tube 1 and 2, and ovaries are not a default shape in the NetLogo Turtle Shapes Library and will need to be created]

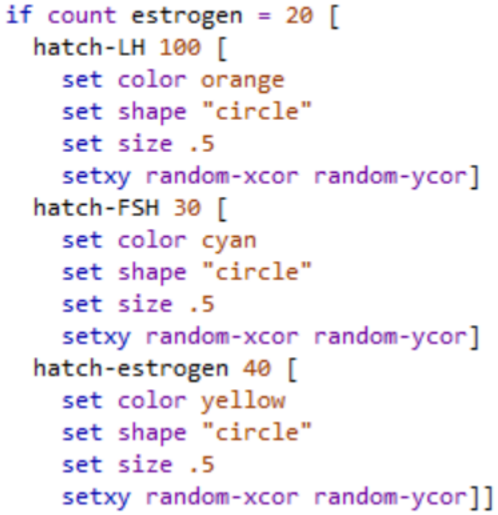


9) Navigate back to the **Code** and create a command called **to follicular-phase.** Initiate Boolean logic with **if not ovulate?** This will run only if ovulate? is set to FALSE. Create command **if count LH= 0** and **if any? FSH on ovaries ask n-of 1 FSH on ovaries** to **hatch-eggs** and **hatch-estrogen** and **ask n-of 3 FSH [die]**. When LH=0 while ovulate? is FALSE, when FSH is on ovaries, eggs will develop and estrogen will be created. Create command **ask eggs** and **if age= 100 [set size .5]** and **if age=500 [set size .75].** This will have eggs get larger as they mature



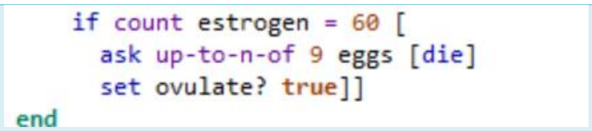
```
to follicular-phase
if not ovulate? [
    if count LH = 0 [
      if any? FSH-on ovaries [
        ask n-of 1 FSH-on ovaries [
          hatch-eggs 1 [
            set color white
            set shape "circle"
            set size .25
            set age 0]
          hatch-estrogen 2 [
            set color yellow
            set shape "circle"
            set size .5
            setxy random-xcor random-ycor]]
        ask n-of 3 FSH [die]]
      ask eggs [
        if age = 100 [set size .5]
        if age = 500 [set size .75]]]
```

10) Within **to follicular-phase if not ovulate?,** create code **if count estrogen = 20** and **hatch-LH 100**, **hatch-FSH 30**, and **hatch-estrogen 40.** This will have LH, FSH, and estrogen spike when ovulate? is FALSE and estrogen = 20



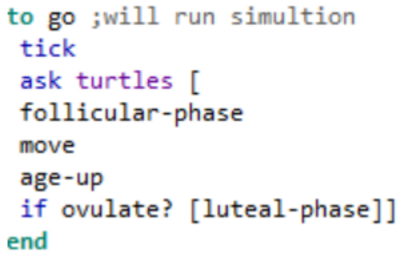
```
if count estrogen = 20 [
  hatch-LH 100 [
    set color orange
    set shape "circle"
    set size .5
    setxy random-xcor random-ycor]
  hatch-FSH 30 [
    set color cyan
    set shape "circle"
    set size .5
    setxy random-xcor random-ycor]
  hatch-estrogen 40 [
    set color yellow
    set shape "circle"
    set size .5
    setxy random-xcor random-ycor]]
```

11) Within **to follicular-phase if not ovulate?,** create code **if count estrogen = 60** and **ask up-to-n-of 9 eggs [die]** and **set ovulate? TRUE**. This will have all but 1 egg die and set ovulate? TRUE when estrogen= 60



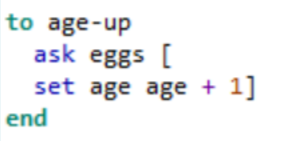
```
      if count estrogen = 60 [
        ask up-to-n-of 9 eggs [die]
        set ovulate? true]]
end
```

12) Create command **to go** and **tick**, **ask turtles [follicular-phase, move, age-up, if ovulate? [luteal-phase]]**



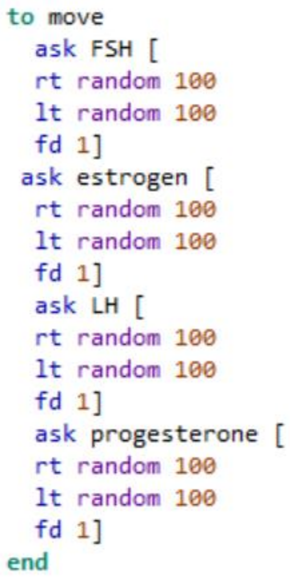
```
to go ;will run simultion
 tick
 ask turtles [
 follicular-phase
 move
 age-up
 if ovulate? [luteal-phase]]
end
```

13) Create command **to age-up [ask eggs [set age age + 1].** This will have eggs age up



```
to age-up
  ask eggs [
  set age age + 1]
end
```

