UNIT PLAN: Constructing Computer Animations in Earth Science

by TEAM MEMBERS

Jovani Cardenas, Peter Tsun

General Overview

(include here description of unit, what class(es) it fits into, when...)

This unit provides students with introduction to key ideas in computer science and skills in logo programming language in Earth Science classes during the astronomy unit. NetLogo allows students to explore coding in a more visually appealing manner. It also hosts a vast library of models such as a climate change model in which it can be used to explore its variables, and later explore its limitations to predict climate change.

Motivation for Unit

(why have you decided to make this?)

For many students, computers are thought of as consumer electronics, not as a medium for putting their own ideas into reality. Students typically do not get to have experience at school or at home to realize that they can actually create useful products themselves. This unit is created to make a small dent in helping children think of themselves as creative thinkers who can turn their ideas into reality. It also serves to demonstrate the benefits and limitations of real-world tools such as computer simulations.

Standards Referenced

Standards utilized from NYSED 9-12 Computer Science Standards (http://www.nysed.gov/common/nysed/files/programs/curriculum-instruction/computer-science-digital-fluency-standards-9-12.pdf)

- 9-12.CT.1 Create a simple digital model that makes predictions of outcomes.
- 9-12.CT.2 Collect and evaluate data from multiple sources for use in a computational artifact.
- 9-12.CT.4 Implement a program using a combination of student-defined and third-party functions to organize the computation.
- 9-12.CT.8 Develop a program that effectively uses control structures in order to create a computer program for practical intent
- 9-12.IC.1 Evaluate the impact of computing technologies on equity, access, and influence in a global society.
- 9-12.IC.3 Debate issues of ethics related to real world computing technologies.
- 9-12.IC.7 Investigate the use of computer science in multiple fields.

In these Earth Science computer programming lessons, students will learn how to design and implement simple programs that recreate observations of planetary orbits. These simple models are attempts to predict how planets will move in the future. We also take a brief look at the benefits of computer simulations as well as its limitations.

Tools Used

(include programming language(s), specific programs/environments, and other tools (digital or otherwise) if necessary)

NetLogo

Resources

(include any links/books/readings to be used during this unit)

1. Thanks to clouds, new climate simulations predict more warming than predecessors (https://www.imperial.ac.uk/news/194738/thanks-clouds-climate-simulations-predict-more/)
2. How reliable are climate models? (https://skepticalscience.com/climate-models.htm)
3. Climate simulations are mostly accurate, study finds (https://apnews.com/article/science-ap-top-news-climate-climate-change-9898308e485f8dea65adb699cb2054a0)
4. Seymour Papert's Mindstorms 1980 https://mindstorms.media.mit.edu/
5. NetLogo and QGIS - Converting GIS files to patches in NetLogo https://youtu.be/_qR9lZy6xjE
6. GIS Tutorial https://static.cambridge.org/content/id/urn:cambridge.org:id:article:S2326376819000056/resource/name/S2326376819000056sup001.pdf
7. b-davies/NetLogo_GIS_tutorial https://zenodo.org/record/1482941#.Yb8p1llOnIU

Lessons

Total length: 2 Weeks (realistically, an ES teacher would try to get it done in 7 or 8 days to keep up with NYS ES curriculum)

- day 1. lesson on iteration (loop)
- day 2. lesson on procedure
- day 3. lesson on variable, passing by parameters to procedures
- day 4. lesson on recursion and conditional
- day 5. lesson on constructing a circle
- day 6. lesson on constructing an elliptical orbit of a planet around the sun
- day 7. lesson on obtaining geographical elevation data and prepare data for NetLogo
- day 8. lesson on drainage pattern with GIS 2D NetLogo modeling
- day 9. lesson on climate change model found within NetLogo library
- day 10. lesson on exploring limitations of models like the climate change model in NetLogo and its implications.

Assesments

(list summative and/or formative assessments used)

- Short response on the limitations of computer simulation models such as used in climate change.
- Verbal explanation of the models explored.
- Verbal explanation of student code and logic.
- A project on 2D or 3D rendering of a geographical area.

Aim: Why do we use iterations?

Objectives:

- Explain the difference between programming with command center and programming in code tab
- Construct a square without iteration command
- Construct a square with iteration command

NYS COMPUTER SCIENCE AND DIGITAL FLUENCY LEARNING STANDARDS

- **9-12.CT.8** Develop a program that effectively uses control structures in order to create a computer program for practical intent
- **9-12.IC.7** Investigate the use of computer science in multiple fields.

Warm Up:

Time: 5 minutes
Go to https://ccl.northwestern.edu/netlogo/ and download NetLogo software for your computer. Skip filling the form and click on Download button right away.

Introduction:

Time: 5 minutes
NetLogo will allow us to create models in Earth Science. We will not be limited to the virtual labs available on the Internet. If the computer models are not available to us, we can create a model by writing a NetLogo program. NetLogo has a command center that allows us to give instructions to the computer in an interactive manner. NetLogo also has a code editor, where we can write a play, and the computer will perform the play.

Lesson Content:

Time: 10 minutes

Today we will learn how to write a basic NetLogo program.

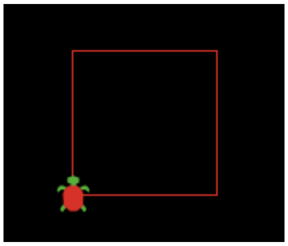
```
to setup
  clear-all
  create-turtles 1 [
    set shape "turtle"
    set color red
    set size 2
    set heading 0
    pen-down
```

```
    ]
  end

  to go
    ask turtle 0 [
      forward 7
      right 90
    ]
  end
```

In this code, a turtle is created when the setup block is called. When the code block is called, the turtle is asked to move forward 7 steps and turn to the right 90 degrees.

You can call the setup block by typing its name at the command prompt at the bottom of the screen.

Type this code in your computer's NetLogo editor.



Lesson Activity

Time: 15 minutes

- Give students a few minutes to draw a square.
- Introduce the GUI buttons that can be used to call setup and go blocks.
- Introduce the command called repeat which allows a square to be drawn with one press on the GO button.

Students are challenged to apply what they have learned to draw a triangle.

```
  to setup
    clear-all
    create-turtles 1 [
      set shape "turtle"
      set color red
      set size 2
      set heading 0
      pen-down
    ]
  end

  to go
    ask turtle 0 [
      repeat 4 [
        forward 7
        right 90
      ]
    ]
  end
```

Closing

Time: 10 minutes Invite students to show others how to draw a triangle. If they do not know how to draw a triangle, teacher will show that how to draw a triangle.

```
to setup
  clear-all
  create-turtles 1 [
    set shape "turtle"
    set color red
    set size 2
    set heading 0
    pen-down
  ]
end

to go
  ask turtle 0 [
    repeat 3 [
      forward 7
      right 120
    ]
  ]
end
```

Feedback

Time: 0 minutes - Homework
Draw a pentagon

Explanation

This lesson shows students the power of iteration/loop. Instead of typing the command multiple times, students can instruct a computer to repeat a command many times.

Aim: How do we use procedures?

Objectives

- Apply knowledge of constructing square and triangle to draw a house
- Apply procedural concept to draw a house
- Extend the procedural concept to the whole house

NYS COMPUTER SCIENCE AND DIGITAL FLUENCY LEARNING STANDARDS

- **9-12.IC.7** Investigate the use of computer science in multiple fields.
- **9-12.CT.4** Implement a program using a combination of student-defined and third-party functions to organize the computation.

Warm Up

Time: 5 minutes
Teacher asks students to draw a square or a triangle.

Lesson Content

Time: 20 minutes

Review how to construct a square and a triangle. In a square, the turn is 90 degrees. For a triangle, the turn is 120 degrees.

Lesson Activity

Time: 15 minutes

Students are asked to draw a house.

```
to setup
  clear-all
  create-turtles 1 [
    set shape "turtle"
    set color sky
    set size 2
    set heading 90
    pen-down
  ]
end

to go
  ask turtle 0 [
    repeat 4 [
      forward 5
      right 90
    ]
    repeat 3 [
      forward 5
      left 120
    ]
  ]
end
```

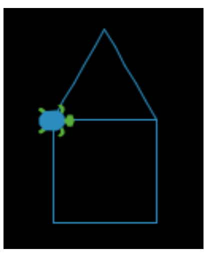
After students have successfully drawn a house. Ask them to draw two more houses. It will take a while and definitely a lot more lines of code.

Teacher introduces the concept of procedure to students to explain the benefit of procedure which allows pieces of code to be reused easily.

```
to setup
  clear-all
  create-turtles 1 [
    set shape "turtle"
    set color sky
```

```
    set size 2
    set heading 90
    pen-down
  ]
end

to go
  ask turtle 0 [
    square
    triangle
  ]
end

to square
  repeat 4 [
    forward 5
    right 90
  ]
end

to triangle
  repeat 3 [
    forward 5
    left 120
  ]
end
```

Students are asked to extend the procedure concept to the construction of a house.



Closing

Time: 5 minutes

Review how programming without procedures results in a lot of repetitive pieces of code. With the use of procedure, a piece of code can be collected in a block. To use this piece of code, all one has to do is to call the block by invoking its name. Once again, the purpose of procedure is to allow use to reuse a piece of code. It also makes a program easier to read and maintain.

```
  clear-all
  create-turtles 1 [
    set shape "turtle"
    set color sky
    set size 2
    set heading 90
    pen-down
  ]
end
```

```
to go
  ask turtle 0 [
    house
  ]
end

to house
  square
  triangle
end

to square
  repeat 4 [
    forward 5
    right 90
  ]
end

to triangle
  repeat 3 [
    forward 5
    left 120
  ]
end
```

Explanation

The lesson is designed for students who will write program in NetLogo to produce virtual models of Earth Science.

Aim: Why do we use variables and parameters?

Objectives

- Use variables to resize a house
- Use parameter for a procedure to take input from outside
- call a procedure with an argument to resize a house

NYS COMPUTER SCIENCE AND DIGITAL FLUENCY LEARNING STANDARDS

- **9-12.CT.2** Collect and evaluate data from multiple sources for use in a computational artifact.
- **9-12.CT.4** Implement a program using a combination of student-defined and third-party functions to organize the computation.

Warm Up

Time: 5 minutes

Teacher asks students to draw a house.