14) Create command **to move ask FSH, estrogen, LH, and progesterone** to randomly move



```
to move
  ask FSH [
  rt random 100
  lt random 100
  fd 1]
 ask estrogen [
  rt random 100
  lt random 100
  fd 1]
  ask LH [
  rt random 100
  lt random 100
  fd 1]
  ask progesterone [
  rt random 100
  lt random 100
  fd 1]
end
```

15) Navigate to the main interface by clicking **Interface.** Under **Button** click **Button**, place it near the top of the white part of the main interface, name it **go**, and click **Ok**. When this button is clicked it will start the simulation

16) While on the main interface, also add a plot to the screen. Under **Button** click **Plot** and add it to the bottom of the white part of the main screen. Right click on the plot and choose **Edit**. Name the plot **Hormone Levels**, the x axis label **Time**, and the y axis label **Amount**

17) Under **pen update commands** change **plot count turtles** to **plot count FSH** and change to default color from black to cyan by clicking on the **black box**, choosing **cyan**, and clicking **Ok**.

18) Click **add pen** and repeat step 17 by naming the pen command, **plot count estrogen** and making the color **yellow**

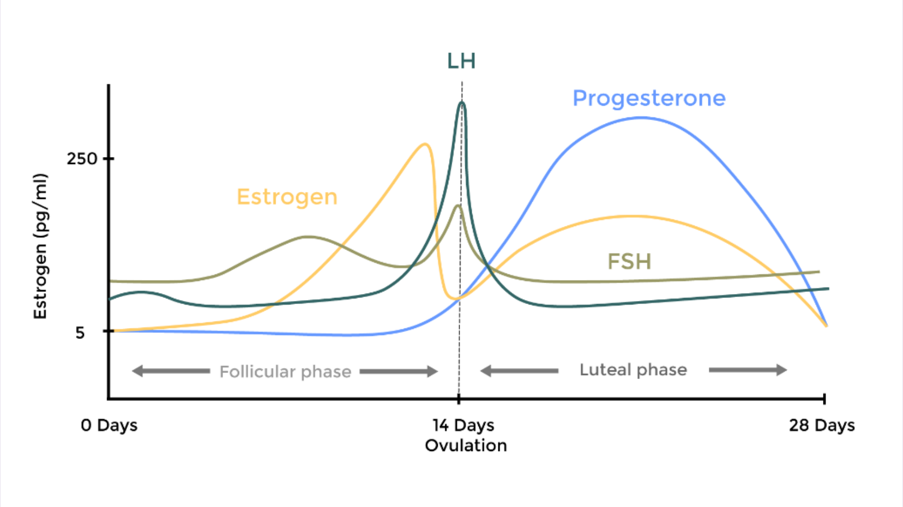
19) Click **add pen** and repeat step 17 by naming the pen command, **plot count LH** and making the color **orange**. Click **Ok** to exit the plot

20) Click **add pen** and repeat step 17 by naming the pen command, **plot count progesterone** and making the color **red**. Click **Ok** to exit the plot

21) This tutorial has helped create half the menstrual cycle, you will need to create the luteal phase and a way for the model to continue cycling. [Hint: setting the time step slower will help]

Module 4: Gene Regulation and Stability Using Boolean Logic

1) To define rules for the luteal phase the simulation you will need to determine what you want the rules to be. Consider this graph of the hormone levels in the menstrual cycle:



a. When does progesterone start to rise?

b. What happens to LH and FSH during the luteal phase?

c. What happens to estrogen during the luteal phase?

d. What happens to the egg during the luteal phase, how does it move?

e. What must occur for the simulation to cycle?

2) Using your considerations from above define **at least five** outcomes (rules) using Boolean Logic. The end goal should be to have your graph look very similar to the

one provided and for the egg to ovulate. Anything else you do to make the model more accurate is up to you! [Hint: look through provided code of follicular phase for assistance]

a.

b.

c.

d.

e.

3) Identify any assumptions you may want to make in order to decrease the complexity of your model.

a.

4) Using these rules and assumptions, determine the code that will need to be added to the model to carry out the rules. Use NetLogo Dictionary (**http://ccl.northwestern.edu/netlogo/docs/index2.html**) for help on syntax and codes that are available. [Hint: the codes **if** and **ifelse** should be very helpful and comprise most of the added components. Also **count**, **ask**, **die**, **any?** and **move** should be helpful. Look at the provided code for the follicular phase for assistance]

5) Once you have a working code, it is important to test that it is accurate. Come up with at least 5 tests (one per rule) to determine if your simulation is working properly.

a.

b.

c.

d.

e.

6) Perform your tests. If your code is not performing how you anticipated try troubleshooting or writing a new code and retry your test.
   a. Describe the outcome of your tests and the troubleshooting process.

7) This model is extremely basic and has many limitations which do not capture the full workings of a lac operon.
   a. In your opinion, what are the model's three biggest limitations? Especially consider any assumptions that were made when answering this question.

   b. Choose one of these limitations and describe what you think should be added to the model to address this limitation.