Lesson Content

Time: 20 minutes

Students try to draw a house. Review how to construct a house with procedures for a square and a triangle. Introduce the steps for declaring global variable. Students are asked draw a larger house. Then they will be asked to draw an even larger house. Then, they will be asked to redraw a house to make it smaller.

```
globals [length-house]

to setup
  clear-all
  create-turtles 1 [
    set shape "turtle"
    set color sky
    set size 2
    set heading 90
    pen-down
  ]
  set length-house 5
end

to go
  ask turtle 0 [
    house
  ]
end

to house
  square
  triangle
end

to square
  repeat 4 [
    forward length-house
    right 90
  ]
end

to triangle
  repeat 3 [
    forward length-house
    left 120
  ]
end
```

Lesson Activity

Time: 15 minutes

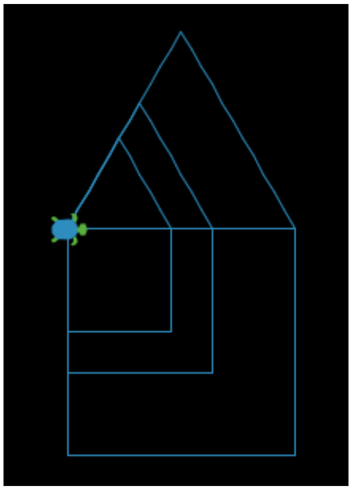
Students are introduced to procedure that take inputs as parameters. Each procedure will have a parameter of side-length. By changing the value of a global variable, called length-house, a call to a square procedure will take an input argument, which has a value contained in the global variable length-house, and draw a square with the size length-house.

```
globals [length-house]

to setup
  clear-all
  create-turtles 1 [
    set shape "turtle"
    set color sky
    set size 2
    set heading 90
    pen-down
  ]
  set length-house 5
end

to go
  ask turtle 0 [
    house
  ]
end

to house
  square 5; length-house
  triangle 5; length-house
end

to square [length-square]
  repeat 4 [
    forward length-square
    right 90
  ]
end

to triangle [length-triangle]
  repeat 3 [
    forward length-triangle
    left 120
  ]
end
```

Students will learn about the GUI interface called slider that allows quick changes for a variable.

```
;globals [length-house]

to setup
  clear-all
  create-turtles 1 [
    set shape "turtle"
    set color sky
    set size 2
    set heading 90
    pen-down
  ]
  ;set length-house 5
end

to go
  ask turtle 0 [
```

```
    house
  ]
end

to house
  square length-from-slider
  triangle length-from-slider
end

to square [length-square]
  repeat 4 [
    forward length-square
    right 90
  ]
end

to triangle [length-triangle]
  repeat 3 [
    forward length-triangle
    left 120
  ]
end
```



Closing

Time: 5 minutes

Students will explain how a variable makes a difference in their work of drawing houses with different sizes. With a variable, you can make a change in one place either by changing text or GUI slider, and the computer automatically passes information to procedures with that variable. Without variables, one has to changes numbers manually in different places. Changing the same values in many places by hand leads to human typing errors and lost productivity.

Explanation

The lesson is designed for students who will write program in NetLogo to produce virtual models of Earth Science.

Aim: How do we use recursion?

Objectives

- Use procedure's parameters for recursion
- Use conditionals to end recursion
- Modify various parameters to create different recursion patterns.

NYS COMPUTER SCIENCE AND DIGITAL FLUENCY LEARNING STANDARDS

- **9-12.IC.7** Investigate the use of computer science in multiple fields.
- **9-12.CT.4** Implement a program using a combination of student-defined and third-party functions to organize the computation.

Warm Up

Time: 5 minutes
Teacher asks students to type an example of recursion into NetLogo.

Lesson Content

Time: 20 minutes

This computer program is unique because the program calls a copy of a procedure, and the procedure calls a copy of the same procedure. This concept of a procedure calling a copy of itself is called recursion. Normally people tend to avoid recursion, because recursion seems like a never-ending invoking of a copy of the same procedure.

However, this program will definitely end because the starting length is small. Every time a copy of the procedure is called, a slightly larger length is fed into the procedure. When the length is greater some threshold, the recursive call to the procedure will not be made. This results in the stopping of continued recursive calls.

Lesson Activity

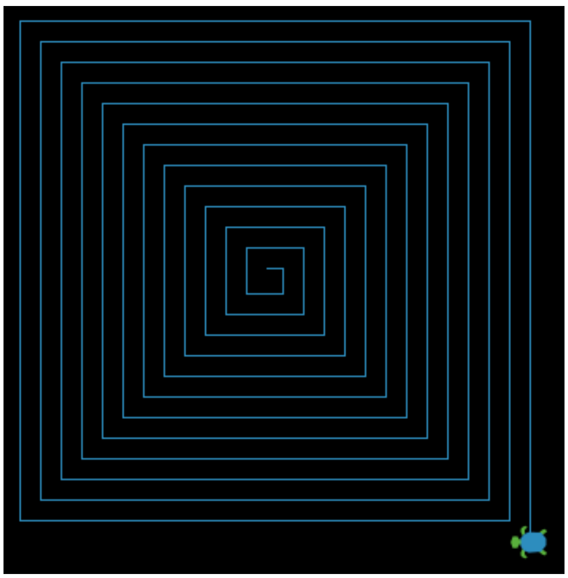
Time: 15 minutes

Students are invited to change various parameters in their recursion program, and discover the effects of their changes.

They can try to integrate the globals variable to give significance to their parameters. Students may even try to add GUI slider to control global variable from the graphical interface.

```
globals [length-side]

to setup
  clear-all
  create-turtles 1 [
    set shape "turtle"
    set color sky
    set size 2
    set heading 90
    pen-down
  ]
```

```
  set length-side .5
end

to go
  ask turtle 0 [
    spiral length-side
  ]
end

to spiral [length-spiral]
  forward length-spiral + 0.3
  right 90
  if (length-spiral < 25) [
    spiral (length-spiral + 0.5)
  ]
end
```



Closing

Time: 5 minutes

Students will explain how a recursive call is made and terminated. Every student is invited to share their recursive graphical patterns.

Explanation

The lesson is designed for students who will write program in NetLogo to produce virtual models of Earth Science.

Aim: How do we construct a circle?

Objectives

- Use basic commands to draw an arc
- Use simple instruction to draw a generic circle

- Use mathematical formula to draw a circle with specfic size.

NYS COMPUTER SCIENCE AND DIGITAL FLUENCY LEARNING STANDARDS

- **9-12.IC.7** Investigate the use of computer science in multiple fields.
- **9-12.CT.4** Implement a program using a combination of student-defined and third-party functions to organize the computation.

Warm Up

Time: 5 minutes
Teacher asks students to draw an arc in NetLogo.

Lesson Content

Time: 20 minutes

Students get a hint for drawing an arc by tracing a pen a little forward and a little to the right.

```
globals [length-side]

to setup
  clear-all
  create-turtles 1 [
    set shape "turtle"
    set color sky
    set size 2
    set heading 0
    setxy -12 0
    pen-down
  ]
  set length-side .5
end

to go
  ask turtle 0 [
    circle
  ]
end

to circle
  repeat 80 [
    forward 0.1
    right 0.5
  ]
end
```

Students are asked to draw a circle. At first their circles are not complete, but eventually they realize they need 360 degrees for a circle if they turn 1 degree for each iteration.

```
globals [radius]

to setup
  clear-all
```

```
  create-turtles 1 [
    set shape "turtle"
    set color sky
    set size 2
    set heading 0
    setxy -12 0
    pen-down
  ]
  set radius 5
end

to go
  ask turtle 0 [
    circle
  ]
end

to circle
  repeat 360 [
    forward 0.1
    right 1
  ]
end
```

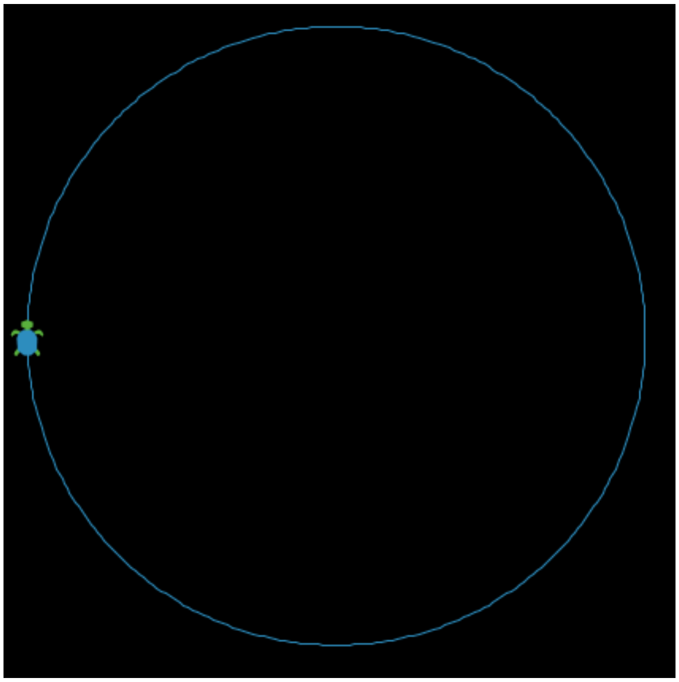
Lesson Activity

Time: 15 minutes

The students are asked to try to draw a circle of exactly the size they want. They may eventually realize they need to use the circumference formula: C = 2 * pi * r, where r = radius. If a turtle turns 1 degree per iteration, then the distance traveled during an iteration will be (1/360) C or 2 * pi * r / 360.

```
globals [radius]

to setup
  clear-all
  create-turtles 1 [
    set shape "turtle"
    set color sky
    set size 2
    set heading 0
  ]

end

to go
  set radius 15
  ask turtle 0 [
    pen-up
    setxy (-1 * radius) 0
    pen-down
    circle
  ]
end

to circle
  repeat 360 [
    forward 2 * pi * radius / 360
```

```
      right 1
    ]
  end
```



Closing

Time: 5 minutes

Students explain their codes to students in other groups. Some students explain their circle program to students in the class. As an exercise, students can incorporate slider variable to adjust the radius of their circles.

Explanation

The lesson is designed for students who will write program in NetLogo to produce virtual models of Earth Science.

Aim: How do we recreate an elliptical orbit?

Objectives

- Use basic commands to draw an ellipse
- Use a stopwatch to measure speeds at aphelion and perihelion
- Compare the speeds at aphelion and perihelion

NYS COMPUTER SCIENCE AND DIGITAL FLUENCY LEARNING STANDARDS

- **9-12.IC.7** Investigate the use of computer science in multiple fields.
- **9-12.CT.4** Implement a program using a combination of student-defined and third-party functions to organize the computation.

Warm Up

Time: 5 minutes
Teacher asks students to write a setup for creating three turtles.

Lesson Content

Time: 20 minutes

Students will code along with teacher who explains the thought process that goes into creating an elliptical orbit during the code-along.

```
; Code for drawing an elliptical orbit
to setup
  clear-all
  create-turtles 3 [
    set shape "butterfly"
    set size 2
    set color pink
    set heading 0
    setxy 9 0
    pen-up
  ]
  ask turtle 0 [
    set color yellow
  ]
  ask turtle 1 [
    setxy 0 0
  ]
  ask turtle 2 [
    setxy 9 0
    pen-down
    set color red
  ]
  draw-grid
  reset-ticks
end

to go
  ;repeat 1500 [
    ask turtle 2 [
      forward .153 / sqrt (distance turtle 1)  ; this is v, a result of Newton’s law of gravity Fg
      left 0.5 / sqrt (distance turtle 1)      ; this is ω, angular speed, also a result of Fg
  ;  ]
  ]
  tick
end

to draw-grid
    ask patches [
    sprout 1 [
      set heading 0
      forward 0.5
      right 90
      set color white
      pen-down
      repeat 4 [ forward 0.5 right 90 forward 0.5 ]
```

```
        die
      ]
    ]
    ask turtle 0 [
      setxy min-pxcor 5
      pen-down
      set heading 90
      forward  (max-pxcor - min-pxcor)
      pen-up
      setxy min-pxcor -5
      pen-down
      forward ( max-pxcor - min-pxcor)
      pen-up
      setxy max-pxcor max-pycor
    ]
  end
```

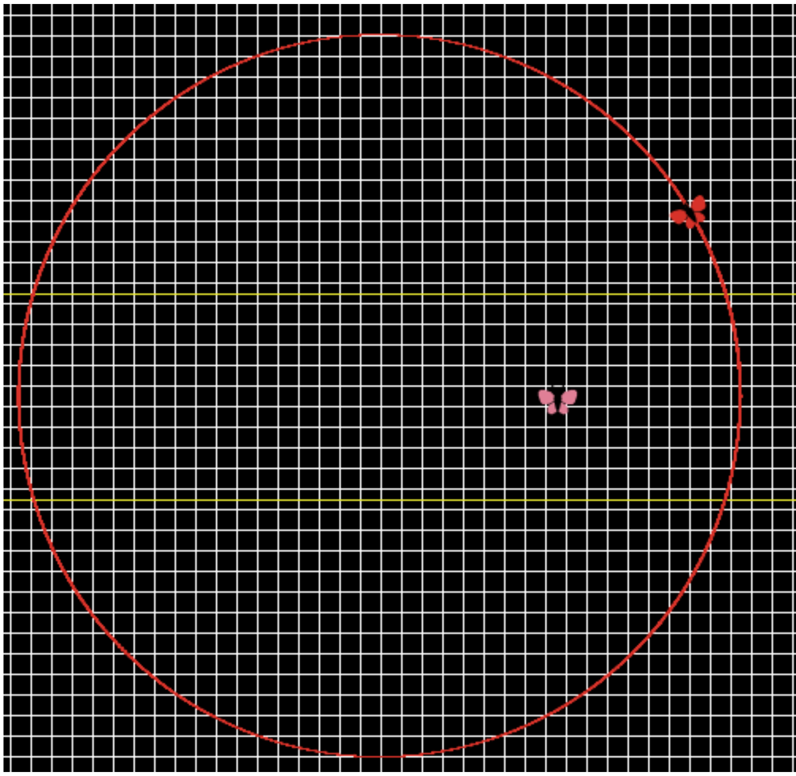
Students are asked to set the max px-cor to 27 and max py-cor to 20 in the settings. Let the students know that they should set the go button to be continuous. Under the view updates in the interactive tab, choose the option on ticks, instead of the option continuous. To run this program, tell students that when running setup, have the speed to be faster. When running the continuous go procedure, have the speed to be much slower, so it is easier to measure the time interval of a portion of an orbit.

Lesson Activity

Time: 15 minutes

The students are asked to measure the time interval when Earth is orbiting closer to the Sun (perihelion). They start the stopwatch when the Earth crosses the first yellow line and travels for 10 units of distance, and stop the clock when the Earth reaches the second yellow line. Then, students will measure the time interval when the Earth is orbiting farthest away from the Sun (aphelion). They start the stopwatch when the Earth crosses the first yellow line and travels for 10 units moving towards the bottom of the screen, and stop the clock when the Earth reaches the second yellow line.

After measuring the time intervals for aphelion and perihelion, compare the speed of the Earth at aphelion and perihelion. Speed is calculated by dividing distance traveled by time interval.



Closing

Time: 5 minutes

Students explain what they discovered in this activity. From their model building and measurement/analysis, they should be able to explain to others that Earth travels faster when it is closer to the Sun (perihelion) and Earth moves slower when it is farthest away from the Sun (aphelion).

Explanation

The lesson is designed for students who will write program in NetLogo to produce virtual models of Earth Science.

Aim: How do we obtain GIS data for NetLogo programs?

Objectives

- Obtain elevation data from USDA GIS interactive map
- Obtain elevation data from USGS GIS library
- Prepare the data from GIS system to be ready for use in NetLogo

NYS COMPUTER SCIENCE AND DIGITAL FLUENCY LEARNING STANDARDS

- **9-12.IC.7** Investigate the use of computer science in multiple fields.
- **9-12.CT.4** Implement a program using a combination of student-defined and third-party functions to organize the computation.

Warm Up

Time: 5 minutes
Teacher directs students to download qGIS software at https://www.qgis.org/en/site/. Once students download a copy of qGIS software to their computers, they should install the qGIS software.

Lesson Content

Time: 20 minutes

Elevation Data from GIS interactive map

- Search for the phrase geospatial data gateway (https://datagateway.nrcs.usda.gov/)
- Click on **Get Data** button, then click on the green link called HERE
- Click on **Order by Interactive Map**
- Once you zoom into an area where you want its elevation data
- Click on Select area button, and select an area of interest with your mouse.
- Click on accept area, and Check National Elevation Dataset
- Click Continue button, and then another continue
- Provide your email, and then click **Place Order**

Students opened their email and download the zip file from USDA. Students are then instructed to go to the browser panel to drag the tif file to the project area. A new layer will be automatically created. Once students are finished or even if they run into obstacles, students are directed to watch a video that explains how to obatin GIS data from USDA and USGS: https://youtu.be/VlKkAYoNoRE (Title: Download Free Elevation Data (DEM) from the USGS National Map and NRCS Geospatial Data Gateway).

Lesson Activity

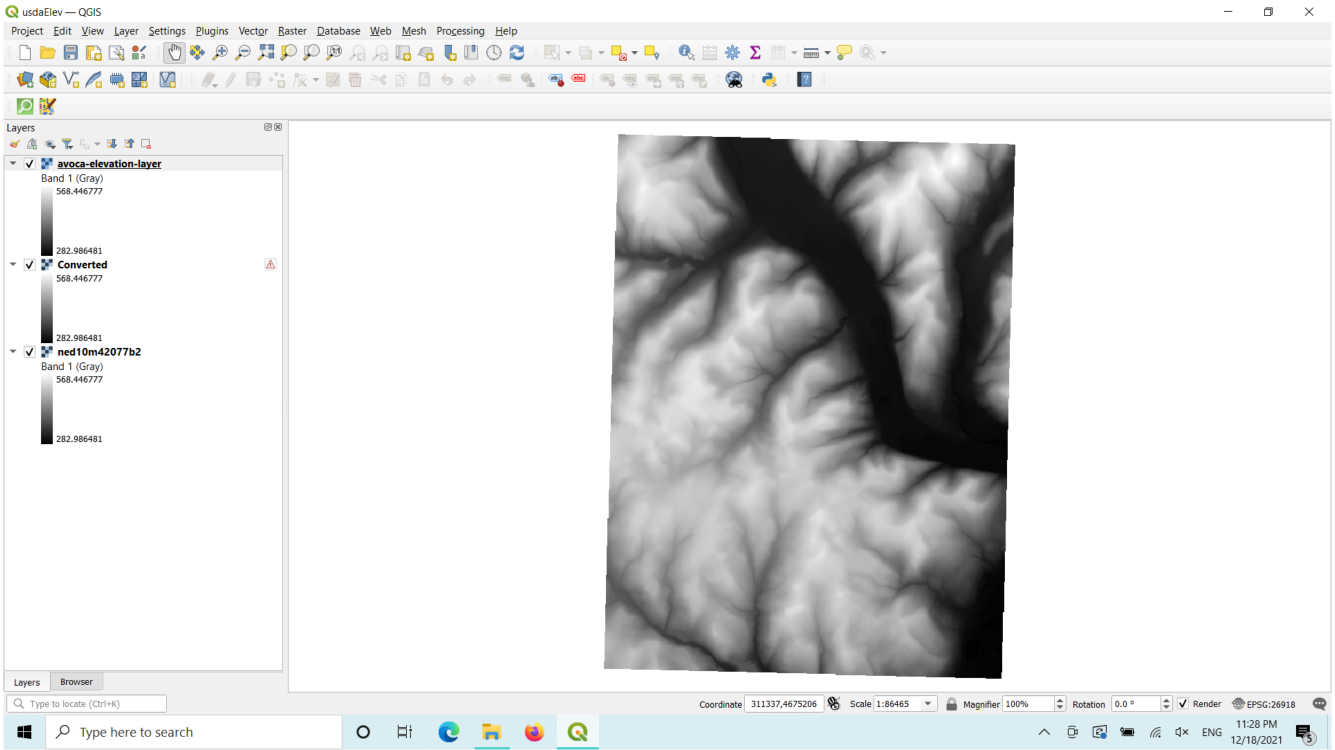
Time: 15 minutes

Students will watch part of a video clip to learn how to convert a raster file to asc file. https://youtu.be/_qR9lZy6xjE (Title: NetLogo and QGIS - Converting GIS files to patches in NetLogo). Students will open GIS data in qGIS and do a vector conversion to translate to a ASCII file. Students will have to name the asc file that they want to save. Once students are finished with the three tasks, they can make a note for themselves of how they obtain GIS file and prepare ASC file so that they can refer to Today's steps for future reference.

Closing

Time: 5 minutes

Students present their notes to the other students in the class. The notes are directions for themselves on how to obtain GIS data for no cost and how to prepare the data in a format that NetLogo can understand. (possibly 2 days given the time it takes to download GIS software and GIS data and the amount of time to replay the video instruction on how to obtain GIS elevation data).



Explanation

The lesson is designed for students who will write program in NetLogo to produce virtual models of Earth Science.

Aim: How do we show streams on NetLogo?

Objectives

- Obtain elevation data from USDA GIS interactive map
- Input the data from GIS into NetLogo
- Make use of existing NetLogo model library to create streams on a land.

NYS COMPUTER SCIENCE AND DIGITAL FLUENCY LEARNING STANDARDS

- **9-12.IC.7** Investigate the use of computer science in multiple fields.
- **9-12.CT.4** Implement a program using a combination of student-defined and third-party functions to organize the computation.

Warm Up

Time: 5 minutes
Teacher directs students to obtain a tif file from the https://datagateway.nrcs.usda.gov

Lesson Content

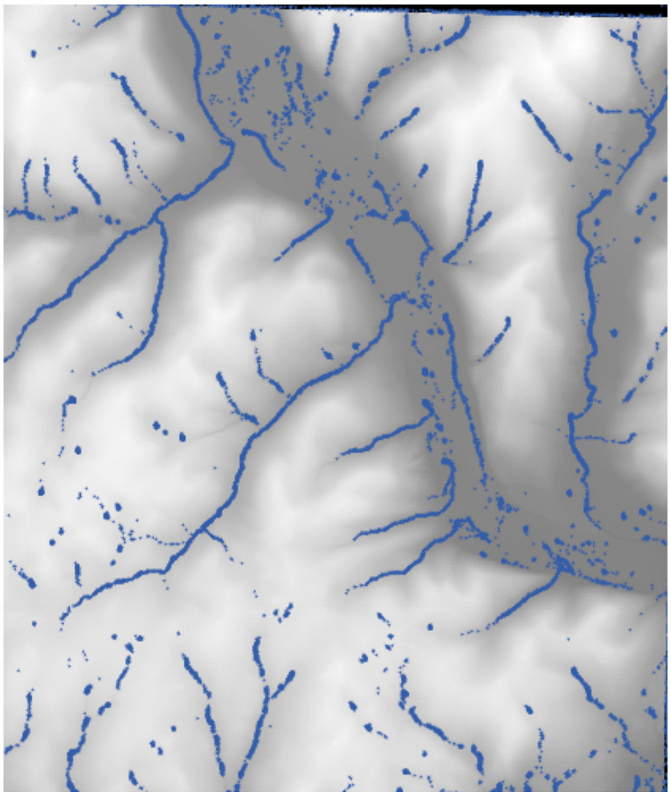
Time: 20 minutes

Teacher walks students through the process of creating asc file and bringing this GIS data into NeLogo. Teacher asks some students to show how to import GIS into NetLogo. Teacher then shows students the model library sample program for GIS gradients.

Lesson Activity

Time: 15 minutes

- Students use the sample program called **GIS Gradient Example** in the models library --> Code Examples --> Extension Examples --> gis as a guide to recreate the stream flow for their own selected map.
- Students examine the sample program and write a detailed explanation of how the sample gradient program works to the best of their abilities.



Closing

Time: 5 minutes

Students show their stream programs on their chosen area, and present their explanations so far of how the NetLogo gradient streams program works.

Explanation

The lesson is designed for students who will write program in NetLogo to produce virtual models of Earth Science.

Aim: How can computer simulations improve our understanding of climate change?

Objectives

- Utilize NetLogo climate change model from model library to manipulate variables and record outcomes
- Simulate Earth's current situation and possible outcome if climate change is not addressed at various future outcomes (5 years from now, 10 years from now, 15 years from now).

NYS COMPUTER SCIENCE AND DIGITAL FLUENCY LEARNING STANDARDS

- 9-12.IC.1 Evaluate the impact of computing technologies on equity, access, and influence in a global society.
- 9-12.IC.7 Investigate the use of computer science in multiple fields.

Warm Up

Time: 5 minutes

Teacher instructs students to obtain computers, sign in, and enter NetLogo. Teacher will instruct students to find the Climate Change Earth Science Model from within the model library.

Lesson Content

Time: 15 minutes

Teacher will probe student background knowledge of climate change and its relationship to carbon dioxide. The instructor will create a list of key details that students mention. The teacher will seek to create a list that includes at minimum the following details:

1. Climate change will have devastating effects on the environment
2. Climate change is occurring due to human activity
3. Climate change is directly related to an increase in greenhouse gases, especially carbon dioxide
4. Carbon dioxide is a byproduct of human industrialization
5. Carbon dioxide traps the Sun's UV rays from escaping Earth's atmosphere

The teacher will add onto the class list if these details are missing.

Lesson Activity

Time: 20 minutes

Students will be tasked to explore the NetLogo Climate Change model. They will be reminded to change one variable at a time. Students will be asked the following questions, which will be displayed on the board:

1. What happens to sunlight rays as they enter Earth's atmosphere? Do they always bounce out of Earth's atmosphere?
2. Run the model with a sun brightness of 1, albedo of 0.6, no clouds and no CO2 added. What is the highest temperature it reaches?
3. Run the model with a sun brightness of higher than 1, albedo of 0.6, no clouds and no CO2 added. What temperature difference, if any, do you notice?
4. Run the model with a sun brightness of 1, albedo of 0.6, no clouds and change the CO2 to a number higher than

0. What is the highest temperature you can reach?
5. Using the previous model settings you used, begin to add clouds. What effect do clouds have on the temperature? Start off by adding a few clouds and progressively add more clouds.
6. Run the model with a sun brightness of 1, no clouds, and no CO2 added. Change the albedo to below 0.6. What effect does this have? Change the albedo to higher than 0.6 what effect does this have?
7. Try to get the highest temperature possible!

Students will be asked to record all their findings in their notebooks.

Closing

Time: 5 minutes

The teacher will go through the list of questions and ask students for their responses, ideas, and reactions to their findings.

Explanation

This lesson serves to allow students to manipulate particular variables and see the effect of carbon dioxide on global temperatures. This lesson serves as a precursor for the next lesson in which students will be questioning and researching the efficiency, reliability, and validity behind computer models such as this one.

Aim: Can we rely on computer models to predict future climate change?

Objectives

- Investigate the efficacy of computer simulations as they pertain to predicting climate change.
- To explore the limitations of computer simulations as they pertain to climate change modeling.

NYS COMPUTER SCIENCE AND DIGITAL FLUENCY LEARNING STANDARDS

- 9-12.IC.1 Evaluate the impact of computing technologies on equity, access, and influence in a global society.
- 9-12.IC.3 Debate issues of ethics related to real world computing technologies.

Warm Up

Time: 5 minutes

Students will enter, sign into their computers and access Google Classroom. In Google Classroom they will access the assignment labeled “Can we rely on computer models to predict future climate change?” Within this assignment are 3 articles in pdf format.

Lesson Content

Time: 10 minutes

The teacher will instruct students to reflect on the previous lesson. The teacher will ask students to write down if they believe the climate change model found in NetLogo is accurate enough to predict the effects of climate change.

Students will share their responses with the class.

Lesson Activity

Time: 20 minutes

Students will be asked to read 3 articles:

1. **Thanks to clouds, new climate simulations predict more warming than predecessors** (https://www.imperial.ac.uk/news/194738/thanks-clouds-climate-simulations-predict-more/)
2. How reliable are climate models? (https://skepticalscience.com/climate-models.htm)
3. **Climate simulations are mostly accurate, study finds** (https://apnews.com/article/science-ap-top-news-climate-climate-change-9898308e485f8dea65adb699cb2054a0)

Students will then be asked to write a short response about their opinion on the current climate change models. They will be asked to support their answers with evidence from the articles or from other credible sources found on the web. They will also be asked to utilize their experience with the climate change model in NetLogo from the previous lesson.

Some student responses will be shared aloud.

Closing

Time: 10 minutes

Students will be asked in pairs to list some factors that should be in every climate change model. A short list will be compiled from student answers and displayed on the board.

Explanation

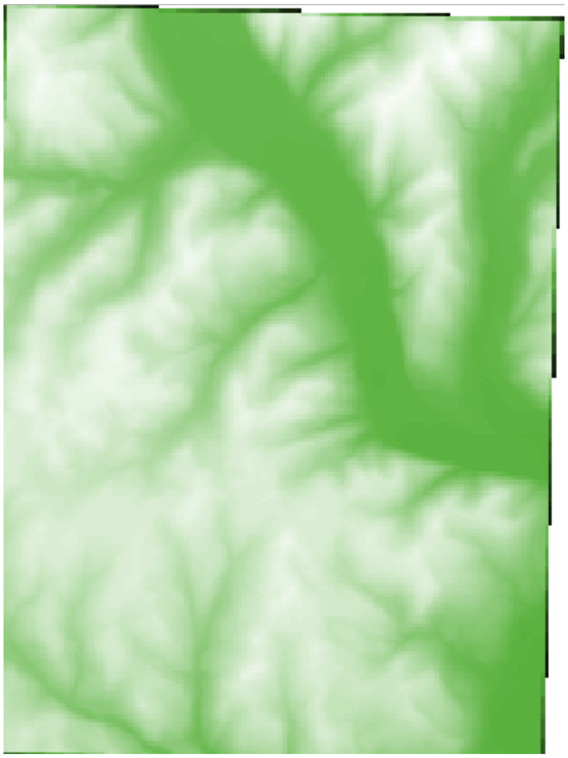
This lesson seeks to allow students to explore the limitations of computer simulations. Students will read articles that share opposing opinions on climate change computer simulations.

Project: Visualization of elevation data Map

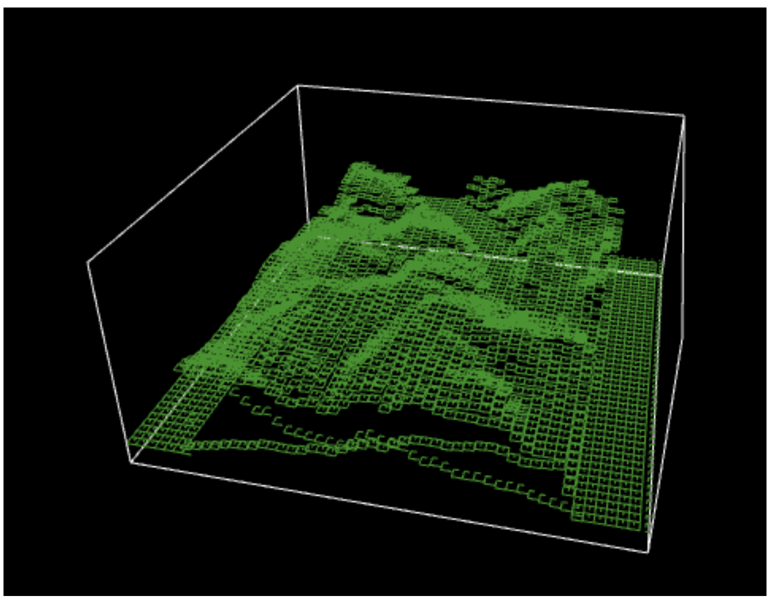
Options:

- option 1) Build a two-dimensional elevation map
- option 2) Build a three-dimensional elevation map

1.



2.



To build a 2-d elevation map, review lesson 7 on how to obtain GIS elevation data from USDA. Then, make use of the sample code below:

```
extensions [ gis ]
globals [ elevation-dataset ]

patches-own [patch-elevation]

to setup
  clear-all

  gis:load-coordinate-system "data/elevation/avoca-elevation-layer.prj"
  ;gis:load-coordinate-system "data/elevationUSGS/avoca-layer.prj"
```

```
set elevation-dataset gis:load-dataset "data/elevation/avoca-elevation-layer.asc"
```